EAI Endorsed Transactions

on Industrial Networks And Intelligent Systems Research Article ICST.ORG

An Investigation of Performance Analysis of Anomaly Detection Techniques for Big Data in SCADA Systems

Mohiuddin Ahmed[1], Adnan Anwar[1], Abdun Naser Mahmood[1], Zubair Shah and Michael J. Maher[1]

[1] School of Engineering and Information Technology, UNSW Canberra, ACT 2600, Australia

Abstract

Anomaly detection is an important aspect of data mining, where the main objective is to identify anomalous or unusual data from a given dataset. However, there is no formal categorization of application-specific anomaly detection techniques for big data and this ignites a confusion for the data miners. In this paper, we categorise anomaly detection techniques based on nearest neighbours, clustering and statistical approaches and investigate the performance analysis of these techniques in critical infrastructure applications such as SCADA systems. Extensive experimental analysis is conducted to compare representative algorithms from each of the categories using seven benchmark datasets (both real and simulated) in SCADA systems. The effectiveness of the representative algorithms is measured through a number of metrics. We highlighted the set of algorithms that are the best performing for SCADA systems.

Keywords: Anomaly detection, SCADA systems, big data.

Received on 19 December 2014, accepted on 12 February 2015, published on 08 May 2015

doi: 10.4108/inis.2.3.e5

1. Big Data Analysis in SCADA Systems

Supervisory Control and Data Acquisition (SCADA) systems are widely used for monitoring and control of Industrial Control System (ICS) of national critical infrastructures, including the emerging energy system, transportation system, gas and water systems, and so on. Generally, ICS is comprised of Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs) with Intelligent Electronic Devices (IEDs), a telemetry system, a Human Machine Interface (HMI) and a supervisory (computer) system. In a SCADA based ICS, communication infrastructures connect the supervisory (computer) systems and the RTUs. The operational process and requirements of SCADA systems, which are used for industrial networks, have characteristics distinct from enterprise networks. The primary objective of a SCADA system is to control real-life physical equipment and devices, e.g., an energy system SCADA may be used for monitoring and control of the generation plants. On the other hand, conventional information based traffic network is used for data processing and transfer [23]. As the primary objective of the SCADA is different from the conventional information network, the operational process and its requirements vary significantly. Since the SCADA is used to control critical infrastructures, the failure severity is very high which requires a high level of reliability. Moreover, the data acquisition, processing, and transmission require real-time operation or atleast near real-time operation. Besides, the data transferred through the SCADA devices are both periodic and aperiodic [23]. For example, in a SCADA based energy transmission system, an RTU sends the information of the voltages and currents of a node every few seconds continually (which is periodic) and it also sends a warning when the current exceeds the maximum rating (which is aperiodic). It is also important to ensure that the transmitted data is received without losing any information within a specific time-frame. A conventional information traffic network can withstand even a high data loss but this is not the case for the SCADA devices as the real-time physical process is highly dependent on the data they receive. In Figure 1, a brief overview of SCADA architecture is given. Next, we briefly discuss the importance and significance of Big Data analysis in a SCADA based ICS.

Typically, big data has three dimensional properties (3V) that include volume, velocity and variety [28]. The term *'volume'* is related with the amount of data and its dimensionality. *'Velocity'* is the processing speed of the data. The last property of big data, *'variety'* refers

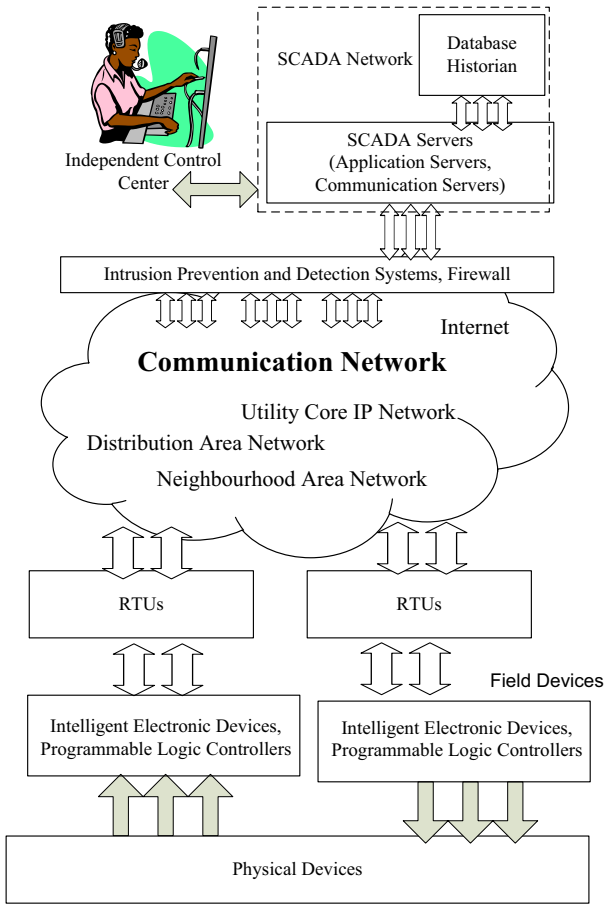

Figure 1. An overview of SCADA Architecture

to the mix of different types of data. Now, we discuss the essence of big data analysis in a SCADA network considering the *3V* properties of big data.

Generally, a SCADA system is dispersed across a large geographic area and is combined of multiple independent systems [23]. Hence, lots of sensor devices and actuators are used to monitor and control of this wide spread large networks. Therefore, the amount of data received in a SCADA is also huge which makes the data analysis a challenging issue. Moreover, the recent trend of using Ethernet and web standards combined with traditional SCADA standards has shifted the SCADA paradigm from event-driven to process-driven, enabling the control of SCADA devices under streaming information exchange. Besides, significant amount of monitoring devices are used to ensure the observability of the processes. All of these technological advancements have provided an improved control performance of the SCADA system; however, big data issue has been emerged with the increased volume of information used in a SCADA network [28].

The second property of the Big data is the *'velocity'* at which the data is processed. In a SCADA system, this property is very crucial as the time requirement of SCADA data exchange is real-time or near real-time. Therefore, those applications which need faster processing, big data is a critical factor and needs significant attention. Even applications which are based on post-event analyses face noticeable challenge to handle the huge amount of data from a SCADA network. Therefore, improved and robust techniques, which are capable of handling big data within sufficient time frame, will add extra value to manage the SCADA network more efficiently and reliably.

In a SCADA system, field devices are responsible to collect different types of data for monitoring a physical system. Therefore, data received from *'variety'* of sources also make the processing very challenging. As a result, the big data issues need to be addressed as all 3V properties of big data is observed in the data received from the SCADA system.

Based on this scenario, performance analysis of anomaly detection techniques is a research requirement. Recently, a number of approaches have been proposed for big data analysis [4–10]. However, for SCADA systems, we are the pioneer to investigate the anomaly detection techniques in big data perspective. Our contribution in this paper are the following:

- We categorize the anomaly detection techniques based on nearest neighbour, clustering and statistics.
- Representative algorithms in each category are applied on benchmark SCADA systems datasets.
- We evaluate the performance of the algorithms using a number of metrics such as *accuracy*, *false positive rate*, *hit rate*, *F-measure* and *MCC*.
- Finally, we highlight the set of techniques that are efficient for big data analysis.

Rest of the papers are organized as follows. Section 2 provides fundamental aspects of anomaly detection and a taxonomy. Section 3 contains the discussion on the different categories of anomaly detection algorithms. Section 4 discusses the proposed criterion to benchmark anomaly detection algorithms and their merits/demerits. Section 5 provides the experimental results and detailed discussion on the performance comparison. We conclude our paper in section 6.

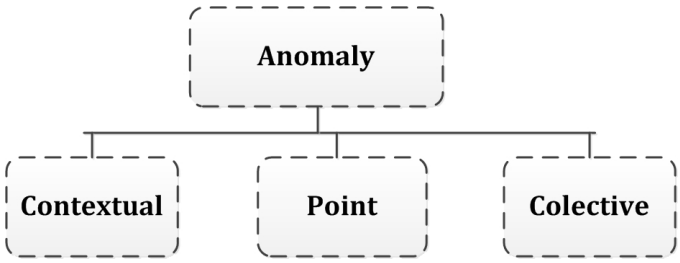

Figure 2. A simple taxonomy of anomaly

2. Anomaly Detection Fundamentals

Anomaly detection is an important data analysis task. The main objective of anomaly detection is to detect anomalous or abnormal data from a given dataset. This is an interesting area of data mining research as it involves discovering new and rare patterns from a dataset. Anomaly detection has been widely studied in statistics and machine learning. It is also known as outlier detection, novelty detection, deviation detection and exception mining [1]. Based on the characteristics of data instances, anomalies are grouped into three categories (Figure 2). These are discussed below:

- ***Point Anomaly*:** When a particular data instance deviates from the normal pattern of the dataset, it can be considered as a point anomaly. For a familiar example, we can consider expenditure on electricity bills. If the usual bill per month is about 100 dollars, and if for one month it is 500 dollars then obviously it is a point anomaly [3].

- ***Contextual Anomaly*:** When a data instance is anomalous in a particular context, but not in other times, then it is termed a contextual anomaly, or conditional anomaly. For example, the expenditure on credit card during a festive period, e.g., Christmas or New Year, is usually higher than the rest of the year. Although, the expenditure during a festive month can be high, it may not be anomalous due to the expenses being contextually normal in nature. On the other hand, an equally high expense during a non-festive month could be considered as a contextual anomaly.

- ***Collective Anomaly*:** Collective anomaly is a pattern in the data when a group of similar data instances behave anomalously with respect to the entire dataset. It might happen that the individual data instance is not an anomaly by itself, but due to its presence in a collection it is identified as an anomaly. For example, a denial of service attack can be considered as a group of network traffic instances affecting the network as well as collective anomaly [2, 24].

One important issue in anomaly detection is how the anomalies are represented as output. Generally there are two categories:

- **Scores:** Scoring based anomaly detection techniques assign a score to each of the data instances. Then the scores are ranked and analyst used to choose the anomalies or use a threshold to select.

- **Binary:** According to these techniques, outputs are considered in binary fashion, i.e. either anomaly or not. Techniques which provide binary labels are computationally efficient since each of the data instances do not have to provide scores.

3. Anomaly Detection Techniques

In this section, we discuss the anomaly detection techniques covered in the scope of this paper. There are various kinds of anomaly detection techniques based on different theories [1, 25]. In this paper, we classify the anomaly detection techniques in two major categories. These are the following:

- **Supervised Learning**: It is the machine learning task of inferring a function from labelled training data [39]. The training data consist of a set of training examples. In supervised learning, the training examples consist of an input object and a desired output value. A supervised learning algorithm learns from the training data and creates a knowledge base which can be used for mapping new and unseen data.

- **Unsupervised Learning**: It tries to find hidden structure in unlabelled data, which distinguishes unsupervised learning from supervised learning [44]. For example, clustering can be considered as unsupervised learning algorithms, where pre-labelled data is not necessary [48].

Supervised learning algorithms require pre-labelled data. Labelled data are rare and difficult to find. However, when pre-labelled data is available, the unseen data cannot be mapped which are not present in the labelled data, such as zero day attacks in the intrusion detection domain [24]. Inspired by this

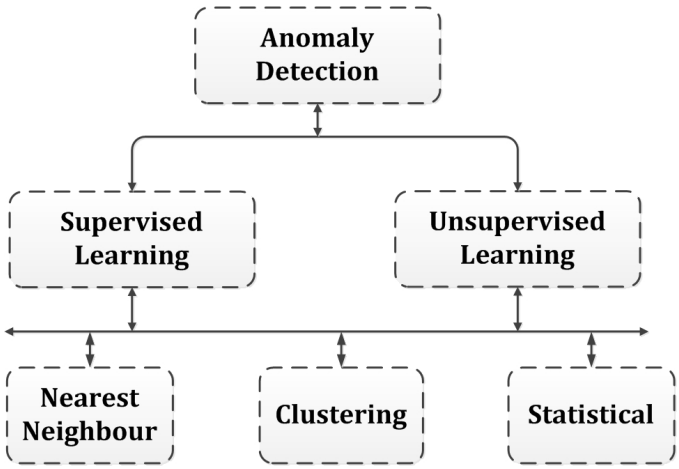

Figure 3. Taxonomy of Anomaly Detection Techniques

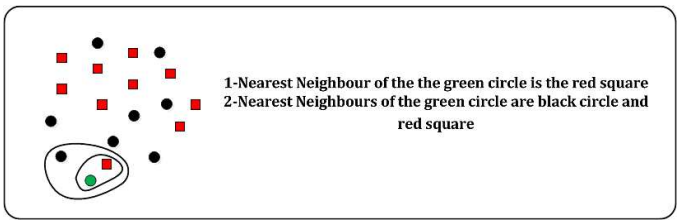

Figure 4. How *k-NN* works

Algorithm 1: Basic *k-NN* Algorithm

Input: $D = \{(x_1,c_1),....,(x_N,c_N)\}$
$x = (x_1,....,x_N)$ *new instance to be classified.*

Begin
  for each labelled instance (x_i,c_i)
   Calculate $d(x_i,x)$, the distance from x_i to x
   Order $d(x_i,x)$ from lowest to highest, $(i=1,.....,N)$
   Let D_x^k be the k-nearest instances to x
   Label x by the most frequent label in D_x^k
  end
End

fact, we emphasize unsupervised anomaly detection algorithms based on nearest neighbours, clustering and statistical approaches. Figure 3 shows a simple taxonomy for anomaly detection in the scope of this paper. The terms *anomaly* and *outlier* are used interchangeably throughout the paper.

3.1. Nearest Neighbor (NN) based Anomaly Detection and Related Works

The concept of nearest neighbor has been widely used in several anomaly detection techniques. The key assumption used in this scenario is *'Normal data instances stay in a dense neighborhoods and the anomalies stay far away from their neighbors'* [20]. Next, we present a couple of anomaly detection techniques [1] based on this idea. Figure 4 shows a simple example of *k-NN* method. The corresponding algorithm is shown in **Algorithm 1**.

Knorr et al [20] presented an algorithm to detect distance-based outliers. They consider a data point O in a dataset T a $DB(p;D)$-outlier if at least a fraction p of the data points in T lies greater than distance D from O. Their index-based algorithm executes a range search with radius D for each data point. If the number of data points in its D-*neighborhood* exceeds a threshold, the search stops and that data point is declared as a non-outlier, otherwise it is an outlier. This concept was further extended by Ramaswamy et al [11] where the anomaly score is based on the *k-nearest neighbor* implementation.

Ramaswamy et al [11] provided outlier definition based on the distance of a point from its k^{th} *nearest neighbor*. They provided a ranking of top-n outliers by the measure of the outlierness of the points. According to them, *top-n* points with the maximum distance to their own k^{th} *nearest neighbor* are considered as outliers. They also exploited index-based and nested-loop algorithms to detect outliers. Furthermore, they proposed a partition-based algorithm to prune and process the partitioned groups to improve efficiency for outlier detection. Their algorithm reduces the cost of computation in large, multidimensional data sets.

Breunig et al [21] proposed to assign each object a degree of being outlier. This degree is called the Local Outlier Factor (LOF). LOF depends on how isolated the object is with respect to the surrounding neighbourhood. The local outlier factor of an object p is calculated using the equation (2), where *MinPts* defines the minimum number of points as a notion of density and *lrd* is the local reachability density (1). (For more details on the mathematical terms please see [21].

$$reach-dist_k(p,o) = \{k-distance(o), d(p,o)\} \quad (1)$$

$$LOF_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \quad (2)$$

This outlier factor of object p calculates the degree to which p can be called as outlier. The outlier factor is the average of the ratio of the local reachability density (lrd) of p and those of p's *MinPts-nearest neighbours*. The author also described mathematically the LOF for objects deep in a cluster along with general bounds (upper, lower, and tight). The **Theorem 1** depicts a general upper and lower bound on *LOF(p)* for any data object p. For the theorem, following terms are necessary.

- $direct_{min}(p)$ = min{reach-dist(p,q)|r∈$N_{MinPts}(p)$}
- $direct_{max}(p)$ = max{reach-dist(p,q)|r∈$N_{MinPts}(p)$}
- $indirect_{min}(p)$ = min{reach-dist(q,o)|q∈$N_{MinPts}(p)$ and o∈$N_{MinPts}(q)$}
- $indirect_{max}(p)$ = max{reach-dist(q,o)|q∈$N_{MinPts}(p)$ and o∈$N_{MinPts}(q)$}

Theorem 1: When p is a data object from the dataset D and $1 \leq MinPts \leq |D|$. Then the *LOF(p)* can be represented by equation (3) [21].

$$\frac{direct_{min}(p)}{indirect_{max}(p)} \leq LOF(p) \leq \frac{direct_{max}(p)}{indirect_{min}(p)} \quad (3)$$

Proof:

Left hand side: $\frac{direct_{min}(p)}{indirect_{max}(p)} \leq LOF(p)$. Following the terms defined above,

$$LOF(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{lrd(o)}{lrd(p)}}{|N_{MinPts}(p)|} \geq \frac{\sum_{o \in N_{MinPts}(p)} \frac{\frac{1}{indirect_{max}(p)}}{\frac{1}{direct_{min}(p)}}}{|N_{MinPts}(p)|} = \frac{direct_{min}(p)}{indirect_{max}(p)} \quad (4)$$

Right hand side: $LOF(p) \leq \frac{direct_{max}(p)}{indirect_{min}(p)}$: analogously proved.

Jin et al [36] proposed an approach for mining only *top-n* local outliers because the LOF [21] values for every data object require a large number of *k-nearest neighbour* searches and can be very computationally expensive. They proposed an efficient microcluster-based local outlier mining algorithm to find the *top-n* local outliers in a large database. A microcluster *MC (n, c, and r)* is a summarized representation of a group of data $p_1,,p_n$, which are so close together that they are likely to belong to the same cluster. Here, $c = \frac{\sum_{i=1}^{n} p_i}{n}$, is the mean center while $r = max(d(p_i,c))$, $i = 1,...,n$, is the radius. Data are compressed into small clusters, and small clusters are represented using some statistical information as microclusters. Three different algorithms are combined to find *top-n* local outliers. First, *k-distance* bounds for each microcluster are computed. Then using these *k-distance* bounds, the LOF bounds are calculated. Finally, given an upper bound and a lower bound for the LOF of each microcluster, *top-n* local outliers are ranked.

He et al [26] introduced a new definition for outlier, the semantic outlier. A semantic outlier is a data point that behaves differently from the other data points in the same class. A measure for identifying the degree of each object being an outlier is presented, which is called the semantic outlier factor (SOF). To mine semantic outliers, an algorithm is also proposed. They used a *SQUEEZER* algorithm, which is used to produce good clusters for categorical datasets, and then used their algorithm to calculate the SOF value for each of the objects. Their proposed outlier definition works by identifying the similarity between a specific set and a record. Given a set of records R and a record t, the similarity between R and t is defined as follows:

$$Sim(t,R) = \frac{\sum_{i=1}^{|R|} similarity(t,t_i)}{|R|} \; where \; \forall \; t_i \in R \quad (5)$$

The semantic outlier factor of a record t is defined as in equation (6).

$$SOF(t) = \frac{pr(cl_i|C_K) * Sim(t,R)}{pr(cl_i|D)} \quad (6)$$

Spiros et al [33] introduced local correlation integral (LOCI) for evaluating outlierness, which is very efficient in detecting outliers and groups of outliers. The main advantage of this approach is an automatic data-dictated cut-off to determine whether a point is an outlier. They introduced the multigranularity deviation factor (MDEF), which at radius r for a point p_i is the relative deviation of its local neighborhood density from the average local neighborhood density in its neighborhood.

Zhange et al [17] proposed a new outlier detection definition, local distance-based outlier factor (LDOF), which is sensitive to outliers in scattered datasets (Figure 5). LDOF uses the relative distances from an object to its neighborhood to measure how much objects deviate from their scattered neighborhood. The higher the violation degree an object has, the more likely the object is an outlier. The local distance-based outlier factor of p_i is defined in equation (7) where $\overline{d}_{p_i}$ the *k-nearest neighbors* are the distance of object p_i and $\overline{D}_{p_i}$ is the *k-nearest neighbor* inner distance of p_i .

$$LDOF_k(p_i) = \frac{\overline{d}_{p_i}}{\overline{D}_{p_i}} \quad (7)$$

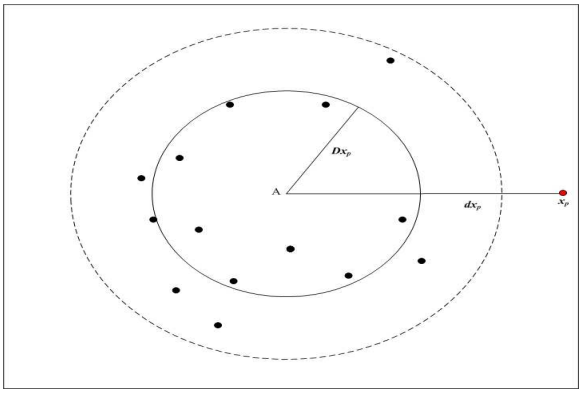

Figure 5. The explicit outlierness of object p_i with the help of LDOF definition. **A** is the center of the neighborhood system of p_i. The dashed circle includes all neighbors of p_i. The solid circle is reformed neighborhood region of p_i. Adapted from [1]

Kriegel et al [47] formulated a local density-based outlier detection method providing an outlier score in the range of [0,1] that is directly interpretable as a probability of a data object for being an outlier. The probabilistic local outlier factor (*PLOF*) of an object $o \in D$ w.r.t. a significance λ and a context set $S(o)$, can be defined as follows in equation (8). To achieve a normalization making the scaling of *PLOF* independent of the particular data distribution, the aggregate value *nPLOF* (9) is obtained during *PLOF* computation.

$$PLOF_{\lambda,S}(o) = \frac{pdist(\lambda, o, S(o))}{E_{s\in S(o)}[pdist(\lambda, s, S(s))]} \tag{8}$$

$$nPLOF = \lambda.\sqrt{E[(PLOF)^2]} \tag{9}$$

Finally, Local outlier probability (*LoOP*) (10), indicating the probability that a point $o \in D$ is an outlier. In equation (10) *erf* is the *Gaussian error function*.

$$LoOP_s(o) = max\left\{0, erf\left(\frac{PLOF_{\lambda,S}(o)}{nPLOF.\sqrt{2}}\right)\right\} \tag{10}$$

3.2. Clustering based Anomaly Detection and Related Works

As discussed earlier that anomaly deviates from the regular characteristics of the data. Consequently, the goal of clustering is to group together similar data and it is used to detect anomalous patterns in a dataset [40]. There are three key assumptions when using clustering to detect anomalies [24]:

1. **Assumption 1:** Once the clusters are created, any new data that do not fit well with existing clusters of normal data are considered as anomalous. For example, if we consider density based clustering algorithms [48] such as *DBSCAN*, we find that it does not include noise inside the clusters. As a result, noise is considered anomalous. For example, in the Figure 6, *C1* and *C2* are clusters containing normal instances and *A1*, *A2* are anomalies.

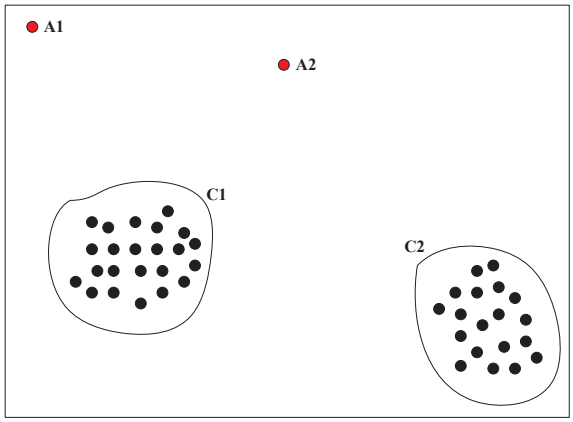

Figure 6. Example of anomaly based on assumption 1

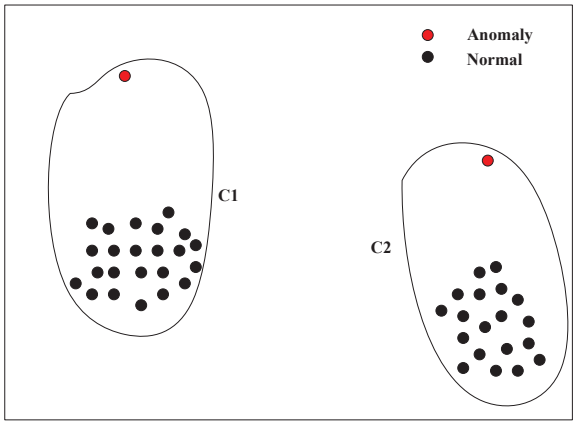

Figure 7. Example of anomaly based on assumption 2

2. **Assumption 2:** In some cases, a cluster contains both normal and anomalous data. It is expected that normal data lie close to the nearest cluster centroid and anomalies are far away from the centroids (Figure 7). Based on this assumption, anomalies are detected using a distance score.
In [40], the authors considered an outlier according to distance of a data instance from the centroid. If the distance is a fixed multiple of mean distances of all other data points from the centroid then it is considered as an outlier. Formally, *'an object in a set of data is an outlier if the distance between the object and the centroid of the dataset is greater than multi times the mean of the distances between centroid and other objects in the dataset'* [40]. They also showed that removing outliers from clusters can significantly improve

clustering objective function.
Svetlona et al [41] presented an outlier removal clustering algorithm (ORC) that provides outlier detection and data clustering simultaneously. Their proposed algorithm has two stages. First, the *k-means* clustering is applied and then *outlyingness factor* o_i for each of the data points, p_i is calculated by taking the ratio of a point's distance to the centroid C and the maximum distance, d_{max} from the centroid to any other point, stated in equation (11). If outlying factor for any point is greater than a threshold T, it is considered as an outlier and removed from dataset. Their experimental data includes synthetic data and some map images. Mean Absolute Error (MAE) is used to evaluate their algorithm performance.

$$o_i = \frac{\|p_i - C\|}{d_{max}} \tag{11}$$

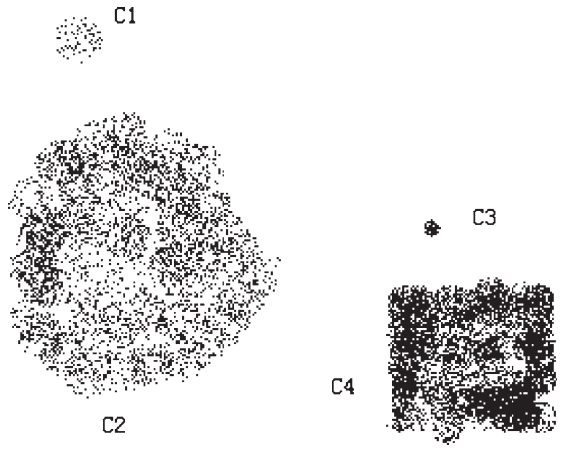

Figure 8. Anomalous clusters *C1,C3*; adapted from [22]

3. **Assumption 3:** In this scenario, it is assumed that in a dataset normal data objects are significantly high in volume than the anomalies. As a result, after clustering the dataset, smaller and sparser clusters are considered as anomalous and thicker clusters are normal. The instances belonging to clusters whose size and/or density is below a threshold are considered anomalous.
He et al [22] proposed a definition for cluster based local anomalies. According to their definition, all the data points in a certain cluster are considered as anomalies rather than a single point, as shown in Figure 8. The clusters *C1* and *C3* are considered as anomalous. They used some numeric parameters, i.e. α, β to identify Small Cluster (SC) and Large Cluster (LC). The clustering technique depends on these parameters but it is not clear how the values can be determined for various datasets. They used the *SQUEEZER* algorithm to cluster data, as it achieves both high quality of clustering and can handle high dimensional data. Then the *FindCBLOF* algorithm determines outlier factor of each individual record in dataset (shown in **Algorithm 2**). CBLOF(t) for each record t is calculated following equation (12):

Algorithm 2: CBLOF Algorithm
Input: Dataset, D
The Parameters, α, β
Output: CBLOF score
Begin
Cluster the Dataset, D
Clusters: $C=\{C_1, C_2, .., C_k\}$ *and* $|C_1| \geq |C_2| \geq ... \geq |C_k|$
Calculate *LC* and *SC* with the α, β
Let C_i be the cluster containing t
  if $C_i \in SC$ **do**
  CBLOF $= |C_i| * \min\{d(t, C_j) | C_j \in LC\}$
  else
  CBLOF $= |C_i| * d(t, C_i)$
End

$$CBLOF(t) = \begin{cases} |C_i| * min(d(t, C_j)) \text{ where } t \in C_i, C_i \in SC \\ \quad \text{and } C_j \in LC \text{ for } j = 1 \text{ to } b \\ |C_i| * (d(t, C_i)) \text{ where } t \in C_i \\ \quad \text{and } C_i \in LC \end{cases} \tag{12}$$

Amer et al [14] introduced Local Density Cluster-Based Outlier Factor (LDCOF) which can be considered as a variant of CBLOF [22]. The LDCOF score (16) is calculated as the distance to the nearest large cluster divided by the average distance to the cluster center of the elements in that large cluster. LDCOF score will be **A** when $p \in C_i \in$SC where $C_j \in$LC and **B** when $p \in C_i \in$LC.

$$distance_{avg}(C) = \frac{\sum_{i \in C} d(i, C)}{|C|} \tag{13}$$

$$A = \frac{min(d(p, C_j))}{distance_{avg}(C_j)} \tag{14}$$

$$B = \frac{d(p, C_i)}{distance_{avg}(C_i)} \tag{15}$$

$$LDCOF(p) = A \mid B; \tag{16}$$

Jiang et al [34] presented a two-phase clustering technique to detect outliers. First, they used a modified *k-means* algorithm to create clusters. If the points in the same cluster are not close enough, the cluster can be split into two smaller clusters and merged when a given threshold exceeds. In the second step, they construct a

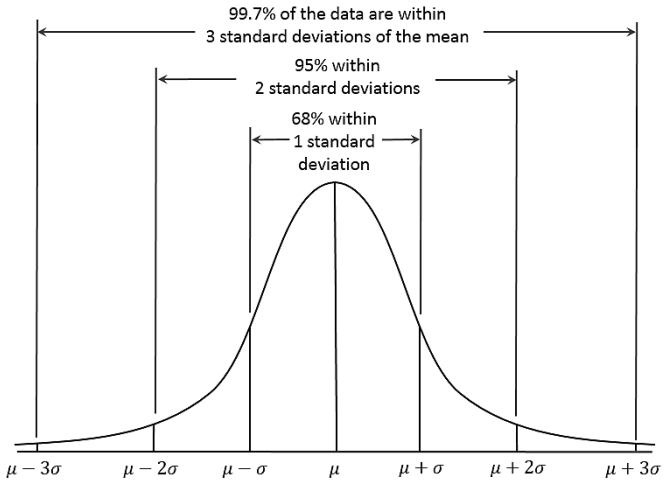

Figure 9. Concept of statistical anomaly detection, adapted from Internet

minimum spanning tree with the cluster centres and remove the longest edge. The smaller sub trees are considered outliers. Their technique considers an entire cluster as an outlier, which may not be applicable for many datasets and may increase *False Positive rate*.

Cluster Based Outlier Detection (*CBOD*) [37] is another technique which consists of two stages. In the first stage, it generates clusters from a given dataset and in the second stage it computes outlier factor as the weighted sum of distances between a particular cluster and rest of the clusters. The outlier factor of cluster C_i, OF(C_i) is defined as the *weighted sum* of distances between cluster C_i and the rest of the clusters. The outlier factor $OF(C_i)$ measures the outlier degree of cluster, the bigger the value is, the bigger the possibility of being an outlier cluster.

$$OF(C_i) = \Sigma_{j \neq i}(C_j) * d(C_i, C_j) \tag{17}$$

Minimum b clusters which satisfy the criteria as follows are labelled as outlier clusters. They used detection rate and false alarm rate to measure performance.

3.3. Statistical based Anomaly Detection and Related Works

The statistical approaches discussed here are considered as the first generation techniques for anomaly detection. Figure 9 portrays the the most commonly used $\mu \pm 3\sigma$ rule for detecting anomalous data. A normally distributed data follows a bell curve and can be mathematically represented in equation (18). Here, μ stands for the mean or average, σ is the standard deviation and σ^2 is the variance. When the μ=0 and $\sigma = 1$, the distribution is called standard normal distribution. The data with values greater than $\mu + 3\sigma$ or less than $\mu - 3\sigma$ is considered anomalous.

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{18}$$

These techniques are also named as model-based techniques. Models are based on probability distribution of the data and anomalies are detected as how well the data fit into the model. Statistical based approaches are categorized into two groups depending on probability distribution as follows:

- **Parametric Approaches:** In these approaches the probability distribution of the data is known (supervised). Then, using the distribution parameters, anomalies are detected. A point is an anomaly if it deviates significantly from the data model. However in many situations prior knowledge of distribution is not possible to attain. As a result, supervised learning techniques are not preferred over the unsupervised learning techniques instead of having less accuracy.

  Wu et al [30] proposed two algorithms for outlying sensors and event boundary detection. The basic idea of outlying sensor detection is as such, each sensor first computes the difference between its reading and the median of the neighboring readings. Each sensor then collects all differences from its neighborhood and standardizes them. A sensor is an outlier if the absolute value of its standardized difference is sufficiently large. The algorithm for event boundary detection is based on the outlying sensor detection algorithm. For an event sensor, there often exist two regions, with each containing the sensor, such that the absolute value of the difference between the reading of the sensor and the median reading from all other sensors in one region is much larger than that in another region. These approaches are not effective because they do not consider the temporal correlation of sensor readings [1].

  Bettencourt et al [29] proposed an anomaly detection technique to identify anomalous events and errors in ecological applications of distributed sensor networks. This method uses spatio-temporal correlation of sensor data to distinguish erroneous measurements and events. A measurement is considered anomalous when its value in the statistical significance test is less than user specified threshold. The disadvantage of this approach is dependence on the user specified threshold [1].

  Jun et al [31] presents a statistical based approach,

which uses alpha-stable distribution. The proposed algorithm consists of collaborative time-series estimation, variogram application and principle component analysis (PCA). Each node detects any temporally abnormal data and transmits the verified data to a local cluster-head, which detects any survived spatial outlier and determines the faulty sensors accordingly. Their approach achieves 94% accuracy when the noise level is alpha = 0.9. Although alpha-stable distribution might be considered for real sensor data and cluster based structure may be susceptible to dynamic changes of network topology [1].

- **Non Parametric Approaches:** These approaches have no knowledge about the underlying data distribution like unsupervised learning methods. A distance measure is used to identify anomalies in this scenario. Anomalies are those points which are distant from their own neighborhood in a dataset. Various detection techniques are available with a wide range of parameters. They resemble anomaly detection using clustering based assumption 2. Parametric methods are not flexible enough like non-parametric methods but due to dimensionality and computational complexity the efficiency might deteriorate in some cases. There are two widely used approaches in this category are discussed as follows-

  - **Histogramming:** This model counts the frequency of occurrence of different data instances and compares the test instance with each of histogram categories to test whether it belongs to any of them [18].
    Sheng et al [32] proposed a histogram-based technique for anomaly detection to reduce communication cost for data collection applications of sensor networks. Rather than collecting all the data in one location for centralized processing, they propose collecting hints about the data distribution and using the hints to filter out unnecessary data and identify potential anomalies. Main drawbacks of this technique are communication overhead and one dimensional data [1].
  - **Kernel Function:** This function is used to estimate the probability distribution function (pdf) of the normal instances. Data instances which lie in the low probability area of pdf are declared as anomalies.
    Palapans et al [15] proposed a technique for online deviation detection in streaming data. They discussed how their technique can be operated efficiently in the distributed environment of a sensor network. In the sensor data, a value is considered as an anomaly if the number of values being in its neighborhood is less than a user specified threshold. This technique can also be implemented for identification for of anomalies in a more global perspective [1].

4. Criteria for Benchmarking Anomaly Detection Algorithms

This section provides a discussion on the key aspects to evaluate anomaly detection algorithms in terms of big data. We propose the following points to be considered while selecting the benchmark anomaly detection techniques in SCADA systems:

- **Size of the Data (Volume)**: Size is an important factor for anomaly detection algorithms. More importantly, in case of big data, it is a crucial parameter to measure the efficiency of the anomaly detection algorithm. Some anomaly detection technique might work well on small dataset but perform poorly on big data and vice-versa!
- **Dimensionality**: It is closely related with the computing efficiency of any data mining techniques. It is quite common that big data has high dimensionality and as the dimensionality increases the data become sparse. As a result similarity/dissimilarity calculation at this situation is challenging.
- **Type of Data**: Handling identical data type and mixed type is completely different. For example, handling only numerical data for anomaly detection is more computationally efficient than dataset with numerical, categorical and binary type of data. Also, in case of big data, it is an important issue to consider the efficiency of the anomaly detection.
- **Velocity**: This criterion deals with complexity of the anomaly detection algorithms.
- **Input Parameter**: Selecting the best possible parameters for any algorithm is a challenge. It is more challenging when input parameters required for big data. A non-optimal value of input parameter causes computational burden. Also more the number of input parameters more it gets complex. In unsupervised fashion, it is also a challenge to provide the best parameter values to the anomaly detection techniques. So, less is better in this case.

In Table 1, we showcase the characteristics of anomaly detection algorithms based on the criterion

Table 1. Characteristics of anomaly detection algorithms

Category	Size	Dimensionality	Variety	Velocity	Input Parameters
NN	Large	High	Yes	High	≥ 2
Clustering	Large	Low and High	Yes	Medium	≥ 2
Statistical	Large	High	Yes	High	≥ 1

discussed above. It is evident that each category has the ability to handle a large volume of data. However, clustering based techniques have greater computational complexity than the others. Also, statistical techniques are better in terms of selection of input parameters.

4.1. Strength and Weakness

We highlight the merits and demerits of the anomaly detection techniques discussed in Section 3.

Nearest Neighbour Techniques: The main advantage of nearest neighbour based techniques is their unsupervised characteristics. However, when anomalies have a large number of close neighbours, it is not possible to identify them correctly. Also, the distance computation requires significant computation and it becomes more complex when the data has mixed type of data such as numerical, categorical, binary etc.

Clustering Techniques: The techniques used to detect anomalies in binary fashion are computationally efficient irrespective of the clustering algorithm since each object in dataset is not required to assign an outlying factor like scoring based output. The *top-N* anomaly concept is absent in these techniques and hence are unsupervised. The main drawback of these techniques is inaccuracy of detecting all the rare class instances. Since not all the data objects are taken into consideration for being outlier, many of them might be missing and normal instances may be detected as anomalies.
The scoring based techniques have the maximum effectiveness in detecting anomaly accurately since all the objects are under consideration as candidate anomalies. But the loophole of these techniques is computational cost. Since all the objects are taken under consideration to assign outlyingness factor. *Top-N* anomalies must have to be specified by data analyst and thus the approach becomes supervised.

Statistical Techniques: Statistical approaches come with strong mathematical background to detect anomalies. But parametric approaches are not feasible when the prior knowledge on the data distribution is not available and hence quite useless in many aspects. In comparison, non-parametric methods are quite useful since the data distribution knowledge is not required. However, these methods might have high computational complexity for high dimensional datasets. Also user-defined parameters are not easy to set.

Table 2. Characteristics of the SCADA datasets

Dataset	Normal	Anomaly
Urban Waster Water Treatment Plant (WTP)	97.5%	2.5%
Single-hop Indoor (SI)	97.35%	2.65%
Single-hop Outdoor (SO)	99.37%	0.63%
Simulated-Data1 (Sim1)	99.02%	0.98%
Simulated-Data2 (Sim2)	99.05%	0.95%
Multi-hop Indoor (MI)	97.86%	2.14%
Multi-hop Outdoor (MO)	98.76%	1.24%

5. Experimental Evaluation on SCADA Systems Big Data

This section starts with a brief discussion on the datasets used. Then we discuss about the evaluation metrics used in the paper. Finally, we showcase the evaluation results showing in figures and tables.

5.1. SCADA Datasets used in this paper

Table 2 contains the description of the characteristics of some of the common SCADA datasets widely used [28]. Figure 10 displays a simple taxonomy of anomalous scenarios in SCADA systems. There are three major categories of anomalies based on the datasets used in this paper. The real anomalies are from water treatment plant. The simulated anomalies are designed by computer software. In real sensor nodes, the anomalies are injected by creating changes in temperature.

The real anomalies in the *WTP* dataset [35] are caused by the inclement weather. It contains data of the daily measures of sensors in a urban waste water treatment plant. Solid overload caused by stormy

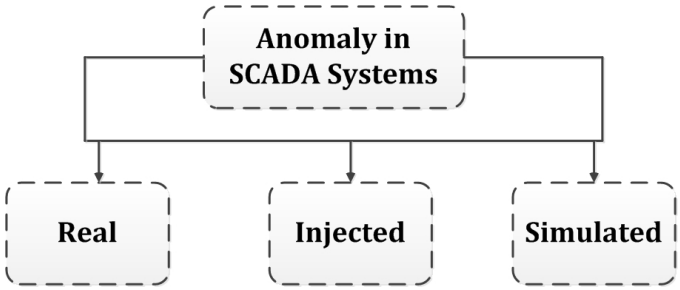

Figure 10. Taxonomy of anomaly in SCADA systems

weather are considered anomalous data in the system.

The simulated anomalies in the *Sim1* and *Sim2* contain man-in-the-middle attacks [38]. Here a water distribution system is simulated using the EPANET library [46]. Anomalies were created using the *man-in-the-middle* attacks. In this scenario, water pumps were turned off when the reserve in the tanks are low.

In the single-hop, multi-hop (indoor and outdoor) datasets, anomalies are injected [45]. For the single-hop scenario, two indoor and two outdoor sensor nodes are used to collect the temperature and humidity data for six hours. Anomalies are introduced by using a kettle of hot water at one of the sensors. The simultaneous raise in the temperature and humidity is considered anomalous in this scenario. In the multi-hop situation, multi-hop routing is used to create a larger sensor network. Like single-hop datasets, anomalies are introduced using the hot water at the temperature and humidity sensors.

5.2. Evaluation Measures

We measure the performance of the anomaly detection algorithms using the standard evaluation criteria [1]. These are briefly discussed here. All of them share some common concept of confusion matrix. The 2×2 matrix contains the number of True Positive (TP), False Positive (FP), True Negative (TN), False negative (FN). Table 3 displays the confusion matrix.

TP: No. of anomalies correctly identified as anomalous.

FP: No. of normal data incorrectly identified as anomalous.

TN: No. of normal data correctly identified as normal.

FN: No. of anomalies incorrectly identified as normal.

Listed below are the five evaluation measures based on confusion matrix.

- **Accuracy** - The accuracy is computed using equation (19).

Table 3. Standard confusion metrics for evaluation of anomaly detection algorithm

Label	Normal	Anomaly
Normal	TN	FP
Anomaly	FN	TP

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

- **FPR** - False Positive Rate also named as *FPR* is another metric which is the proportion of non-relevant data that are retrieved, out of all non-relevant data available. The lower the value is better the anomaly detection technique is. Equation (20) shows the way to calculate *FPR*.

$$FPR = \frac{FP}{FP + TN} \tag{20}$$

- **Recall** - Recall is the fraction of the data that are relevant to the query that are successfully retrieved. In the case of anomaly detection, *recall* is also known as *TPR*, *Hit Rate*, can be calculated using (21).

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

- **F-1** - *F-1* score is the harmonic mean of precision ($TP/TP + FP$) and recall. Equation (22) shows the way to calculate *F-1*.

$$F\text{-}1 = \frac{2TP}{2TP + FP + FN} \tag{22}$$

- **MCC** - The Matthews correlation coefficient is a popular measure in machine learning to identify the quality of binary (two-class) classifications. It considers the true and false positives and negatives for calculating the measure. The *MCC* provides a value between -1 and +1. A *MCC* score of +1 represents a perfect anticipation and -1 indicates complete opposite scenario between

observation and prediction (23).

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (23)$$

Last but not least, we also consider the run time (in seconds) as an important evaluation criteria for anomaly detection algorithms.

5.3. Experimental Results

This section contains the performance analysis of anomaly detection techniques based on the evaluation measures discussed in the previous section. For simplicity, we scale all the metrics between 0 and ±100. The representative algorithms are the following and standard values are considered for the input parameters for all the techniques:

- **Nearest Neighbour:**
  - **k-NN**: Each data instance is given score for being anomalous based on the average distance to the nearest neighbours [11].
  - **LOF**: LOF provides anomaly score to the data instances based on the local density of the data points [21].
  - **COF**: The connectivity based outlier factor is a modification of the *LOF* approach which can handle outliers deviating from low density patterns [43]
  - **aLOCI**: Calculates the outlier score based on local correlation integral [33].
  - **LoOP**: The LoOP score represents the probability that the object is a local density outlier [47].
  - **INFLO**: Calculates the outlier score based on Influenced outlierness, proposed by Jin et al [19].
- **Clustering:**
  - **CBLOF**: CBLOF creates clusters from the given dataset and then it categorizes the clusters into small clusters and large clusters using the parameters α and β. The anomaly score is then calculated based on the size of the cluster the point belongs to as well as the distance to the nearest large cluster centroid [22].
  - **LDCOF**: This local density based anomaly detection algorithm sets the anomaly score based on the distance to the nearest large cluster divided by the average cluster distance of the large cluster [14].
  - **CMGOS**: This method calculates the anomaly score based on a clustering result. The outlier score of an instance is dependent on the probability of how likely its distance to the cluster center is [14].
- **Statistical:**
  - **HBOS**: Calculates an outlier score by creating an histogram with a fixed or a dynamic binwidth [18].
  - **LIBSVM**: Computes the outlier score using one-class SVMs [42]. This operator extends the semi-supervised one-class SVM such that it can be used for unsupervised anomaly detection.

Table 4. Performance of Anomaly Detection Techniques on Real SCADA Dataset (WTP: Water Treatment Plant)

Technique	Recall	FPR	Accuracy	F-1	MCC	Run Time
k-NN	85.71	0.38	97.39	85.71	85.32	≤1
LOF	78.57	0.58	97.38	78.57	77.98	≤1
COF	57.14	1.16	97.35	57.14	55.97	≤1
aLOCI	85.71	0.38	97.39	85.71	85.32	≤69
LoOP	42.85	1.55	97.33	42.85	41.29	≤1
INFLO	57.14	1.16	97.35	57.14	55.97	≤1
CBLOF	92.85	0.19	97.40	92.85	92.66	≤1
LDCOF	85.71	0.38	97.39	85.71	85.32	≤1
CMGOS	57.14	1.16	97.35	57.14	55.97	≤1
HBOS	28.57	1.94	97.32	28.57	26.62	≤1
LIBSVM	85.71	0.38	97.39	85.71	85.32	≤1

Table 5. Performance of Anomaly Detection Techniques on Simulated SCADA Datasets

Results on Sim1 Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	64.70	0.34	99.03	64.70	64.35	≤4
LOF	0	0.98	99.01	0	-0.98	≤4
COF	0	0.98	99.01	0	-0.98	≤5
aLOCI	0	0.98	99.01	0	-0.98	≤10
LoOP	0.98	0.97	99.01	0.98	0.009	≤4
INFLO	0	0.98	99.01	0	-0.98	≤3.5
CBLOF	0	0.98	99.01	0	-0.98	≤2
LDCOF	0	0.98	99.01	0	-0.98	≤2
CMGOS	18.62	0.79	99.02	18.62	17.82	≤2
HBOS	30.39	0.682757957	99.02	30.39	29.70	≤2
LIBSVM	74.50	0.25	99.03	74.50	74.25	≤322
Results on Sim2 Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	63	0.35	99.05	63	62.64	≤3
LOF	0	0.96	99.03	0	-0.96	≤4
COF	2	0.94	99.03	2	1.05	≤3
aLOCI	0	0.96	99.03	0	-0.96	≤23
LoOP	0	0.96	99.03	0	-0.96	≤4
INFLO	0	0.96	99.03	0	-0.96	≤4
CBLOF	0	0.96	99.03	0	-0.96	≤2
LDCOF	0	0.96	99.03	0	-0.96	≤4
CMGOS	97	0.02	99.05	97	96.97	≤2
HBOS	27	0.70	99.04	6	7.31	≤1
LIBSVM	68	0.30	99.05	68	67.69	≤220

We categorize the performance of the anomaly detection algorithms based on the taxonomy of anomaly in SCADA systems (Figure 10). For the real

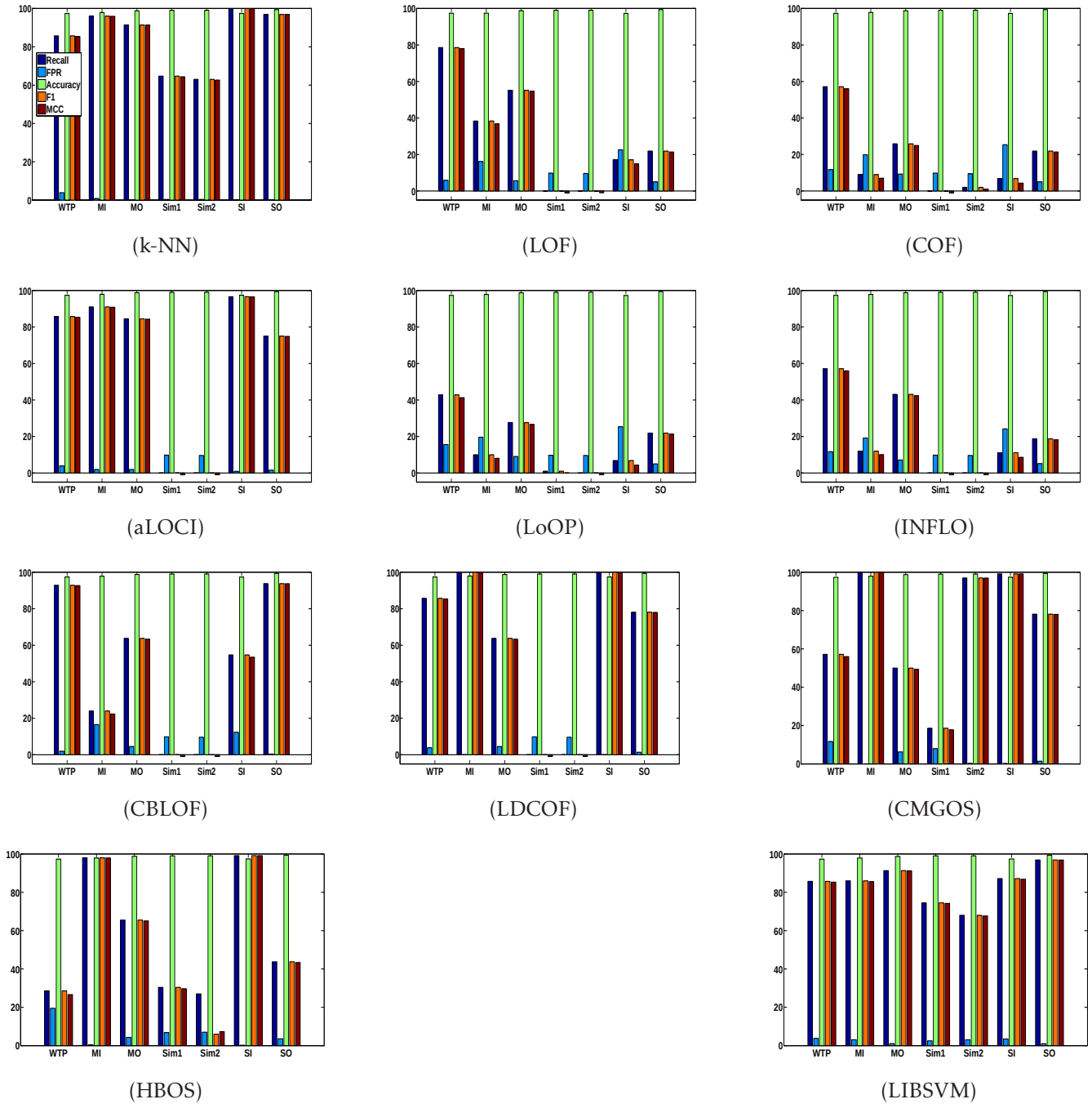

Figure 11. Performance of anomaly detection techniques on SCADA datasets, scaled to 0±100

SCADA dataset *WTP*, from Table 4 it is evident that the clustering based anomaly detection technique *CBLOF* performs best and second best performance is attained by the nearest neighbour based technique *k-NN*. Statistical based approach *HBOS* does not perform well here.

For the simulated datasets, it is surprising that semi-supervised anomaly detection technique *LIBSVM* has better *recall* than others, however suffers from unacceptable run time. On the other hand, nearest neighbour based method *k-NN* has very low run time complexity and acceptable *recall*. Clustering based approaches are not well suited for the simulated datasets here and statistical approach *HBOS* outperforms clustering techniques. Table 5 displays the results on simulated datasets.

Table 6. Performance of Anomaly Detection Techniques on Datasets with Injected Anomalies

Results on Multi-hop Indoor (MI) Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	96	0.08	97.90	96	95.91	≤1
LOF	38.33	1.61	97.43	38.33	36.72	≤1
COF	9	1.98	97.82	9	7.01	≤1
aLOCI	91	0.19	97.90	91	90.80	≤11
LoOP	10	1.96	97.83	10	8.03	≤1
INFLO	12	1.91	97.83	12	10.08	≤1
CBLOF	24	1.65	97.84	24	22.34	≤1
LDCOF	100	0	97.91	100	100	≤1
CMGOS	100	0	97.91	100	100	≤1
HBOS	98	0.04	97.91	98	97.95	≤1
LIBSVM	86	0.30	97.90	86	85.69	≤39
Results on Multi-hop Outdoor (MO) Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	91.37	0.10	98.77	91.37	91.27	≤1
LOF	55.17	0.56	98.76	55.17	54.61	≤1
COF	25.86	0.92	98.75	25.86	24.93	≤1
aLOCI	84.48	0.19	98.77	84.48	84.28	≤13
LoOP	27.58	0.90	98.75	27.58	26.67	≤1
INFLO	43.10	0.71	98.76	43.10	42.39	≤1
CBLOF	63.79	0.45	98.76	63.79	63.33	≤1
LDCOF	63.79	0.45	98.76	63.79	63.33	≤1
CMGOS	50	0.62	98.76	50	49.37	≤1
HBOS	65.51	0.43	98.76	65.51	65.08	≤1
LIBSVM	91.37	0.10	98.77	91.37	91.27	≤39

Table 7. Performance of Anomaly Detection Techniques on Datasets with Injected Anomalies (Single-Hop)

Results on Single-hop Indoor (SI) Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	100	0	97.41	100	100	≤1
LOF	17.09	2.25	97.30	17.09	14.83	≤1
COF	6.83	2.53	97.28	6.83	4.30	≤1
aLOCI	96.58	0.09	97.41	96.58	96.48	≤114
LoOP	6.83	2.53	97.28	6.83	4.30	≤9
INFLO	11.11	2.41	97.29	11.11	8.69	≤14
CBLOF	54.70	1.23	97.35	54.70	53.46	≤1
LDCOF	100	0	97.41	100	100	≤1
CMGOS	99.14	0.02	97.41	99.14	99.12	≤1
HBOS	99.14	0.02	97.41	99.14	99.12	≤1
LIBSVM	87.17	0.34	97.40	87.17	86.83	≤22
Results on Single-hop Outdoor (SO) Dataset						
Technique	**Recall**	**FPR**	**Accuracy**	**F-1**	**MCC**	**Run Time**
k-NN	96.87	0.01	99.36	96.87	96.85	≤2
LOF	21.87	0.49	99.36	21.87	21.37	≤1
COF	21.87	0.49	99.36	21.87	21.37	≤1
aLOCI	75	0.15	99.36	75	74.84	≤16
LoOP	21.87	0.49	99.36	21.87	21.37	≤2
INFLO	18.75	0.51	99.36	18.75	18.23	≤2
CBLOF	93.75	0.03	99.36	93.75	93.71	≤1
LDCOF	78.12	0.13	99.36	78.12	77.98	≤1
CMGOS	78.12	0.13	99.36	78.12	77.98	≤1
HBOS	43.75	0.35	99.36	43.75	43.39	≤1
LIBSVM	96.87	0.01	99.36	96.87	96.85	≤16

For the datasets with injected anomalies in multi-hop senario, we found the performance (Table 6) of clustering based approaches is the best considering the evaluation measures. Nearest neighbour based approaches are the next best. Among the *HBOS* and *LIBSVM* approach, the latter has the better results in terms of anomaly detection but attains high computational burden (run time).

Finally, for the datasets in single-hop scenario, it is seen that, the clustering-based methods perform consistently well, but the nearest neighbour methods are quite variable (Table 7). *LIBSVM* performs better than *HBOS* but still suffers from high run time complexity.

It is interesting to observe that, for all the anomaly detection techniques the *Recall* and *F-1* values are identical. Since, top N anomalies detected by the techniques are matched with the actual N number of anomalies in the dataset, the *Recall* and *F-1* scores will always yield exactly the same values. Finally, we summarise the performance of each of the anomaly detection techniques in Figure 11. In Table 8 we also summarize the performance on different SCADA datasets. We suggest the usage of these techniques analysing the results discussed earlier. The sign (√) indicates the affirmative gesture to apply the techniques and the sign (×) discourages the usage.

Table 8. Characteristics of anomaly detection algorithms

Category	Real	Simulated	Injected
NN	√	√	√
Clustering	√	×	√
Statistical	×	√	×

6. Conclusion and Future Works

This paper gives a detailed discussion on the popular anomaly detection techniques on SCADA systems and analysed their performance. We come to a conclusion that *nearest neighbour* and *clustering* based approaches are more suitable for SCADA systems than statistical and semi-supervised support vector machine based approaches. In future we will investigate the following:

- How to find the most suitable input parameter values?
- How to incorporate the idea of contextual anomaly in big data perspective?
- How can incorporation of multi-view clustering [16], hierarchical clustering [12] and co-clustering [13] improve the efficiency of clustering-based anomaly detection techniques?
- How to reduce the run time complexity of semi-supervised support vector machine based anomaly detection?

References

[1] M. Ahmed, A. N. Mahmood, J. Hu, Outlier detection, in: The State of the Art in Intrusion Prevention and Detection, CRC Press, USA, 2014, pp. 3–23.

[2] M. Ahmed and A. N. Mahmood, "Network traffic pattern analysis using improved information-theoretic co-clustering based collective anomaly detection," in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2014.

[3] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, 2015.

[4] I. A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, May 2014, pp. 1–5.

[5] X. Miao and D. Zhang, "The opportunity and challenge of big data's application in distribution grids," in *Electricity Distribution (CICED), 2014 China International Conference on*, Sept 2014, pp. 962–964.

[6] L. Wang, J. Zhan, C. Luo, Y. Zhu, Q. Yang, Y. He, W. Gao, Z. Jia, Y. Shi, S. Zhang, C. Zheng, G. Lu, K. Zhan, X. Li, and B. Qiu, "Bigdatabench: A big data benchmark suite from internet services," in *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, Feb 2014, pp. 488–499.

[7] S. Pandey and V. Tokekar, "Prominence of mapreduce in big data processing," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*, April 2014, pp. 555–560.

[8] Z. Zheng, J. Zhu, and M. Lyu, "Service-generated big data and big data-as-a-service: An overview," in *Big Data (BigData Congress), 2013 IEEE International Congress on*, June 2013, pp. 403–410.

[9] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *Access, IEEE*, vol. 2, pp. 652–687, 2014.

[10] C.-S. Leung, R. MacKinnon, and F. Jiang, "Reducing the search space for big data mining for interesting patterns from uncertain data," in *Big Data (BigData Congress), 2014 IEEE International Congress on*, June 2014, pp. 315–322.

[11] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec.*, vol. 29, no. 2, pp. 427–438, May 2000.

[12] A. N. Mahmood, C. Leckie, and P. Udaya, "An efficient clustering scheme to exploit hierarchical data in network traffic analysis," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, no. 6, pp. 752–767, Jun. 2008.

[13] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03, USA: ACM, 2003, pp. 89–98.

[14] M. G. Mennatallah Amer, Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer, Shaker Verlag GmbH, Aachen, 2012, pp. 1–12.

[15] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Distributed deviation detection in sensor networks," *SIGMOD Rec.*, vol. 32, no. 4, pp. 77–82, Dec. 2003.

[16] X. H. Dang and J. Bailey, "A framework to uncover multiple alternative clusterings," *Machine Learning*, pp. 1–24, 2013.

[17] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Advances in Knowledge Discovery and Data Mining*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5476, pp. 813–822.

[18] M. Hofmann and R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013.

[19] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 577–593.

[20] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proceedings of the 24rd International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 392–403.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000.

[22] Z. He, X. Xu, S. Deng, Discovering cluster based local outliers, Pattern Recognition Letters 2003 (2003) 9–10.

[23] B. Galloway and G. Hancke, "Introduction to industrial control networks," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 2, pp. 860–880, Second 2013.

[24] M. Ahmed and A. Mahmood, "Network traffic analysis based on collective anomaly detection," in *Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on*, June 2014, pp. 1141–1146.

[25] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58.

[26] Z. He, S. Deng, and X. Xu, "Outlier detection integrating semantic knowledge," in *Proceedings of the Third International Conference on Advances in Web-Age Information Management*, ser. WAIM '02. London, UK, UK: Springer-Verlag, 2002, pp. 126–131.

[27] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, Dec 2010, pp. 269–274.

[28] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *Emerging Topics in Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 267–279, Sept 2014.

[29] L. Bettencourt, A. Hagberg, and L. Larkey, "Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks," in *Distributed Computing in Sensor Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4549, pp. 223–239.

[30] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng, "Localized outlying and boundary data detection in sensor networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 8, pp. 1145–1157, Aug 2007.

[31] M. C. Jun, H. Jeong, and C.-C. J. Kuo, "Distributed spatio-temporal outlier detection in sensor networks," pp. 273–284, 2005.

[32] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier detection in sensor networks," ser. MobiHoc '07, 2007, pp. 219–228.

[33] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "Loci: fast outlier detection using the local correlation integral," in *Data Engineering, 2003. Proceedings. 19th International Conference on*, March 2003, pp. 315–326.

[34] M. Jiang, S. Tseng, and C. Su, "Two-phase clustering process for outliers detection," *Pattern Recognition Letters*, vol. 22, no. 6âĂŞ7, pp. 691 – 700, 2001.

[35] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[36] W. Jin, A. K. H. Tung, and J. Han, "Mining top-n local outliers in large databases," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 293–298.

[37] S.-y. Jiang and Q.-b. An, "Clustering-based outlier detection method," in *Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 02*, ser. FSKD '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 429–433.

[38] A. Almalawi, Z. Tari, I. Khalil, and A. Fahad, "Scadavt-a framework for scada security testbed based on virtualization technology," in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, Oct 2013, pp. 639–646.

[39] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of machine learning, The MIT Press, 2012.

[40] M. Ahmed, A. N. Mahmood, A novel approach for outlier detection and clustering improvement, in: Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, 2013, pp. 577–582.

[41] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, P. Fränti, Improving k-means by outlier removal, in: Proc. 14th Scandinavian Conference on Image Analysis (SCIA'05), 2005, pp. 978–987.

[42] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ser. ODD '13. New York, NY, USA: ACM, 2013, pp. 8–15.

[43] J. Tang, Z. Chen, A. W. Fu, and D. W. Cheung, "Capabilities of outlier detection schemes in large datasets, framework and methodologies," *Knowl. Inf. Syst.*, vol. 11, no. 1, pp. 45–84, Dec. 2006.

[44] C. M. Bishop, Neural networks for pattern recognition, Oxford University Press, Inc., New York, NY, USA, 1995.

[45] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2010 Sixth International Conference on*, Dec 2010, pp. 269–274.

[46] Lewis A. Rossman, "The EPANET Programmer's Toolkit for Analysis of Water Distribution Systems," in *Proceedings of the 29th Annual Water Resources Planning and Management Conference, 1999*, June 1999, pp. 1–10.

[47] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 1649–1652.

[48] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.