Unsupervised Anomaly Detection Based on Minimum Spanning Tree Approximated Distance Measures and Its Application to Hydropower Turbines

Imtiaz Ahmed, Aldo Dagnino, and Yu Ding, *Senior Member, IEEE*

***Abstract*— Anomalies are data points or a cluster of data points that lie away from the neighboring points or clusters and are inconsistent with the overall pattern of the data. Anomaly detection techniques help distinguish the anomalous observations from the regular ones, and thus provide the basis for developing a standard performance guideline for process control. The process of identifying anomalies becomes complicated in the absence of labeled training data as in supervised learning. Moreover, Euclidean distance between two points is less likely able to reflect the intrinsic structural distance imposed by the underlying manifold structure. In this paper, the authors propose a minimum spanning tree (MST)-based anomaly detection method. The merit of the method is that an MST provides a new distance measure, capable of capturing the relative connectedness of data points/clusters in a complicated manifold, and could be a better (dis)similarity metric, than the simple Euclidean distance, to identify anomalies in unsupervised learning settings. The proposed method is compared with 13 popular anomaly detection methods on 20 benchmark data sets, demonstrating a considerable improvement in its ability of identifying anomalies. Furthermore, the MST-based anomaly detection is applied to the data set from a hydropower turbine and demonstrates remarkable detection competence.**

***Note to Practitioners*—This paper is motivated by the problem of unsupervised anomaly detection in a hydropower generation plant, which operates with turbine systems that are instrumented with dozens of sensors. Each turbine has subcomponents or functional areas such as several bearing systems, a generator, and so on. Sensors collect various types of data in real time such as temperature of oil inside the bearing systems, temperature of the bearings, ambient temperature, vibrations in each functional areas, a variety of harmonics in functional areas, temperature of the coil in the generator, and many more. In total, each turbine collects more than 200 attributes from its sensors. The sensor data are then stored in a control system and kept as time stamped historical data points. When a service/maintenance engineer suspects that there is a malfunction in a turbine, she/he extracts a data set from the control system that contains the collected sensor data for that turbine for the selected period of time (few weeks to few months), and then stores this data in a relational databases or simply in a comma separate value (csv) file for further analysis. The objective is to efficiently identify and isolate anomalies in the turbines. Toward this goal, we propose a new solution for tackling this challenging problem, which is an unsupervised method based on the concept of MST. The proposed method can be used as a competitive tool to aid the practitioners in their search of anomalies for making their systems better.**

***Index Terms*— Geodesic distance, hydropower generation plant, minimum spanning tree (MST), process automation, unsupervised anomaly detection.**

I. Introduction

ANOMALY detection techniques are supposed to identify anomalies from loads of seemingly homogeneous data and doing so can possibly lead us to timely, pivotal, and actionable information. Detection of anomalies can be linked to numerous real-life applications including but not limited to credit card fraud detection, cyber security, medical image analysis, surveillance, and industrial process safety. Accurate and timely detection of anomalies can save us from potential human, financial, and informational loss.

There are three broad categories of anomaly detection approaches, depending on the labels of the data in a training set. *Supervised anomaly detection* comes into play when we have appropriately labeled training data in advance (both normal and abnormal) so that we can train a model based on these labeled data and use it to decide the labels of future data. Support vector machine (SVM) [1] or artificial neural network [2] are the examples of this approach. However, when we have only normal instances and no anomalous data, we can still use the normal data to train a model and classify future observations as anomalies if they deviate from the normalcy. This normal-data-only approach is known as *semisupervised anomaly detection*. One-class SVM [3] falls under this category. The most difficult scenario is the absence of any label of the data. As a result, it is not possible to conduct a supervised training. One, therefore, has to rely entirely on the structure of the data set and detect the anomalies, if any, in an unsupervised manner. This last category is known as *unsupervised anomaly detection*.

Manuscript received April 20, 2018; accepted June 11, 2018. This paper was recommended for publication by Associate Editor K. Liu and Editor L. Shi upon evaluation of the reviewers' comments. The work of I. Ahmed and Y. Ding was supported in part by NSF under Grant CMMI-1545038. *(Corresponding author: Yu Ding.)*

I. Ahmed and Y. Ding are with the Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: imtiazavi@tamu.edu; yuding@tamu.edu).

A. Dagnino is with the Information Systems Group, ABB, Cary, NC 27518 USA (e-mail: aldo.dagnino@us.abb.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2018.2848198

In this paper, it is the last category of unsupervised anomaly detection we are concerned with. Our problem is motivated by such a need encountered in a hydropower generation plant, which operates with turbine systems that are instrumented with dozens of sensors. Each turbine has subcomponents or functional areas such as several bearing systems, a generator, and so on. Sensors collect various types of data in real time such as the temperature of oil inside the bearing systems, vibrations in each functional areas, a variety of harmonics in functional areas, and many more. In total, each turbine collects more than 200 attributes from its sensors. The sensor data are then stored in a control system and kept as time stamped historical data points. Anomaly can be triggered from various sources and can cause a range of problems. For instance, an anomaly can be overheating of bearing oil and metal components, vibrations from bearings, low active power, or a combination of values from several components producing an out of control situation. To protect the health of components, it is vital to identify the anomalies as they appear.

When a service/maintenance engineer suspects that there is a malfunction in a turbine, she/he extracts a data set from the control system that contains the collected sensor data for that turbine for the selected period of time (few weeks to few months), and then stores this data in a relational database or simply in a csv file for further analysis. Staring at a spreadsheet of data, a service/maintenance engineer often wonders if there is an automated, efficient way to isolate the anomalies in the turbines. This problem falls under unsupervised anomaly detection because the historical data set in the spreadsheet almost surely have both normal data and anomalies. It is just that the service/maintenance engineers do not know which is what. What makes this problem more challenging is the number of attributes in the data space, amounting to a few hundreds and making a low-dimensional visualization difficult to carry out. This paper was in fact derived from the discussions with industrial collaborators on how to develop analytic algorithms that could adapt to analysis of large volumes of multidimensional data.

One fundamental issue in anomaly detection is proving the distinctness of anomalous observations relative to the normal observations. It is the absence of any learned rule from training data in the unsupervised setting that makes it a harder problem. The most commonly used dissimilarity matrix in the literature is still the Euclidean distance or some of its statistical variants such as the Mahalanobis distance [4]. Simply put, if $\|A - B\|_2 > \|A - C\|_2$, it implies that A and C are more similar. When a minority of data points are dissimilar from a majority of data points, then the minority of data points is considered an anomaly. Zimek *et al.* [5] observed that the Euclidean distance-based metrics lose its effectiveness in structured data spaces. To solve this problem, several schools of thought have been researched from different perspectives; we will provide a detailed account of the alternative approaches in the next section of literature review. Here, we wish to stress one important revelation of why Euclidean distance does not work well in structured data spaces, as first reported in [6].

It was argued in [6] that when a data space embeds an inherent structure forming a nonlinear manifold, the Euclidean distance for any two arbitrary points on the nonlinear manifold is unlikely to reflect their intrinsic similarity; please refer to the illustrative example in [6, Fig. 3]. The conclusion in [6] is that a geodesic distance must be used in a nonlinear manifold to reflect accurately the distance between the data points. We note that although a complicated structure happens more frequently in a high-dimensional space [5], it could indeed happen to low-dimensional spaces too, so that the Euclidean-based distance metric loses its effectiveness in low-dimensional spaces as well. As a matter of fact, the illustrative example in [6, Fig. 3] is in a 3-D space.

Geodesic distance is the minimum possible distance between two points in a curved surface like the surface of the earth. It was also shown [6] that as the number of data instances increases, the shortest path distances among data instances provide the best approximation to the geodesic distances. This important insight plays a vital motivational role for our research reported in this paper, as what we propose here is to use a minimum spanning tree (MST) to provide an approximation of geodesic distances in a structured space and then use it as the (dis)similarity metric. More specifically, we model the data observations as a network of nodes where edges represent the Euclidean distance from one another. An anomalous node would be the one which is less connected to its neighboring nodes. An MST is a measure that can capture the relative connectedness among the nodes, while at the same time, approximates the geodesic dissimilarities among observations forming a nonlinear manifold. It has been shown in the literature [7]–[9] that MST is indeed a capable approximation of geodesic distances in a high-dimensional data space embedding complicated structures.

There are several positive aspects associated with the proposed approach. First, the distance between two nodes in an MST, with the exception of the immediate neighboring nodes, is no longer the direct Euclidean distance between them; rather, it is the new dissimilarity metric which takes into account the overall connectivity among data points reflecting the complexity in a structured data space. So, the MST-based dissimilarity measure is a good candidate to approximate the geodesic distance and has the potential to overcome the limitation of direct Euclidean distances. Next, to take into account the presence of clusters of different shapes and densities, we develop MST locally and compare a node's connectedness with its neighboring cluster only. Doing so enhances the detection ability of the local, point-wise anomalies. The numerical analysis in comparing our proposed method with 13 other anomaly detection methods on 20 benchmark data sets demonstrates the superiority of the MST-based approach and supports the claimed merit.

We are aware that MST has been used to find anomalies [10]–[14]. The objective of most of them [10]–[12], [14] is to isolate the clustered anomalies by removing the links of the global MST one by one. Our attention in this paper is more about local, pointwise anomalies. What was done in [10]–[12] and [14] is similar to Stage 1 of the proposed method (more details later), which is really a preprocessing step in our method. Our main attention is on Stage 2, the local anomaly detection. The method in [13] falls in the category of semisupervised anomaly detection, which is fundamentally different from the unsupervised problem we deal with. Apart from that, our proposed approach uses neighborhood-based local MST which is rather different from the concept of k-point MST used in [13].

The rest of the paper unfolds as follows: Section II summarizes some of the existing approaches in the anomaly detection literature with their relative merits and drawbacks. Section III describes the main idea of our proposed approach and the steps of the algorithm developed. Section IV presents the comparative performance of our method with respect to 13 other alternatives on 20 benchmark data sets. Section V analyzes the hydropower data set and flags the potential anomalies. Finally, we conclude the paper in Section VI.

II. Literature Review

Anomaly detection methods in the literature can be categorized into four major domains depending on their criteria of identifying anomalies. They are: *distance and density-based methods*, *clustering-based methods*, *subspace-based methods*, and *ensemble-based methods*.

We want to stress that a common thread in almost all the methods is the use of Euclidean distance-based dissimilarity metric. This is certainly true in the distance-based methods in which a point is considered an anomaly if it lies further away from most of the points [15]. The distance-based criterion entails a number of variants to handle the complexity in real life. For instance, the k-nearest neighbor (k-NN)-based methods compare a candidate data point with its k-NNs, rather than all the data points, because it was believed doing so enhances the detection capability [16]–[18]. The density-based criterion is another variant, in which a point is considered an anomaly if the density around it is considerably lower than the density around its neighbors. The local outlier factor (LOF) [19], which is by far the most cited work in the anomaly detection literature, along with some of its variants [20]–[22] falls under this category. In the density-based methods, the data density surrounding a candidate point is calculated based on the pairwise Euclidean distances.

The density-based methods provide a different perspective for identifying anomalies, which is to consider the data clustering tendency. According to [23], there are three categories of clustering-based anomaly detection algorithms. Methods in [24]–[26] fall under the first category, which identify the instances that do not belong to any regular cluster as anomalies. The second group of clustering technique is a variation of the first group and uses a clustering algorithm to detect clusters, and then calculate an anomaly score by taking the distance from a point to its nearest cluster center. Both of these groups do not take into account the anomalies that can also form clusters, and in those cases, such methods will fail to detect these anomalous clusters. The third category of clustering-based algorithms [27]–[30] were introduced to tackle this problem which assumes that normal observations belong to large and dense clusters, whereas the anomalous observations belong to small and sparse clusters or lies further away from the cluster centroid. The clustering-based methods look at the distance measure differently, mostly in the form of between-cluster versus within-cluster distance comparison. Nonetheless, the Euclidean pairwise distances are still fundamental to these methods.

The use of Euclidean distance-based metric loses its effectiveness in a data space embedding inherent structures, which is mostly likely of high dimensionality. Several approaches were carried out to resolve this issue from different directions. A greater effort of addressing the detection issue is the school of methods generally known as the *subspace-based methods*, following the thought that one should look for anomalies only in relevant subspaces rather than searching them in the entire space [5]. Of course, the challenge lies in which subspace to search. For instance, principal components analysis (PCA) [31] renders the subspace that has the largest variances most relevant, while multidimensional scaling [32] selects the subspace that preserves the interpoint distances in their low-dimensional representation. They are both capable of preserving the original data space structure in linear vector spaces, but they tend to lose the data structure in the presence of nonlinear manifolds [6].

Many other variants of subspace-based methods exist. A grid-based subspace clustering [33] was proposed for searching sparse, rather than dense, grid cells to report the objects contained within those sparse grid cells as anomalies. High-dimensional outlying space miner [34] ranks a point as an anomaly in any subspace if the sum of its distance from its k-NNs crosses a predetermined threshold. Subspace clustering [35] can be also helpful as anomalies are found in abnormally few clusters or low-dimensional clusters. Subspace outlying degree (SOD) [36] detects an anomaly based on its deviation to the subspace spanned by a set of reference points. High contrast subspaces (HicS) [37] relies on finding those subspaces where attributes are correlated (statistically dependent). GLOSS [38] suggested that global neighborhoods should be considered when detecting anomalies locally in selected subspaces. Subspace methods undoubtedly made progress in the unsupervised anomaly detection literature. But fundamentally, finding out the right subspaces to explore is still a difficult problem to solve.

Aware that we cannot rely on any single detection technique to detect different kinds of anomalies, people simply decide to aggregate them together, forming the school of ensemble-based anomaly detection algorithms [39], [40]. One unsolved concern of such techniques is how to combine scores from totally different methods. In addition, we also need to ensure the diversity and accuracy of the chosen methods; otherwise, final solution would be biased from the similar errors.

In a nutshell, a blank in the anomaly detection research waiting to be filled in is a method which is capable of detecting local, pointwise anomalies in a structured data space. Instead of making incremental improvements over any one of the existing methods, our research shows that we need to rethink the fundamental issue of how we differentiate data instances in unsupervised learning settings. The current reliance on Euclidean distances appears to run out of steam. In this paper, we introduce a suitable dissimilarity metric

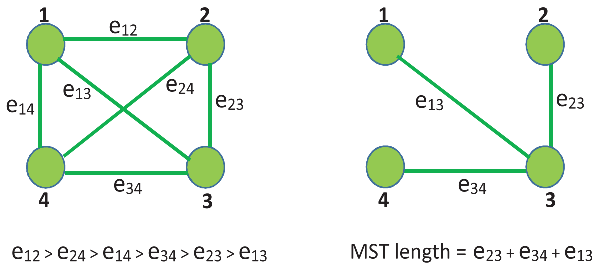
1
2
4
3
e12
e13
e14
e23
e24
e34
e12 > e24 > e14 > e34 > e23 > e13
1
2
4
3
e13
e23
e34
MST length = e23 + e34 + e13

Fig. 1. Formation of an MST. Left: complete graph. Right: MST.

that approximates the intrinsic distance of data points in the original feature space. Our proposed approach is an important attempt to provide a solution to this challenging anomaly detection problem in an unsupervised setting.

III. MST-Based Anomaly Detection Method

As mentioned earlier, our main focus is to come up with an MST-based distance metric which reflects the dissimilarity among anomalies and normal data points in structured data spaces. We would first like to provide a brief background on MST.

To understand the MST, let us consider a connected undirected graph $A = (V, E)$, where V denotes the collection of vertices or nodes and E represents the collection of edges connecting these vertices as pairs. For each edge $e \in E$, there is a weight associated with it. It could be either the distance between the chosen pair of nodes or the cost to connect them together. An MST is a subset of the edges in E that connects all the vertices together, without any cycles and with the minimum possible total edge weight. This total edge weight is the sum of the weights of the individual edges, also known as the total length or total cost of the MST. If we use the Euclidean distance between a pair of nodes as the edge weight, then it is called the Euclidean MST. Consider the example in Fig. 1, where $V = \{1, 2, 3, 4\}$ and $E = \{e_{12}, e_{13}, e_{14}, e_{23}, e_{24}, e_{34}\}$. Edges in E are all different in length. If we want to connect all the nodes in V without forming a cycle, there could be 16 such combinations with only one having the minimum total edge length; that one is the MST for this connected graph.

If we consider data instances as vertices and the Euclidean distance between any pairs of data points as the edge weight, then we can construct an MST to connect all the nodes. There are three well-known algorithms ([41]–[43]) that can find the MST, given a graph. Although the distance between a pair of immediately connected nodes is still Euclidean, the distance between a general pair of nodes (i.e., data points) is not. Rather, it is the summation of many small-step, localized Euclidean distances hopping from one data point to another point. As the MST reflects the connectedness among the data points in a nonlinear manifold, the MST-based distance is the geodesic distance between two data points, which, according to [6], provides a better metric to differentiate them.

Using MST helps address another complexity often encountered in anomaly detection, which is to detect pointwise anomalies in the presence of anomalous clusters. Fig. 2 presents an illustrating example in which there are well-separated four clusters whose structures are not difficult to identify. The star-shaped symbols represent the local anomalies relative to their nearest cluster. Cluster 4 itself is an anomalous cluster whereas clusters 1, 2, and 3 are regular clusters. Existing anomaly detection methods, such as LOF [19], adjust their view field on anomalies by setting different values of the nearest neighborhood, k—when a small k is used, the local anomalies are detected but the anomalous cluster 4 will be unidentified, while when a large k is used, all the instances can be separated into two parts, namely, Group 1 and Group 2, in which Group 2 contains cluster 4. Under that circumstance, one has to pay the price of not detecting the local anomalies in Group 1. The reason that using MST can help is because MST can be used as a clustering tool [10], [12] to isolate the anomalous clusters first. Then, it can be refined to define the dissimilarity distances in a local setting. This thought points to a two-stage procedure, which is to remove the global anomalous clusters first and then detect the local pointwise anomalies later; both stages use MST as the common methodological foundation.

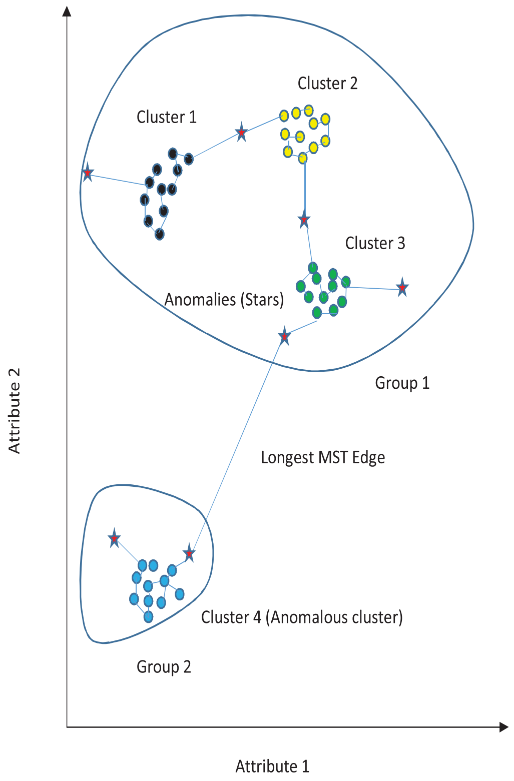

Fig. 2. Pointwise anomalies versus anomalous clusters.

Specifically, our detection algorithm proceeds as follows. The first stage is to identify the anomalous clusters, if they exist. The procedure is similar to the existing work of how MST is used [10]–[12]. First, we build a global MST using all the data points. The specific MST construction algorithm we use is in [41]. After the formation of the global MST, we then look for a long edge and treat it as the connecting edge between the anomalous cluster and the rest of the MST. Once this edge is disconnected, it separates the MST into two groups, and the smaller group is considered an anomalous cluster.

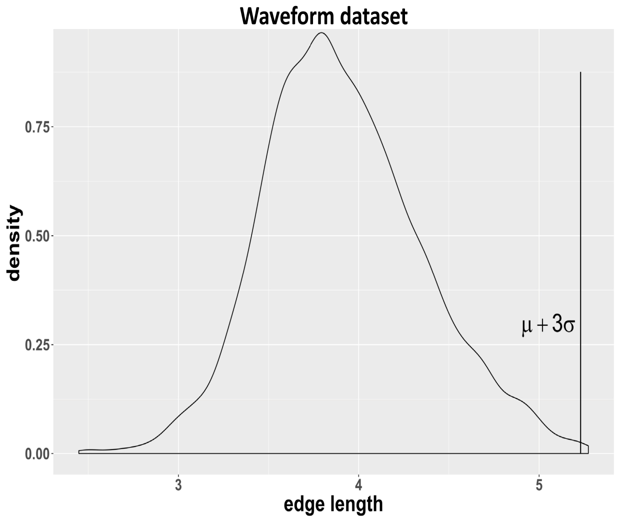

Fig. 3. Distribution of MST edges.

For this purpose, we suggest using the control limit approach, $\mu + q \cdot \sigma$, as commonly used in the statistical quality control [44], to flag the unusually long edges. Here, μ represents the average of the edge weight and σ represents their standard deviation. The choice of q depends on the data set and the distribution of edge lengths. Following the practice in statistical quality control, we suggest plotting the empirical distribution of the edges in the MST and then selecting q corresponding to an α probability, such that an edge is considered unusually long only if it is longer than $(1 - \alpha)100\%$ of all edges. Fig. 3 presents an example of the empirically estimated density curve of the MST edges. In practice, the distribution does not always follow exactly the normal distribution, and the edge length distribution is different for different applications. For this reason, using the edge length distribution specifically estimated for each application to set the corresponding q adapts the selection of parameters to individual circumstances and is thus more robust. In the application cases we experiment with, we find that $q = 3$ works reasonably well for most of the data sets. In the example shown in Fig. 3, $q = 3$ is corresponding roughly to $\alpha = 0.27\%$. This edge deletion procedure will be iterated on the larger remaining group and see if there is another, less dominating anomalous cluster, until there is no anomalous long edge detected. This procedure is equivalent to a Phase I analysis in the statistical quality control [44].

Once the clustering decomposition stops in the first stage, we then move on to the second stage of identifying pointwise anomalies, which is also the main contribution of our work. In the second stage, we go into the neighborhood level for each data point to determine its possibility as an anomaly. The neighborhood is determined by the number of NNs and parameterized by k. We will come back later to discuss the procedure of selecting the value of k, but for now, let us assume we have a predetermined value for k.

Denote by R the rest of data points after the anomalous clusters are removed in Stage 1. For any given data point in R, first isolate its k-NNs and treat them as this data point's neighborhood. Then, build an MST in this neighborhood. Considering the nature of these neighborhood MSTs, they are referred to as local MSTs (LoMST). The total edge length of the LoMST associated with the original data point is called the LoMST score for this data point and is considered the metric measuring its connectedness with the rest of the points in the neighborhood, as well as how far away it is from its neighbors. For this reason, the LoMST is used as the differentiating metric to signal the possibility that the said data point may be an anomaly.

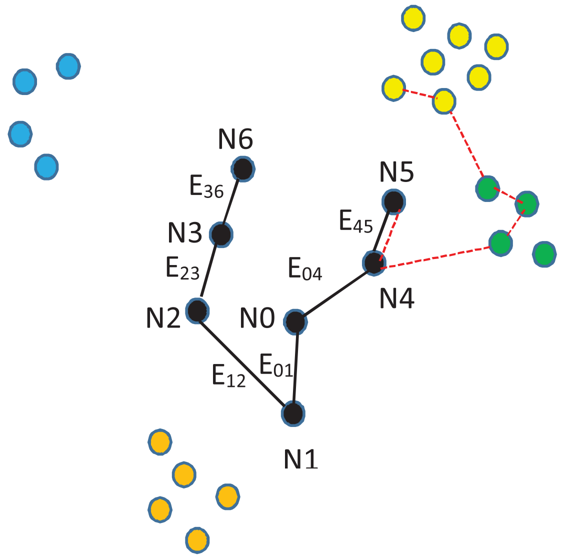

Fig. 4. Local MST and LoMST score. The total edge weight of the local MST for $N0$ is its LoMST score, i.e., $W_{N0} = E_{01} + E_{12} + E_{23} + E_{04} + E_{45} + E_{36}$.

Consider the illustrating example in Fig. 4. Suppose that we have chosen $k = 6$ and start with data point $N0$. Then, we can locate its neighbors as $N1$, $N2$, $N3$, $N4$, $N5$, and $N6$. The MST construction algorithm connects $N0$ to its neighbors in the way as shown in Fig. 4. For $N0$, the total edge weight is $W_{N0} = E_{01} + E_{12} + E_{23} + E_{04} + E_{45} + E_{36}$, which is deemed the LoMST score for $N0$. This procedure will be repeated for other data points. Fig. 4 shows another MST, which is for $N5$ in the dotted edges.

The LoMST score for a selected observation will be compared with its neighbor's score. Comparison can be done in two ways. We can either compare W with the mean of the neighbor's scores or with the mean-to-standard deviation ratio of the neighbor's scores. Our analysis suggests that both comparison approaches have their own advantages depending on the structure of the data set. When there are numerous anomalies, almost forming an anomalous cluster within a neighborhood, it would be better to use the mean-to-standard deviation ratio, as the mean of the neighbor's LoMST scores are severely contaminated by other anomalies. However, when the anomalies are very few, pointwise scattered around, the mean of the LoMST score works just fine. Considering that we have a first stage to remove some of the anomalous clusters, in this second stage, we then use the mean comparison of LoMST scores as our default approach.

The step of comparison will be repeated $|R|$ times covering all the nodes in R. Then, the comparison score will be scaled between 0 and 1 using the maximum and minimum

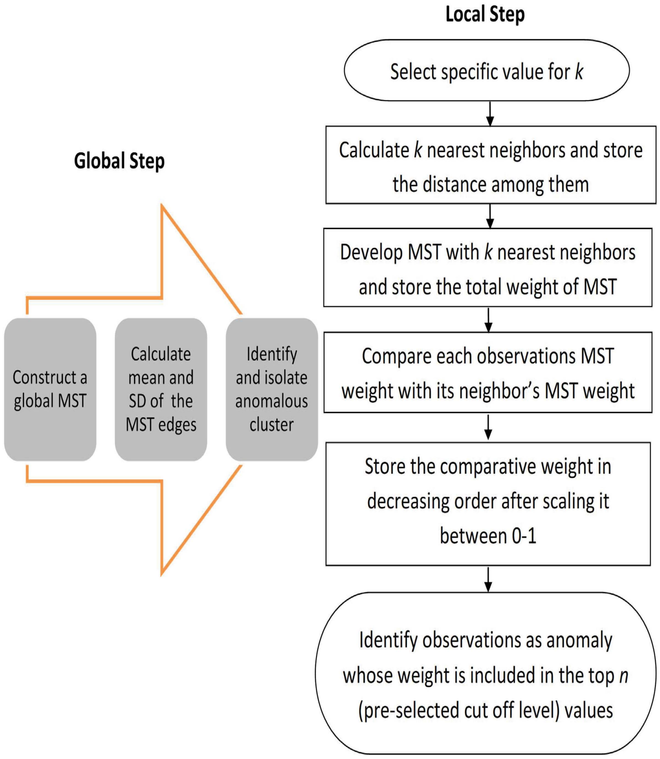
Local Step
Select specific value for k
Calculate k nearest neighbors and store the distance among them
Develop MST with k nearest neighbors and store the total weight of MST
Compare each observations MST weight with its neighbor's MST weight
Store the comparative weight in decreasing order after scaling it between 0-1
Identify observations as anomaly whose weight is included in the top n (pre-selected cut off level) values
Global Step
Construct a global MST
Calculate mean and SD of the MST edges
Identify and isolate anomalous cluster

Fig. 5. Flowchart of the proposed method.

Algorithm 1 MST-Based Method

```
Input: Data set (rows represent observations and columns represent attributes), number of NNs, $k$, coverage probability, $\alpha$, and cut-off level for identifying anomalies, $n$.
Output: anomaly index set, $TO$
Stage 1: Steps for identifying distant anomalous observations
1. Develop set of vertices $V$, where each vertex represent a separate observation from the data set
2. Construct edges by calculating Euclidean distance between each pair of vertices using their attribute values from the data set and store them in $E$
3. Construct a global MST using $V$ and $E$, let $S$ be the set of edges of the resulting MST, where $S \subseteq E$
4.   Calculate the mean, $\mu$, and the standard deviation, $\sigma$, of the edges in $S$
5.   Calculate the longest edge from $S$ and store its length in $Lo$
6.   IF $Lo \geq \mu + q \cdot \sigma$ THEN remove this edge from the global MST
7.   From the two disconnected trees, let $O_1$= {vertices contained in the smaller tree} and $R$= {vertices contained in the larger tree}
8. REPEAT steps 4-7 in $R$ until no such anomalous edge exists
Stage 2: Steps for identifying local anomalous observations
1. For each vertex $r_i \in R$
2.   Determine its $k$-NNs and save them in $N_i$
3.   Construct a local MST using edges contained in $E_{uv}$, where $u, v \in N_i$ and $E_{uv} \subseteq E$
4. Set $T = \oslash$, $O_2 = \oslash$
5. For each vertex $r_i \in R$
6.   Calculate the total length of $r_i$'s LoMST, $W_{r_i}$
7.   Calculate the mean $(\overline{W}_{N_i})$ of the total length of the LoMSTs associated with all vertices in $N_i$
8.   Calculate the LoMST score for $r_i$ as $T_i$ = $W_{r_i}$ - $\overline{W}_{N_i}$
9. Normalize the scores stored in $T$ to be between 0 and 1
10. Rank the normalized scores in $T$ in decreasing order
11. Identify the top $n$ scores and store the corresponding observations as point anomalies in $O_2$
12. $TO$= $O_1$ U $O_2$
```

value of the scores. From now on, we call the normalized scores as the LoMST scores. After that, these LoMST scores will be sorted in decreasing sequence, where a greater score implies a higher possibility to be an anomaly. To compile a list of anomalies, we follow the common approach in the unsupervised learning setting, which is to select a prescribed cutoff value n and flag the top n instances on our rank list as anomalies. One main reason behind such a detection procedure is that unsupervised detection methods tend to have a lower detection capability and higher false alarm rate, as compared to general supervised learning algorithms. As a result, unsupervised detection methods are typically used as a screening tool, flagging potential anomalies to be further analyzed by either a human operator or some more expensive procedure. A cutoff is therefore used to ensure the subsequent, more expensive, or time-consuming step practical and feasible.

For better technical understanding, the algorithm steps are summarized in Algorithm 1. In addition, a flowchart of the proposed method separated into two stages is also highlighted in Fig. 5.

Now, let us get back to the issue of selecting a suitable value for k. The difficulty in choosing k in an unsupervised setting is that methods like cross validation that work for supervised learning do not apply here. Our approach is then based on the following observations, illustrated in Fig. 6. When we plot the average LoMST scores for a broad range of k (here 1–100), we observe that at small k values, the average LoMST score tends to fluctuate, but as we keep increasing k, the average LoMST score will become stable at certain point. Our understanding is that when a proper k is chosen, the structure of the data is revealed and the label of the instances will become almost fixed, thus reflected in a less fluctuating LoMST score. If one keeps increasing k, there is the possibility that the data structure becomes mismatch with the assigned number of clusters and the the current assignments of anomalies and normal instances become destabilized again. Consequently, the average LoMST score could fluctuate once again. Based on this observation, our policy in choosing k is to select a range of k where the average LoMST scores are stable. If there are more than one stable ranges, we will then select the first one.

Let us look at the examples in Fig. 6. For the *Cardiotocography* data set, we can choose a k range from 27–47, and for the *Glass* data set, we can choose a k range from 70–95.

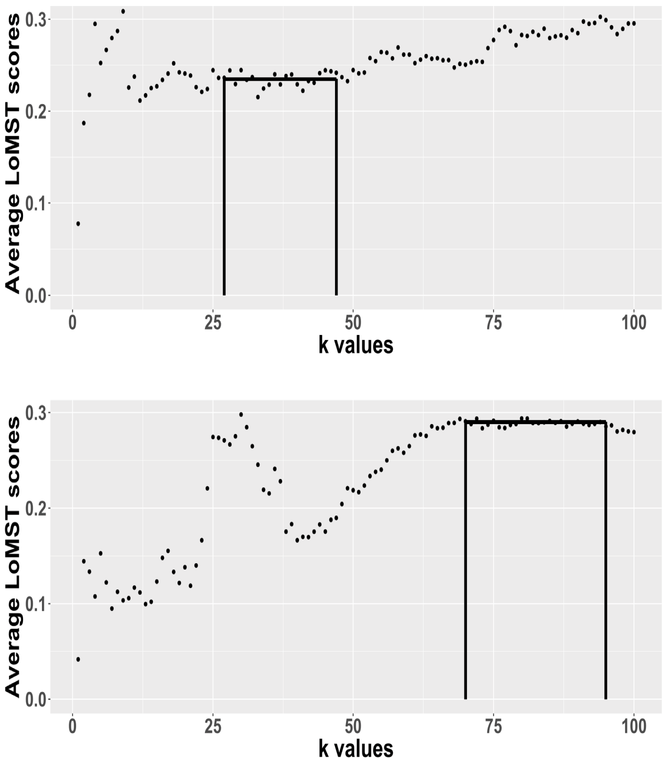

Fig. 6. Select the range of k values for Cardioctography and Glass data sets.

Within the identified stable range, choosing the appropriate k value matters but matters less. What we suggest to select is the k value that returns the maximum standard deviation of the LoMST scores, because by maximizing the standard deviation among the LoMST scores, it increases the separation between the normal instances and anomaly instances and facilitates the detection mission. For the *Glass* data set, it takes slightly more than 1 min to select the k value and for the *Cardiotocography* data set, it takes about 10 min, both on a typical desktop computer.

IV. Performance Comparisons

In this section, we wish to evaluate the performance of the MST-based method on a set of benchmark testing data sets compared to an array of well-established methods in the literature. Our comparative study is entirely based on the data sets and methods used in a comprehensive survey [45], as this survey established the benchmark for all subsequent anomaly detection research, and it is also timely and up to date.

There are 20 test data sets in [45] for performance evaluation. Several versions of these data sets are stored in the online repository of [45]. These versions mainly differ in terms of the preprocessing steps used. We use in this comparative study the normalized version of the data sets in which all the missing values are removed and categorical variables are converted into numerical format. We do not perform any preprocessing by ourselves, and instead, we use the same form as stored in the repository. Table I summarizes the basic characteristics of the 20 data sets used in our study. Note that for these 20 test sets, the anomalies are known.

TABLE I

Data Sets Used in the Performance Evaluation Study

Data set	Number of observations (N)	Number of anomalies ($\|O\|$)	Number of attributes (p)
Annthyroid	7,200	347	21
Arrhythmia	450	12	259
Cardiotocography	2,126	86	21
HeartDisease	270	7	13
Page Blocks	5,473	99	10
Parkinson	195	5	22
Pima	768	26	8
SpamBase	4,601	280	57
Stamps	340	16	9
WBC	454	10	9
Waveform	3,443	100	21
WPBC	198	47	33
WDBC	367	10	30
ALOI	50,000	1,508	27
KDDcup99	60,632	200	41
Shuttle	1,013	13	9
Ionosphere	351	126	32
Glass	214	9	7
Pen digits	9,868	20	16
Lymphography	148	6	19

As our method is dependent on the parameter k, we mainly focus on the nearest neighborhood-based approaches for a fair assessment. Campos *et al.* [45] provided detailed experimentation on 12 popular nearest neighborhood approaches based on the 20 aforementioned data sets. These 12 methods are *connectivity-based outlier factor (COF), local density factor (LDF), k-NN, outlier detection using indegree number (ODIN), LOF, k-NN weight (KNNW), simplified LOF, local outlier probabilities (LoOP), influenced outlierness (INFLO), local distance-based outlier factor (LDOF), fast angle-based outlier detection (FABOD), and kernel density estimation outlier score (KDEOS)*. Traditional statistical process control (SPC)-based approach could also be applied in the anomaly detection setting. We implemented one of the popular methods in SPC, the Hotelling T^2 control chart [46]. We tested two versions while using the Hotelling T^2 control chart: one with PCA that reduces the data dimension first and the other without PCA. It turned out that the T^2 control chart without PCA performs slightly better than the PCA version. Hence, we only include the T^2 result without PCA in the comparison tables to save space. In summary, we compare our MST-based approach with a total of 13 competing methods.

We chose the same performance metric, called *precision at n* ($P@n$), as used in [45]. It is defined as the proportion of correct anomalies identified in the top n ranks. Just like in [45], we chose the number of anomalies as our value of n. For a database DB of size N, consisting of anomaly set $O \subset DB$ and normal point sets $I \subseteq DB$, such that $DB = O \cup I$, $P@n$ can be formalized as

$$P@n = \frac{\{o \in O \mid \text{rank}(o) \leq n\}}{n}, \quad \text{where } n = |O|. \tag{1}$$

Admittedly, this $P@n$ metric focuses on the detection capability, which is critical for anomaly detection applications. We would like to add that following the top n rank detection procedure, the false alarm rate is implied by the $P@n$ metric, as the number of false positives or false alarms is simply

TABLE II
PERFORMANCE COMPARISON BASED ON THE BEST k VALUE

Anomaly detection methods Performance (number of data sets)	LoMST	COF	LDF	KNN	ODIN	LOF	KNNW	SLOF	LoOP	INFLO	LDOF	FABOD	KDEOS	SPC
Better (uniquely best result)	6	0	3	1	1	0	1	0	0	0	0	0	0	1
Equal (equal to the existing best result)	7	5	5	2	1	3	2	2	2	2	1	2	2	0
Close (within 20% of the best result)	5	10	6	7	8	8	7	6	6	5	7	5	1	1
Worse (not within 20% of the best result)	2	5	6	10	10	9	10	12	12	13	12	13	17	18
Mean relative rank	2.2	3.3	3.8	5.0	7.7	5.1	4.5	5.9	5.8	6.2	7.9	7.0	8.9	11.7

TABLE III
PERFORMANCE COMPARISON BASED ON THE PRACTICAL k CHOSEN ACCORDING TO OUR SELECTION POLICY

Anomaly detection methods Performance (number of datasets)	LoMST	COF	LDF	KNN	ODIN	LOF	KNNW	SLOF	LoOP	INFLO	LDOF	FABOD	KDEOS	SPC
Better (uniquely best result)	5	2	1	1	2	0	0	0	0	0	0	2	0	1
Equal (equal to the existing best result)	5	1	4	5	1	3	4	1	1	2	1	2	0	0
Close (within 20% of the best result)	7	11	4	6	8	7	9	8	7	7	6	6	3	3
Worse (not within 20% of the best result)	3	6	11	8	9	10	7	11	12	11	13	10	17	16
Mean relative rank	2.8	4.2	5.7	5.3	6.7	6.1	4.3	6.5	5.6	5.8	7.6	4.9	11.7	8.7

$n - n \times P@n$. Suppose that we set $n = 10$ and if we have 8 of them identified correctly as the anomalies, then $n \times P@n = 8$, and the number of false alarms is $10 - 8 = 2$. For this reason, we do not present the false alarm rate explicitly.

There is no guideline in the literature for how best to select k. Campos *et al.* [45] simply tried a range of k values (from 1 to 100) to obtain all the results and then choose the best k value for each method. In the first comparison, we follow the same approach, labeled as the "best k" comparison. The results are presented in Table II. To better reflect the detection capability as they are compared to one another, we break down the comparative performance into four major categories, namely *Better*, *Equal*, *Close*, and *Worse*, as explained in the table. Please note that the "best" k value in Table II may be different for respective methods.

LoMST shows a superior performance and clearly outperforms other methods. In 13 of the 20 data sets, LoMST either exhibits a uniquely best detection performance or is tied with some other methods to achieve the best detection capability. In only two data sets, LoMST performs in the *Worse* category, meaning that its detection capability is 20% lower than the best alternative. If we rank each of the 14 methods in a scale of 1 to 14 according to its actual performance in relative to others, then the average rank for the LoMST method is 2.2, while some of the closest competitors are COF(3.3), LDF(3.8), KNNW(4.5), KNN(5.0), and LOF(5.1).

Understandably, the "best k" is not practical, as in reality, people do not know the anomalies while selecting k. Since we have come up with a strategy to select a practical value of k, we use the same practical k in the other 12 alternative methods that need this value (SPC does not need to know k). The performance comparison based on the practical k is presented in Table III. We use the same performance breakdown as in Table II. Our LoMST method continues to exhibit a superior performance for being uniquely best in five of the 20 data sets, and tieing for the best in another five data sets. The number of cases in the *Worse* category is three. The average rank of the LoMST method is 2.8, slightly lower than that under the best k condition, while some of the closest competitors are COF(4.2), KNNW(4.3), FABOD (4.9), KNN(5.3), and LDF(5.7). Table IV provides the number of true positive detections of 14 methods under the best k setting, in which the best performance in every row is highlighted in boldface. To save space, we omit the same table under the practical k setting as it conveys the same message.

In Section III, we mentioned both the mean-based and the mean-to-standard deviation-based comparison statistics. Tables II and III present the comparison results using the mean-based statistic, which is our recommended default option. We also explore what if we use the mean-to-standard deviation ratio as the comparison statistic, and the results are presented in Tables V and VI, respectively, depending on how k is selected. This time we included a smaller number of alternative methods in the comparison to save space, because the performance of other methods lags too far behind the LoMST and the few top competitors. Our comparison still shows that the LoMST performs the best among the alternatives but its relative performance is slightly worse when using the mean-to-standard deviation ratio. This is not surprising. As explained in Section III, our MST-based approach has a first stage of operation that removes the anomalous clusters, so it generally performs well when using the mean-based comparison statistic.

We further conduct some statistical test and see if the performance difference between the proposed method and its competitors is significant. For this purpose, we use a nonparametric method, the Friedman test [47]. Let n_a be the number of anomaly detection methods and n_d be the number of data sets. We define a matrix $\mathbf{Ra}$ whose entries in each row represent the detection method's rank for that specific data set. If there are tied values, we assign to each tied value the average of the ranks that would have been assigned without ties. For example, suppose we have two tied methods both with rank 7. If there had been no tie, then one should have been assigned rank 7 and the other rank 8. The Friedman test then uses the average of the two ranks, which is 7.5,

TABLE IV

NUMBER OF TRUE POSITIVE DETECTIONS OF THE 14 METHODS IN THE BEST *k* SETTING

	LoMST	COF	LDF	KNN	ODIN	LOF	KNNW	SLOF	LoOP	INFLO	LDOF	FABOD	KDEOS	SPC
WBC	**8**	**8**	**8**	**8**	4	**8**	**8**	6	4	4	4	7	1	5
Waveform	**35**	27	29	23	9	21	22	20	17	15	11	8	8	0
Heart	**5**	4	4	4	4	4	4	4	4	4	2	2	1	0
WDBC	6	6	**7**	6	5	6	6	6	6	6	5	5	1	4
Glass	**3**	**3**	**3**	1	2	**3**	1	**3**	**3**	**3**	2	1	1	1
Spambase	**78**	55	48	76	65	55	76	44	55	55	52	64	41	68
WPBC	**14**	11	13	10	12	10	9	10	10	11	13	10	**14**	5
Stamps	**6**	5	4	3	3	4	4	4	4	4	5	**6**	4	2
Parkinson	**4**	3	3	3	2	3	2	2	2	3	2	3	3	1
Lymphography	**6**	**6**	**6**	**6**	5	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	4
Ionosphere	108	107	104	107	99	105	108	104	101	101	101	108	97	**119**
PIMA	**8**	7	3	6	3	3	6	3	3	3	3	6	5	2
Shuttle	**7**	**7**	**7**	6	**7**	5	6	6	5	5	3	3	6	3
Cardiotocography	30	28	28	30	18	22	**31**	27	26	23	25	29	15	18
Arrhythmia	**5**	**5**	**5**	3	3	3	3	3	3	3	3	3	2	0
Page blocks	46	45	**48**	46	40	45	47	47	45	43	44	45	12	38
Pendigits	**4**	3	1	0	0	1	1	1	1	0	1	1	1	0
KDD	39	5	4	**87**	9	3	43	0	5	2	0	0	10	0
ALOI	314	346	226	261	**357**	319	265	316	336	345	0	0	193	69
Annthyroid	39	47	**60**	36	55	53	41	41	51	54	57	35	46	22

TABLE V

PERFORMANCE COMPARISON BASED ON BEST *k* FOR ALTERNATIVE NEIGHBORHOOD COMPARISON STATISTIC

Anomaly detection methods Performance (number of datasets)	LoMST	COF	LDF	KNN	KNNW
Better(uniquely best result)	7	1	3	1	1
Equal(equal to the best result)	4	5	5	4	2
Close(within 20% of the best result)	7	9	6	5	9
Worse(not within 20% of the best result)	2	5	6	10	8
Mean relative rank	3.0	3.1	3.5	4.8	4.5

TABLE VI

PERFORMANCE COMPARISON BASED ON PRACTICAL *k* FOR ALTERNATIVE NEIGHBORHOOD COMPARISON STATISTIC

Anomaly detection methods Performance (number of datasets)	LoMST	COF	LDF	KNN	KNNW
Better(uniquely best result)	6	0	1	1	0
Equal(equal to the best result)	2	2	5	5	3
Close(within 20% of the best result)	8	12	7	8	10
Worse(not within 20% of the best result)	4	6	7	6	7
Mean relative rank	3.8	4.4	5.6	5.0	4.2

TABLE VII

P-VALUES OF PAIRWISE COMPARISON OF LOMST METHOD WITH THE COMPETING METHODS

	Best k	Practical k
COF	1.28×10^{-1}	6.47×10^{-2}
LDF	3.18×10^{-2}	2.96×10^{-4}
KNN	3.78×10^{-4}	2.24×10^{-3}
ODIN	2.03×10^{-8}	6.11×10^{-6}
LOF	5.66×10^{-5}	2.01×10^{-5}
KNNW	1.95×10^{-3}	4.02×10^{-2}
SLOF	2.16×10^{-6}	3.92×10^{-6}
LoOP	2.02×10^{-6}	2.77×10^{-4}
INFLO	8.13×10^{-7}	9.79×10^{-5}
LDOF	1.30×10^{-9}	1.44×10^{-8}
FABOD	1.02×10^{-7}	1.14×10^{-2}
KDEOS	1.28×10^{-11}	7.04×10^{-17}
SPC	3.08×10^{-18}	1.05×10^{-9}

as the rank value for both of these methods. Under the null hypothesis that all methods perform the same, the Friedman statistic

$$\chi_F^2 = \frac{12n_d}{n_a(n_a+1)}\left(\sum_{l=1}^{n_a}\overline{Ra_l^2} - \frac{n_a(n_a+1)^2}{4}\right) \tag{2}$$

follows a chi-squared distribution with $n_a - 1$ degrees of freedom, where $\overline{Ra_l}$ is the average value of column $l = 1, 2, \ldots, n_a$. We have done the tests for both best k and practical k settings and found the p-values (1.27×10^{-12} and 1.07×10^{-10}, respectively) significant enough to reject the null hypothesis.

To find out whether our method is significantly different from other methods, we also conducted some *post hoc* analysis. Fig. 7 presents the *post hoc* analysis on the ranking data for the practical k setting, showing that the LoMST's ranking is significantly higher (lower in numeric sense) than other competing algorithms. The detailed pairwise comparisons using respective p-values for both best k and practical k settings are presented in Table VII. The p-values are calculated using Conover *post hoc* test [48]. We have used the false discovery rate approach [49] to adjust the p-values for multiple comparisons. Other than between COF and LoMST, which shows a marginal significance, all other pairwise comparisons have shown a sufficiently significant difference, suggesting that the LoMST is superior and produces a better performance.

We summarize LoMST's performance with respect to the data size in Table VIII. It is evident that LoMST performs well and comes out as the best method in a wide range of scenarios. Looking at the two extremes, the case of the highest number of observations ($N = 60,632$, *KDDcup99* data), which is also the one having the second highest N/p ratio, versus the case of the highest number of attributes ($p = 259$, *Arrhythmia* data), which is also the one having the lowest N/p ratio, LoMST performs on top in both cases. It is reassuring that LoMST delivers this level of success if not

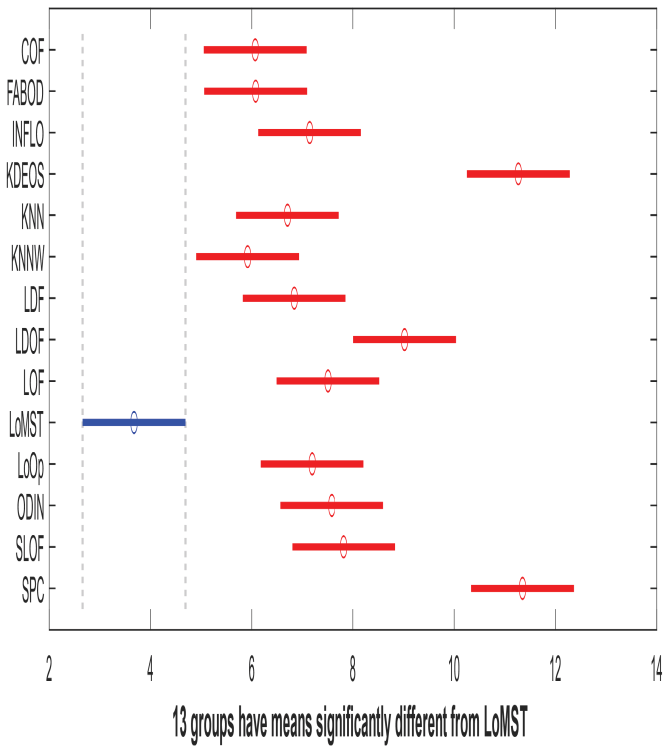

Fig. 7. *Post hoc* analysis on the ranking data obtained by the Friedman test. This analysis is under the practical k setting.

TABLE VIII

PERFORMANCE OF THE LOMST UNDER DIFFERENT N/p RATIOS. THE RATIOS ARE ROUNDED UP TO THE NEXT INTEGER

Data set	N/p	N	p	$\|O\|$	Rank (Best k)	Rank (Practical k)
Arrhythmia	2	450	259	12	1	1
SpamBase	81	4,601	57	280	1	1
KDDcup99	1,479	60,632	41	200	3	2
WPBC	6	198	33	47	1	3
Ionosphere	11	351	32	126	2	3
WBC	51	367	30	10	1	1
ALOI	1,852	50,000	27	1,508	7	7
Parkinson	9	195	22	5	1	1
Annthyroid	343	7,200	21	347	11	10
Waveform	164	3,443	21	100	1	1
Cardiotocography	102	2,126	21	86	2	3
Lymphography	8	148	19	6	1	1
Pendigits	617	9,868	16	20	1	2
HeartDisease	21	270	13	7	1	1
PageBlocks	548	5,473	10	99	4	7
Stamps	38	340	9	16	1	6
Shuttle	113	1,013	9	13	1	1
WDBC	13	454	9	10	2	2
Pima	96	768	8	26	1	1
Glass	31	214	7	9	1	1

beyond. We do notice that on two of the data sets, when the number of anomalies are too numerous (over a few hundreds to more than one thousand), LoMST did not do well enough. In hindsight, it makes intuitive sense, as LoMST is designed to find the local, pointwise anomalies, which, when existing, should be of a relatively small amount. This shortcoming does not appear to diminish the practicality of LoMST, because for engineering or industrial applications, it is very unlikely that operators will wait to accumulate hundreds or over a thousand of anomalies before applying an anomaly detection method.

We want to note that in practice, one does not know the number of the true anomalies to set the cutoff n. As explained earlier, n is usually chosen to be larger than the perceived number of anomalies but small enough to make the subsequent identification operations feasible. For anomaly detection problems, the false alarm rate is generally high, in order to boost the detection capability. This is a common phenomenon in all anomaly detection methods. To see this point, consider the following example. In the WBC data set, there are 10 true anomalies and 213 normal instances. When using $n = 20$ as the cutoff, meaning that the LoMST method flags 20 instances as anomalies, 10 of the 20 are truly anomalies and 10 are falsely tagged as anomalies. As such, the detection rate is 10/10 (100%), whereas the false alarms are 10/20 (50% of all alarms, but 4.7% relative to the total number of normal instances). We would argue that despite the relatively high proportion of the false alarms, the anomaly detection method is still practically useful, particularly as a prescreening tool. By narrowing down the candidate anomalies from the whole set to 20, which is an order of magnitude decrease in data amount, it helps human experts a great deal to follow up with each circumstance and decide how to improve their processes. We believe that a fully automated anomaly detection is not yet realistic in the near future, due to the challenging nature of the problem and the relatively lack of advancement in the state of the art. Therefore, a useful prescreening tool, as the current anomaly detection offers, would be valuable in filling the void, while the researchers strive for the ultimate, full automation goal.

Another note is about the computational complexity of LoMST, which comes from two major sources. First, we need to conduct the k-NN search based on the chosen k. Then, for each observation, we need to build a local MST using its k-NNs. For the first step, we use the fast approximate NN searching approach [50], [51] with time complexity $O(pN \log N) +$ $O(kp \log N)$. The first time complexity component, $O(pN \log N)$, represents the time to build the tree structure, whereas the second component, $O(kp \log N)$, represents the k-neighborhood query time for a single observation. In the second step, building the local MST has the time complexity of $O(|V| \log |E|)$ where $|V|$ is the number of vertices and $|E|$ is the number of edges. The $|V|$ and $|E|$ depend on k but usually remain small. The neighborhood search and the local MST step will be repeated N times, while the tree structure building is a one-time action. As such, the total complexity of the LoMST algorithm is approximately $O(pN \log N) + O(N[kp \log N + |V| \log |E|])$. In case of the *ALOI* data set which has the largest N/p $(= 1852)$ ratio and the second largest N $(= 50\,000)$, LoMST takes approximately 2 min to finish anomaly detection on a typical desktop computer. Being a local MST method, our method's MST construction can stay local and thus be computationally efficient.

V. APPLICATION TO THE HYDRO TURBINE DATA

The hydropower data initially received was time stamped (a total of 7 months' worth of data) and divided into different functional areas (turbines, generators, bearings, etc.). The data was collected at 10-min intervals each day. But it was not always continuous and some days from each of these seven months were missing. After combining all data across all functional areas, there are 9508 observations (rows in a table) and 222 attribute variables (columns in a table). Each row has a time stamp assigned to it. Attribute variables are primarily temperatures, vibrations, pressure, harmonic values, active power, and so on. Before applying the anomaly detection method, we conducted some basic preprocessing and statistical analysis in order to clean the data. To maintain the similarity with the 20 benchmark data sets, we normalized the data and removed the duplicate rows as well as the rows with missing values. In addition, we also did correlation analysis and plotted histogram, density, and box plots. The data preprocessing did yield a small number of data records that are so far off from other data records. When checking with the domain expert who provided the data, it was confirmed that those records were due to a recording mistake. After removing them, the total number of observations comes down to 9219. This hydropower data set was studied in a preliminary effort [52], which presents additional details of the data preprocessing step.

Now, the data is ready for applying an anomaly detection algorithm. Besides LoMST, we have also applied two other popular anomaly detection methods on the same hydropower data. The two other methods are: LOF [19] and SOD [36] methods, and both of them were previously applied to the hydropower data set in our preliminary study [52]. LOF represents the 12 neighborhood-based methods considered for comparison in Section IV and is arguably the most popular method in the anomaly detection literature. SOD is a representative of the subspace methods, and we are curious to see how a subspace method could do to the real application data. However, we want to mention here that finding the right subspace is usually even harder than finding the candidate anomalies, and for this reason, the neighborhood based methods often outperform the subspace based methods. For instance, if we compare SOD based on the practical k approach with LoMST and the other 13 methods in Section IV, its ranking in the four categories would be 0, 0, 8, and 12 and its average ranking would be 6.7, much worse than the top competitors.

For all three methods, we need to specify the value of the k-NN. In this case, we were lucky enough to get a suggestion from the domain expert about the possible size of an anomaly cluster. Based on the domain expert's suggestion, we decide to consider the value of k in a range of 10–20 and find the anomaly scores for each k in the range. Then, we took the average of these scores as the final anomaly score for each of the instances. We followed this principle for both the LOF method and our LoMST method. For SOD, we need to select two parameters instead of one: one is k, while the other one is the number of reference points for forming the subspace. To maintain the comparability with LoMST and LOF, we choose $k = 15$ for SOD, which is the middle point of the above-suggested range. Concerning the number of reference points, it should be smaller than k but not too small a value that may render instability in SOD. We explore a few options and finally settle on 10. Below 10, the SOD method becomes unstable.

TABLE IX

SUMMARY OF THE TOP 100 ANOMALIES RETURNED BY THE THREE METHODS. EVENTS FOLLOWED BY ASTERISK (*) ARE THE COMMON ONES IDENTIFIED BY ALL THREE METHODS IN THE TOP 30 TIMESTAMPS

LoMST	LOF	SOD
1/12/2016 11:20*	1/12/2016 11:30*	9/14/2015 8:00
9/14/2015 1:00*	9/14/2015 1:00*	1/12/2016 11:30*
1/2/2016 9:10*	9/14/2015 1:10*	9/13/2015 7:00*
1/11/2016 12:00*	1/12/2016 11:20*	7/4/2015 8:30
1/2/2016 9:30*	1/9/2016 6:50*	7/4/2015 8:20
7/4/2015 11:20	1/2/2016 9:10*	9/14/2015 1:50
7/4/2015 11:10	9/14/2015 8:00	7/4/2015 5:40
7/4/2015 11:30	1/2/2016 9:20*	1/11/2016 12:00*
1/9/2016 6:50*	1/9/2016 18:30	9/14/2015 1:00*
7/4/2015 10:40	9/14/2015 8:10*	10/3/2015 14:40
1/2/2016 9:20*	9/13/2015 7:00*	7/4/2015 5:50
7/4/2015 9:40	9/14/2015 2:00	10/13/2015 8:15*
9/13/2015 7:00*	1/11/2016 14:40	9/14/2015 1:10*
1/11/2016 1:30*	1/11/2016 13:50	11/2/2015 9:56
7/4/2015 9:50	1/11/2016 12:00*	7/4/2015 6:30
9/16/2015 10:50*	1/11/2016 13:00	7/4/2015 4:30
9/14/2015 14:10	9/16/2015 10:50*	1/2/2016 9:20*
9/14/2015 13:50	9/17/2015 11:30	9/14/2015 2:00
7/4/2015 5:20	10/3/2015 14:40	9/14/2015 8:10*
9/14/2015 1:10*	1/2/2016 21:40	7/4/2015 4:20
1/12/2016 11:40	4/16/2015 23:10	1/11/2016 1:30*
1/12/2016 11:30*	10/4/2015 3:10	1/2/2016 21:40
9/14/2015 13:20	10/13/2015 8:15*	7/4/2015 4:40
7/4/2015 4:50	10/14/2015 23:35	9/16/2015 10:50*
9/14/2015 8:10*	10/14/2015 23:15	1/2/2016 1:30*
4/16/2015 23:10	1/2/2016 9:30*	1/11/2016 14:40
4/16/2015 16:00	4/16/2015 16:00	1/2/2016 9:10*
10/13/2015 8:15*	11/2/2015 9:56	1/12/2016 11:20*
7/4/2015 5:30	1/11/2016 1:30*	1/9/2016 6:50*
7/4/2015 9:10	1/11/2016 11:50	9/14/2015 13:05
....	....	....
9/13/2015 19:10	10/13/2015 5:45	7/4/2015 0:00
7/4/2015 4:40	1/2/2016 21:00	7/4/2015 5:30
7/4/2015 6:20	1/9/2016 18:20	7/4/2015 6:20
7/4/2015 5:00	1/9/2016 18:40	7/4/2015 6:50
7/4/2015 13:50	9/14/2015 0:40	7/4/2015 7:00
....	....	....
9/14/2015 8:00	9/14/2015 2:10	7/4/2015 7:50
1/9/2016 18:30	9/14/2015 8:20	10/13/2015 5:45
1/11/2016 13:00	9/14/2015 8:30	9/16/2015 11:00
1/11/2016 11:50	9/14/2015 8:40	10/13/2015 6:35
7/4/2015 9:30	10/14/2015 8:15	10/13/2015 8:25
....	....	....
10/14/2015 7:25	1/9/2016 18:00	10/4/2015 4:30
10/14/2015 7:35	1/11/2016 11:40	10/4/2015 4:20
7/4/2015 10:10	10/13/2015 6:35	1/2/2016 21:50
7/4/2015 10:20	10/4/2015 23:10	1/11/2016 13:50
7/4/2015 10:30	9/13/2015 19:30	9/13/2015 21:40
....	....	....
7/4/2015 10:50	1/9/2016 18:10	1/11/2016 12:10
1/11/2016 13:40	1/11/2016 13:40	1/9/2016 18:30
1/11/2016 13:50	9/13/2015 19:40	10/4/2015 3:10
10/4/2015 3:10	10/14/2015 7:55	1/11/2016 11:30
1/9/2016 18:40	1/11/2016 11:30	10/14/2015 7:25

By applying the three methods, the top 100 anomalies that are identified by them are shown in Table IX. We noticed that after the top 30 time stamps, no new anomaly prone day emerged and similar data patterns keep repeating themselves with slight differences in the time stamps. To save space, we skip some rows after the top 30 stamps.

The performance of the three methods are reasonably consistent as 14 out of the top 30 probable anomalies identified

TABLE X
MOST ANOMALY PRONE DAYS IDENTIFIED BY ALL THREE METHODS

July 4th, 2015
September 13th, 2015
September 14th, 2015
October 13th, 2015
October 14th, 2015
January 2nd, 2016
January 9th, 2016
January 11th, 2016
January 12th, 2016

by these methods are common, represented by an asterisk (*) in Table IX. This similarity continues even if we go beyond 30 time stamps. By closely looking at these top 100 time stamps, we find that there are some particular days and certain time chunks in these days which are more prone to anomaly. These more anomaly prone days are listed in Table X.

As we move out from the range of top 30 time stamps, there are slight differences in the time stamps returned by individual methods, but they are very close time wise (within 10–50 mins range) in the same day. The most possible explanation behind this phenomenon is anomalies appeared in a small cluster.

These three methods work differently, especially that SOD is from another family of methods and completely different from LOF and LoMST. In spite of their differences, they have returned similar results for the hydropower data set. This serves as a way to cross validate the detecting outcomes, as the true anomalies are unknown. We reported the top 100 time stamps as anomalies to the data provider. The domain expert checked the physical system and agreed that these present valid concerns and the method provides valuable tips for trouble shooting.

The three methods do have differences in their detection outcomes. LOF method completely missed the July 4th time stamps, although almost half of the 100 top anomaly prone timestamps returned by both SOD and LoMST method belong to this day. We investigate the issue and find that most of the timestamps in July corresponds to low active power, whereas the timestamps from July 4th are marked with abnormally high active power. The rest of the attributes behaves identically as other days of July. We know that when the number of attributes increases, nearest neighborhood methods usually fall short of detecting anomalies if abnormal values only happen to one or few dimensions. This is where the subspaces method can do better (assuming that the abnormal value subspace is successfully identified). It is therefore not surprising to see that the SOD method detects these anomalies correctly, but it is truly encouraging to see that LoMST is capable of detecting these anomalies as well, even though LoMST is a neighborhood-based method. It supports our claim that MST approximates the intrinsic distance among observations in a structured data space. On the other hand, LOF and LoMST, being a local method, successfully identified point anomalies on the April 16th, while SOD method failed to identify them. In a nutshell, LoMST method attains the merit of subspace-based methods without losing the benefits of local neighborhood-based methods.

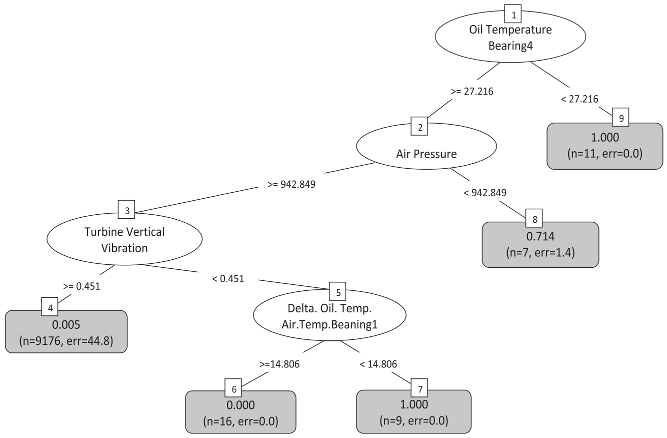

Fig. 8. Decision tree based on the anomalies identified by LoMST method.

Anomaly detection does not immediately reveal the root causes causing the anomalies. Finding out which variables contribute to the anomalies provides a nice interpretability of each anomaly and helps the domain expert to verify the root causes and then fix them (if genuine). Given that the anomalies are identified now, the original unsupervised learning problem is translated into a supervised learning problem, and for that, we build a classification and regression tree [53] using the R package rpart with the package's default parameter values. We at first discard the July 4th time stamps from the top 100 timestamps, as the reason for their happening is straightforward. We proceed with the remaining of the top 100 timestamps and assign them a response value of 1, and all other data records (outside the top 100 timestamps) in the data set a response value of 0 (meaning normal condition). The resulting tree is shown in Fig. 8.

From this decision tree, we can see that the variable *Oil Temperature of Bearing 4*, *Air Pressure*, *Turbine Vertical Vibration*, and *Delta Oil temp–Air Temp of Bearing* 1 can correctly classify 25 anomalies based on the right combination of their conditions. One such condition is when the oil temperature of bearing 4 is less than 27.216 °C, the turbine generator almost surely behaves strangely, and this condition consistently leads to 11 anomalous observations. When we report this finding to the domain expert, he deems this a key finding. During the subsequent preventive maintenance operation of the said turbine, it is confirmed that bearing 4 indeed needs repair to avoid future damage or costly interruption of the turbine operation.

VI. CONCLUSION

We proposed a new dissimilarity metric based on the concept of MST for isolating local, pointwise anomalies from the normal observations in a structured data space. Rather than applying MST to the entire data set, we choose to follow a two-stage procedure. At first, a global MST is used to separate the distant anomalous clusters from the rest of the data set. Then, we build local MSTs for the remaining instances to detect pointwise anomalies. The proposed MST-based method is effective and registers the best performance when it is compared with a wide array of methods on 20 benchmark data sets. The superiority of the proposed method inspires us to apply it to a real life hydropower data set. The MST-based method is successful in detecting different families of anomalies, achieving the merit of subspace-based methods without losing the benefits of local neighborhood-based methods. The validity of the anomalies detected is cross validated by two other anomaly detection methods and confirmed by the domain experts and maintenance operators who provided us the hydropower data in the first place. Root causes and threshold values for key attributes that contribute to the anomalies are determined in the form of a decision tree. The knowledge generated from the anomaly detection analyses helps service engineers continuously monitor the turbine operation and potentially diagnose and predict the malfunctions of turbines in time.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[3] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[4] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, p. e0152173, 2016.

[5] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statist. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[6] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[7] M. Yu *et al.*, "Hierarchical clustering in minimum spanning trees," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 25, no. 2, p. 023107, 2015.

[8] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2334–2342.

[9] J. Costa and A. Hero. (2003). "Manifold learning with geodesic minimal spanning trees." [Online]. Available: https://arxiv.org/abs/cs/0307038

[10] 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* (Lecture Notes in Computer Science), vol. 5476. Berlin, Germany: Springer, 2009, pp. 813–822.

[11] P. K. Sujatha, R. Arun, P. Shanthoosh, I. E. P. Jebahar, and A. Kannan, "Network level anomaly detection system using MST based genetic clustering," in *Advances in Network Security and Applications* (Communications in Computer and Information Science), vol. 196. Berlin, Germany: Springer, 2011, pp. 113–122.

[12] X. Wang, X. L. Wang, and D. M. Wilkes, "A minimum spanning tree-inspired clustering-based outlier detection technique," in *Advances in Data Mining. Applications and Theoretical Aspects* (Lecture Notes in Computer Science), vol. 7377. Berlin, Germany: Springer, 2012, pp. 209–223.

[13] A. O. Hero, "Geometric entropy minimization (GEM) for anomaly detection and localization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 585–592.

[14] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern Recognit. Lett.*, vol. 22, nos. 6–7, pp. 691–700, 2001.

[15] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proc. 24th Int. Conf. Very Large Data Bases*, 1998, pp. 392–403.

[16] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, vol. 29, no. 2, pp. 427–438.

[17] F. Angiulli and C. Pizzuti, "Outlier mining in large high-dimensional data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 2, pp. 203–215, Feb. 2005.

[18] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Proc. 13th Pacific-Asia Conf. Knowl. Discovery Data Mining*, Berlin, Germany: Springer, 2009, pp. 813–822.

[19] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, vol. 29, no. 2, pp. 93–104.

[20] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1649–1652.

[21] E. Schubert, A. Zimek, and H.-P. Kriegel, "Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

[22] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *Proc. 5th Int. Conf. Mach. Learn. Data Mining Pattern Recognit.*, Berlin, Germany: Springer, 2007, pp. 61–75.

[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[25] L. Ertöz *et al.*, "Minds—Minnesota intrusion detection system," in *Next Generation Data Mining*, H. Kargupta, J. Han, P. S. Yu, R. Motwani, and V. Kumar, Eds. Cambridge, MA, USA: MIT Press, 2004, ch. 3, pp. 199–218.

[26] D. Yu, G. Sheikholeslami, and A. Zhang, "*FindOut*: Finding outliers in very large datasets," *Knowl. Inf. Syst.*, vol. 4, no. 4, pp. 387–412, 2002.

[27] A. M. Pires and C. Santos-Pereira, "Using clustering and robust estimators to detect outliers in multivariate data," in *Proc. Int. Conf. Robust Statist.*, 2005, pp. 1–2.

[28] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda, "Towards NIC-based intrusion detection," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 723–728.

[29] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1641–1650, 2003.

[30] M. Amer and M. Goldstein, "Nearest-neighbor and clustering based anomaly detection algorithms for RapidMiner," in *Proc. 3rd RapidMiner Community Meeting Conf. (RCOMM)*, 2012, pp. 1–12.

[31] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[32] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[33] C. C. Aggarwal and P. S. Yu, "An effective and efficient algorithm for high-dimensional outlier detection," *Int. J. Very Large Data Bases*, vol. 14, no. 2, pp. 211–221, 2005.

[34] J. Zhang, M. Lou, T. W. Ling, and H. Wang, "HOS-miner: A system for detecting outlying subspaces of high-dimensional data," in *Proc. 30th Int. Conf. Very Large Data Bases (VLDB)*, 2004, pp. 1265–1268.

[35] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "Outrank: Ranking outliers in high dimensional data," in *Proc. IEEE 24th Int. Conf. Data Eng. Workshop*, Apr. 2008, pp. 600–603.

[36] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proc. 13th Pacific–Asia Conf. Knowl. Discovery Data Mining*, Berlin, Germany: Springer, 2009, pp. 831–838.

[37] F. Keller, E. Müller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking," in *Proc. IEEE 28th Int. Conf. Data Eng. (ICDE)*, Apr. 2012, pp. 1037–1048.

[38] B. van Stein, M. van Leeuwen, and T. Bäck, "Local subspace-based outlier detection using global neighbourhoods," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2016, pp. 1136–1142.

[39] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 157–166.

[40] A. Zimek, R. J. G. B. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: Challenges and research questions a position paper," *ACM SIGKDD Explorations Newslett.*, vol. 15, no. 1, pp. 11–22, 2014.

[41] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, 1957.

[42] J. B. Kruskal, Jr., "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, 1956.

[43] J. Nešetřil, E. Milková, and H. Nešetrilová, "Otakar Borůvka on minimum spanning tree problem translation of both the 1926 papers, comments, history," *Discrete Math.*, vol. 233, nos. 1–3, pp. 3–36, 2001.

[44] D. C. Montgomery, *Introduction to Statistical Quality Control*. Hoboken, NJ, USA: Wiley, 2009.

[45] G. O. Campos *et al.*, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining Knowl. Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[46] H. Hotelling, "The generalization of student's ratio," *Ann. Math. Statist.*, vol. 2, no. 3, pp. 360–378, 1931.

[47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[48] W. Conover, *Practical Nonparametric Statistics*. Hoboken, NJ, USA: Wiley, 1999.

[49] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," *J. Roy. Statist. Soc. B*, vol. 57, pp. 289–300, Jan. 1995.

[50] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[51] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[52] I. Ahmed, A. Dagnino, A. Bongiovi, and Y. Ding, "Outlier detection for hydropower generation plant," in *Proc. 14th IEEE Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2018.

[53] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.

Imtiaz Ahmed received the B.Sc. and M.Sc. degrees in industrial and production engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Industrial and Systems Engineering Department, Texas A&M University, College Station, TX, USA.

He was a Faculty Member with the Bangladesh University of Engineering & Technology. His research interests are in data analytics, machine learning, and quality control.

Aldo Dagnino received the M.A.Sc. and Ph.D. degrees from the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada, in 1984 and 1987, respectively.

He has over 20 years of experience in the software industry. He is currently leading the Advanced Analytics Global IS group at ABB, Cary, NC, USA. From 2009 to 2016, he led the Industrial Analytics Research work at ABB Corporate Research, where he worked in Industrial Analytic applications and successfully completed several projects in this field that resulted in commercial products and services for ABB. He is currently an Adjunct Assistant Faculty Member with the Department of Computer Science, North Carolina State University, Raleigh, NC, USA, where he is involved in collaborative research with other faculty members and teaches graduate courses in the department. His applied research areas include software engineering, software architectures, data mining, and knowledge-intensive systems applied to industrial systems.

Yu Ding (M'01–SM'11) received the B.S. degree from the University of Science and Technology of China, Anhui, China, in 1993, the M.S. degree from Tsinghua University, Beijing, China, in 1996, the M.S. degree from Penn State University, State College, PA, USA, in 1998, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2001.

He is currently the Mike and Sugar Barnes Professor of Industrial and Systems Engineering and a Professor of Electrical and Computer Engineering with Texas A&M University, College Station, TX, USA. His research interests are in system informatics and quality and data science.

Dr. Ding is a Fellow of IIE and ASME and a member of INFORMS.