Anomaly Detection in Edge Nodes using Sparsity Profile

Aekyeung Moon
ETRI
akmoon@etri.re.kr

Xiaoyan Zhuo
UMass Lowell
xiaoyan_zhuo@student.uml.edu

Jialing Zhang
UMass Lowell
jialing_zhang@student.uml.edu

Seung Woo Son
UMass Lowell
seungwoo_son@uml.edu

Yun Jeong Song
ETRI
yjsong@etri.re.kr

***Abstract*—Edge devices with attentive sensors enable various intelligent services by exploring streams of sensor data. However, anomalies, which are inevitable due to faults or failures in the sensor and network, can result in incorrect or unwanted operational decisions. While promptly ensuring the accuracy of IoT data is critical, lack of labels for live sensor data and limited storage resources necessitates efficient and reliable detection of anomalies at edge nodes. Motivated by the existence of unique sparsity profiles that express original signals as a combination of a few coefficients between normal and abnormal sensing periods, we propose a novel anomaly detection approach, called ADSP (Anomaly Detection with Sparsity Profile). The key idea is to apply a transformation on the raw data, identify top-K dominant components that represent normal data behaviors, and detect data anomalies based on the disparity from K values approximating the periods of normal data in an unsupervised manner. Our evaluation using a set of synthetic datasets demonstrates that ADSP can achieve 92%–100% of detection accuracy. To validate our anomaly detection approach on real-world cases, we label potential anomalies using a range of error boundary conditions using sensors exhibiting a straight line in Q-Q plot and strong Pearson correlation and conduct a controlled comparison of the detection accuracy. Our experimental evaluation using real-world datasets demonstrates that ADSP can detect 83%–92% of anomalies using only 1.7% of the original data, which is comparable to the accuracy achieved by using the entire datasets.**

***Index Terms*—Transform Coding, Lossy Compression, Anomaly Detection, IoT**

I. Introduction

Internet of Things (IoT) enables connectivity for an extremely large number of devices, which consist of fine-grained sensors and actuators. Rapid advances in wireless sensors have been a key enabler for discovering actionable insights from raw data and ultimately making smart decisions. Under those knowledge discovery processes, IoT systems need to support time-sensitive applications by allowing data storage and processing to occur at or near edge nodes while minimizing use of limited storage, computing, and bandwidth. Performing data analysis locally, at or near edges, allows faster time-to-action than doing it at a remote data center or cloud.

The discovery paradigm where actionable knowledge is extracted from a steady stream of data collected from sensor nodes is increasingly adopted by various applications [1]. For example, traditional farms, which used to rely on human expertise, have been transformed into IoT-enabled agriculture, enabling precise and profitable operations such as the reduction of non-essential pesticides use [2], [3]. Similarly, an IoT platform called Waggle [4] improves the ability to collect real-time urban environmental data and ultimately enabling sustainable urban growth and smart city planning. For instance, microclimate data collected by Waggle nodes can be used as inputs to various simulation models to predict urgent climate conditions or air pollution levels. Furthermore, it can also use IoT devices or sensors in environments with unreliable network connectivity, such as rural agricultural plants or ecological research sites.

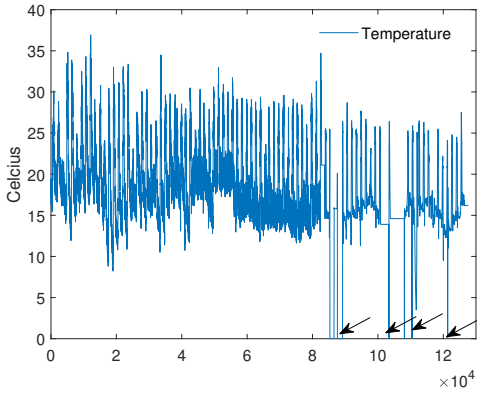


Fig. 1. Original datasets showing anomalous periods. Black arrows indicate apparent anomalous periods.

While the volume of data generated by IoT-enabled applications has increased at an unprecedented rate, it is practically infeasible to send all raw data to the remote cloud or data centers for post-processing. Due to the limited network bandwidth available, instead of transmitting all raw data, edge (sensor) nodes locally filter (such as hourly average) and perform analysis and periodically send significantly relevant aggregated data to remote nodes for long-term storage and big data analysis. Furthermore, the computing capability of most connected devices or physical sensors is limited. Under these inherently unfavorable circumstances, yet moving computation close to data is promising. There are, however, several challenges to exploit IoT datasets effectively and reliably in edge nodes.

First, data collected at sensor nodes are inherently inaccurate because of the faults or failures in sensors and unreliable network and can eventually generate incorrect or unwanted

978-1-7281-6251-5/20/$31.00 ©2020 IEEE

control operations or decisions in actuator nodes or gateways. These unexpected data frequently occur, especially in microclimate datasets collected in rural or urban settings. Figure 1 shows data samples containing several instances of actual anomaly data, which are considerably dissimilar from the remainder of the data. The presence of such anomalies must be detected and notified before control operations are performed [5], [6]; An effective anomaly detection empowers decision-makers to adequately react and take timely actions to correct anomaly situations [7].

Second, edge nodes usually have limited resources and capabilities in terms of processing power, bandwidth, energy, and storage [4], [8]. The scarcity of storage resources, in particular, could be a significant bottleneck as sensor nodes continuously collect data. Since sensor nodes send collected data to the gateways or cloud periodically, data volume during transmission needs to be minimized. Nevertheless, the transmitted data should be represented in the highest possible precision. Otherwise, analytic decisions derived from the collected data will have limited significance.

Lastly, the current state-of-the-art anomaly detection techniques are increasingly accommodating supervised machine learning (ML) methods where they learn to distinguish between healthy and faulty states after a training phase [9], [10]. In other words, ML-based anomaly detection models learn to classify normal and anomalous behaviors from *labeled* training data. However, it is often difficult to obtain large-scale datasets with proper labels. Furthermore, the annotation process requires domain knowledge from experts. Labeling the representative data patterns in real-world scenarios is also another challenging tasks as precise labeling depends on applications. Because of that, contextual anomaly such as [11] is often imprecise, and there is no single generic rule that applies to all IoT datasets of interest. For instance, in IoT farm datasets, the continuous change due to interaction among crop growth and operations of actuators (such as a heater, CO_2 generator for growing plants, fan, etc.) makes it almost impossible to distinguish between normal states and abnormal states except noticeably significant deviations.

Motivated by aforementioned challenges, we propose a novel approach, called ADSP (Anomaly Detection with Sparsity Profile), to detect data anomaly using a sparse representation of streams of sensor data. Our approach exploits lossy data compression (or approximation) to obtain unique sparsity profiles between normal and abnormal states during sensing periods. Our mechanism systematically collects original data, transforms them, obtains approximated data, and detects anomalies in an unsupervised manner. Many transformation methods provide a mechanism to reconstruct the signal close to the original using a minimal number of significant components [12]. After applying a transformation on the original datasets, we detect anomalies by inspecting on K-dominant components. Sensor nodes only need to process a small number of transformed data points, which optimally compacts a certain amount of signal energy (information).

We evaluate the effectiveness of ADSP against several state-of-the-art unsupervised anomaly detection techniques using a set of synthetic and real-world IoT datasets. Our experimental results demonstrate that ADSP can: 1) detect anomaly with high accuracy, 83%–92% and 92%–100% of detection accuracy for real-world datasets and synthetic datasets, respectively; 2) achieve competitive approximation ratios, 98.3% on average, with less influence on errors from approximation; and 3) provide the characterization criteria between normal and anomalous and a hypothesis for labeling data with a range of error thresholds.

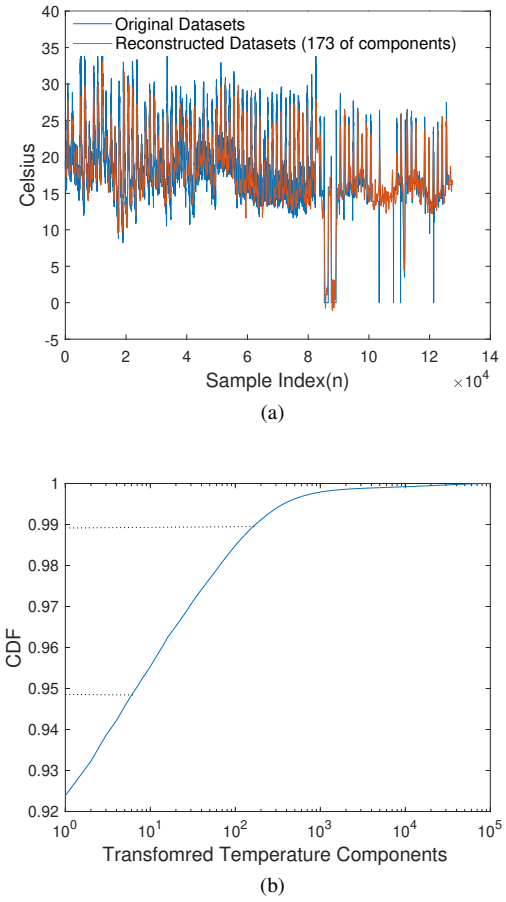


Fig. 2. The variation of "temperature" values. (a) Original and reconstructed data from approximated data. (b) Cumulative distribution function (CDF) in the sequences of DCT components.

II. Preliminaries

A. Lossy Compression

In lossy compression algorithms [13], the theoretical underpinning for our approximation mechanism, a signal can be sparse or compressible after applying signal transform with a suitable basis, e.g., discrete cosine transform (DCT), Fourier, or Wavelet basis [14]. In other words, unlike original signals, signals in a transformed domain often contain *only* a few significant components [12].

To demonstrate that similar sparsity exists in real-world IoT datasets, let us consider Figure 2a that depicts a case where

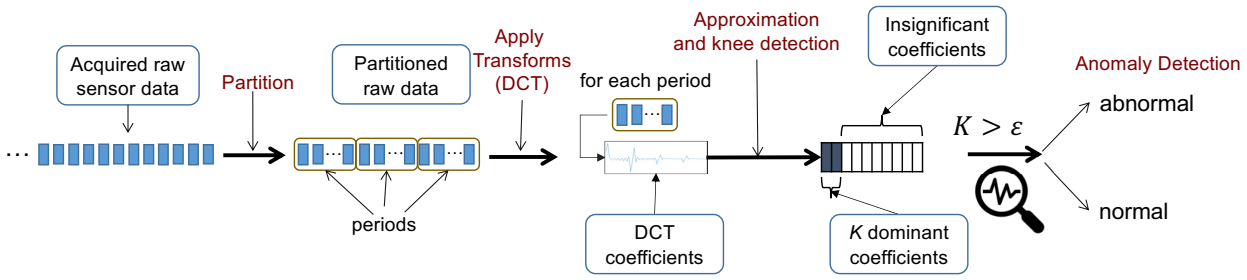


Fig. 3. Overview of our anomaly detection mechanism.

DCT components containing 99%-percentile of the energy attained by the original data account for only 0.14% (173 out of 127,667) of the entire data points. In signal theory, the energy spectrum of a given signal is the sum of the squared transform coefficients [14]. Figure 2b illustrates such high correlation through the energy cumulative distribution function (CDF) for coefficients after applying the transform on the original temperature data. Note that the x-axis is represented by a log scale to emphasize that most of the energy is concentrated on a small number of transformed components. Therefore, even though we sampled only a small number of dominant components, the approximated data can be reconstructed close to the original with a relatively small error [15]. As shown in Figure 2a, NRMSE (Normalized Root Mean Square Error) between reconstructed data points and original data points is only 0.1039.

While we can exploit such property in IoT datasets, applying conventional lossy compression techniques in the context of detecting anomalies in IoT datasets is challenging because selecting a suitable number of dominant coefficients is highly data-dependent. In other words, application exhibits different characteristics in the signal domain, so does the number of dominant coefficients that attain the same or similar energy compaction. Even in the same application, it varies in different sampling periods.

B. Anomaly Detection

Anomaly detection is the process of finding the patterns in data points whose behavior is not normal as expected [11], [16]. Those unexpected behaviors are also referred to as outliers [13]. Anomaly detection, which has been extensively studied, can be classified into three categories [9]: unsupervised, supervised, and semi-supervised. Under the assumption that the majority of the instances in the datasets are normal, the unsupervised method finds anomalies by looking for the instance that seems to be the least fit in the unlabeled datasets. Supervised anomaly detection techniques, on the other hand, require datasets that have been labeled as "normal" and "abnormal" and involve training a classifier. Semi-supervised anomaly detection techniques construct a model representing normal behavior from a given normal training datasets and then obtain the likelihood of a test instance generated by the learned model.

While the anomaly detection technique can be applied as supervised learning, this is not viable in many real-world scenarios because there are few or no labeled examples of anomalous behavior. In fact, in many cases, it is infeasible to label them manually [9], [17]. Moreover, it is hard to define normal data behavior in real-world scenarios where data patterns depend on the application domain as well as the data itself. Prior studies address this by learning anomaly conditions, building a prediction model, and comparing predicted values and measured values to determine whether there were anomalies or not. For instance, Haque et al. [5] used 30 samples as a history to build a prediction model. Nevertheless, in the real world IoT deployment where there is a frequent influence from the surrounding environment such as the microclimate in rural and urban settings, it becomes more difficult to distinguish normal data from anomalies.

III. Proposed Approach: ADSP

The relationship among the dominant components (denoted as K in this paper), anomalies, and scarcity of storage resources at the sensor nodes inspired us to design our data anomaly detection scheme, called ADSP, that includes overall processes from approximation to detection and reconstruction. We envision that the sensor or edge node can employ an efficient approximation algorithm that exploits the sparsity of sensing data to detect anomalies without relying on full datasets, which would incur a storage burden at sensor nodes. Figure 3 shows an overview of our approach. The sensor node transforms collected raw data into an approximated form in a fixed interval (or period) and detects data anomalies using profiles of these sparsely sampled data. ADSP detects anomalies by investigating only the K-dominant values, which are determined automatically based on the curvatures of the transformed coefficients, for approximated (sampled) data points without inspecting full original datasets.

A. Data Approximation

Our approach begins by approximating raw data to extract compact sparse representations. To illustrate ADSP's approximation mechanism, let us consider X, which denotes the transformed version of the original datasets (D). We first formulate this as the energy (or information) contained in the number of coefficients, denoted as K, of the entire sorted

transformed components ($X = \{X_1, X_2, ...X_n\}$), which is calculated as:

$$E(X_k) = \frac{\sum_{i=1}^{k} X_i^2}{\sum_{i=1}^{n} X_i^2}, N = 1, 2, ..., n, k \leq n. \quad (1)$$

Note that the sum of energy stored in the entire transformed components is 1.0 (or 100%-percentile). We then select the K-dominant components ($\{X_1, X_2, ...X_k\}$) from the transformed components to approximate the original datasets. One should expect K varies spatially (depending on data) and temporally. Thus, we need a systematic method to determine K. It should be mentioned that, unlike lossy compression techniques used in image and video data, we ignore the non-significant components other than the K-dominant components during approximation. We employ the DCT transformation mechanism, particularly the type-II DCT, as it is known to generate sparse representation effectively as compared with other signal transformation methods [14]. We later evaluate the effect of transform methods other than DCT in terms of data approximation in Section IV-B4.

The data approximation procedure based on the transformed data, in some more details, is as follows. First, we transform the original data, D, into DCT basis components to decorrelate data. In this way, the original data points $D(t)_i$ at t^{th} sampling period (P), where $1 \leq i \leq N$, are converted into the transformed data points, $X(t)$. After the transform, we acquire the full N-sample signal $X(t)$, sort $|X(t)|$ in descending order, and determine the K largest components (using Equation 1). As K-dominant components are selected from the sorted form $S(t)$ of $X(t)$, the approximated data points ($AD(t)$) include K-dominant components and their indices only. The index indicates the position of the selected component in $X(t)$, which is required to reconstruct data reliably later. Once K-dominant components are selected, non-significant components ($N-K$) are discarded. If we keep performing these processes every hour on data collected every second, it will locate only K-dominant components from 3,600 data points (i.e., 60 times 60), resulting in a significant data reduction.

While our approximation mechanism reduces data requirements by maintaining K components (and indices associated with them when the reconstruction is required), selecting the right K, especially in a systematic way, is not an easy task. One of the main reasons is that K is application-dependent. Even in the same data, it also varies in the spatial and temporal domains. In the following subsection, we will describe our novel mechanism to obtain optimal K.

B. Finding Optimal K

We adopt Kneedle algorithm [18] on the CDF of the coefficients' energy compaction [14] to determine K. Specifically, we first fit CDF into a smoothing spline such that it preserves the overall behavior of the energy distribution. Then we normalize the points in the best-fit curve to the unit square as a preprocessing step to eliminate anomalies (which can make the analysis more complicated). Next, we find the knee points (K) of the normalized curve, which is, in general, the point of maximum curvature of the normalized curve. In other words, the point is the local maxima, where it depicts the maximum distance between the normalized curve and the line $y = x$; mathematically, it is defined as a function of its first and second derivatives. We use a 1D interpolate function for obtaining smoothing curves, but a more sophisticated one such as polynomial interpolation function can also be applied.

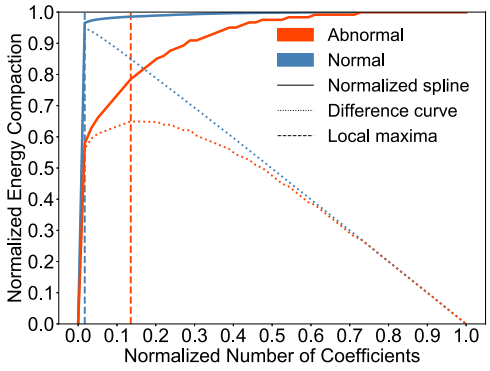
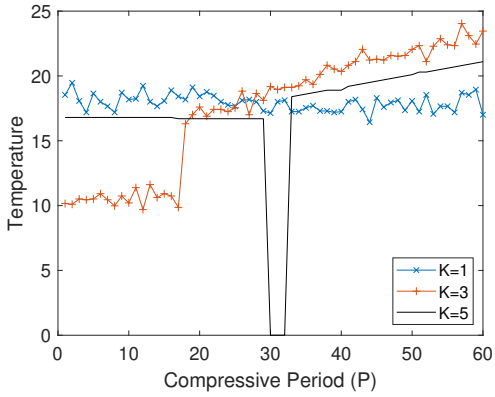


Fig. 4. Finding anomaly using knee detection algorithm.

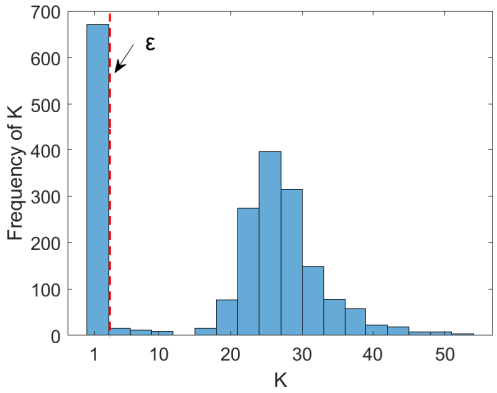
Figure 4 shows the use of knee detection algorithm in our anomaly detection on two selected periods in "humidity" dataset: normal (blue) and abnormal (red). The dotted lines depict the curve where each represents the difference between the normalized spline (solid line) and $y = x$. The dashed vertical lines depict the position of local maxima (maximum curvature of the normalized curve, i.e., maximum distance between spline and difference curve). From these curves, we can see that the abnormal period has a large K value (in this case, K=8) while the normal period shows a stable K value of 1. We attribute such disparate K values to the observed pattern that normal periods exhibit a pattern of high energy compaction and abnormal periods show a low energy compaction rate.

C. Data Anomaly Detection

Our anomaly detection mechanism utilizes the fact that the energy (information) is more dispersed when data has more unexpected patterns, which in turn requires more dominant components to maintain the same amount of energy. To illustrate how our anomaly detection mechanism exploits that property, let us consider the graph in Figure 5a, where the x-axis represents per-minute sampling points and the y-axis represents three examples of the temperature data that required different K values. In other words, to make all three temperature curves have the same curvatures, each requires different K values. The reason for such variation is that, when DCT (or any other equivalent transforms) is applied to data with an anomaly, the correlation between those data and K shows a distinctive difference. Figure 5a indicates that the sampling periods with anomalies require three or more dominant components, whereas periods with normal data require only one component. Figure 5b shows the histogram



(a) K-dominant coefficients



(b) Histogram of K-dominant coefficients

Fig. 5. Detection of anomalies using K-dominant coefficient. (a) K values for each period and (b) histogram of K.

of K considering all periods in the temperature dataset. Since there exist dominant K values, we can classify each period as normal or abnormal behavior according to K.

Our anomaly detection approach consists of two methods: binary detector and difference detector. The binary detector compares only K values and decides whether this period is normal or abnormal depending on if K is above a certain threshold, denoted as ε. The ε is determined by the average of K_n, where K_n indicates the majority of K values in normal periods. As an example, ε for Figure 5b is equal to 1. The ε, determined by the normal periods, will be used for classifying among normal and abnormal. Our methods reference K-dominant components of normal datasets as shown in Figure 3. The difference detector, on the other hand, calculates $|(K_n - K)|/Mean(X)$, where X denotes transformed coefficients, and classifies normal and abnormal state according to $K > K_n$. When $K > \varepsilon$, the K could be represented K_a.

D. Data Reconstruction

We have thus far discussed how our approach approximates the original data using top-K transformed components and how the variation in K per sampling period can be used for anomaly detection. In this subsection, we discuss how to reconstruct approximated data in sensor nodes or gateways/clouds to full data when there is a need for the whole data such as measuring reconstruction errors. As discussed in Section III-A, after sensor data are transformed, the selected K-dominant components along with the corresponding indices are maintained at edge devices or sent to the cloud or gateways if needed. Note that both K-dominant transformed components and their indices are required to define the approximated data points. Using $AD(t)$ (Approximated Data at period t), we generate $X(t)$ by replacing the indices of data points except for the indices of K-dominant with zeros. We then reconstruct data points $D'(t)$, which are generated from $X(t)$ by applying inverse transformation methods, i.e., IDCT (Inverse Discrete Cosine Transform) in our case. We use the reconstructed data to compare it against the original data and measure the quality of data approximation (later in Section IV-B5).

IV. Evaluations

A. Setup

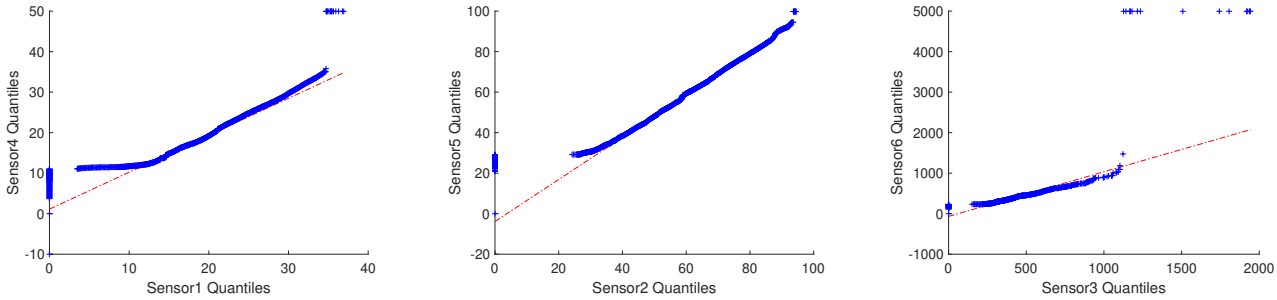
1) Datasets: We use multiple synthetic and real-world datasets in our evaluation. For generating synthetic datasets, we use utility functions provided in the PyOD library [19], which is an open-source Python toolbox. We generated both training (3,000*N) and test (10,000*N) datasets using the library, where N is the number of data points in each sampling P (described in Section III-A). When generating synthetic datasets, we use a range of contamination ratios from 0.1 to 0.4. The rate of contamination means the proportion of anomalies in the datasets.

For real-world datasets, we used the followings:

- **F1**: The IoT farm system deployed in Gangwon Province, South Korea, from October 1st to December 31st, 2017 (90 days of data collection) [20]. We collected the following three microclimate datasets in the deployed system: temperature, humidity, and CO (carbon monoxide). The data collection period is per minute.
- **F2**: The IoT farm system deployed in Chungnam Province, South Korea, from March 1st to December 31st, 2019 (304 days of data collection). The data collection period is per minute.
- **U**: The Waggle project is also targeting climate applications but in an urban environment setting [4], [21]. The Waggle dataset is time-series data, including air temperature, relative humidity, barometric pressure, UV light, IR light, and so on, collected at several urban locations in the US [4]. We evaluated temperature, humidity, and CO extracted in the big Chicago datasets measured from February 3rd, 2017 to July 4th, 2019.

The unit for temperature (C or Celsius), humidity (RH or Relative Humidity in %), and CO (ppm) are the same for all three datasets (F1, F2, and U).

Table I and Table II show the statistical properties of the both synthetic and real-world datasets in terms of STD (standard deviation), NSTD (normalized standard deviation),



(a) Temperature (b) Humidity (c) CO

Fig. 6. The normality test using Q-Q (Quantile-Quantile) plot for our evaluated datasets: (a) temperature, (b) humidity, and (c) CO.

TABLE I
THE CHARACTERISTICS OF SYNTHETIC DATASET AND APPROXIMATION RATIOS (AR).

Contamination	No. of Data points (N=60)	STD	NSTD	Skewness	Kurtosis	AR
0.1	10,000*N	2.2365	0.4141	-3.0276	12.8917	67.54
0.2	10,000*N	2.9440	0.6134	-2.0943	6.6107	56.99
0.3	10,000*N	3.4071	0.8112	-1.5455	4.2248	50.2
0.4	10,000*N	3.7157	1.0317	-1.1573	3.0262	45.18

TABLE II
THE CHARACTERISTICS OF REAL-WORLD DATASET AND ARS.

	No. of Data points	STD	NSTD	Skewness	Kurtosis	AR
F1-Temp	127,668	5.1532	0.2871	-0.2983	5.3136	99.86
F1-Humidity	127,668	15.5245	0.2700	-1.0632	5.4268	99.89
F1-CO	127,668	87.4290	0.2188	-1.4900	17.8790	99.42
F2-Temp	440,684	4.4801	0.2135	0.5871	3.3394	99.86
F2-Hum	440,684	13.7558	0.1542	-1.5751	6.5466	99.95
F2-CO	440,684	59.5058	0.1454	1.4967	6.0620	99.86
U-Temp	46,203,212	11.0143	0.8309	-0.0975	2.2491	85.13
U-Hum	8,762,131	18.81023	0.3036	-0.2474	3.0318	99.25
U-CO	19,401,868	1.0097e+04	1.2764	62.86	1.0397e+04	34.56

skewness, and kurtosis. As shown in Table I, the higher the contamination ratio, the higher STD. NSTD is calculated as $\frac{STD(x)}{Mean(x)}$. Skewness is a measure of data asymmetry around the mean value. In general, negative skewness means that more data are scattered in the left of the mean, whereas positive skewness means the opposite. Therefore, the normal distribution, where data is symmetric about its mean, gives zero skewness. Measures of kurtosis indicate how outlier-prone a distribution is. As the kurtosis of any normal distribution is 3, distributions with kurtosis higher than 3 are more outlier prone. As shown in Table II, U-CO exhibits higher STD than other datasets and has the highest kurtosis value among all datasets, which means it is more outlier prone. In the case of skewness, U-Temperature and U-CO only have positive values, which indicates they are mainly scattered in the right of the mean. AR means the compression ratio of all datasets without considering P.

2) Labeling Rule for Datasets: For the synthetic datasets, the utility function in PyOD labels each data point based on the ratio of contamination. For instance, 10% of datasets are set to anomaly data when the contamination is 0.1. In the case of real-world datasets, however, there is no easy way to label each data point using the same mechanism as the synthetic datasets. However, we already deployed multiple sensors of the same types in our farm environment (F1 and F2) for evaluating the integrity of sensors, so we use an analytical comparison of identical sensors in real-world scenarios as ground truth for labeling datasets. The hypothesis is that the same types of sensors are deployed closer than others to evaluate how sensor values are spatially and temporally correlated.

To characterize normal period behavior precisely, we first test the normality of datasets using the quantile-quantile (Q-Q) plot of sensors. Figure 6 shows a normal Q–Q plot that compares datasets (temperature, humidity, and CO). The linearity of the points in Figure 6 suggests that the data are normally distributed. Especially the plot of "temperature" produces an approximately straight line, which means it follows a normal distribution more than the others. We also use a Q-Q plot to determine whether two sets of sample data come from the same distribution or not. As shown in Figure 6, [Sensor 1, Sensor 4] in the F1 dataset shows almost the same distribution. To confirm this, we also measured the Pearson correlation coefficient between sensors and chose three pairs of a sensor for each farm as the guidelines for analyzing the strength of the correlation [22]. Our measurement confirmed that the correlations between the same types of sensors are high. Specifically, we have 0.8285, 0.8487 and 0.406 for three groups [Sensor 1, Sensor 4], [Sensor 2, Sensor 5], [Sensor 3, Sensor 6] of F1. We then used the result from this step as our ground truth for training and test.

3) Evaluated Schemes: We compare our approach with several unsupervised anomaly detection techniques. Since unsupervised anomaly detection does not require any labels, there is no distinction between a training and a test dataset. Typically, distances or densities are used to estimate normal or outlier. We choose two state-of-the-art unsupervised anomaly detection methods [23]: local outlier factor (LoF) [24] and autoencoder (AE) [25], [26]:

- **LoF**: It measures the local deviation of the density of a given sample with respect to its neighbors. The anomaly score depends on how isolated the object is with respect to the surrounding neighborhood.

- **AE**: It can detect outlying objects in the data by calculating reconstruction errors.
- **ADSP**: This scheme is our proposed method described in Section III.

4) Evaluation Metrics: We use the following metrics to assess the overall anomaly detection rates and the quality of the approximated data.

- Accuracy can be calculated as:
  Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$ where TP, TN, FP, and FN refer to true positive, true negative, false positive and false negative, respectively.
- Approximation Ratio (AR) is given by:
  $AR = \frac{|D|-|D'|}{|D|} \times 100\%$,
  where $|D|$ is the size of D, $|D'|$ is the approximated data size.
- The error rate is assessed using PSNR (Peak Signal-to-Noise Ratio), which measures the overall distortion between the original data and the reconstructed data. PSNR is expressed in terms of the logarithmic decibel scale:
  $PSNR = 20 \cdot log10(\text{value range}) - 10 \cdot log10(MSE)$,
  where value range and MSE refer to data value range and the mean squared compression error, respectively.

The objective of our performance metrics is to achieve higher AR, PSNR, and detection accuracy.

B. Results

1) Detection Accuracy using Synthetic Data: Figure 7 shows the detection accuracy of the evaluated schemes while varying the ratio of data contamination. The contamination ratio of 0.1 means that 10% of the anomaly data are included in training and test datasets, respectively. As shown in Figure 7, ADSP not only can detect 92%-100% of anomalies but also improves the detection accuracy as the contamination ratio increases. The reason for ADSP's higher detection accuracy with higher contamination ratios is that, when there were more anomalies, K would increase accordingly, consequently adapting well with increasing contamination ratios. In ADSP, the threshold values ε of K for distinguishing normal and abnormal from training datasets we obtain are 9, 13, 17 and 21 for the contamination of 0.1, 0.2, 0.3 and 0.4, respectively. We did not evaluate the contamination of 0.5 or beyond because such a high contamination ratio makes it difficult to distinguish between normal and abnormal conditions.

AE and LoF, on the other hand, showed decreasing detection accuracies as the contamination ratio increases. The reason for the decreasing accuracy is that an AE-based anomaly detector uses the reconstruction error as an anomaly score. Because of this, AE shows better detection accuracy than ADSP when the contamination ratio is 0.1, i.e., relatively few anomalies exist. However, this means that AE becomes unreliable in the practical unsupervised case, where the training data may contain more anomalous examples [26], thus suffering from the increasing contamination ratios. In LoF, because the degree of anomaly depends on how isolated the object is with respect

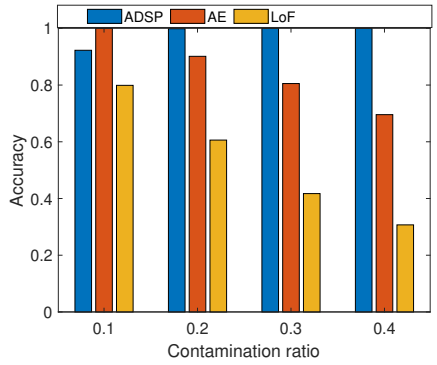


Fig. 7. Comparison of anomaly detection techniques for synthetic datasets with varying contamination rates.

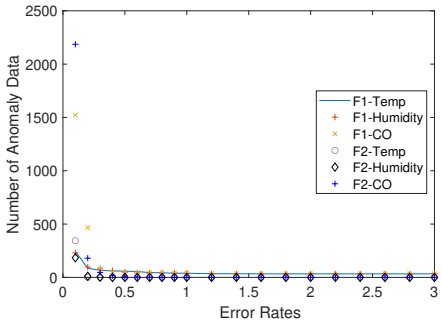


Fig. 8. The number of anomaly data based on error rates.

to the surrounding neighborhood, it is more difficult to decide the anomaly threshold for each period than to find anomaly points.

2) Detection Accuracy using Real-World Data: Since the measurement from the paired sensors with high linearity and correlation in Q-Q plots and Pearson correlation provides us with a mechanism to control confidence on our dataset, we can apply statistical methods in our classification. More specifically, we use NRMSE (the normalized version of root mean square error) to characterize anomalies. That is, we classify the sensor data in the case of anomaly conditions according to the NRMSE values. Let $x = \{x_1, x_2, x_3, ..x_n\}$ be the first sensor data, $\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, ...\hat{x}_n\}$ be the second sensor data, and N be the number of data points.

- NRMSE for each period P_t can be calculated as:
  $NRMSE(t) = \frac{RMSE(t)}{Mean(x)} = \frac{1}{\bar{x}}\sqrt{\frac{\sum_{n=1}^{N}(x(n)-\hat{x}(n))^2}{N}}$.

Any data points whose NRMSE is larger than the error threshold α will be considered as an anomaly.

To give overall characteristics about anomalous data, Figure 8 presents results for F1 and F2 while varying NRMSE values. As expected, there are more data anomalies as we increase α. In the case of temperature, 86 periods are labeled as anomaly states when α is 2.0. Comparatively, there are more periods with data anomalies in CO than the other two datasets. This is because the CO sensor has relatively higher data variation than that of the other sensors. As we expected,

F2 datasets have fewer anomaly data in the regard of NRMSE than F1. Because there are little anomaly data points in F2 datasets, the anomaly detection accuracy in F2 datasets has less significance. Therefore, we evaluate the detection accuracy using the F1 dataset.

Based on the labeling from using sensors with strong relationships, we measure detection accuracy using the F1 dataset. In our context, TP is the case where K is greater than ε and NRMSE is greater than α. Recall that we automatically set ε, which is a threshold for distinguishing between normal and abnormal.

Figure 9 shows that the accuracy while increasing α values. Recall that α is the error threshold determined using unsupervised schemes. Specifically, we evaluate the error rate from 0.1 to 0.5 because the number of anomaly data based on NRMSE shown in Figure 8 becomes steady beyond 0.5. Overall, higher α means more tolerable sensor variations, thus resulting in fewer anomaly points. As shown in Figure 9, ADSP's detection accuracy is higher than those of AE and LoF. In the CO data, ADSP has even higher accuracy than others. We attribute this to ADSP's detection models, which are more reliable than the other schemes when there are more anomaly data points like CO by determining K value automatically with the varying characteristic of data. ADSP shows 83%-92% the accuracy of anomaly detection in the case of temperature data.

3) Variations of K: ADSP's high detection accuracy on synthetic and real-world datasets presented so far is based on our hypothesis that an anomaly can be detected by the relationship among the K-dominant coefficients. To confirm this, Figure 10 shows the variation in K values for F1 datasets from our knee detection algorithm (described in Section III-A). In Figure 10, the primary x- and y-axis (in blue color) represents the index of data points and its values, respectively, whereas the secondary x- and y-axis (in orange color) represents the compressive period and the K-dominant components required to approximate every period, respectively. As we can see, K is notably different in periods where there are anomalies. For example, in the case of temperature data (shown in Figure 10a), anomalies occurred around the $1{,}422^{th}$ periods (or 85,320 sample index). The datasets in this period are anomalous due to unexpected network outages.

We also measured the variation of K for the F2 and U datasets and observed that our mechanism works well with them as well. While we observe similar trends in overall (i.e., relatively higher K values in abnormal periods), each dataset showed minor differences. First, as shown in Table II, F2 has less variation than F1, thus smaller variations in K values in both K values themselves and their range of fluctuation. As compared with the other datasets, the U dataset showed the largest K values, which is on a par with the statistical characteristic of datasets; all datasets in U showed the highest variations in Table II.

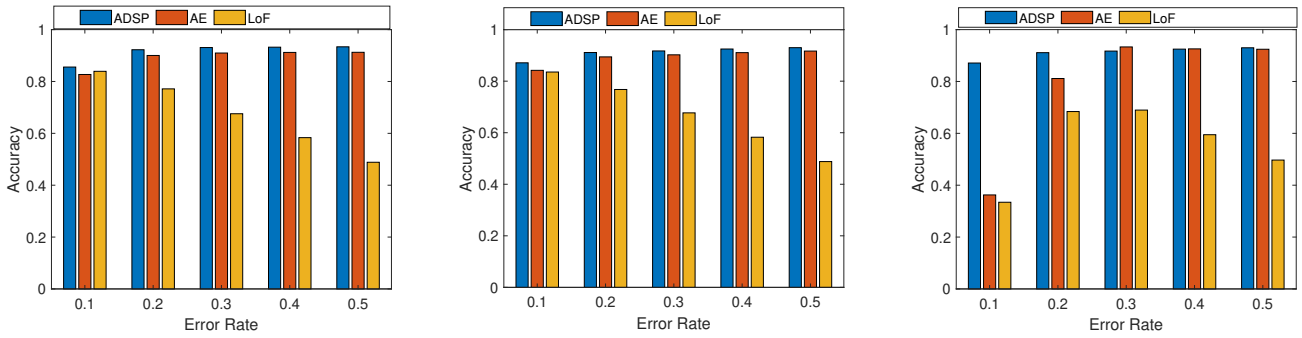
4) Approximation Ratios (AR): In our evaluated datasets, the data are collected every minute, and we approximate every 60 raw data points. In other words, the approximation period is set to 1 hour. Note that the approximation ratios for our approach vary depending on periods, whereas the hourly average has fixed 1/60 ratios (or 98.3%). In other words, only 1.7% of data is required. While one can use more complex data approximation methods, we compare our algorithms with hourly averaged data as it is one of the most commonly used techniques in widespread edge deployments. We also measured the data approximation approach of ADSP with FWHT, a faster version of the Walsh-Hadamard Transform (WHT), in addition to DCT. ADSP is also a transform-based method because it adopts DCT transformations. In terms of compressibility, the F1 dataset shows higher approximation ratios than FWHT as shown in Table III. Among different transformation techniques, we observed that our approach shows better approximation ratios. As invested in [12], DCT and FWHT show better compression ratios than other transformation methods. We also observe that our approach can achieve up to 98.3% of approximation ratios, which is the same as the hourly average. Note that the approximation ratios presented in Table III are based on all sampling periods, including both normal and abnormal cases.

TABLE III
COMPARISON OF ERROR RATES (IN TERMS OF PSNR) AND APPROXIMATION RATIO FOR OUR APPROACHES WITH DIFFERENT TRANSFORMS AND HOURLY AVERAGES.

	Temperature		Humidity		CO	
	PSNR	AR	PSNR	AR	PSNR	AR
ADSP	32.41	98.3%	31.86	98.3%	34.08	98.1%
FWHT	32.14	98.1%	31.69	98.5%	34.00	98.2%
Hourly average	31.77	98.3%	31.08	98.3%	33.54	98.3%

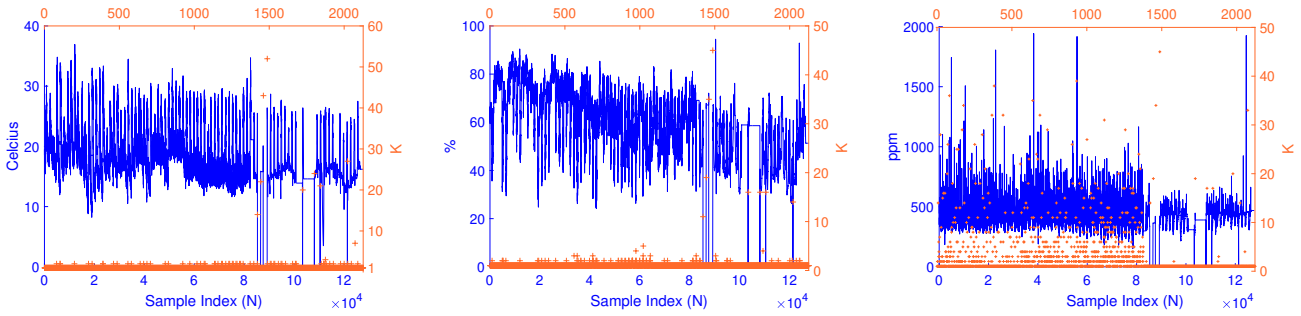
5) Error Rates: Since our approximation mechanism is based on lossy compression techniques, it also minimizes errors when data is reconstructed even using only K-dominant coefficients. Table III shows the error rates between the reconstructed data using our approach and the conventional hourly average value of data. In our approach, K varies depending on data. So, for a fair comparison, we measured error rates using a fixed energy compaction rate of 95%-percentile. In the case of our approximation approach, the error rate is slightly lower (i.e., slightly higher PSNR), while both ADSP and the hourly average have the same approximation ratios (98.3%). The results show that sparse signals extracted from the original datasets (collected every minute) can be more accurate than hourly averages because PSNR of our approach is higher than that of hourly averages. Recall that higher PSNR represents less error that affects data quality.

Table III also shows the relationship between AR and PSNR for ADSP and FWHT when energy is 0.95 (95%-percentile) (in terms of Equation 1). Overall, PSNR of 95%-percentile, while it has comparable approximation ratios, is higher than that of the hourly average. Overall, our approach achieves a quite competitive approximation ratio while maintaining a relatively low error rate.



(a) Temperature (b) Humidity (c) CO

Fig. 9. Comparison of anomaly detection accuracy for F1 datasets. (a) temperature, (b) humidity, and (c) CO.



(a) Temperature (b) Humidity (c) CO

Fig. 10. Variation of K-dominant components in F1 datasets. We also see similar variations of K in the other datasets: F2 and U.

V. Related Work

Statistical methods such as exponentially weighted moving average or cumulative sum [27] have been used for detecting anomalous behaviors in time series data. Anomaly detection for AWS IoT data [6] uses PEWMA (Probabilistic Exponentially Weighted Moving Average) proposed by Carter and Streilein [28], but they did not consider the impact of approximated data on detecting data anomalies. In other words, it requires all original data points in their anomaly detection model. Recent studies proposed several methods to detect anomalies while using compression techniques. For instance, Kartakis and McCann [8], [29] presented that anomalies can be identified by analyzing significant changes in the compression rate. Kartakis et al. [29] also used compression rate fluctuation to detect anomalies. However, these prior studies mainly used lossless compression methods with a high sampling rate so that the compression rate is usually lower compared to lossy compression techniques [30]. In several prior studies such as analyses of turbulent flow data [31] and climate data [32], data reconstructed from lossy compression allows meaningful analysis to be carried out. Kartakis et al. [30] recently used adaptive compressive sensing mechanisms in smart water networks but for reducing data transfer overheads. Shah et al. [33] proposed a compression-based approach for detecting anomalies in edge-attributed networks.

We proposed an anomaly detection using lossy compression based on an approximated form out of original datasets with a high compression capability in our previous research [20]. Our proposed mechanism is similar to approaches based on compressive sensing [20], [30], [34]–[36] in that, it shares the same notion that data can be reconstructed from a small number of samples or projections. It has been shown that compressive sensing is beneficial for solving the distributed outlier detection problem in distributed computing [37]. However, our previous approach does not consider K for the normal datasets and the validation of labeling approach.

VI. Conclusions

In this work, we proposed a novel approach to detect anomalies using a sparse representation of IoT datasets. The motivation behind our method is that microclimate datasets collected at rural and urban sensor nodes exhibit a certain degree of sparsity. We use discrete transformations, specifically DCT, to reveal the signal's local behavior, which effectively leads to such sparse representation. Our technique also exploits the high correlation between the number of dominant transform coefficients and data anomalies.

Our experimental results showed the farm datasets generate a higher approximation ratio than the hourly average method. For validation of our unsupervised approach, we first measured two state-of-the-art unsupervised anomaly detection techniques, AE and LoF using the synthetic datasets generated by the PyOD library. The results showed that

our approach achieves a high accuracy as the contamination ratio is increased compared with others. We then labeled data collected at two sensors with a Q-Q plot and a strong Pearson correlation using a set of statistical rules to cover a range of error boundary conditions for real-world datasets. Our proposed approach successfully detected anomalies while obtaining 98.3% of approximation ratios, i.e., requiring only 1.7% of original data. We also observed that the accuracy of anomaly detection can be 92%–100% for synthetic datasets and 83%–92% for real-world datasets.

ACKNOWLEDGEMENTS

This material is in part based upon work supported by the National Science Foundation under Grant No. 1751143. This work is also supported by the Korea Innovation Foundation (INNOPOLIS) grant funded by the Korea government (MSIT) (2020-DD-UP-0278).

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] K. Kirkpatrick, "Technologizing Agriculture ," in *Commun. ACM*, 2019.

[3] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "FarmBeats: An IoT Platform for Data-Driven Agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation*, 2017, pp. 515–529.

[4] P. Beckman, R. Sankaran, C. Catlett, N. Ferrier, R. Jacob, and M. Papka, "Waggle: An Open Sensor Platform for Edge Computing ," in *IEEE SENSORS*, 2016.

[5] S. A. Haque, M. Rahman, and S. M. Aziz, "Sensor Anomaly Detection in Wireless Sensor Networks for Healthcare," *Sensors*, vol. 15, no. 4, pp. 8764–8786, 2015.

[6] J. Renshaw, "Anomaly Detection Using AWS IoT and AWS Lambda," https://aws.amazon.com/ko/blogs/iot/anomaly-detection-using-aws-iot-and-aws-lambda/.

[7] L. Martí, N. Sanchez-Pi, J. M. Molina, and A. C. B. Garcia, "Anomaly Detection Based on Sensor Data in Petroleum Industry Applications," *Sensors*, 2015.

[8] S. Kartakis and J. A. McCann, "Real-time Edge Analytics for Cyber Physical Systems using Compression Rates," in *11th International Conference on Autonomic Computing (ICAC '14)*, 2014, pp. 154–159.

[9] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, "Anomaly Detection using Autoencoders in High Performance Computing Systems ," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[10] H. Ren, B. Xu, C. Y. Yujing Wang, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-Series Anomaly Detection Service at Microsoft ," in *KDD*, 2019.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[12] A. Moon, J. Kim, J. Zhang, H. Liu, and S. W. Son, "Understanding the Impact of Lossy Compression on IoT Smart Farm Analytics," in *2017 IEEE Big Data*, 2017, pp. 4602–4611.

[13] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications," in *IEEE Access*, 2018, pp. 4875–4894.

[14] J. Zhang, A. Moon, X. Zhuo, and S. W. Son, "Towards Improving Rate-Distortion Performance of Transform-Based Lossy Compression for HPC Datasets ," in *IEEE HPEC*, 2019.

[15] M. A. Razzaque, C. J. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, p. 5, 2013.

[16] J. Florbäck, "Anomaly Detection in Logged Sensor Data," Master's thesis, Chalmers University of Technology, 2015, master's thesis in Complex Adaptive Systems.

[17] V. Vercruyssen, W. Meert, G. Verbruggen, K. Maes, R. Baumer, and J. Davis, "Semi-Supervised Anomaly Detection with an Application to Water Analytics," in *IEEE International Conference on Data Mining*, 2018.

[18] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems Workshops*, ser. ICDCSW '11. USA: IEEE Computer Society, 2011, p. 166–171. [Online]. Available: https://doi.org/10.1109/ICDCSW.2011.20

[19] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html

[20] A. Moon, X. Zhuo, J. Zhang, and S. W. Son, "AD^2: Improving Quality of IoT Data through Compressive Anomaly Detection," in *2019 IEEE Big Data*, 2019.

[21] https://www.mcs.anl.gov/research/projects/waggle/downloads/datasets/index.php.

[22] I. P. S. Mary and L. Arockiam, "Imputing the Missing Values in IoT using ESTCP Model," in *International Journal of Advanced Research in Computer Science*, 2017.

[23] M. Goldstein and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data ," in *PLoS ONE 11(4): e0152173. https://doi.org/10.1371/journal.pone.0152173*, 2016.

[24] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers ," in *ACM sigmod record*, 2000.

[25] C. C. Aggarwal, "Outlier analysis ," in *Data mining*, 2015, pp. 75–79.

[26] N. Merrill and A. Eskandarian, "Modified Autoencoder Training and Scoring for Robust Unsupervised Anomaly Detection in Deep Learning," in *IEE Access*, 2020, pp. 101 824–101 833.

[27] D. Rolnick, "Tackling Climate Change with Machine Learning ," in *ICML*, 2019.

[28] K. M. Carter and W. W. Streilein, "Probabilistic Reasoning for Streaming Anomaly Detection," in *IEEE Statistical Signal Processing Workshop*, 2012.

[29] S. Kartakis, M. M. Jevric, G. Tzagkarakis, and J. A. Mccann, "Energy-based adaptive compression in water network control systems," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, April 2016, pp. 43–48.

[30] S. Kartakis, G. Tzagkarakis, and J. A. McCann, "Adaptive Compressive Sensing in Smart Water Networks," in *2nd International Electronic Conference on Sensors and Applications*, 2015.

[31] S. Li, K. Gruchalla, K. Potter, J. Clyne, and H. Childs, "Evaluating the Efficacy of Wavelet Configurations on Turbulent-Flow Data," in *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, 2015, pp. 81–89.

[32] A. H. Baker, H. Xu, J. M. Dennis, M. N. Levy, D. Nychka, and S. A. Mickelson, "A Methodology for Evaluating the Impact of Data Compression on Climate Simulation Data," in *The 23rd International Symposium on High Performance Parallel and Distributed Computing*, 2014, pp. 203–214.

[33] N. Shah, A. Beutel, B. Hooi, L. Akoglu, S. Günnemann, D. Makhija, M. Kumar, and C. Faloutsos, "EdgeCentric: Anomaly Detection in Edge-Attributed Networks," in *IEEE ICDM Workshop on Data Mining for Cyber Security*, 2016, pp. 327–334.

[34] M. M. Abo-Zahhad, A. I. Hussein, and A. M. Mohamed, "Compressive Sensing Algorithms for Signal Processing Applications: A Survey," *International Journal of Communications, Network and System Sciences*, vol. 8, no. 6, pp. 197–216, 2015.

[35] S. Budhaditya, D.-S. Pham, M. Lazarescu, and S. Venkatesh, "Effective Anomaly Detection in Sensor Networks Data Stream," in *Proceedings of the 9th IEEE International Conference on Data Mining*, 2009.

[36] E. J. Candes and M. B. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[37] Y. Yan, J. Zhang, B. Huang, X. Sun, J. Mu, Z. Zhang, and T. Moscibroda, "Distributed Outlier Detection Using Compressive Sensing," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 3–16. [Online]. Available: http://doi.acm.org/10.1145/2723372.2747641