AD^2: Improving Quality of IoT Data through Compressive Anomaly Detection

Aekyeung Moon
UMass Lowell
aekyeung_moon@uml.edu

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

***Abstract*—With recent technological advances in sensor nodes, IoT enabled applications have great potential in many domains. However, sensing data may be inaccurate due to not only faults or failures in the sensor and network but also the limited resources and transmission capability available in sensor nodes. In this paper, we first model streams of IoT data as a handful of sampled data in the transformed domain while assuming the information attained by those sampled data reveal different sparsity profiles between normal and abnormal. We then present a novel approach called AD^2 (Anomaly Detection using Approximated Data) that applies a transformation on the original data, samples top k-dominant components, and detects data anomalies based on the disparity in k values. To demonstrate the effectiveness of AD^2, we use IoT datasets (temperature, humidity, and CO) collected from real-world wireless sensor nodes. Our experimental evaluation demonstrates that AD^2 can approximate and successfully detect 64%–94% of anomalies using only 1.9% of the original data and minimize false positive rates, which would otherwise require the entire dataset to achieve the same level of accuracy.**

***Index Terms*—Transform Coding, Compressive Sensing, Anomaly Detection, IoT**

I. Introduction

Rapid advances in wireless sensor nodes have been key enablers to discover actionable knowledge from raw data and ultimately make smart decisions [1]. This discovery paradigm where knowledge is extracted from a steady stream of data collected from IoT sensor nodes is increasingly adopted by various domains [2], [3]. For instance, IoT-enabled agriculture has transformed traditional farms, which have been relying on human expertise, thereby enabling more precise and productive operations such as the reduction of non-essential pesticides use [4], [5]. A similar IoT platform called Waggle [6] collects real-time urban environmental data, which can be used for enabling a smart city.

There are several major challenges to exploit collected IoT datasets effectively, and more importantly reliably. First, like existing IoT environments, sensor nodes have limited resources and capabilities in terms of processing power, bandwidth, energy, and storage [7], [8]. The scarcity of resources could be a significant bottleneck as sensor nodes continuously collect data. The sensor nodes should be capable of managing data efficiently so that storage and transmission costs are reduced [6], [9]. Since sensor nodes send collected data to the gateways or cloud periodically, data volume during transmission need to be minimized, but, paradoxically, the transmitted data should be represented in the highest possible precision. Otherwise, analytic decisions derived from the collected data will have limited significance.

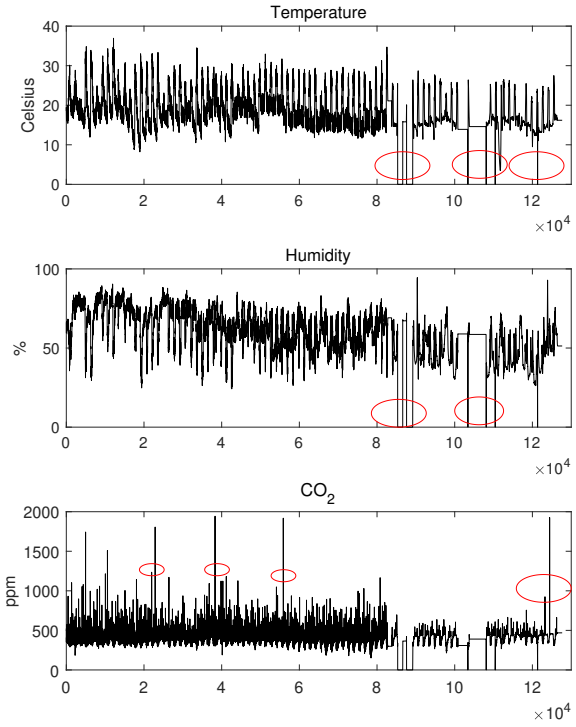


Fig. 1. Original datasets (collected at farms) showing anomalous periods. Red circles indicate apparent anomalous periods.

Second, data collected at sensor nodes are frequently inaccurate, which could potentially generate incorrect or unwanted control operations in actuator nodes or gateways. Figure 1 shows the collected data samples containing several instances of actual anomaly data, which are considerably dissimilar from the remainder of the data and expected patterns. The presence of anomalies in the collected datasets must be detected and notified before control operations are performed on potentially inaccurate data [10], [11]. Detecting anomalies in wind velocity measurements from the San Francisco wind monitoring datasets [11] exemplifies such a scenario. The effective anomaly detection empowers the decision maker to adequately react and take actions to correct anomaly situations timely [12].

Lastly, the boundary between normal and outlier is often

The first author conducted part of this research while at ETRI.

imprecise and there is no single rule that can be applied to any datasets generically. To elaborate on this, let us consider the temperature in Waggle datasets [13]. We find a number of temperature data points to be illegitimate because the range varies from -303.28 to 326.87 and there is a surge in temperature to 326 degrees. Similarly, in IoT farm datasets, the continuous change due to interaction among crop growth and operations of actuators (such as heater, CO_2 generator, fan, etc.) make it almost impossible to distinguish between normal states and abnormal states. Therefore, labeling time-series data as normal or abnormal is of the utmost importance.

Motivated by the aforementioned issues, we propose a novel approach, called AD^2 (Anomaly Detection using Approximated Data), to detect the anomaly of sensor data using sparse data representation. AD^2 collects, transforms, obtains approximated data and detects anomaly data state. In this paper, we argue that the data transformation is useful to reveal the correlation of data and detect anomalies since it makes data in a concise format. More specifically, in AD^2, we apply a transformation on the original datasets and approximate them. Sensor nodes maintain an approximated data points out of transformed datasets. Finally, we model the collected data using a simple statistical criterion and characterize normal and anomaly states in a specific sampling period. Our evaluation using several real-world datasets show that AD^2 can detect the majority of anomalies with a wide range of error thresholds. Specifically, AD^2 can achieve competitive approximation ratios, 98.1% on average, and the accuracy of detection data anomalies up to 95% using only 1.9% of the original data.

II. Related Works

A. Data Approximation

The theoretical underpinning for our approximation mechanism is compressive sensing. Compressive sensing is a sampling theory where certain signals can be recovered from a few samples [14], [15]. A signal can be sparse or compressible after applying signal transform with a suitable basis, e.g., DCT (Discrete Cosine Transform), Fourier, or Wavelet basis. In fact, compressive sensing techniques exploit the fact that sparse signal has only a few significant components and a greater number of insignificant components [14], [16].

Our data approximation mechanism based on compressive sensing works as follows. Let X' denotes the transformed version of original datasets, X. X' consists of the DCT signal components. Then the question is how many components are required to approximate original datasets with minimal information loss. There is a trade-off between approximation ratios and information loss. We formulate this as the energy (or information) contained in the highest k of the entire sorted transform components ($S = \{S_1, S_2, ...S_n\}$) and find an optimum k value, which is calculated as:

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

Note that the sum of energy stored in the entire DCT components is 1.0 (or 100%). We select the maximum k-dominant components ($\{S_1, S_2, ...S_k\}$) from the transformed components to approximate the original datasets where $E(x_k) \leq \delta$. The amount of energy, denoted as δ, has $0 \leq \delta \leq 1.0$. For instance, δ of 0.95 requires smaller k values than δ of 0.99, thus introducing more errors with less storage space requirement. After applying DCT [17], the original temperature values are transformed into sparse DCT signal components. Moon et al. [18] applied transformation on IoT data and showed that by keeping only 3.16% (245 out of 77,590) of the transformed signal components, they could represent 99.9% of information (energy) attained by the original data. Similarly, the temperature datasets in the Waggle require only 14% (6,872,724 out of 46,203,212) of components for representing 99% of information of the original data [13].

B. Anomaly Detection

Anomalies are points or patterns in datasets that differ from the expected "normal" behavior [19], [20]. Prior techniques based on machine learning to detect anomalies generally fall into three categories: unsupervised, supervised, and semi-supervised [21], [22]. There are various techniques to detect anomalies in sensor data by using fuzzy association rules [23]. Even though these techniques can work with the different amounts of labeled data for training, it is essentially difficult to classify into the normal state and the anomaly state because of domain-specific problem characteristics. Statistical methods such as exponentially weighted moving average or cumulative sum [24] have been used for detecting anomalous behaviors in time series data. In anomaly detection for AWS IoT data, [11] uses PEWMA (Probabilistic Exponentially Weighted Moving Average) proposed by Carter and Streilein [25], 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 [7], [26] presented that anomalies can be identified by analyzing significant changes in the compression rate. Kartakis et al. [26] 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 [27]. In our approach, we detect anomaly data periods using an approximated form with a capability to vary approximation ratios depending on datasets.

C. Characterizing Anomaly

Prior studies learned anomaly conditions, built a prediction model, and compared predicted values and measured values to determine if there were anomalies or not. For instance, Haque et al. [10] used 30 samples as a history to build a prediction model. If the microclimate affected by surrounding environment keeps changing, it becomes more difficult to distinguish normal from an anomaly.

As already shown in Figure 1, the real-world IoT datasets indeed exhibit some anomalies. For example, the humidity

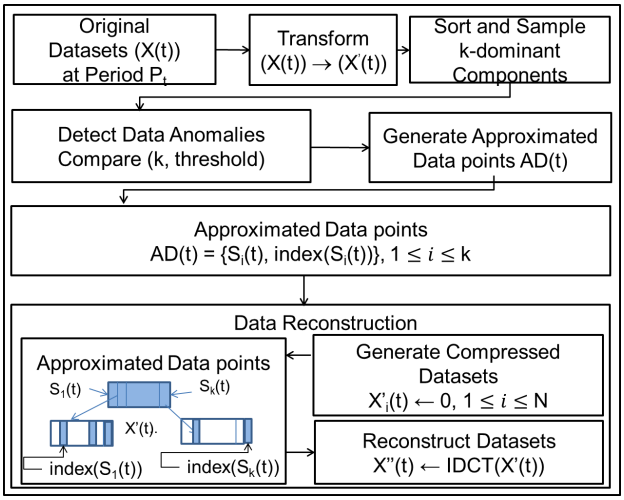


Fig. 2. Overview of the AD^2 framework.

occasionally goes down to nearly zero, which is an anomaly. We have learned that this situation typically occurs when there are network failures. Also, the temperature can experience a dramatic decrease when a sensor fails. The CO_2 sensor particularly, experiences more frequent and sudden changes than others. However, this does not mean such changes are all related to sensor failures. Certain external conditions can change sensor values temporarily, which a native method would predict as an anomaly, thus causing to increase the number of false positives. In urban microclimate data, transportation system generates CO_2 in the air. In rural farm data, the photosynthesis from crops keeps consuming CO_2 [24]. In these cases, the sudden change in CO_2 should be considered as normal to a certain degree.

III. Design of AD^2

We envision an expanded the capability of the sensor node by using a simple yet efficient approximation algorithm to represent original data points with a low sampling rate. Figure 2 gives the overall framework of AD^2 where the sensor node collects data points, transforms them into an approximated form, and detects data anomalies using sparse sample data. Note that AD^2 detects anomaly by investigating the k value for approximated (sampled) data points without inspecting full original data points. This mitigates storage and computation usage in the resource-constrained sensor nodes.

A. Notations

Before describing AD^2 in detail, let us define terminologies and notations used in the rest of the paper. We first define the period for performing the data transform as it is also duration for data sampling and anomaly detection. While data can be sampled and processed in any rates, in this paper, we assume that data has been gathered every second and they are transformed every minute or every hour. The transformed results (denoted as $X^{'}$) have the same number of data points as original datasets (denoted as X). Below is a summary of our notations:

- X: Original datasets ($\{X_1, ..., X_n\}$).
- P: Data transformation (approximation) period.
- P_t: t^{th} data transformation (sampling) period.
- $X(t)$: Original datasets at the sampling period P_t.
- $X'(t)$: Transformed components form of ($X(t)$) at P_t.
- $S(t)$: Sorted components form of $X'(t)$ at P_t.
- k: The number of dominant components in $X'(t)$. $k \geq 1$.
- $AD(t)$: The approximated data points comprised of k-dominant components.

Algorithm 1 Data approximation procedure.

Input: $X(t)$: original data points at period P_t
I: the length of $X(t)$
Output: k: the number of dominant coefficients.
$AD(t)$: approximated data points at Period t
1: $X'(t) \leftarrow$ DCT-Transform ($X(t)$)
2: $S(t) \leftarrow \text{sort}|X'(t)|$
3: $SX_n \leftarrow 0$
4: **for** $i = 1, 2, \ldots, I$ **do**
5: $SX_n \leftarrow SX_n + X_i'^2$
6: **end for**
7: $k \leftarrow 1$
8: $S(t) = S_k(t)$
9: **while** $S(t)/SX_n < \delta$ **do**
10: $k \leftarrow k + 1$
11: $S(t) = \sum_{k=1}^{k} S_k(t)$
12: **end while**
13: $AD(t) \leftarrow \{(S_1(t), \text{index}(S_1(t))) \ldots, (S_k(t), \text{index}(S_k(t)))\}$

Note that $AD(t)$ needs two additional parameters, α and ε. The first one (α) is the error threshold used in labeling the normal and anomaly periods according to the value difference between two identical sensors. The second one (ε), on the other hand, is the threshold value for predicting anomalies.

B. Data Approximation

Before discussing our approximation method in detail, we want to mention that while DCT is used as our transform mechanism, the AD^2 framework is *agnostic* to the choice of transform signals and is designed to operate with any signal transforming such as wavelets. Also, one can use higher amount of energy than 99% used in our experiment, thereby reconstructing data more precisely. However, it also means higher storage and communication overheads. We leave finding of the optimal transform basis and the number of components as our future work.

Algorithm 1 outlines the data approximation procedure based on the transformed data, which is similar to compressive sensing in a broader sense. First, we transform the original data, X, into DCT basis components to make the data sparse. In this way, the original data points $X(t)_i$ at t^{th} sampling period, 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. For example, assuming $X(t) = X(t)_1, ..., X(t)_n$ and k is 1, then the $AD(t)$ includes only one component and its index. The index indicates the position of the selected component in the original data points $X(t)$ and $X^{'}(t)$, which is required to

reliably reconstruct data later. In other words, $(N-k)$ non-significant components are discarded and only the values of k components and their indices are maintained. If AD^2 executes these processes per hour for data collected every second, it will locate the k-dominant components out of 3,600 data points (i.e., 60 times 60), resulting in a significant data reduction.

C. Data Anomalies Detection

The more anomalous data points there are, the larger k is needed. In our evaluated datasets, the sampling period requires 3 or 6 dominant components whereas normal data requires 1 component to maintain the same amount of energy (information). This is because the energy is more dispersed when data has more unexpected patterns, which in turn requires more number of dominant components to maintain the same amount of energy.

Our anomaly detection approach based on the correlation between k and the degree of anomalies consists of two methods: binary detector and difference detector. The binary detector only compares if the two values k and the threshold ε which is the boundary condition to decide whether this period is normal or abnormal. To simplify the process of finding the value ε, it might reference k-dominant components of normal periods. The difference detector calculates the equation denoted as $|(k-\varepsilon)|/Mean(X)$

D. Characterizing Anomaly Data

To characterize normal sensor behavior precisely, we deployed two identical sensors and used the result from this step as our ground truth for training and prediction. While one can employ more complex methods such as machine learning or more recent neural network based techniques, we found simple statistical methods using two identical sensors more straightforward. More specifically, we use NRMSE (the normalized version of root mean square error) and NDIFF (the normalized version of difference) to characterize anomalies. That is, we classify the sensor data in the case of anomaly conditions according to the NRMSE and NDIFF 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: $NRMSE(t) = \frac{RMSE(t)}{Mean(x)} = \frac{1}{\bar{x}}\sqrt{\frac{\sum_{n=1}^{N}(x(n)-\hat{x}(n))^2}{N}}$.
- NDIFF for each period P_t is given by: $NDIFF(t) = \frac{|x-\hat{x}|}{Mean(x)}$.

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

- $NRMSE(t) > \alpha$, set true to Anomaly(t) for period t.
- $NDIFF(t) > \alpha$, set true to Anomaly(t) for period t.

Based on this assumption, we use the following evaluation metrics to measure the performance of AD^2 for the simple binary detector.

- TP (True Positive): $((k > \varepsilon) \wedge (\text{NRMSE} > \alpha)) \vee ((k > \varepsilon) \wedge (\text{NDIFF} > \alpha)) \rightarrow (k > \varepsilon) \wedge Anomaly(t)$.
- FP (False Positive): $((k > \varepsilon) \wedge (\text{NRMSE} \leq \alpha)) \vee ((k > \varepsilon) \wedge (\text{NDIFF} \leq \alpha)) \rightarrow (k > \varepsilon) \wedge \neg Anomaly(t)$.
- TN (True Negative): $((k \leq \varepsilon) \wedge (\text{NRMSE} \leq \alpha)) \vee ((k \leq \varepsilon) \wedge (\text{NDIFF} \leq \alpha)) \rightarrow (k \leq \varepsilon) \wedge \neg Anomaly(t)$.
- FN (False Negative): $((k \leq \varepsilon) \wedge (\text{NRMSE} > \alpha)) \vee ((k \leq \varepsilon) \wedge (\text{NDIFF} > \alpha)) \rightarrow (k \leq \varepsilon) \wedge Anomaly(t)$.

For instance, TP is the case where k is greater than ε and NRMSE is greater than α. In our experiment, we set ε to 1 because most normal data points required only one dominant components (i.e., k=1). As a result, we need to maintain only one data point per sampling period (e.g., hourly average).

E. Data Reconstruction

So far, we mainly discussed how AD^2 approximates the original data using top-k transformed components and how the variation in k per sampling periods can be used for anomaly detection. In this subsection, we would like to discuss how to reconstruct data when there is a need for fully reconstructed data. As discussed in Section III-B, after sensor data are transformed, the selected k-dominant components along with the corresponding indices are maintained, which can be sent to the server if needed. Note that both k-dominant transformed components and their indices are required to define the approximated data points in AD^2. After receiving AD', it generates $X'(t)$ by replacing the indices of data points except the indices of k-dominant with zeros. It then reconstructs data points $X''(t)$, which are generated from $X'(t)$ by applying inverse transformation methods, 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.

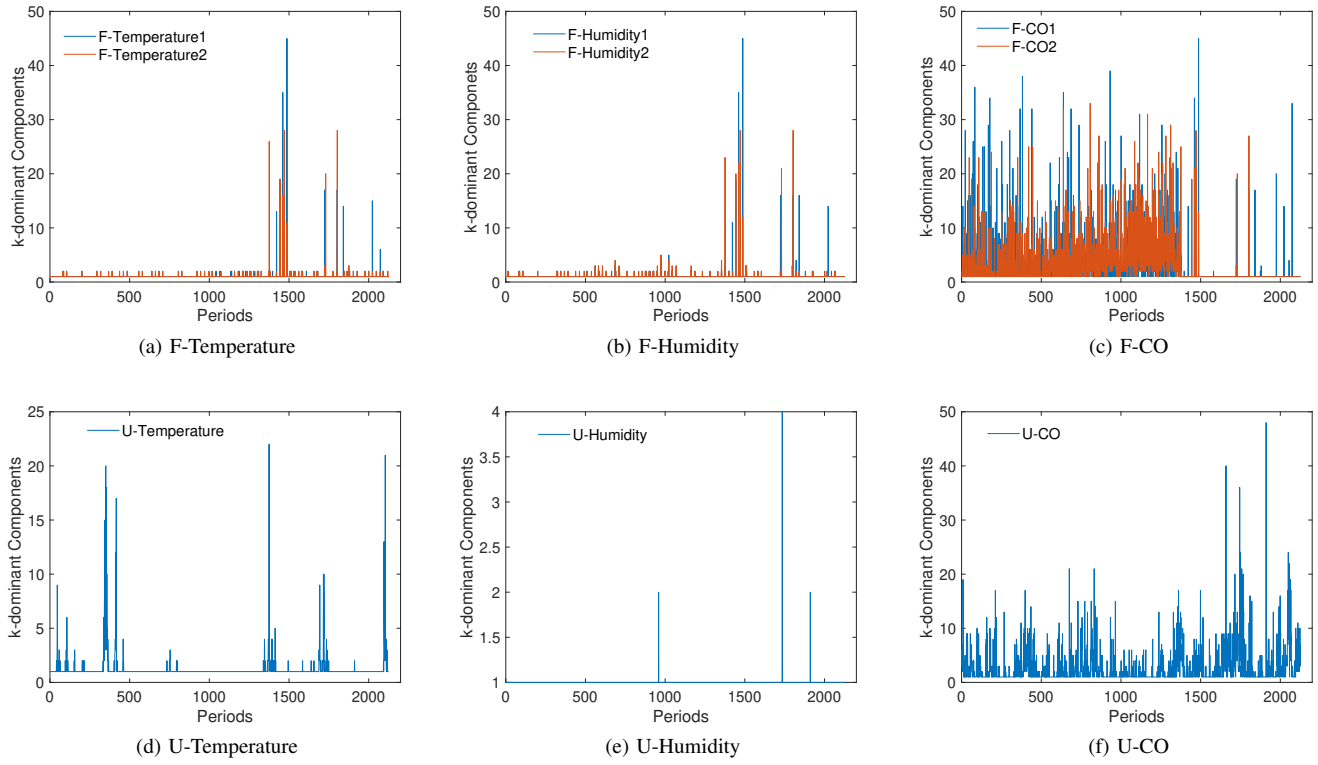
IV. Evaluations

A. Datasets

In our evaluation, we used two sets of real-world IoT data. The first one is from the IoT farm system deployed in Gangwon Province, South Korea, from October 1st to December 31st, 2017 (90 days). We collected the following three microclimate datasets in the deployed system: temperature, humidity, and CO. The data collection period is set to every minute. As a result, we have 127,667 data points per each data during 90 days of sampling. The second dataset we evaluated is from the Waggle project, which is also targeting climate applications but in an urban environment setting [6], [13]. The Waggle dataset is time-series data collected at several urban locations in the US that includes air temperature, relative humidity, barometric pressure, UV light, IR light, and so on [6], [8]. We evaluated temperature, humidity, and CO extracted from the big Chicago datasets measured between February 3rd, 2017 and July 4th, 2019.

We choose the following performance metrics to assess the quality of the approximated data and the overall anomaly detection rates.

- Approximation Ratio (AR), where $|D|$ is the size of D, $|D'|$ is the approximated data size, is given by:



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

(d) U-Temperature (e) U-Humidity (f) U-CO

Fig. 3. Variation of k-dominant components. (a), (b), and (c) are from two identical sensors whereas (d), (e), and (f) are from a single sensor.

$AR = \frac{|D|-|D'|}{|D|} \times 100\%$,

- Error rate is assessed using PSNR (Peak Signal-to-Noise Ratio), which measures the overall distortion between the original data and the reconstructed data.
  $PSNR = 20 \cdot log10(\text{value range}) - 10 \cdot log10(MSE)$, where value range and MSE refer to data value range and the mean squared compression error, respectively.
- FPR (False Positive Rate) and ACC (Accuracy) can be calculated as:
  $FPR = \frac{FP}{FP+TN}$,
  $ACC = \frac{TP+TN}{TP+TN+FP+FN}$,
  where TP, FP, TN, and TF refer to true positive, false positive, true negative and true positive respectively.

The objective of AD^2 in terms of the above performance metrics is to achieve higher AR, PSNR, and ACC and lower FPR. In our evaluation, we did not compare AD^2 with existing anomaly detection methods as our method is based on approximated data. Instead, we evaluate AD^2 using a range of error threshold values (α).

B. Results

a) Data Characteristics and Approximation Ratios: Table I shows statistical properties of the original datasets in terms of STD (standard deviation), NSTD (normalized standard deviation), skewness, and kurtosis. NSTD is calculated as $\frac{STD(x)}{Mean(x)}$. Skewness is a measure of data asymmetry around the mean value. Negative skewness means that more data are scattered to the left of the mean whereas positive skewness means opposite. Normal distribution, where data is symmetric about its mean, gives zero skewness. Measures of kurtosis in Table I 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.

TABLE I
THE CHARACTERISTICS OF DATASET AND APPROXIMATION RATIOS.

	No. of Data points	STD	NSTD	Skewness	Kurtosis	AR
F-Temp.	127,668	5.1532	0.2871	-0.2983	5.3136	99.86
F-Humidity	127,668	15.5245	0.2700	-1.0632	5.4268	99.89
F-CO	127,668	87.4290	0.2188	-1.4900	17.8790	99.42
U-Temp.	46,203,212	11.0143	0.8309	0.5782	2.2491	85.13
U-Humidity	8,762,131	18.81023	0.3036	-0.2474	3.0318	99.25
U-CO	19,401,868	10097	1.2764	62.8617	10397	34.56

In our datasets, U and F denote Urban datasets (i.e., Waggle) and Farm datasets, respectively. U-CO shows higher STD than other datasets and has the highest kurtosis value among all datasets. In the case of skewness, U-Temperature and U-CO have positive values only. In terms of compressibility, the farm datasets show higher approximation ratios than the urban datasets. U-CO particularly shows the lowest approximation ratios. As shown in Table I, the urban datasets show some variances in the approximation ratio depending on the characteristics of data. The approximation ratio of U-CO is about 29% lower than W-Humidity.

b) Variation in k-dominant Components: Figure 3 shows the variation in k values. Recall that we use the fixed information compaction rate of 99% for all sampling periods during

TABLE II
COMPARISON OF ERROR RATES (IN TERMS OF PSNR) BETWEEN AD^2 COMPRESSION AND HOURLY AVERAGES.

	AD^2		Hourly Average	
	Error Rate	AR_M	Error Rate	AR_M
Temperature	33.72 38.23	98.15% 98.15%	31.77 35.28	98.3%
Humidity	33.31 33.65	98.13% 98.09%	31.08 30.46	98.3%
CO	35.74 44.63	95.5% 94.89%	33.54 38.37	98.3%

approximation. Farm datasets are from two identical sensors and the data transformation period P is set to an hour. Because the collection period of urban datasets are not clearly defined, transforms are applied for every 60 data points, which are eqivalent to 770,053 periods. As shown in Figure 3, the x-axis represents the transformation period of data points, which corresponds to 2,127 periods, and the y-axis represents the k-dominant components required to approximate every sampling period (one hour). We can observe that k is notably different between two identical sensors in case there are anomalies. For example, in the case of temperature data (shown in Figure 3a), anomalies occurred around the $1{,}400^{th}$ periods where k is larger than 5. Typically, periods with higher anomalies require more components to approximate data maintained by the same amount of energy, hence resulted in a lower approximation ratio.

c) Approximation Ratios and Error Rates: As previously described, the original farm datasets are collected every minute. Table II shows the error rate between the reconstructed data using AD^2 and the conventional hourly average value of data. Note that the approximation ratios for AD^2 vary depending on periods, whereas the hourly average has fixed 1/60 ratios (or 98.3%). We compare our algorithms with hourly average data because it is one of the most commonly used techniques in sensor nodes. In the case of the AD^2 approximation, the error rate is slightly lower than the hourly average, which achieves 98.1% of approximation ratios. Therefore, our approach based on compressive sensing can be more accurate than conventional hourly averages because AD^2 shows higher PSNR than hourly averages. Higher PSNR represents less error that affects the data quality.

We also observed that AD^2 can achieve 98.3% of approximation ratios for the time periods with the normal state only. This ratio is same as the hourly average. As shown in Table II, when all sampling periods including normal and abnormal cases were considered, AD^2 can achieve 98.1% of approximation ratios. On the other hand, when periods with only abnormal cases were considered, AD^2 can achieve 90.4% of approximation ratios. Anomaly periods require more k-dominant components to maintain the same amount of information or approximation, so approximation ratios for periods with anomalies are lower than those with normal periods. Overall, AD^2 achieves quite competitive approximation ratios while maintaining a relatively low error rate.

d) Anomaly Detection Rate: Figure 4 shows that the accuracy converges to 1 with increasing α values. More specifically, in the case of temperature, AD^2 can achieve the accuracy of 33.8% when α is 0.1, but 92% when α is 0.2. 86 periods are labeled as anomalous periods when α is 2.0 in the case of temperature. Beyond that, accuracy is as high as 95%. Similarly, the accuracy ranges from 64% to 94% for the humidity data. However, for the CO data, since it has more anomalous periods than others, the detection accuracy is inferior to the other two datasets. We can observe that, as α increases, it converges quickly. On the other hand, FPR for the three datasets is close to zero. Specifically, temperature, humidity, and CO, the error converges to 3.5%, 4%, and 3.3%, respectively. Lastly, in the case of NRMSE and NDIFF, we observed that they both converges faster with an increasing threshold of α. However, NRMSE converges faster than NDIFF. Consequently, NRMSE could be more suitable for detecting anomalies than NDIFF. Overall, it is necessary to select a proper threshold of α within a tolerable range, which is tightly coupled with applications.

V. CONCLUSIONS

In this paper, we proposed an approach, called AD^2, to detect data anomalies at sensor nodes using highly approximated data with minimal information loss. We approximate the original datasets into the k-dominant components, thereby significantly relieving the limitation of sensor node resource. The relationship between k-dominant components and data anomalies provides a dependable method to detect data anomalies using sparse sampling data. We first characterize normal and abnormal data collected at two identical sensors using two statistical metrics, NRMSE and NDIFF. Our experimental results using the farm datasets showed a higher approximation ratio than the urban datasets. Our proposed approach successfully detected anomalies while obtaining 98.1% of approximation ratios. We also observed that the accuracy of anomaly detection can be close to 1 and the false positive rate can be close to zero when less tight anomaly boundary (i.e., higher α) is used.

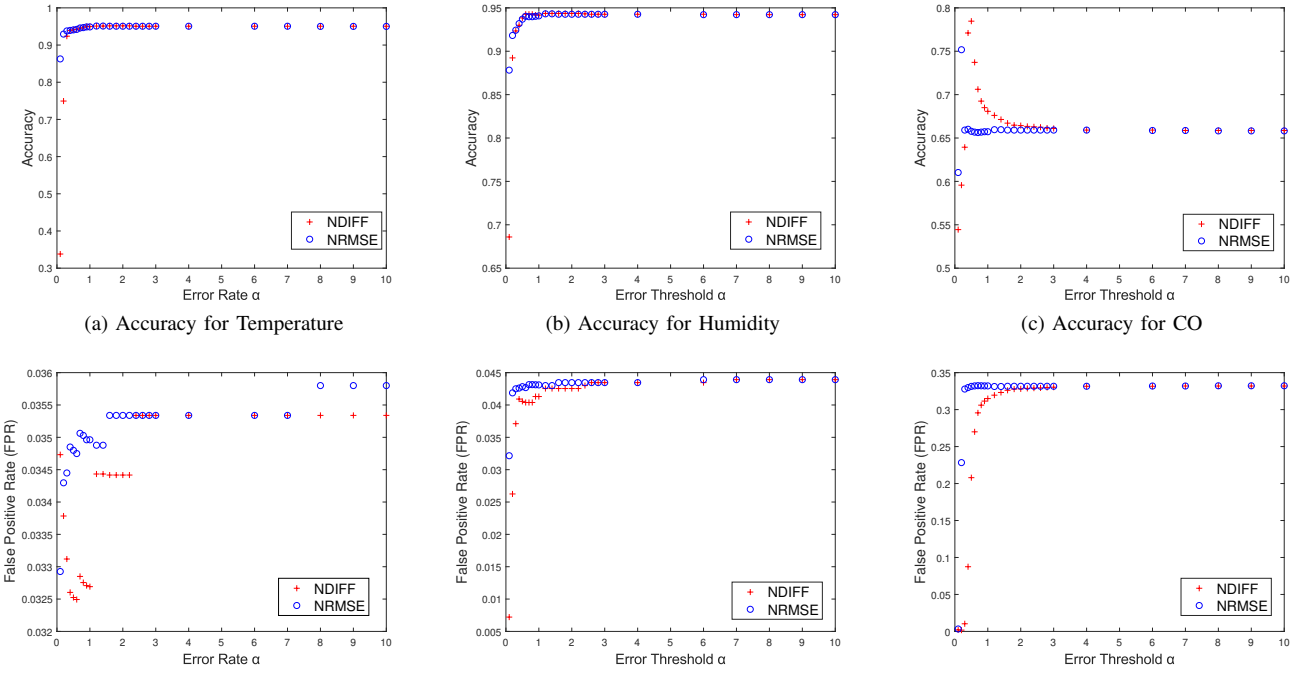
Although using two identical sensors can mitigate the false decision of data anomalies due to the effect of the operation of actuators around IoT sensor nodes collecting microclimate to some degree, a more generic solution to characterize normal and abnormal will be considered in our future work. We also plan to develop estimation and imputing methods for missing time-series data generated continuously from IoT sensors.

ACKNOWLEDGEMENTS

This material is in part based upon work supported by the National Science Foundation under Grant No. 1751143. This work was also supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIP) (No. CRC-15-01-KIST).

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.



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

(d) False Positive Rate for Temperature (e) False Positive Rate for Humidity (f) False Positive Rate for CO

Fig. 4. Performance of anomaly detection for three datasets.

[2] 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.

[3] A. Moon, J. Kim, J. Zhang, and S. W. Son, "Evaluating Fidelity of Lossy Compression on Spatiotemporal Data from an IoT Enabled Smart Farm," *Computers and Electronics in Agriculture*, vol. 154, pp. 304–313, Nov. 2018.

[4] K. Kirkpatrick, "Technologizing Agriculture ," in *Communications of the ACM*, 2019.

[5] 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.

[6] 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.

[7] 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.

[8] http://www.mcs.anl.gov/project/waggle-open-platform-intelligent-attentive-sensors.

[9] T. Bose, S. Bandyopadhyay, S. Kumar, A. Bhattacharyya, and A. Pal, "Signal Characteristics on Sensor Data Compression in IoT – An Investigation," in *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2016, pp. 1–6.

[10] 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.

[11] 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/.

[12] 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.

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

[14] R. Sustika and B. Sugiarto, "Compressive Sensing Algorithm for Data Compression on Weather Monitoring System," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, pp. 974–980, 2016.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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

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

[21] M. A. Hayes and M. A. Capretz, "Contextual anomaly detection framework for big sensor data," *Journal of Big Data*, 2015.

[22] 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 ICDM(International Conference on Data Mining)*, 2018.

[23] C.-H. Weng, "Mining fuzzy specific rare item sets for education data ," *Knowledge-Based Systems. 24(5)*, 2011.

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

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

[26] 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.

[27] 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.