Fault Detection of ICT systems with Deep Learning Model for Missing Data

Kengo Tajiri*, Tomoharu Iwata†, Yoichi Matsuo*, Keishiro Watanabe*
*NTT Network Technology Laboratories, NTT Corporation, Tokyo 180-8585, Japan
†NTT Communication Science Laboratories, NTT Corporation, Kyoto 619-0237, Japan
Email: *,†{kengo.tajiri.bk, tomoharu.iwata.gy, yoichi.matsuo.ex, keishiro.watanabe.ry}@hco.ntt.co.jp

***Abstract*—Fault detection is one of the most important tasks in information and communications technology (ICT) systems. Unsupervised anomaly detection methods, which are based on machine learning for fault detection in the ICT systems, use various kinds of data such as traffic data, memory usage data, CPU usage data, and text log data. The problem of deploying unsupervised anomaly detection methods in real ICT systems is that these data may have missing values. When a record has missing values, existing unsupervised anomaly detection ignores the records or imputes missing values with specific values. However, both operations lead to decreased performance of the anomaly detection methods. In this paper, we propose an unsupervised anomaly detection method that can handle records with missing values without imputation by using a neural network that can process variable length inputs. We experimented with 22 benchmark datasets to evaluate the performance of the proposed method for various kinds of data. The experimental results reveal that the proposed method performs better than existing methods in terms of area under the receiver operating characteristic (AUROC) on average for two cases in which 1) neither training nor test data include incomplete data, and 2) both training and test data include incomplete data. Moreover, we experimented with data from a Wi-Fi service that have missing values. The results show that the proposed method outperformed existing unsupervised anomaly detection methods.**

***Index Terms*—anomaly detection, deep learning, missing data**

I. Introduction

Fault detection is one of the most important tasks in the operation of information and communications technology (ICT) systems. If operators cannot notice the anomalous states of ICT systems due to failures, large-scale faults may occur and impact many services. Because, operators need to detect the anomalous states by using anomaly detection methods before serious problems appear, many anomaly detection methods based on machine learning for detecting failures in ICT systems have been developed in recent years [1]–[3].

In many cases for real systems, unsupervised anomaly detection methods, which are trained with only normal data and extract the intrinsic property of the data, are more commonly used, because of the lack of anomalous records and the difficulty of defining the kinds of anomalies. Especially, deep learning based unsupervised anomaly detection methods, which detect anomalous records by learning the characteristics of normal data, have been successfully used. For example, Autoencoder (AE) [4], [5] learns the intrinsic property of normal data by executing dimensional compression in latent layers so that the output is close to the input. Some studies have used an AE based anomaly detection method to detect anomalous states of ICT systems [6], [7].

To detect anomalies in ICT systems, unsupervised anomaly detection methods based on machine learning use various kinds of data such as traffic data, memory usage data, central processing unit (CPU) usage data, and text log data. These methods require that no records have missing values. On the other hand, records having missing values may be mixed in datasets from ICT systems for various reasons such as system maintenance, the absence of traffic by chance, or errors in collecting information about the state of each network instrument due to faults in monitoring equipment or applications. In this case, these records with missing values have to be ignored or imputed with estimated values in a specific way. Ignoring records with missing values causes the anomalous records to be overlooked for two reasons: an anomaly detection model is not trained enough due to the shortage of training records and anomalous test records with missing values are ignored in the detection phase. Imputing missing values in estimated values in a specific way distorts the intrinsic property of training data, which leads to decreased performance, because estimated values do not consider the ICT system status that makes records of missing values for the reasons above.

We propose a new deep learning based unsupervised anomaly detection method that uses only existing values in incomplete records. The AE based anomaly detection method cannot use incomplete records without imputation, because AE is composed of neural networks, whose encoder requires the dimension of input data to be fixed. In the proposed method, to overcome missing values, we represent a record by a set of the combination of a non-missing value and the index of the value, instead of representing a vector as usual. More specifically, the proposed method is constructed with set transformers [8], which enables us to handle a variable number of non-missing attributes, and is trained to minimize the reconstruction error of non-missing values of normal records in the same way as AE. The reconstructions of the existing values are then outputted. In the detection phase, the reconstruction error of each record is calculated as the anomaly score of the record.

We experimented with 22 benchmark datasets to evaluate the usefulness of the proposed method for various kinds of data and found that the proposed method performs better than existing methods in terms of area under the receiver operating characteristic (AUROC) on average for two cases in which 1) neither training nor test data include incomplete

978-3-903176-32-4

data, and 2) both training and test data include incomplete data. Moreover, we experimented with data from a Wi-Fi service that have missing values. The results reveal that the proposed method generates fewer false alerts than existing unsupervised anomaly detection methods.

II. Related works

In this section, we review existing methods for unsupervised anomaly detection and missing data processing.

A. Unsupervised anomaly detection

Anomaly detection methods such as Local Outlier Factor (LOF) [9], One Class Support Vector Machine (OCSVM) [10], and Isolation Forest (IF) [11] has been extensively studied. AE is one of the most widely used unsupervised anomaly detection methods based on deep learning. Denoising Autoencoder (DAE) [12], which is a derivative of AE, is also used for anomaly detection. Since DAE is trained to reconstruct original data from the data synthesized by adding noise, it can train a robust model from noisy data. However, all the above methods require that all records have no missing values in either the training or evaluation phase. Therefore, if missing data are used for anomaly detection, the missing attributes have to be filled with some values.

B. Processing missing data

Many methods have also been developed for processing missing data. The simplest methods involve filling each missing attribute with a single value using the mean of all data values or near data point values with k-nearest neighbors [13]. A regression model using existing attributes is also used to fill each missing attribute with a single value. However, filling a missing attribute with a single value shifts the distribution of data and decreases the variance of data. The multiple imputation method (MICE) [14], with which a model is trained with the data in which the missing attributes are filled with different values is better than the imputation of a missing attribute with a single value. Furthermore, there are also methods for filling missing values by using, for example, a deep learning model, such as a context encoder [15] and a generative adversarial network [16] based on adversarial learning. These methods can estimate values close to true values with a deep learning model. However, there is a computational cost for training deep learning models for estimating missing values.

Unlike the above methods, the full information maximum likelihood method (FIML) [17] uses incomplete records without imputation for updating only the likelihood model parameters related to the existing values in the training phase. With FIML, it is hypothesized that each likelihood model parameter is related to some attributes of input data. However, the parameters can be related to all attributes of input data with the proposed method. Therefore, our method can learn the complicated relationship between the attributes.

III. Proposed method

To overcome missing values without using imputations, we build a method with an encoder-decoder model using deep learning models for sets to handle records represented by sets with variable sizes. In this section, first, we introduce the properties of deep learning models for sets. Second, we explain our problem formulation of anomaly detection with missing values. Third, we introduce a set transformer that constitutes the proposed deep learning model. Finally, we detail the encoder and decoder of the proposed method.

A. Deep learning model for sets

We introduce the properties of deep learning models for sets, since the proposed method handles values of non-missing attributes as a set. According to Zaheer et al. [18], deep learning models for sets require the permutation invariant or permutation equivariant property because sets do not have the information about the order of the elements. Since we use the permutation equivariant property in the proposed method, we introduce the definition of the permutation equivariant property. The following equation holds for models $f(\cdot)$ satisfying the permutation equivariant property,

$$\pi(f(\{x_1, \ldots, x_n\})) = f(\{x_{\pi(1)}, \ldots, x_{\pi(n)}\}), \qquad (1)$$

where $\pi(\cdot)$ is a permutation function and $X = \{x_1, \ldots, x_n\}$ is a set. Zaheer et al. [18] proposed deep sets, which is a deep learning model for sets preserving permutation invariant or equivariant property.

B. Unsupervised anomaly detection with missing data

Next, we represent a record with missing values by a set as follows, $\boldsymbol{u}_m = \{(x_{mn}, r_{mn})\}_{n=1}^{N_m}$, where r_{mn} is the attribute index of the n-th non-missing value in the m-th record, x_{mn} is its observed value, and N_m is the number of non-missing attributes in the m-th record. The number of non-missing values can differ across records. Suppose that we are given a set of normal data with missing values $\mathcal{U} = \{\boldsymbol{u}_m\}_{m=1}^{M}$. Our task is to learn an anomaly score function that can detect anomalies in the test data using a given set $\mathcal{U}$.

Since the proposed method is built with an encoder-decoder model using deep learning models for sets to handle records represented by sets with a variable size, an encoder transforms set $\boldsymbol{u}_m = \{(x_{m1}, r_{n1}), \ldots, (x_{mN_m}, r_{mN_m})\}$ into latent representation $\boldsymbol{Z}_m$. Then, a decoder calculates the set of values for the non-missing attribute, $\hat{\boldsymbol{u}}_m = \{(\hat{x}_{m1}, r_{n1}), \ldots, (\hat{x}_{mN_m}, r_{mN_m})\}$, where $\hat{x}_{mn}$ is the reconstruction of the n-th non-missing attribute. The encoder-decoder function must be permutation equivariant since when elements in input set $\boldsymbol{u}_m$ are permutated, the elements in output set $\hat{\boldsymbol{u}}_m$ need to be permutated in the same way so that non-missing attribute indices r_{mn} correspond between $\boldsymbol{u}_m$ and $\hat{\boldsymbol{u}}_m$. When a permutation equivalent function is used, we only need to reconstruct the set of values for non-missing attributes, $\{\hat{x}_{m1}, \ldots, \hat{x}_{mN_m}\}$.

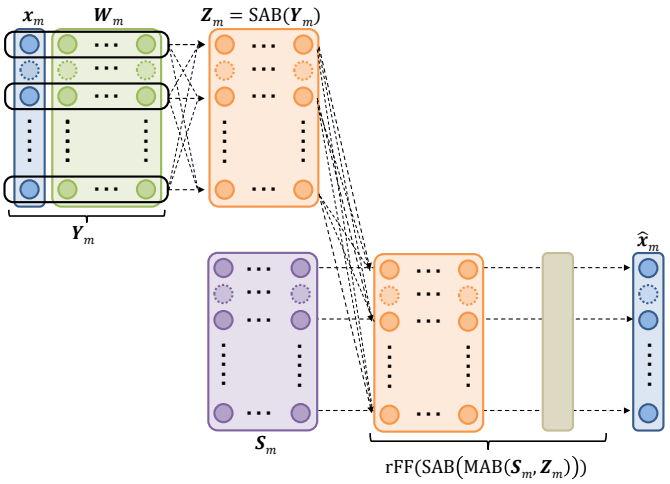

Fig. 1. Overview of proposed method. The broken-line circles represent missing values. The proposed method calculates the set of the non-missing values $\hat{\boldsymbol{x}}_m$ in the m-th record from the set of the non-missing values $\boldsymbol{x}_m$ and the embedding vectors in $\boldsymbol{W}_m$ and $\boldsymbol{S}_m$ corresponding to the non-missing values in the m-th record with the permutation equivariant models.

The encoder and decoder are trained by minimizing the following reconstruction errors over the given records,

$$\mathcal{L}(\mathcal{U}) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N_m}\sum_{n=1}^{N_m}(x_{mn} - \hat{x}_{mn})^2, \tag{2}$$

where we used the squared error assuming the observed values are continuous. We can use other metrics for the reconstruction error, such as cross-entropy loss when the observed values are categorical.

In the detection phase, the reconstruction error $\frac{1}{N_m}\sum_{n=1}^{N_m}(x_{mn} - \hat{x}_{mn})^2$ for each record is used for its anomaly score. The overview of the proposed method is shown in Figure 1.

C. Set transformer

For our encoder-decoder model, we decided to use a set transformer [8] to minimize the reconstruction error in eq. (2). Before explaining the details of the proposed model, we describe the properties of a set transformer. Lee et al. [8] used the transformer [19] as a kind of attention mechanism to construct a deep learning model preserving the permutation equivariant property. The deep learning model is called a Multihead Attention Block (MAB) and shown as follow,

$$\boldsymbol{Z} = \mathrm{MAB}(\boldsymbol{X}, \boldsymbol{Y}), \tag{3}$$

where $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \in \mathbb{R}^{n\times d}$ are sets of n elements of d-dimensional vectors represented as matrices. The MAB calculates the relationship between the elements of $\boldsymbol{X}$ and that of $\boldsymbol{Y}$ with attention mechanisms. The permutation of elements corresponds to that of rows of the matrices. The relationship between the input $\boldsymbol{X}$ and the output $\boldsymbol{Z}$ preserves the permutation equivariant property and $\boldsymbol{Z}$ does not depend on the permutation of the elements of $\boldsymbol{Y}$. Lee et al. [8] also define the Set Attention Block (SAB) as follows,

$$\mathrm{SAB}(\boldsymbol{X}) = \mathrm{MAB}(\boldsymbol{X}, \boldsymbol{X}). \tag{4}$$

The SAB calculates self-attention for $\boldsymbol{X}$.

D. Details of proposed model

We explain the encoder, decoder, and property of our proposed model.

1) Encoder: The encoder of the proposed method needs to be able to handle sets with variable sizes. We use a transformer for the encoder to effectively encode the information in the given record with missing values.

We assume an embedding vector for each attribute. Let $\boldsymbol{w}_i \in \mathbb{R}^D$ be the embedding vector of the i-th attribute. Then, each element in the input set (x_{mn}, r_{mn}) is represented by the multiplication of the value and its attribute's embedding vector:

$$\boldsymbol{y}_{mn} = \boldsymbol{w}_{r_{mn}} x_{mn}. \tag{5}$$

The representations for all elements are written in a matrix form, $\boldsymbol{Y}_m = [\boldsymbol{y}_{m1}, \ldots, \boldsymbol{y}_{mN_m}]^\top \in \mathbb{R}^{N_m\times D}$. By using a SAB, representations $\boldsymbol{Y}_m$ are encoded:

$$\boldsymbol{Z}_m = \mathrm{SAB}(\boldsymbol{Y}_m). \tag{6}$$

The encoder satisfies the permutation equivariant property for the order of the elements of $\boldsymbol{Y}_m$, that is the row of the matrix $\boldsymbol{Y}_m$. The output $\boldsymbol{Z}_m$ is also handled as a matrix $\boldsymbol{Z}_m \in \mathbb{R}^{N_m\times D}$ in the decoder.

2) Decoder: We make the encoder-decoder permutation equivalent function by inputting non-missing value indices $\{r_{mn}\}_{n=1}^{N_m}$ into the decoder. As in the same way with the encoder, we assume an embedding vector for each attribute for the decoder. Let $\boldsymbol{s}_i \in \mathbb{R}^D$ be the embedding vector of the i-th attribute for the decoder. The embedding vectors for all non-missing attributes in the m-th record are written in a matrix form, $\boldsymbol{S}_m = [\boldsymbol{s}_{mr_{m1}}, \ldots, \boldsymbol{s}_{mr_{mN_m}}]^\top \in \mathbb{R}^{N_m\times D}$ by aligning with non-missing value indices $\{r_{mn}\}_{n=1}^{N_m}$. We decode latent representation $\boldsymbol{Z}_m$ using embedding matrix $\boldsymbol{S}_m$ with the following procedure. First, the embedding matrix $\boldsymbol{S}_m$ is transformed by a MAB into latent representation $\boldsymbol{Z}_m$:

$$\boldsymbol{S}'_m = \mathrm{MAB}(\boldsymbol{S}_m, \boldsymbol{Z}_m), \tag{7}$$

where the number of rows in $\boldsymbol{S}'_m$ is N_m. By this transformation, encoded information for each non-missing attribute is obtained in each row of $\boldsymbol{S}'_m$. Then, we reconstruct the non-missing values by

$$\hat{\boldsymbol{x}}_m = \mathrm{rFF}(\mathrm{SAB}(\boldsymbol{S}'_m)), \tag{8}$$

where the rFF means a row-wise feedforward network, which processes each row of $\mathrm{SAB}(\boldsymbol{S}'_m)$ identically and independently and $\hat{\boldsymbol{x}}_m = \{\hat{x}_{m1}, \ldots, \hat{x}_{mN_m}\}$. Using a SAB and row-wise feedforward network, we can learn the interaction between attributes, and their nonlinear relationship.

3) Property of proposed method: Our encoder-decoder function satisfies the permutation equivariant property for the order of the elements of $\boldsymbol{u}_m$. Therefore, the proposed method satisfies the requirement for a deep learning model for sets and handles variable length records, that is, records with missing values. Although the proposed method minimizes the reconstruction error to obtain the characteristics of normal data in the middle of the model as AE does, the proposed method obtains the characteristics by not using dimensional reduction but the attention mechanisms to satisfy the permutation equivariant property.

IV. Experiments

We experimented with 22 benchmark datasets and Wi-Fi service data.

A. Benchmark datasets

We conducted two experiments using two different missing datasets. The first experiment involved complete data without missing values. The second experiment involved both training and test data including missing values. In the second experiment, we used the original labels, which determine whether an incomplete test record including missing values is normal or an anomaly. The rate of the missing attributes of each training and test record was fixed to 0.2. This is because if the rate of the missing attributes is too large, the original labels become meaningless.

1) Data: We evaluated anomaly detection methods including the proposed method with 22 benchmark datasets [20] used for unsupervised outlier detection[1]. Each attribute was normalized to the range of $[0, 1]$. We used 80 % of the normal records as training data, 10% as validation data, and the other 10%, and all the anomalous records as test data. The validation data were used for determining the number of epochs in the training phase for deep learning based methods. We used AUROCs of test data as the evaluation metric and then discuss the average AUROCs of each dataset and all datasets for five sets in Section IV-A3.

Some values in the records were deleted artificially. We chose the missing completely at random setting [21]. In the second experiment, floor($0.2n$) attributes in each record were randomly chosen and the values were deleted, where n is the number of attributes and floor($\cdot$) is a floor function. With the proposed method, we do not impute missing values and uses only non-missing values in records. In the comparison methods, we complemented the missing values with two methods: mean and MICE. Mean means that missing values were complemented with the means of existing values for each attribute. MICE was introduced in Section II-B. We used scikit-learn 0.23 [22] for the mean and MICE imputation.

2) Comparison methods for anomaly detection: The comparison methods were LOF, OCSVM, IF, AE, and DAE.

With LOF, a record is determined to be normal or an anomaly on the basis of the ratio of the local density of the record to that of the neighboring records. We set the number of neighboring points to 1, 3, 5, 15, and 35. Section IV-A3 presents the highest results among the above parameters. Parameters of the below methods were selected in the same manner as LOF.

OCSVM is an expansion of the SVM for unlabeled datasets. We used the radial basis function (RBF) kernel and set the kernel hyperparameter to $10^{-3}, 10^{-2}, 10^{-1}$, and 1.

IF is a random forest method. We set the number of decision trees to 1, 5, 10, 20, and 30.

AE is a deep learning based unsupervised anomaly detection method. The number of hidden layers was set to 3, and rectified linear unit (ReLU) functions were used as the activation functions between layers. We set the ratio of the number of each hidden layer to that of the dimensions of input to $\{0.75, 0.5, 0.75\}$ and $\{0.5, 0.25, 0.5\}$. In the training phase, the AE models were optimized with Adam [23].

DAE is an expansion of AE. In the training phase of DAE, the training records plus random noise are input for an AE model. Random noise was generated from the Gaussian where the mean was 0 and variance was 0.1. Other hyperparameters were the same as those for AE.

With the proposed method, the number of dimensions of each element of the input D was set to 128. ReLU functions were used as the activation function. In the training phase, the models of the proposed method were optimized with Adam [23].

We built the models of LOF, OCSVM, and IF with the package of scikit-learn 0.23 [22] and those of AE, DAE, and the proposed method with pytorch 1.4.0 [24].

3) Results: Table I lists the AUROCs with the complete datasets. The AUROCs of the proposed method were the best for 13 datasets and statistically highest for 18 datasets except for ALOI, Cardiotocograph, Pima, and SpamBase. These results indicate that the proposed method can perform better than conventional unsupervised anomaly detection methods in many cases.

Table III lists the AUROCs in the second experiment, in which incomplete data were used. The average AUROCs of the proposed method were also the highest. In the second experiment, the AUROCs of the proposed method were also the best for 10 datasets and statistically highest for 17 datasets among 11 methods. Table III shows that the proposed method achieved the best results for the highest number of datasets, even when there were missing records in the training and test datasets. These results indicate that the proposed method can use incomplete data without imputing missing values for training a model and detecting anomalous records.

B. Wi-Fi dataset

We show the results of experiments with the Wi-Fi dataset.

1) Data: The data were collected from an actual Wi-Fi service between February 1, 2018, and July 20, 2019, and include five attributes: association log, 2.4 GHz up, 2.4 GHz down, 5 GHz up, and 5 GHz down. The details of each attribute are shown in Table II. Each record in the data was

[1]These datasets are available at https://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/ along with the detailed information of each dataset.

generated every hour, and the data have 12,840 records. The five attributes have 33, 19, 19, 19, and 19 missing values, respectively. The reason these values were lost is not known.

We used 7,272 records from February 1, 2018, to November 30, 2018, as training data, 1,488 records from December 1, 2018, to January 31, 2019, as validation data, and 4,080 records from February 1, 2019, to July 20, 2019, as test data. The Wi-Fi service had a problem, and the traffic volume on the service decreased from March 23 to 26, 2019, which was part of the test period. In this experiment, we defined the records in the period as anomaly ones and other records as normal ones.

2) Exoerimental settings: The comparison methods were LOF, OCSVM, IF, AE, and DAE. Since the variance of the four attributes except the association log is large, these attributes were log-transformed. Then, all attributes were normalized to fit into the 0-1 range. In LOF, OCSVM, and IF experiments, the reversed outputs of the decision_function [22] were defined as the anomaly scores of input data. In the experiments with the proposed method, the reconstruction errors were defined as the anomaly score of input data. The anomaly score of each test record was normalized using those of the validation records. We also used mean and MICE methods to fill the missing values to input records including missing attributes into comparison methods.

The settings of the comparison methods in this experiment are described below. In LOF, the number of neighboring points was set to 35. In OCSVM, the RBF kernel was used and the hyperparameter of the kernel was set to 10^{-1}. In IF, the number of decision trees was set to 30. In AE and DAE, the number of hidden layers was set to three, and ReLU functions were used as the activation functions between layers. We set the number of nodes of each hidden layer to $\{4, 3, 4\}$. Random noise was generated from the Gaussian where the mean was 0 and variance was 0.1, for adding to input data in DAE. With the proposed method, the number of dimensions of each element of the input D was set to 32. ReLU functions were used as the activation functions. In the training phase, the models of both the proposed method and AE were optimized with Adam [23].

3) Results: Fig. 2 shows the anomaly score of test data with the comparison methods. All graphs show a peak derived from the anomalous records generated from March 23 to 26, 2019. The peak derived from the anomalous data is prominent in the graphs of LOF and the proposed method in Fig. 2. However, there are low anomaly score points in the peak of anomalous records in the graph of LOF. Many other peaks are shown, and the peaks derived from anomalous records are buried in the graph of the other four methods in Fig. 2.

For a quantitative discussion, the precision of the results for the proposed and comparison methods was calculated in the case of all anomalous records being detected, which is the case of recall = 1. The results are shown in Table IV. The number of false alerts is important for the operators of ICT systems because a large number of false alerts unnecessarily burdens the operators. The results show the proposed method has better precision than the comparison methods. Specifically, false alerts account for only 2 % of total alerts in the proposed method, but about 30 % to 40 % of total alerts in other anomaly detection methods.

	LOF	OCSVM	IF	AE	DAE	Proposed
ALOI	**0.657**	0.515	0.511	0.554	0.557	0.563
Annthyroid	0.624	0.482	**0.606**	0.601	0.593	**0.650**
Arrhythmia	0.654	0.697	0.700	**0.763**	**0.762**	**0.770**
Cardiotocograph	0.705	0.724	0.651	**0.747**	**0.764**	0.701
Glass	0.749	0.524	0.608	0.689	0.724	**0.924**
HeartDisease	0.539	0.653	**0.716**	**0.761**	**0.737**	**0.748**
Hepatitis	**0.693**	0.554	0.599	**0.710**	**0.734**	**0.780**
Ionosphere	0.872	0.753	0.788	**0.962**	**0.961**	**0.963**
KDDCu99	0.874	0.944	0.932	0.993	**0.993**	**0.994**
Lymphography	**0.960**	0.933	**0.937**	**0.998**	**0.989**	**0.993**
PageBlocks	0.847	0.804	0.820	0.908	0.908	**0.948**
Parkinson	0.719	0.815	0.771	0.820	0.812	**0.948**
PenDigits	**0.925**	0.448	0.559	0.793	0.826	**0.902**
Pima	0.550	0.550	0.586	**0.703**	**0.686**	0.645
Shuttle	0.944	0.522	0.598	0.761	0.745	**0.993**
SpamBase	0.615	0.507	**0.722**	**0.770**	**0.768**	0.712
Stamps	0.848	0.639	**0.906**	**0.826**	0.839	**0.919**
WBC	0.778	0.921	**0.977**	**0.958**	0.945	**0.982**
WDBC	0.833	0.844	0.844	**0.892**	**0.875**	**0.883**
WPBC	**0.461**	**0.507**	**0.494**	**0.528**	**0.561**	**0.506**
Waveform	**0.676**	0.510	**0.566**	**0.673**	**0.680**	**0.627**
Wilt	0.494	0.483	0.463	0.324	0.347	**0.919**
average	0.728	0.651	0.698	0.761	0.764	**0.821**

TABLE I
AUROCS ON 22 COMPLETE DATASETS BY USING UNSUPERVISED ANOMALY DETECTION METHODS (LOF, OCSVM, IF, AE, DAE AND PROPOSED METHOD). BOLD VALUES MEAN THAT THE VALUES ARE NOT STATISTICALLY DIFFERENT FROM THE BEST RESULT AMONG THE ANOMALY DETECTION METHODS WITH A T-TEST (P-VALUE = 0.05).

Attribute name	Explanation	Number of missing values
Association log	Number of times devices were connected to the access point	33
2.4 GHz up	Traffic volume uploaded on 2.4 GHz band	33
2.4 GHz down	Traffic volume downloaded on 2.4 GHz band	19
5 GHz up	Traffic volume uploaded on 5 GHz band	19
5 GHz down	Traffic volume downloaded on 5 GHz band	19

TABLE II
DATASET DETAILS

V. CONCLUSIONS

We proposed a new unsupervised anomaly detection method to handle missing data in ICT systems. Since in the proposed method, a deep learning model for sets with attention mechanisms handles a record as the combination of a set of non-missing values and a set of embedding vectors of non-missing attributes, the proposed method can be processed without compensating for missing data, unlike conventional unsupervised anomaly detection methods based on deep learning. We conducted anomaly detection experiments using 22 complete and incomplete benchmark datasets and data from a Wi-Fi service having missing values. These experimental results revealed that the proposed method outperforms other anomaly detection methods.

	LOF		OCSVM		IF		AE		DAE		proposed
	mean	MICE	mean	MICE	mean	MICE	mean	MICE	mean	MICE	
ALOI	0.580	**0.604**	0.514	0.514	0.510	0.510	0.541	0.546	0.541	0.548	0.548
Annthyroid	0.599	**0.633**	0.479	0.483	0.590	0.583	0.607	**0.642**	0.615	**0.646**	**0.653**
Arrhythmia	0.652	0.667	0.690	0.702	0.679	0.675	**0.759**	**0.747**	**0.755**	**0.752**	**0.761**
Cardiotocograph	0.693	0.711	0.710	0.708	0.658	0.649	**0.765**	0.650	**0.752**	0.647	0.668
Glass	0.546	0.695	0.525	0.537	0.602	0.613	0.695	**0.757**	0.710	**0.770**	**0.851**
HeartDisease	0.551	0.558	0.646	0.659	0.740	0.713	**0.755**	**0.800**	**0.790**	**0.738**	**0.806**
Hepatitis	0.618	0.426	0.557	0.577	0.622	0.613	**0.732**	**0.774**	**0.800**	**0.763**	**0.763**
Ionosphere	0.895	0.850	0.726	0.788	0.736	0.811	0.920	**0.959**	0.922	**0.957**	**0.955**
KDDCu99	0.816	0.753	0.929	0.934	0.888	0.939	**0.989**	**0.987**	**0.988**	**0.987**	**0.990**
Lymphography	0.927	0.847	0.903	0.887	0.893	**0.943**	**0.991**	**0.984**	**0.998**	**0.980**	0.958
PageBlocks	0.765	0.752	0.769	0.796	0.794	0.797	**0.886**	**0.918**	**0.883**	**0.922**	**0.919**
Parkinson	0.733	**0.772**	**0.748**	0.750	0.717	0.750	0.802	**0.855**	**0.839**	**0.803**	**0.865**
PenDigits	**0.814**	**0.867**	0.447	0.463	0.458	0.524	0.739	0.796	0.724	0.795	**0.749**
Pima	0.575	0.580	0.548	0.535	0.584	0.588	**0.698**	**0.673**	0.679	0.664	0.667
Shuttle	0.895	0.905	0.517	0.507	0.595	0.614	0.776	**0.879**	0.783	0.896	**0.956**
SpamBase	0.648	0.670	0.510	0.514	0.726	0.720	0.723	**0.800**	0.728	**0.788**	0.716
Stamps	0.794	0.823	0.626	0.629	0.839	0.865	0.882	0.878	**0.892**	0.877	**0.927**
WBC	0.868	0.625	0.930	0.925	**0.964**	**0.964**	**0.972**	**0.963**	**0.967**	0.941	**0.979**
WDBC	0.842	0.834	0.831	0.836	0.836	0.839	**0.886**	**0.894**	**0.887**	**0.903**	**0.897**
WPBC	0.486	0.485	0.470	**0.501**	0.478	0.488	**0.564**	0.505	**0.576**	0.478	**0.520**
Waveform	**0.599**	**0.640**	0.513	0.511	0.542	0.567	**0.631**	0.517	**0.695**	0.519	**0.591**
Wilt	0.486	0.598	0.484	0.535	0.462	0.492	0.366	0.494	0.378	0.510	**0.691**
average	0.699	0.695	0.640	0.649	0.678	0.694	0.758	0.774	0.768	0.768	**0.792**

TABLE III
AUROCS ON 22 DATASETS WITH ALL INCOMPLETE RECORDS BY USING UNSUPERVISED ANOMALY DETECTION METHODS.

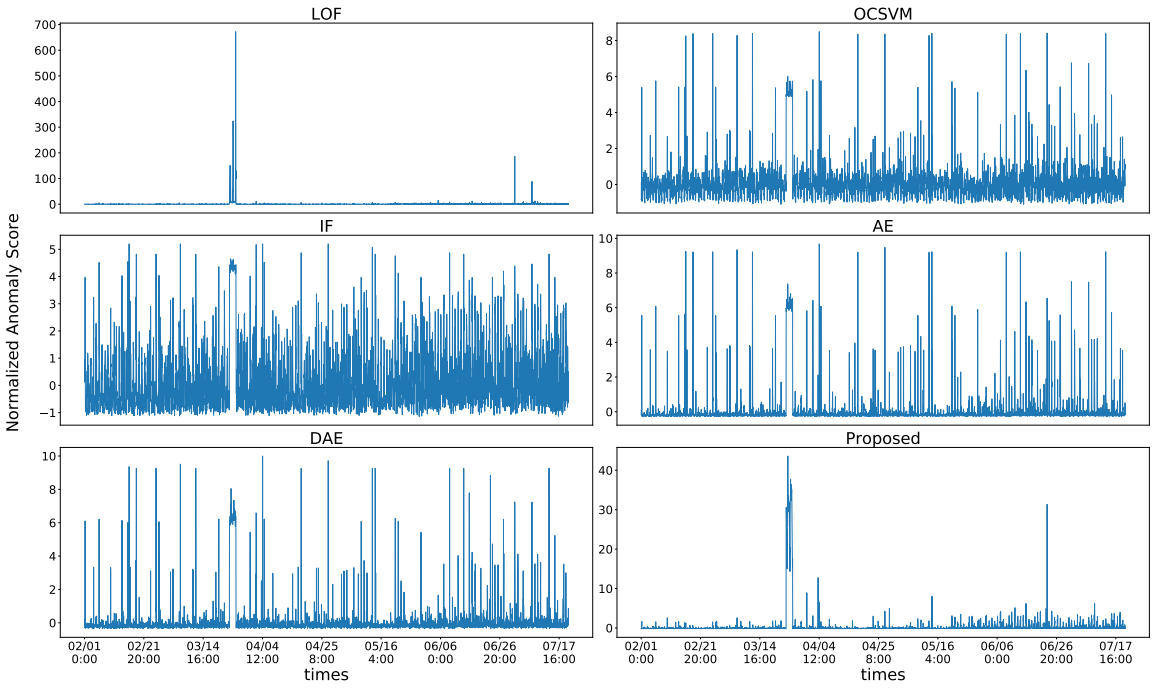

Fig. 2. Anomaly score by using unsupervised anomaly detection methods (LOF, OCSVM, IF, AE, DAE and proposed method). The five methods except the proposed method use MICE to fill missing values.

	LOF		OCSVM		IF		AE		DAE		proposed
	mean	MICE	mean	MICE	mean	MICE	mean	MICE	mean	MICE	
Precision	0.138	0.692	0.628	0.692	0.701	0.701	0.701	0.701	0.651	0.651	**0.982**

TABLE IV
PRECISION BY USING UNSUPERVISED ANOMALY DETECTION METHODS (LOF, OCSVM, IF, AE, DAE AND PROPOSED METHOD). IN THIS EXPERIMENT, WE FIXED RECALL = 1.

REFERENCES

[1] R. A. Bauder and T. M. Khoshgoftaar, "The detection of medicare fraud using machine learning methods with excluded provider labels," in *The Thirty-First International Flairs Conference*, 2018.

[2] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[3] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, 2018.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, p. 4.

[6] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*. IEEE, 2018, pp. 1–5.

[7] K. Yang, J. Zhang, Y. Xu, and J. Chao, "Ddos attacks detection with autoencoder," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–9.

[8] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer," *arXiv preprint arXiv:1810.00825*, 2018.

[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[10] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[12] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[13] G. E. Batista, M. C. Monard *et al.*, "A study of k-nearest neighbour as an imputation method." *HIS*, vol. 87, no. 251-260, p. 48, 2002.

[14] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?" *International journal of methods in psychiatric research*, vol. 20, no. 1, pp. 40–49, 2011.

[15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[16] J. Yoon, J. Jordon, and M. Van Der Schaar, "Gain: Missing data imputation using generative adversarial nets," *arXiv preprint arXiv:1806.02920*, 2018.

[17] C. K. Enders and D. L. Bandalos, "The relative performance of full information maximum likelihood estimation for missing data in structural equation models," *Structural equation modeling*, vol. 8, no. 3, pp. 430–457, 2001.

[18] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in neural information processing systems*, 2017, pp. 3391–3401.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[20] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.

[21] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[22] "scikit-learn 0.23." [Online]. Available: https://scikit-learn.org/stable/

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] "pytorch 1.4.0." [Online]. Available: https://pytorch.org/docs/1.4.0/