Robust Unsupervised Anomaly Detection with Variational Autoencoder in Multivariate Time Series Data

UMAPORN YOKKAMPON1, (Member, IEEE), ABBE MOWSHOWITZ2, SAKMONGKON CHUMKAMON1, AND EIJI HAYASHI1, (Member, IEEE)

1Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka 8208502, Japan
2Department of Computer Science, The City College of New York, New York 10031, USA

Corresponding author: Umaporn Yokkampon (e-mail: umaporn@ieee.org)

The Japanese Government has supported this work for the Religion revitalization project.

ABSTRACT Accurate detection of anomalies in multivariate time series data has attracted much attention due to its importance in a wide range of applications. Since it is difficult to obtain accurately labeled data, many unsupervised anomaly detection algorithms for multivariate time series data have been developed. However, building such a system is challenging since it requires capturing temporal dependencies in each time series and must also encode the inter-correlations between different pairs of time series. To meet this challenge, we propose a Multi Scale Convolutional Variational Autoencoder (MSCVAE) to detect anomalies in multivariate time series data. Firstly, multi scale attribute matrices are constructed from multivariate time series to characterize multiple levels of the system states at different time steps. Then, given the attribute matrices, a convolutional variational autoencoder is employed to generate reconstructed attribute matrices, and also an attention-based ConvLSTM network is used to capture the temporal patterns. In addition, a new ERR-based threshold setting strategy is developed to optimize anomaly detection performance instead of relying on the traditional ROC-based threshold setting strategy with an imbalanced dataset. Finally, the proposed framework is assessed by means of experiments on four datasets. The experimental results show that our proposed framework is superior to competing algorithms in terms of model performance and robustness, demonstrating that our model is effective in detecting anomalies in multivariate time series.

INDEX TERMS Anomaly detection, Multivariate time series, Convolutional variational autoencoder, Threshold setting strategy

I. INTRODUCTION In data mining, a time series [1] is a sequence of data points collected over time [2]. Such a sequence forms the basis of methods for tracking changes over time. Time series data can track changes in milliseconds, days, months, or even years. Time series data plays an important role in virtually all areas of science, engineering, commerce, and industry. Since data points in time series are collected at intervals, there is a relationship between successive observations, proportional or not, which distinguishes time series data from other kinds of data. There are two types of time series data. One is a univariate time series based on a single time dependent variable (or dimension). The other is a multivariate time series based on two or more time dependent, interrelated variables (or dimensions).

In recent years, research on time series data has focused on detecting and analyzing anomalies [3-6]. Anomaly detection is the identification of unexpected data points, i.e., events, or items that differ significantly from what is expected. Three types of time series anomalies have been identified, namely, point anomalies, contextual anomalies, and collective anomalies [7]. Point anomalies are points that exist far outside the range of the entire data set. Contextual anomalies are values that deviate significantly from most of the data points in the same context. Collective anomalies occur when a subset of the data points deviates substantially from the entire dataset. Anomaly detection is...
critical in many real world applications, such as analysis of potentially fraudulent transactions, sensor network faults, abnormal equipment behavior, etc.

Great strides have been made recently in multivariate time series anomaly detection systems [8-11]. This progress is also reflected in many areas of application [12-16]. Multivariate time series anomaly detection refers to anomaly detection of time series data with multiple sequences. The occurrence of anomalies in multivariate time series data typically involves multiple features. Sequential analysis of individual features cannot accurately locate all the anomalies, because several variables have to be examined simultaneously when analyzing segments of data. Moreover, encoding the inter-correlations between different pairs of time series also needs to be considered. Clearly, detecting anomalous parts of multivariate time series is a challenging problem.

Machine learning techniques are increasingly being adopted to detect anomalies because they can capture different characteristics of time series and detect anomalies effectively. Various anomaly detection methods for multivariate time series data have been developed. One such method employs unsupervised anomaly detection based on a Convolutional encoder-decoder with an attention-based Convolutional Long-Short Term Memory (ConvLSTM) network to detect and diagnose anomalies from signature matrices [17]. Liang et al. [18] make use of the multi-channel signature matrix transform to capture hidden non-linear anomalies. They also employ a forgetting mechanism to assign different weights, and use a novel threshold strategy to optimize performance. Another method uses a fully convolutional adversarial autoencoder to detect anomalies in video scenes and localizations [19]. Yet another method applies online unsupervised anomaly detection to industrial robot performance; this approach is based on Variational Autoencoder and Convolutional neural network with Sliding window to better recognize normal patterns of data for real-time anomaly detection [20]. Despite considerable progress, VAE based anomaly detection for imbalanced data has not received much attention.

In this paper, we propose an unsupervised anomaly detection method, i.e., Multi-Scale Convolutional Variational Autoencoder (MSCVAE), to detect anomalies in multivariate time series data. This model incorporates a new threshold setting strategy designed to fill a gap in existing methods.

The main contributions of our proposed framework are as follows.

- The novel MSCVAE framework is designed to detect anomalies in multivariate time series data. MSCVAE constructs multi-scale attribute matrices to characterize multiple levels of the system states across different time steps and then uses a convolutional variational autoencoder to extract the characteristics of the time series input. Specifically, we use an attention-based Convolutional Long-Short Term Memory (ConvLSTM) network to capture the temporal patterns and also to reconstruct the attribute matrices.
- We propose a novel threshold setting strategy based on a confusion matrix to optimize threshold selection of anomaly detection, which will help to improve the model robustness under conditions of imbalance between normal and abnormal data in multivariate time series.
- Experiments have been conducted on four datasets in order to verify the effectiveness of the proposed framework and the new threshold setting strategy. The results demonstrate that our method is superior to competing models in terms of anomaly detection performance and robustness under different ratios of imbalanced datasets.

The paper is structured as follows. Section II analyzes related work. Section III describes the proposed methodology framework. The experiment is detailed in Section IV. Finally, conclusions are presented in Section V.

II. RELATED WORK

Anomaly detection is challenging, and many approaches have been taken in various applications. In past years, many classical unsupervised approaches have been developed [21-26], including Principal Component Analysis (PCA) [27], which finds a low-dimensional projection that captures most of the variance in the data. The anomaly score is the reconstruction error of this projection. It is a linear algebra technique that can automatically achieve dimension reduction. Lee et al. [28] proposed online over-sampling PCA which makes use of online platforms for large-scale problems. By over-sampling the minority class of the target instance, their proposed algorithm allows them to determine the anomaly of the target instance.

One of the latest techniques for dimensionality reduction is Autoencoder [29], which is a popular approach for anomaly detection. Autoencoders consists of an encoder and decoder which reconstruct data samples and use the reconstruction error as the anomaly score [30]. Zhou et al. [31] proposed a deep autoencoder that combines robust PCA and deep autoencoders. It splits data into two parts: one part can be reconstructed by autoencoders; the other is the noise (outliers) in the data. Deep Autoencoding Gaussian Mixture Model (DAGMM) [32] jointly considers the Deep Autoencoder and Gaussian mixture Model to model the density distribution of multi-dimensional data.

Recently, Generative Adversarial Networks (GANs) [33] and LSTM-based approaches [34] have also shown promising performance for multivariate anomaly detection [35,36]. Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks (MAD-GAN) [37] proposed an unsupervised anomaly detection method based on generative adversarial networks (GAN) by considering complex dependencies among different time series variables. The LSTM-based Encoder-Decoder [38]...
models time series temporal dependency by means of LSTM networks and achieves better generalization capability than traditional methods. OmniAnomaly [39] is a stochastic recurrent neural network designed to avoid potential misleading by uncertain instances, which uses stochastic variable connection and normalizing flow to get reconstruction probabilities to determine anomalies. Ryota et al. [40] introduced a convolutional neural network and environment-dependent anomaly detector to detect the object, its attributes, and actions in the image. An environment-specific model can identify unusual attributes likely to explain abnormal patterns. Hu et al. [41] proposed a time series anomaly detection technique using six meta-features. This technique is a One-class Support Vector Machine (OC-SVM) system designed to identify the abnormal states of a univariate or multivariate time series based on local dynamics. Recently, a novel computational approach, namely Local Recurrence Rate based Discord Search (LRRDS), was proposed to identify discords from multivariate time series. This approach reduces the dimensionality of a time series and can detect variable-length discords using the given time series as the normality reference [42].

U-Net [43] shares some of the design characteristics of the system architecture described in this paper. It is a fully convolutional neural network incorporating so-called skip channels between encoding and decoding layers. This approach allows for integrating high and low level features in a way that prevents information loss and reduces the depth of sequential layers. However, U-net is limited in that the rate of learning may be diminished in the middle layers of a high depth case. This means the system is at risk of ignoring layers with abstract features, thus limiting extraction of some of the complex features that could help image segmentation in medical images. Moreover, U-Net requires rather substantial training time because of a large number of hyper-parameters. By contrast, our system uses a more effective method consisting of the multi-scale MSCVAE model that has an attention-based ConvLSTM network. This method allows for capturing interesting features and key patterns to assist with anomaly detection in multivariate time series data.

Some Variational Autoencoders (VAEs) [44-48] have taken a probabilistic approach, and autoencoders have been combined with Gaussian mixture modeling [19]. Bayer and Osendorfer [49] used Variational Inference to learn the underlying distribution of sequences and introduced recurrent stochastic networks. The core of these models is an RNN extended with a latent variable. Söllch et al. [50] used a Stochastic Recurrent Network (STORN) to detect robot anomalies using unimodal signals. Park et al. [51] presented the combination of LSTMAE using multimodal sensory signals, where LSTM is used to replace the feed-forward network in VAE. Pereira et al. [52] proposed applying a self-attention mechanism to VAE to improve the encoding-decoding process for energy data. Despite the intrinsic unsupervised setting, most of them may still not be able to detect anomalies effectively since most of the methods cannot capture temporal dependencies across different time steps in multivariate time series data.

Our goal is to develop specific approaches for multivariate time series data, which take account of the correlation between the series. Multivariate time series data usually contain noise and mask the true anomalies. Also, we need to consider the characteristics of the data itself in addition to the correlation between series. Combining the advantages of the above models, the MSCVAE model we propose can not only capture the time pattern effectively but also has the ability to deal with noise. Although these methods use a variety of technologies and operate in different ways, common to all is the need to have indicators to judge anomalies.

In anomaly detection problems, thresholding (i.e., setting a threshold) is an effective strategy for evaluating anomaly scores. The scores output by the model are separated into outliers and normal data using the threshold. Most systems select the threshold with the aid of the receiver operating characteristics (ROC) curve. The ROC curve is a plot in a unit square of true positive rate (TPR) versus false positive rate (FPR). Using the ROC curve, the threshold that optimizes specificity as well as sensitivity can be identified. However, the imbalanced class distribution of a data set poses a problem because most unsupervised learning techniques are designed for balanced class distribution [53]. To solve this problem, we propose a new threshold setting strategy based on a confusion matrix, thus selecting a better threshold, improving model performance and robustness.

## III. OVERVIEW OF THE METHODOLOGY

In this section, we first introduce the problem to be tackled in section III-A, and then elaborate on how to generate attribute matrices in section III-B. Next, we explain the proposed Multi-scale convolutional variational autoencoder (MSCVAE) in detail in section III-C - III-G. Specifically, we first show how to generate multi-scale system attribute matrices. Then, we encode the spatial information in attribute matrices via a convolutional variational encoder and model the temporal information via attention-based ConvLSTM. We reconstruct attribute matrices based upon a convolutional variational decoder. Finally, we detail the strategy of threshold setting in section III-H. Figure 1 presents an overview of the MSCVAE framework. The terminology and notation used in the paper are summarized in Table 1.

### A. PROBLEM STATEMENT

In this work, we focus on multivariate time series, given the historical data of n time series \( X = (x_1, x_2, ..., x_n) \in \mathbb{R}^{n \times T} \), of length \( T \), and assume that there are anomalies in the data. We aim to detect anomalous events at certain time steps after \( T \). We use only the normal dataset for training to characterize the various time series patterns under normal conditions. For the validation, we also use only the normal dataset. Both normal and abnormal data are used for testing.
For multivariate time series segment $X^*$, we define two time series, namely, $x_i = \{x_i^{l-w}, x_i^{l-w-1}, ..., x_i^{l}\}$, $x_j = \{x_j^{l-w}, x_j^{l-w-1}, ..., x_j^{l}\}$, and their correlation $m_{ij}' \in M'$ can be computed as follows:

$$m_{ij}' = \sum_{t=0}^{w} \gamma_{t} x_{i}^{t} x_{j}^{t}$$

where $w$ is the sliding window size, assigned the value 10 at each time step. Also, the interval between two segments is set to 10. To further explain the attribute matrices generation process, the algorithm’s pseudocode is shown in Table 2.

### C. VARIATIONAL AUTOENCODER (VAE)

Autoencoder is a typical neural network that transforms inputs into outputs with the minimum possible error. It consists of three sequential layers: the latent variable layer, the encoder layer, and the decoder layer. The encoder encodes or compresses the input data $x \in \mathbb{R}^{d}$ into a latent variable $z \in \mathbb{R}^{c}$, while the decoder decodes or reconstructs the encoded data (latent variable) $z$ back to the input dimension. The objective of Autoencoder is to get a reconstructed value from the latent variable that is close to the original input data. A disadvantage of the autoencoder is that a hidden layer may not be continuous and poses serious computational challenges. Variational Autoencoder (VAE) is a type of Autoencoder designed to address this issue.

VAE is an unsupervised learning network for dimension reduction, which can learn to represent complex data without supervision using a deep neural network. VAE relies on probability distributions of observations in latent variables and makes strong assumptions about the distribution of latent variables. Therefore, instead of generating an encoder that outputs a single value to describe each latent variable, it creates an encoder that operates with a probability distribution for each latent variable.

VAE has three main layers, namely, encoder, decoder, and loss function. The input denoted by $x$ is generated from a latent variable $z$. VAE defines latent variable $z$ as a random variable distributed according to a prior function $p(z)$, defined as a multivariate unit Gaussian distribution $N(0,I)$. The first process samples data $x$ from the conditional likelihood distribution $p(x|z)$. Then, the decoder $p(x|z)$ functions as a generator model corresponding to the distribution of the decoded variable given the encoded one. However, the encoder is defined by the conditional probability function $p(z|x)$, i.e., the distribution of the encoded variable given the decoded one.

The loss function of VAE has two terms: the first term is a reconstruction loss term, which can maximize the reconstruction likelihood, and the second term is a regularization term, which ensures that the distribution learning $q(z|x)$ is like the true prior distribution $p(z)$. The loss function is given by Eq. 2.

### B. GENERATE ATTRIBUTE MATRICES

Given the importance of correlations between the different pairs of time series for characterizing the system state [54], we generate an $n \times n$ attribute matrix $M'$ utilizing the pairwise inner product of multivariate time series within this segment. This is designed to illustrate the inter-correlations between different pairs of time series in a multivariate time series segment from time $t-w$ to $t$. We adopt the method for calculating the attribute matrix proposed in [17] to capture the similarity of shape and value scale correlations between two time series. Examples of attribute matrices are shown in Figure 1, part A. The pseudocode of the algorithm for generating the attribute matrix is introduced in Table 2.

---

**TABLE 1.** Terminology and notation used in the paper

| Notations | Descriptions |
|-----------|--------------|
| $X$       | Multivariate time series dataset |
| $T$       | Length of time series |
| $n$       | Dimension of time series |
| $w$       | Size of sliding window |
| $t$       | Time step |
| $M'$      | Attribute matrix |
| $p(z)$    | Prior distribution |
| $p(z|x)$  | Probabilistic encoder |
| $p(x|z)$  | Probabilistic decoder |
| $P^b$     | Input of the first layer of the encoder |
| $N(0,I)$  | A multivariate unit Gaussian distribution |
| $z$       | Latent variable |
| $*$       | Convolutional operation |
| $o$       | Hadamard product |
| $\oplus$  | Deconvolutional operation |
| $\otimes$ | Concatenation operation |
| $f(\cdot)$| Activation function unit |
| $\hat{z}_t$ | Output, reconstruction at $t$ time step |

**TABLE 2.** The pseudocode of attribute matrices generation

| Algorithm | Generating attribute matrices |
|-----------|------------------------------|
| **Input:** | Multivariate time series $X$, length of time series $T$, window size $w$, dimension of time series $n$ |
| **Output:** | Attribute matrix $M$ |
| 1: for $i$ in $T$ do |
| 2: for $j$ in $T$ do |
| 3: $X_i = x[i-w+1], x[i-w+2], ..., x[i]$ |
| 4: $X_j = x[i-w+1], x[i-w+2], ..., x[i]$ |
| 5: $m_{ij}'[i][j] = \frac{X_i \times X_j}{w}$ |
| 6: return $M$ |
\[ L(x_i) = -E_{z \sim q(z|x)} \left[ \log p(x_i | z) \right] + KL(q(z | x) \| p(z)) \] (2)

The regularization term of VAE is the Kullback-Leibler divergence which measures the difference between the encoder’s distribution \( q(z | x) \) and \( p(z) \), and constitutes a measure of how close \( p \) is to \( q \), as described in Figure 1. part B(a).

D. CONVOLUTIONAL ENCODER

We use convolution encoders [55] to encode the spatial form of the system attribute matrices. Four convolutional encoder layers are applied to extract the values of the attribute matrices in our framework. \( P^{l0} \) represents the input of the first layer and assumes that \( P^{l-1} \in \mathbb{R}^{n \times n \times O} \) denotes the feature maps in \((l-1)\)-th layer. The output of \( l \)-th layer is given by:

\[ P^{l} = f(W^{l}_f \ast P^{l-1} + b^{l}_f) \] (3)

where \( \ast \) denotes the convolutional operation, \( W^{l}_f \) is the filter kernel of layer \( l \), \( b^{l}_f \) is a term of bias, \( P^{l} \) is the output of layer \( l \), \( P^{l-1} \) is the output of the \((l-1)\) layer (input of layer \( l \)) and \( f(\cdot) \) is an activation function. Figure 1. part B(b) illustrates the detailed encoding process of attribute matrices.

E. ATTENTION-BASED CONVOLUTIONAL LSTM

The spatial feature maps, extracted by the convolutional encoder, depend on the previous time steps. The traditional Convolutional LSTM performance might degrade as the length of the sequence increases. Xingjian et al. [56] introduced the LSTM model to the ConvLSTM to capture the temporal information in a video sequence. Here we have used an attention-based ConvLSTM to obtain the temporal characteristics which can be adaptive with different time steps. ConvLSTM consists of four convolutional layers of four ConvLSTM blocks. Details of ConvLSTM are shown in Figure 1. part B(c). At time \( t \), the results of \( P^{l} \) calculated by Eq. 3. in layer \( l \) of the convolutional encoder are used as input to the \( l \)-th layer of ConvLSTM. The other input is the time \( t-1 \) hidden state \( H^{l-1} \) in hidden layer \( l \). The ConvLSTM cell is formulated as:

\[ i^{l} = \sigma(W^{l}_i \ast P^{l} + W^{l}_hi \ast H^{l-1} + W^{l}_ci \circ C^{l-1} + b^{l}_i) \]
\[ f^{l} = \sigma(W^{l}_f \ast P^{l} + W^{l}_hf \ast H^{l-1} + W^{l}_cf \circ C^{l-1} + b^{l}_f) \]
\[ o^{l} = \sigma(W^{l}_o \ast P^{l} + W^{l}_ho \ast H^{l-1} + W^{l}_co \circ C^{l-1} + b^{l}_o) \]
\[ G^{l} = \tanh(W^{l}_g \ast P^{l} + W^{l}_hg \ast H^{l-1} + W^{l}_cg \circ C^{l-1} + b^{l}_g) \]
\[ C^{l} = f^{l} \circ C^{l-1} + i^{l} \circ G^{l} \]
\[ H^{l} = o^{l} \circ \tanh(C^{l}) \] (4)

where \( \sigma \) is the sigmoid function, \( W^{l}_i \) is the filter kernel of the input gate, \( W^{l}_c \) is the filter kernel of the input gate process for the input of the hidden layer at the previous time step, and \( W^{l}_h \) is the filter kernel of the cell state at the previous time step in the input gate. The input \( P^{l} \), the cell state \( C^{l} \), the hidden state \( H^{l} \), the candidate memory \( G^{l} \), and the gates \( i^{l}, f^{l}, o^{l} \) are all 3D tensors. The symbol “\( \ast \)” denotes the convolutional operator, and “\( \circ \)” denotes the
Hadamard product. Figure 1. part B.(c) illustrates the temporal modeling procedure.

**F. CONVOLUTIONAL DECODER**

We can reconstruct the input data using the extracted attribute output produced by the decoder. For the convolutional decoder, we follow the procedure discussed in [17] which is formulated as:

\[
\hat{p}^{l-1} = \begin{cases} 
    f((\hat{W}^{t,l} \otimes \hat{H}^{t,l} + \hat{b}^{l,t})), & l = 4 \\
    f((\hat{W}^{t,l} \otimes (\hat{H}^{t,l} \oplus \hat{p}^{l-1}) + \hat{b}^{l,t})), & l = 3, 2, 1
\end{cases}
\]

where \( \otimes \) denotes the deconvolutional operation, \( \oplus \) is the concatenation operation, \( f(\cdot) \) is the activation function unit (same as in the encoder), \( \hat{W}^{t,l} \) and \( \hat{b}^{l,t} \) are the filter kernel and bias parameter, respectively, of the \( l \)-th deconvolutional layer. We reconstruct the attribute matrix of each layer by reversely deconvolving the deconvolution from layer \( l = 4 \) to layer \( l = 1 \). We then deconvolve the result \( o^{l-4} \) of the ConvLSTM at the last layer \( l = 4 \) and reconstruct the previous layer matrix \( \hat{p}^{l-3} \). After that, the deconvolution operation is performed as concatenation to reconstruct the attribute matrix sequence of the \( l-1 \)-th layer. Figure 1. part B.(d) illustrates the decoding procedure.

**G. LOSS FUNCTION**

Mean Square Error (MSE) is used as a loss function to optimize the parameter values in our model. MSE is the most commonly used loss function suitable for anomaly detection problems. MSE is given by the following:

\[
L_{MSE} = \frac{1}{N} \sum_{t=1}^{N} \sum_{c=1}^{k} \left| P^{t,0} - \hat{P}^{t,0} \right|^2
\]

where \( P^{t,0} \in \mathbb{R}^{n 	imes n} \). We employ the mini-batch stochastic gradient descent method and the Adam optimizer to minimize the above loss. After sufficient training epochs, the learned neural network parameters are utilized to infer the reconstructed attribute matrices of validation and test data. Finally, we perform anomaly detection and diagnosis based on the residual attribute matrices, elaborated in the next section.

**H. THRESHOLD SETTING STRATEGY**

VAEs map each group of attribute matrices to anomaly scores during the anomaly detection process. The threshold setting aims to give the best boundary to distinguish normal and abnormal samples, and then detected samples are labeled normal or abnormal by means of VAEs.

The confusion matrix is commonly used to show how each test value predicts classes compared to their actual classes, including True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

Based on the ROC-based threshold setting strategy [57], the best threshold is always sought as the dot on the upper left corner or point (0,1). However, the ROC-based threshold setting strategy is insensitive to imbalanced datasets, leading to poor performance in anomaly detection. According to the false positive rate formula, when a large number of normal samples are wrongly judged as anomalies, the false positive rate (FPR) changes a little. Therefore, anomaly detection performance may be widely divergent for two threshold values that are close together. In this case, it is not easy to select an optimal threshold among different discrete points. Moreover, Eq. 8 is often adopted to calculate the distance between a point on the ROC curve and the point (0,1). The threshold corresponding to the minimum distance is selected as the best threshold. However, TN is much larger than TP, which makes \( \frac{FP^2}{FP^2 + TN^2} \) much smaller than \( \frac{FN^2}{FN^2 + TP^2} \).

Therefore, when choosing the minimum distance, the ROC-based strategy focuses more on the latter, only choosing the smaller FN as much as possible. Eventually, it leads to low F1.

\[
\text{Distance} = \sqrt{(1 - TP)^2 + FPR^2}
\]

\[
= \frac{FN^2}{FN^2 + TP^2} + \frac{FP^2}{FP^2 + TN^2}
\]

We adopt a threshold setting strategy using a confusion matrix to avoid the problems described above. The confusion matrix can effectively reflect the anomaly detection results, even for an imbalanced dataset. Therefore, we introduce a new error rate (ERR) defined as a function of TP, FP, FN, and TN, as shown in the formula below. The aim of the new threshold setting strategy is to minimize ERR, which means fewer samples are misjudged. As a result, the optimal threshold can be selected according to the minimum ERR.

\[
\text{ERR} = \frac{FP}{FP + TP + TN}
\]

As indicated above, Precision, Recall, F1-Score, and ERR are utilized as evaluation metrics for anomaly detection. Moreover, the geometric mean or G-mean [58] of sensitivity and specificity, is suitable for evaluating the quality of binary (two-class) classifications for a balanced as well as imbalanced dataset problem. Therefore, G-mean is used as another evaluation metric in the following experiments.

\[
\text{G-mean} = \sqrt{\text{Recall} \times \text{Specificity}}
\]

\[
= \frac{TP}{TP + TN} \times \frac{TN}{TP + FN}
\]

**IV. EXPERIMENTS AND DISCUSSION**

**A. DATASETS DESCRIPTION**

To illustrate the performance of the proposed method, we have carried out extensive experiments on four publicly available benchmark datasets, namely, Satellite, Wafer (UCR), EEG, and Opt. Descriptions of these data sets are
given below, and the details of the experiments are listed in Table 3. The second dataset can be downloaded from the website: http://www.cs.ucr.edu/~eamonn/discords/, and the other three datasets are from the UCI public database. Details of the above datasets are given below.

Satellite: This dataset was generated from data purchased from NASA by the Australian Centre for Remote Sensing. It consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image together with the classification associated with the central pixel in each neighborhood. The data set is used to predict the category of the image of the observed region of soil. Soil classes “red soil”, “grey soil”, “mixture class”, and “very damp grey soil” constitute the normal class. The class of anomalies consists of “cotton crop”, “damp grey soil”, and “soil with vegetation stubble”. This classification is based on multi-spectral values, consisting of 36 time series and 6435 instances.

Wafer: The Wafer dataset is related to semiconductor microelectronics fabrication, using data collected from various sensors during the processing of silicon wafers for semiconductor fabrication. Each time series in this dataset contains measurements recorded by one sensor in the course of processing one wafer by one tool. The dataset contains 152 attributes, and there is a large class imbalance between normal and anomaly.

EEG: This dataset is from one continuous EEG measurement with the Emotiv EEG Neuroheadset. The duration of the measurement was 117 seconds. The eye state was detected via a camera during the EEG measurement and later added manually to the file after analyzing the video frames. This dataset consists of 14 EEG attribute values and one indicating the eye state.

Opt: The Opt dataset is used for preprocessing programs made available by NIST to extract normalized bitmaps of handwritten digits from a preprinted form, which contains 64 parameters. It is a character recognition dataset for integers handwritten digits from a preprinted form, which contains 64

| Datasets | #Training | #Validation | #Testing | Length | Anomaly rate |
|----------|-----------|-------------|----------|--------|--------------|
| Satellite | 3,000 | 1,000 | 2,435 | 36 | 32% |
| Wafer | 4,000 | 1,000 | 2,164 | 152 | 11% |
| EEG | 8,000 | 2,000 | 4,980 | 14 | 44.9% |
| Opt | 2,500 | 1,000 | 1,716 | 64 | 3% |

TABLE 4. Detailed structural comparison of the algorithms

| Model | Kernel | Dense | CNN | LSTM | Dropout |
|-------|--------|-------|-----|------|---------|
| KNN   | x      | x     | x   | x    | x       |
| LOF   | x      | x     | x   | x    | x       |
| iForest | x    | x     | x   | x    | x       |
| OCSCVM |       |       |     |      |         |
| GMM   | x      | x     | x   | x    | x       |
| PCA   | x      | x     | x   | x    | x       |
| ABOD  | x      | x     | x   | x    | x       |
| HBOS  | x      | x     | x   | x    | x       |
| CBLOF | x      | x     | x   | x    | x       |
| MOGAAL |      |       |     |      |         |
| DAGMM |       |       |     |      |         |
| GAN   | x      | x     | x   | x    | x       |
| MAD-GAN |     |       |     |      |         |
| OmniAnomaly | |       |     |      |         |
| Autoencoder | | x    | x   | x    |         |
| VAEs  | x      | x     | x   | x    | x       |
| LSTM-AE |       |       |     |      |         |
| MSCVAE | x      | x     | o   | o    | o       |

Our framework. The computer configuration is Intel(R) Core(TM) i7-9750H CPU 2.60GHz 6-Core Processor with NVIDIA GeForce RTX 2060 GPU, 16G Memory.

Furthermore, seventeen algorithms were used to verify the superiority of the proposed framework, including classical anomaly detection algorithms and deep architecture models. The classical anomaly detection algorithms include K-Nearest Neighbor (KNN), Local Outlier Factor (LOF), Isolation Forest (iForest), One-class Support Vector Machine (OCSCVM), Gaussian Mixture Model (GMM), Principal Component Analysis (PCA), Angle-based Outlier Detector (ABOD), Histogram-based Outlier Score (HBOS), Cluster-based Local Outlier Factor (CBLOF), Multiple-Objective Generative Adversarial Active Learning (MOGAAL).

The deep architecture models include Deep Autoencoding Gaussian Mixture Model (DAGMM), Generative Adversarial Network (GAN), Multivariate Anomaly Detection with GAN (MAD-GAN), OmniAnomaly, Autoencoder, Variational Autoencoder (VAEs), and Long Short-Term Memory Autoencoder (LSTM-AE). Table 4 presents the details of all the algorithms.

C. PERFORMANCE EVALUATION

All anomaly detection models in experiments were trained with the corresponding training subsets, consisting of normal samples. Then the models were verified using a validation method consisting of normal samples and testing subsets, including both normal and abnormal data. The model evaluation metrics, i.e., precision, recall, F1-Score, and G-mean, were in the range of 0 to 1. Higher precision, recall, F1-Score, G-mean, and lower ERR indicate better model performance. Table 5, Table 6, Table 7, and Table 8 show the
confusion matrix elements and evaluation metrics of anomaly detection models on four datasets. The bold fonts in Table 5, Table 6, Table 7, and Table 8 indicate that the MSCVAE is superior to the other models.

The comparison experiments show that 10 classical anomaly detection algorithms and 7 deep architecture models have been implemented. Generally speaking, the ability of classical anomaly detection algorithms is limited when facing modeling issues for multivariate time series, and this conclusion is supported by results for the two comprehensive indexes, F1 and G-mean. DAGMM reduces the dimension and extracts features by using neural networks, slightly improving compared to GMM. GAN achieves similar results to DAGMM on the Satellite dataset (shown in Table 5) but performs well on the other three datasets. MAD-GAN attempts to map data into the latent space and detect anomalies via discriminant results and reconstruction errors generated from the mapping process. MAD-GAN achieves good results on the Opt dataset and gives slightly better results than OmniAnomaly. LSTM-AE performs better on

### TABLE 5. Anomaly detection results of Satellite dataset.

| Model   | Confusion matrix value | Performance evaluation |
|---------|------------------------|------------------------|
|         | TP | FP | FN | TN | ERR | Precision | Recall | F1-Score | G-mean |
| KNN     | 45 | 75 | 68 | 51 | 0.4386 | 0.3750 | 0.3982 | 0.3863 | 0.4015 |
| LOF     | 54 | 66 | 56 | 63 | 0.3607 | 0.4500 | 0.4099 | 0.4696 | 0.4896 |
| iForest | 51 | 69 | 58 | 61 | 0.3812 | 0.4250 | 0.4679 | 0.4454 | 0.4686 |
| OCSVM   | 55 | 65 | 70 | 49 | 0.3846 | 0.4583 | 0.4400 | 0.4490 | 0.4349 |
| GMM     | 72 | 48 | 73 | 46 | 0.2892 | 0.6000 | 0.4966 | 0.5434 | 0.4929 |
| PCA     | 112 | 8 | 11 | 108 | 0.0351 | 0.9333 | 0.9106 | 0.9218 | 0.9207 |
| ABOD    | 44 | 76 | 73 | 46 | 0.4578 | 0.3667 | 0.3761 | 0.3713 | 0.3766 |
| HBOS    | 48 | 72 | 70 | 49 | 0.4260 | 0.4000 | 0.4068 | 0.4034 | 0.4059 |
| CBLOF   | 51 | 69 | 62 | 57 | 0.3898 | 0.4250 | 0.4513 | 0.4378 | 0.4519 |
| MOGAAL  | 52 | 68 | 78 | 41 | 0.4224 | 0.4333 | 0.4000 | 0.4160 | 0.3879 |
| DAGMM   | 80 | 40 | 95 | 24 | 0.2778 | 0.6667 | 0.4571 | 0.5424 | 0.4140 |
| GAN     | 85 | 37 | 82 | 37 | 0.2298 | 0.6967 | 0.5215 | 0.5965 | 0.5173 |
| MAD-GAN | 82 | 37 | 87 | 42 | 0.2298 | 0.6891 | 0.5125 | 0.5878 | 0.5220 |
| OmniAnomaly | 84 | 37 | 79 | 39 | 0.2313 | 0.6942 | 0.5153 | 0.5915 | 0.5142 |
| Autoencoder | 83 | 37 | 78 | 41 | 0.2298 | 0.6917 | 0.5155 | 0.5907 | 0.5206 |
| VAEs    | 105 | 9 | 31 | 94 | 0.0433 | 0.9211 | 0.7721 | 0.8400 | 0.8394 |
| LSTM-AE | 107 | 11 | 70 | 53 | 0.0651 | 0.9052 | 0.6000 | 0.7216 | 0.7049 |
| MSCVAE  | 122 | 4 | 6 | 107 | 0.0172 | 0.9683 | 0.9531 | 0.9606 | 0.9585 |

### TABLE 6. Anomaly detection results of Wafer dataset.

| Model   | Confusion matrix value | Performance evaluation |
|---------|------------------------|------------------------|
|         | TP | FP | FN | TN | ERR | Precision | Recall | F1-Score | G-mean |
| KNN     | 58 | 48 | 37 | 67 | 0.2775 | 0.5472 | 0.5105 | 0.5771 | 0.5964 |
| LOF     | 60 | 46 | 56 | 48 | 0.2987 | 0.5660 | 0.5172 | 0.5405 | 0.5139 |
| iForest | 54 | 52 | 43 | 61 | 0.3114 | 0.5094 | 0.5567 | 0.5320 | 0.5412 |
| OCSVM   | 61 | 45 | 56 | 48 | 0.2922 | 0.5755 | 0.5214 | 0.5471 | 0.5187 |
| GMM     | 60 | 46 | 45 | 59 | 0.2788 | 0.5660 | 0.5714 | 0.5687 | 0.5666 |
| PCA     | 61 | 45 | 42 | 62 | 0.2679 | 0.5755 | 0.5922 | 0.5837 | 0.5858 |
| ABOD    | 50 | 50 | 45 | 65 | 0.3030 | 0.5000 | 0.5263 | 0.5128 | 0.5454 |
| HBOS    | 49 | 57 | 41 | 63 | 0.3373 | 0.7075 | 0.5034 | 0.5882 | 0.5346 |
| CBLOF   | 35 | 73 | 62 | 42 | 0.4932 | 0.4623 | 0.5444 | 0.5000 | 0.3562 |
| MOGAAL  | 45 | 41 | 50 | 54 | 0.3813 | 0.3113 | 0.3474 | 0.3284 | 0.4716 |
| DAGMM   | 75 | 31 | 74 | 30 | 0.2279 | 0.4245 | 0.4737 | 0.4478 | 0.4975 |
| GAN     | 120 | 7 | 15 | 68 | 0.1170 | 0.9449 | 0.8889 | 0.9160 | 0.8977 |
| MAD-GAN | 112 | 12 | 21 | 65 | 0.1864 | 0.9032 | 0.8421 | 0.8716 | 0.8431 |
| OmniAnomaly | 107 | 10 | 33 | 60 | 0.2575 | 0.9145 | 0.7643 | 0.8327 | 0.8094 |
| Autoencoder | 100 | 12 | 49 | 49 | 0.0745 | 0.8929 | 0.6711 | 0.7663 | 0.7342 |
| VAEs    | 90 | 21 | 12 | 87 | 0.1061 | 0.8108 | 0.8824 | 0.8451 | 0.8431 |
| LSTM-AE | 86 | 9 | 64 | 51 | 0.0616 | 0.9053 | 0.5733 | 0.7020 | 0.6981 |
| MSCVAE  | 142 | 2 | 0 | 66 | 0.0095 | 0.9861 | 1.0000 | 0.9930 | 0.9852 |
low dimension datasets than high dimension datasets (shown in Tables 6 and 8). As an excellent sequence to sequence model, Autoencoder fails at temporal data and performs well on the Opt dataset as shown in Table 8. VAEs achieve better results than Autoencoder on four datasets, which suggests that a high dimension dataset is a challenge for the training of VAEs.

MSCVAE is superior to the other algorithms discussed above, except in the EEG dataset, which gives slightly better recall, F1-Score, and G-mean values than our proposed framework. Also, the Opt dataset gives slightly better recall values than in our proposed framework. Nevertheless, it is clear that our proposed framework can effectively improve anomaly detection performance on multivariate time series. In other words, MSCVAE is much better than baseline methods, as it can handle both inter-sensor correlations and temporal patterns of multivariate time series effectively.

The reasons for the superior performance of our framework can be summarized as follows: (1) As a deep learning model with temporal patterns, MSCVAE can achieve good performance with no need for supervised training. (2) VAEs with an attention-based ConvLSTM network framework can effectively identify anomalies. (3) Our framework can model both inter-sensor correlations and temporal patterns of multivariate time series effectively.

### TABLE 7. Anomaly detection results of EEG dataset.

| Model     | Confusion matrix value | Performance evaluation |
|-----------|-------------------------|------------------------|
|           | TP | FP | FN | TN | ERR | Precision | Recall | F1-Score | G-mean |
| KNN       | 164| 92 | 94 | 140| 0.2323| 0.6406 | 0.6357 | 0.6381 | 0.6193 |
| LOF       | 152| 104| 123| 111| 0.2834| 0.5938 | 0.5527 | 0.5725 | 0.5342 |
| iForest   | 184| 72 | 123| 111| 0.1778| 0.7188 | 0.6840 | 0.7010 | 0.6791 |
| OCSVM     | 163| 93 | 96 | 138| 0.2360| 0.6367 | 0.6293 | 0.6330 | 0.6132 |
| GMM       | 170| 86 | 113| 121| 0.2281| 0.6641 | 0.6007 | 0.6308 | 0.5926 |
| PCA       | 189| 67 | 76 | 158| 0.1618| 0.7383 | 0.7132 | 0.7255 | 0.7077 |
| ABOD      | 160| 96 | 111| 123| 0.2533| 0.6250 | 0.5904 | 0.6072 | 0.5758 |
| HBOS      | 178| 78 | 77 | 157| 0.1889| 0.6953 | 0.6980 | 0.6967 | 0.6829 |
| CBLOF     | 157| 99 | 103| 131| 0.2558| 0.6133 | 0.6038 | 0.6085 | 0.5865 |
| MOGAAL    | 92 | 164| 146| 88 | 0.4767| 0.3594 | 0.3866 | 0.3725 | 0.3674 |
| DAGMM     | 194| 62 | 173| 61 | 0.1956| 0.7578 | 0.5286 | 0.6228 | 0.5120 |
| GAN       | 297| 10 | 33 | 150| 0.0219| 0.9674 | 0.9000 | 0.9325 | 0.9186 |
| MAD-GAN   | 297| 26 | 27 | 140| 0.0562| 0.9195 | 0.9167 | 0.9181 | 0.8793 |
| OmniAnomaly| 296| 14 | 38 | 142| 0.0310| 0.9548 | 0.8862 | 0.9193 | 0.8982 |
| Autoencoder| 225| 34 | 142| 89 | 0.0977| 0.8687 | 0.6131 | 0.7188 | 0.6660 |
| VAEs      | 259| 16 | 23 | 192| 0.0543| 0.9418 | 0.9184 | 0.9300 | 0.9208 |
| LSTM-AE   | 251| 14 | 117| 108| 0.0375| 0.9472 | 0.6821 | 0.7930 | 0.7770 |
| MSCVAE    | 296| 9  | 34 | 151| 0.0197| 0.9705 | 0.8970 | 0.9323 | 0.9201 |

### TABLE 8. Anomaly detection results of Opt dataset.

| Model     | Confusion matrix value | Performance evaluation |
|-----------|-------------------------|------------------------|
|           | TP | FP | FN | TN | ERR | Precision | Recall | F1-Score | G-mean |
| KNN       | 82 | 23 | 13 | 52 | 0.1465| 0.7810 | 0.8632 | 0.8200 | 0.7736 |
| LOF       | 54 | 51 | 32 | 33 | 0.3696| 0.5143 | 0.6279 | 0.5654 | 0.4967 |
| iForest   | 78 | 28 | 24 | 40 | 0.1918| 0.7358 | 0.7647 | 0.7500 | 0.6707 |
| OCSVM     | 80 | 25 | 13 | 52 | 0.1592| 0.7619 | 0.8062 | 0.8081 | 0.7622 |
| GMM       | 82 | 23 | 14 | 51 | 0.1474| 0.7810 | 0.8542 | 0.8159 | 0.7673 |
| PCA       | 90 | 16 | 12 | 52 | 0.1013| 0.8491 | 0.8824 | 0.8654 | 0.8214 |
| ABOD      | 81 | 24 | 21 | 44 | 0.1611| 0.7714 | 0.7941 | 0.7826 | 0.7168 |
| HBOS      | 77 | 28 | 19 | 46 | 0.1854| 0.7333 | 0.8021 | 0.7662 | 0.7061 |
| CBLOF     | 82 | 23 | 13 | 52 | 0.1465| 0.7810 | 0.8632 | 0.8200 | 0.7736 |
| MOGAAL    | 51 | 54 | 58 | 7  | 0.4821| 0.4857 | 0.4679 | 0.4766 | 0.2317 |
| DAGMM     | 84 | 22 | 44 | 21 | 0.1732| 0.7925 | 0.6563 | 0.7179 | 0.5661 |
| GAN       | 97 | 6  | 6  | 66 | 0.0061| 0.9989 | 0.9417 | 0.9652 | 0.9166 |
| MAD-GAN   | 104| 25 | 2  | 39 | 0.1488| 0.8062 | 0.9811 | 0.8851 | 0.6688 |
| OmniAnomaly| 105| 28 | 3  | 34 | 0.1677| 0.7895 | 0.9722 | 0.8714 | 0.6073 |
| Autoencoder| 90 | 9  | 28 | 43 | 0.0634| 0.9091 | 0.7627 | 0.8295 | 0.7942 |
| VAEs      | 89 | 11 | 4  | 66 | 0.0663| 0.8900 | 0.9570 | 0.9223 | 0.9057 |
| LSTM-AE   | 92 | 3  | 35 | 40 | 0.0222| 0.9684 | 0.7244 | 0.8288 | 0.8209 |
| MSCVAE    | 99 | 1  | 4  | 66 | 0.0060| 0.9900 | 0.9612 | 0.9754 | 0.9730 |
D. ABLATION STUDY

We have conducted an extensive study to illustrate the impact of different components on the model results, using two key modules of our model: multi-scale attribute matrices and an attention-based ConvLSTM. Three variants of MSCVAE are considered in the evaluation:

- MSCVAE\(_w\): MSCVAE framework without attention-based ConvLSTM.
- CVAE\(_a\): MSCVAE framework without the multi-scale attribute matrices.
- CVAE\(_w\): MSCVAE framework without both multi-scale attribute matrices and an attention-based ConvLSTM.

F1-Scores and G-means on four datasets are reported in Fig. 2. We observe that our proposed framework, MSCVAE (marked in blue), is obviously superior to the other three competing models on anomaly detection tasks, which indicates the importance of multi-scale attribute matrices and an attention-based ConvLSTM. However, MSCVAE\(_w\) (marked in pink) and CVAE\(_w\) (marked in yellow) methods can obtain approximately equal results with all four datasets. Besides, the results of CVAE\(_a\) (marked in green) are much better than MSCVAE\(_w\) and CVAE\(_w\), which indicates the importance of an attention-based ConvLSTM. Furthermore, we see that under all conditions, both multi-scale attribute matrices and attention-based ConvLSTM can effectively improve the anomaly detection performance of multivariate time series.
(b) Wafer dataset

(c) EEG dataset

FIGURE 3. F1-Score and G-mean comparison all four datasets under different anomaly rate. (continued)
E. ROBUSTNESS EVALUATION
Anomaly detection often suffers from the dataset imbalance problem, which means there are more normal samples than anomaly samples. To combat this problem, we used the two criteria, F1-Score, and G-mean, with different rates of anomalies to evaluate our model’s robustness. Fig. 3 shows the chart of F1-Score and G-mean comparisons with different rates of anomalies. The comparison results indicate that the F1-Score and G-mean of most algorithms tend to increase as anomaly rates increase. Our proposed framework’s F1-Score and G-mean values are highest on all datasets under different anomaly rates. Therefore, we are justified in concluding that our proposed framework has good robustness even in the face of dataset imbalance problems.

F. THRESHOLD SETTING STRATEGY COMPARISON
In order to verify the effectiveness of the proposed threshold setting strategy, we compared two threshold setting strategies based on four datasets, shown in Table 9, Table 10, Table 11, and Table 12. Table 9, Table 10, Table 11, and Table 12 show the top three thresholds on four datasets. The rank indicates the suitability of the threshold. Therefore, the highest rank is the best threshold. As explained in formulas 14 and 15, the threshold with the lower distance and the ERR
As stated, the ROC-based strategy pays more attention to FN and ignores FP, so it selects the best threshold with lower FN. In contrast, the ERR-based strategy focuses on FP, and its best selected threshold makes for similarly excellent FP performance. Consider the experimental results in Table 9. The first threshold of the ERR-based strategy has 4 FP and 6 FN, and the corresponding F1-Score and G-mean are 0.9606 and 0.9585. However, the first threshold of ROC-based strategy has 28 FP and 11 FN, and the corresponding F1-Score and G-mean are just 0.8632 and 0.8204. ROC-based strategy fails to achieve the best threshold because of its excessive attention to FN.

Further investigation was conducted to analyze the same proposed framework using the two threshold setting strategies under different anomaly rates, shown in Fig. 4. The ERR-based strategy achieves a better threshold than the ROC-based strategy under a low anomaly rate on four datasets. Besides, it is clear that the difference between the two strategies decreases as the anomaly rate increases. In short, the proposed threshold setting strategy is superior to the traditional ROC-based strategy and can achieve an appropriate threshold even when confronted with a dataset imbalance problem.

![Graphs showing F1-Score comparisons between two threshold setting strategies on four datasets under different anomaly rates.](image-url)

**FIGURE 4.** F1-Score comparisons two threshold setting strategies on four datasets under different anomaly rate.
V. CONCLUSION

In this paper, we proposed a novel MSCVAE framework to solve the anomaly detection problem for multivariate time series data. The framework employs multi scale (resolution) system attribute matrices which transform multivariate time series into multi scale attribute matrices. This approach allows for characterizing the state of the entire system in different time segments, and adopts the convolutional variational autoencoder to generate reconstructed attribute matrices, which makes our framework more robust by taking advantage of VAEs. An attention-based Convolutional Long-Short Term Memory (ConvLSTM) network is used to capture the temporal patterns. The framework can model both inter-sensor correlations and temporal dependencies of multivariate time series. Finally, a new ERR-based threshold setting strategy is adopted, instead of a traditional ROC-based threshold setting strategy, to achieve better model performance. To verify the effectiveness of the proposed framework, experiments on four datasets were implemented. The results demonstrate that MSCVAE can outperform state-of-the-art baseline methods.

The work reported here justifies the following conclusions.

1) Multi-scale attribute matrices provide an effective preprocessing method for characterizing system states at different time segments of multivariate time series with no need of prior knowledge.

2) CNN structure is embedded in the encoder, and the decoder of the VAEs model is adopted to extract the characteristics of the time series. An attention-based Convolutional Long-Short Term Memory (ConvLSTM) network is used to capture the temporal patterns and reconstruct the attribute matrices, providing an effective unsupervised anomaly detection method. Combined with the proposed ERR-based threshold setting strategy, the MSCVAE based framework can achieve excellent performance.

3) Experiments on four datasets indicate that our framework is outperforms competing models in detection accuracy and robustness under imbalanced datasets. An extensive experiment was also conducted to verify that the proposed threshold setting strategy can acquire an optimal threshold in the anomaly detection task, thus contributing to the superior anomaly detection performance of our model.

REFERENCES

[1] P. J. Brockwell, and R. A. Davis, “Time series: theory and methods,” Springer Science & Business Media, May, 13, 2009.

[2] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM computing surveys (CSUR), vol. 41, no. 3, pp. 15, 2009, DOI: 10.1145/1541880.1541882.

[3] L. Li, J. Yan, H. Wang, and Y. Jin, “Anomaly detection of time series with smoothness-inducing sequential variational autoencoder,” IEEE transactions on neural networks and learning systems, vol. 32, no. 3, pp. 1177–1191, 2020.

[4] N. Lattep, S. Amizadeh, and I. Flint, “Generative and scalable framework for automated time-series anomaly detection,” in Proc. of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1939–1947, Aug. 2015, DOI: 10.1145/2783258.2788611.

[5] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, “DeepAR: A deep learning approach for unsupervised anomaly detection in time series,” IEEE Access, vol. 7, pp. 1991–2005, 2019, DOI: 10.1109/ACCESS.2018.2886457.

[6] H. Izakian, and W. Pedrycz, “Anomaly detection and characterization in spatio-temporal time series data: A cluster-centric approach,” IEEE Transactions on Fuzzy Systems, vol. 22, no. 6, pp. 1612–1624, 2014, DOI: 10.1109/TFUZZ.2014.2302456.

[7] R. Chalapathy, and S. Chawla, “Deep learning for anomaly detection: A survey,” arXiv:1901.03407, 2019. [Online]. Available: http://arxiv.org/pdf/1901.03407.pdf.

[8] S. Du, T. Li, Y. Yang, and J. S. Homg, “Multivariate time series forecasting via attention-based encoder–decoder framework,” Neurocomputing, vol. 388, pp. 269–279, 2020, DOI: 10.1016/j.neucom.2019.12.118.

[9] M. Goldstein, and S. Uchida, “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,” PloS one, vol. 11, no. 4, pp. e0152173, 2016, DOI: 10.1371/journal.pone.0152173.

[10] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, “Clustering-based anomaly detection in multivariate time series data,” Applied Soft Computing, vol. 100, pp. 106919, 2021. DOI: 10.1016/j.asoc.2020.106919.

[11] M. Canizo, I. Triguero, A. Conde, and E. Onieva, “Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study,” Neurocomputing, vol. 363, pp. 246–260, 2019, DOI: 10.1016/j.neucom.2019.07.034.

[12] S. Plakias, and Y. S. Boutalis, “Exploiting the generative adversarial framework for one-class multi-dimensional fault detection,” Neurocomputing, vol. 332, pp. 396–405, 2019, DOI: 10.1016/j.neucom.2018.12.041.

[13] K. Hundman, V. Constantiou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding,” in Proc. of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 387–395, Jul. 2018, DOI: 10.1145/3219819.3219845.

[14] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, “An empirical study on network anomaly detection using convolutional neural networks,” in Proc. of IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1595–1598, Jul. 2018.

[15] S. Zvavak, and M. Isekefiyeli, “Anomaly-based intrusion detection from network flow features using variational autoencoder,” IEEE Access, vol. 8, pp. 108346–108358, 2020, DOI: 10.1109/ACCESS.2020.3001350.

[16] B. Yan, and G. Han, “Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system,” IEEE Access, vol. 6, pp. 41238–41248, 2018, DOI: 10.1109/ACCESS.2018.2858277.

[17] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, “A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data,” in Proc. of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 1409–1406, Jul. 2019, DOI: 10.1609/aaai.v33i01.33011409.

[18] H. Liang, L. Song, J. Wang, L. Guo, X. Li, and J. Liang, “Robust unsupervised anomaly detection via multi-time scale DC-GANs with forgetting mechanism for industrial multivariate time series,” Neurocomputing, vol. 423, pp. 444–462, 2021, DOI: 10.1016/j.neucom.2020.10.084.

[19] N. Li, and F. Chang, “Video anomaly detection and localization via multivariate gaussian fully convolution adversarial autoencoder,” Neurocomputing, vol. 369, pp. 92–105, 2019, DOI: 10.1016/j.neucom.2019.08.044.

[20] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, “Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder,” IEEE Access, vol. 8, pp. 47072–47081, 2020, DOI: 10.1109/ACCESS.2020.2979892.

[21] B. Min, J. Yoo, S. Kim, and D. Shin, “Network Anomaly Detection Using Memory-Augmented Deep Autoencoder,” IEEE Access, vol.
convolutional auto-encoders for hierarchical feature extraction,” in J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked (ICMLA)
International Conference on Machine Learning and Applications
algorithms—the Numenta anomaly benchmark,” in A. Lavin, and S. Ahmad, “Evaluating real-time anomaly detection
recurrent neural network,” in Proc. dual-stage attention-based recurrent neural network for time series
adversarial networks,” in Proc. MAD-GAN: convolutional networks for time series classification,
F. Karim, S. Majumdar, H. Darabi, and S. Chen, “Lstm fully auto-encoding Gaussian process regression for unsupervised
unsupervised anomaly detection,” in Proc. of the 23rd ACM SIGKDD International Conference on Knowledge
detection: A survey,” In Anomaly detection using autoencoders in high performance
processing systems,” in Advances in neural information processing systems, 27, 2014.
Advances in neural information processing systems, 1997.
Advances in neural information processing systems, 2015. [Online]. Available: http://arxiv.org/pdf/1511.0634.pdf.
Advances in neural information processing systems, 2018. [Online]. Available: http://arxiv.org/pdf/1901.04997.pdf.
A dual-stage attention-based recurrent neural network for time series prediction,” LCIA, 2017.
Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828–2837, Jul, 2019. DOI: 10.1145/3292500.3330672.
"Joint detection and recounting of abnormal events by learning deep generic knowledge,” In Proc of the IEEE International Conference on Computer Vision, pp. 3619-3627, 2017.
"Detecting anomalies in time series data via a meta-feature based approach," IEEE Access, vol. 6, pp. 27760-27776, 2018, DOI: 10.1109/ACCESS.2018.2840086.
“A novel computational approach for discord search with local recurrence rates in multivariate time series,” Information Sciences, vol. 477, pp. 220-233, 2019.
"U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention, pp. 234-241, Oct. 2015.
"Auto-encoding variational bayes,” Proc. Int. Conf. Learn. Represent., April, 2014.
"Learning sparse representation with variational auto-encoder for anomaly detection,” IEEE Access, vol. 6, pp. 33353–33361, 2018, DOI: 10.1109/ACCESS.2018.2848210.
"A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection,” Knowledge-Based Systems, vol. 199, pp. 105187, 2020, DOI: 10.1016/j.knosys.2019.105187.
"Unsupervised anomaly detection using variational auto-encoder based feature extraction," in Proc. 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 1–7, Jun. 2019.
"Variational autoencoder based anomaly detection using reconstruction probability,” Special Lecture on IE, vol. 2, no. 1, pp. 1–18, 2015.
NIPS Workshop Advances Variational Inf, 2014.
"Variational inference for on-line anomaly detection in high-dimensional time series,” Proc ICML 2016.
"A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder,” IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1544-1551, 2018.
"Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention,” In IEEE international conference on machine learning and applications (ICMLA), pp. 1275-1282, 2018.
"Learning from Imbalanced Data Sets. Berlin: Springer, 2018.
"Detecting anomalies in time series data via a meta-feature based approach,” In Proc. of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.
"Convolutional LSTN network: A machine learning approach for precipitation nowcasting,” In Advances in neural information processing systems, pp. 802-810, 2015.
"Receive operating characteristic (ROC) curve analysis for medical diagnostic test evaluation,” Caspian journal of internal medicine, vol. 4, no. 2, pp. 627-635, 2013.
"Geometric means and medians with applications to target detection,” IET Signal Processing, vol. 11, no. 6, pp. 711-720, 2017.
UMAPORN YOKKAMPON (Member, IEEE) received the B.S. and M.S. degrees from King Mongkut's University of Technology North Bangkok, Bangkok, Thailand, in 2016 and 2019, respectively. She is currently pursuing the Ph.D. degree in Computer Science and Systems Engineering with Kyushu Institute of Technology, Fukuoka, Japan. Her current research interests include data mining, anomaly detection, and time series analysis.

ABBE MOWSHOWITZ received the Ph.D. degree from University of Michigan in 1967. He has been professor of computer science at the City College of New York and member of the doctoral faculty at the Graduate Center of the City University of New York since 1984. His current research interests lie in two areas are organizational and managerial issues in computing, and network science. In addition to teaching and research, He has acted as consultant on the uses and impacts of information technology (especially computer networks) to a wide range of public and private organizations in North America and Europe.

SAKMONGKON CHUMKAMON received the Ph.D. degree in Computer Science and Systems Engineering from Kyushu Institute of Technology in 2017. He was a postdoctoral researcher at Guangdong University of Technology, China, in 2017-2019. Presently he is a postdoctoral researcher in Kyushu Institute of Technology since 2019. His research interests include factory automation robots and social robots.

EIJI HAYASHI (Member, IEEE) received the Ph.D. (Dr. Eng.) degree from Waseda University, Japan, in 1996. He is currently a Lecturer with the Department of Intelligent and Control Systems, Kyushu Institute of Technology. His research interests include Intelligent mechanics, Mechanical systems and Perceptual information processing. He is a member of The Institute of Electrical and Electronics Engineers (IEEE) and The Japan Society of Mechanical Engineers (JSME).