IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Time-series to Image-transformed Adversarial Autoencoder for Anomaly Detection

JIYOUNG KANG[1], MINSEOK KIM[2], JINUK PARK[2], and SANGHYUN PARK[2], (Member, IEEE)

[1]IP Prosecution Team, Samsung Electronics, 33 Seongchon-gil, Seocho-gu, Seoul, 06765, South Korea
[2]Department of Computer Science, Yonsei University, Seodaemun-gu, Seoul 03722, South Korea

Corresponding author: Sanghyun Park (sanghyun@yonsei.ac.kr)

This research was supported by the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. RS-2023-00229822).

ABSTRACT The automation of systems and the accelerated digital transformations across various industries have rendered the manual monitoring of systems difficult. Therefore, the automatic detection of system anomalies is essential in diverse industries. Various deep learning-based techniques have been developed for anomaly detection in multivariate time-series data with promising performance. However, there are several challenges: 1) difficulty in understanding the relationships among time-series data due to their complexity and high-dimensionality, 2) limitation in distinguishing anomalies from normal data that exhibit similar distributional patterns, and 3) lack of intuitive interpretation of anomaly detection results. To address these issues, we propose a novel approach referred to as the time-series to image-transformed adversarial autoencoder (T2IAE), which adopts image transformation techniques and convolutional neural network (CNN)-based adversarial learning. Image transformation techniques were used to effectively capture the local features of adjacent time points. Two CNN-based adversarial autoencoders competitively learned to distinguish between normal and abnormal data. We experimentally analyzed five real-world multivariate time-series datasets, wherein the proposed model achieved superior anomaly detection performance compared with state-of-the-art methods. Moreover, the proposed model enables humans to intuitively interpret the detection results, facilitating appropriate explanations of the results and enhancing the model's usability.

INDEX TERMS Anomaly Detection, Unsupervised Learning, Multivariate Time-series Data, Image Transformation

I. INTRODUCTION

With the advancement of Industry 4.0 driven by the Internet of Things, various industries are automating and digitizing their systems [1]. These real-world systems comprise several interconnected sensors that generate a significant amount of time-series data. Owing to the high-dimensionality and complexity of sensor data, monitoring them manually is becoming increasingly difficult. Therefore, approaches that can rapidly and automatically detect anomalies and notify human operators have been extensively researched, leading to the emergence of anomaly detection as a major research area in various application domains, such as manufacturing [2], healthcare [3], finance [4], security [5], social analysis [6], drug development [7], and IoT networks [8, 9].

Time-series anomaly detection aim to identify the data points that significantly deviate from normal patterns within a chronologically ordered dataset. In the case of multivariate time series, it is central to model both the interactions between variables and the effects that occur over time. Anomalous time-series data are infrequent and costly to label due to the diverse manifestations of anomalies, such as unpredictable fluctuations, missing data, and seasonal variations. Therefore, anomaly detection in time-series data often uses unsupervised learning, wherein models are trained solely on normal data. Traditional unsupervised learning approaches include distance-based methods, such as the local outlier factor [10] and K-nearest neighbors [11]; density-based methods such as the density-based spatial clustering of applications with noise [12] and ordering points to identify the clustering structure [13]; and clustering-based methods such as K-means [14]. However, these approaches exhibit poor performance and require high computational costs when handling high-dimensional

complex data, rendering their application to real-world multivariate time-series data difficult.

In recent years, various deep learning-based techniques have been proposed for the detection of anomalies in multivariate time-series data. These techniques employ architectures such as autoencoders (AEs) [15], recurrent neural networks (RNNs) [16], long short-term memory (LSTM)-based approaches [17], variational autoencoders (VAEs) [18], graph neural networks [19], generative adversarial networks (GANs) [20], and hybrid approaches [21, 22]. The most crucial aspect of a well-established time series anomaly detection model is its ability to accurately identify anomalies. To achieve this factor, it is essential to precisely capture inter-variable and temporal dependencies. Also, the interpretability of the detected anomalies is another important factor. If the model can provide human-recognizable explanations for anomalies, it can aid in detecting, explaining, and preventing them, thus enhancing its applicability in real-world scenarios. Although recent advanced deep learning models demonstrate promising performance for anomaly detection, several challenges remain unaddressed.

The main challenges in multivariate time-series anomaly detection include the high-dimensionality of the series and the presence of anomalies that closely resemble normal patterns. Time-series data often exhibit intricate relationships between different variables, making them difficult to learn, particularly in high-dimensional contexts. This inherent complexity impedes the accurate identification of anomalies. Additionally, distinguishing anomalies that are similar to normal data distributions is challenging. Unsupervised learning methods, which are trained exclusively on normal data, struggle to detect these subtle anomalies accurately.

Another significant challenge is the lack of interpretability in detected anomalies. Providing an intuitive explanation for why an observation was identified as an anomaly is crucial for assisting human operators in troubleshooting and solving real-world problems. However, interpretability in multivariate time-series data is difficult to achieve. Previous deep learning-based studies, while achieving acceptable performance, further complicate the task of making the detection process transparent and understandable.

To address these issues, we propose a novel approach referred to as the time-series to image-transformed adversarial autoencoder (T2IAE), which utilizes image transformation techniques and convolutional neural network (CNN)-based adversarial learning. By transforming multivariate time-series data into images, our model learns complex relationships through temporal and spatial information between variables. Anomalies tend to exhibit stronger correlations with adjacent time points [23], and the proposed model facilitates learning these correlations between consecutive time-series data. By employing an adversarial learning approach with two AEs, our model effectively captures subtle differences between normal data and anomalies. Consequently, the active combination of these two approaches precisely detects anomalies. Additionally, this model requires less computational cost compared to other models by utilizing a simple and efficient autoencoder architecture. Moreover, the model presents the detection results in a human-recognizable image format, enabling users to visually inspect and intuitively interpret the detected anomalies. The effectiveness of the proposed model is demonstrated with respect to anomaly detection in time-series data by considering five real-world datasets using three image transformation techniques. The primary contributions of this study can be summarized as follows.

- We propose T2IAE, a novel approach that learns complex spatial–temporal patterns. The proposed approach effectively detects anomalies by using images that capture subtle changes in temporal information within variables and the correlations between variables in multivariate time-series data.
- We performed empirical studies using publicly available real-world datasets to evaluate the anomaly detection performance of the proposed model. The experimental results demonstrate that the developed approach outperforms other state-of-the-art methods.
- The model's effective time-series to image transformation enables an intuitive interpretation of the results. This allows for clear explanation of the detected anomalies, ultimately enhancing the model's usability.

The remainder of this paper is organized as follows. Section II briefly discusses the related studies on unsupervised anomaly detection using multivariate time-series data. Section III describes the proposed T2IAE model in detail. Section IV introduces the experimental environment and settings. Section V discusses the obtained experimental results and intuitive interpretability for evaluating the performance of the proposed model. Finally, Section VI summarizes the study findings and concludes the paper.

II. RELATED WORK

One of the traditional models for unsupervised anomaly detection is the Isolation Forest (IF) model [24]. IF utilizes randomly generated binary trees to detect anomalies based on the degree of isolation of data points. Although the model is computationally efficient and easy to train, it is sensitive to the distribution of normal data. Moreover, its performance varies significantly depending on the type of anomaly. A one-class support vector machine (OCSVM) is an algorithm that identifies a hyperplane that separates normal and abnormal univariate data [25]. Although this can be trained with

relatively fewer data points, it is not suitable for data in which the boundary between anomalies and normal data is unclear. This algorithm transforms the data into a higher-dimensional space and identifies a hyperplane that maximizes the distance between the transformed and original data. Autoregressive integrated moving average (ARIMA) is a representative statistical model used for time-series forecasting, wherein past values and forecasting errors are leveraged to estimate the current value [26]. This model combines autoregressive (AR) and moving average (MA), and non-stationary data are transformed into stationary data via differencing. This method effectively captures the characteristics of trends, seasonality, and autocorrelation in time-series data. However, the model is unsuitable for multivariate time-series data because it requires multiple hyperparameters for AR and MA.

Deep learning neural networks have gained significant attention in recent years owing to their ability to capture complex nonlinear relationships in time-series data [27]. AEs are neural network models trained to condense input data into a lower-dimensional latent space and to recreate output data that are highly similar to the original input. They can extract important features from the data and detect anomalies based on reconstruction errors. AE-based models enable distinct distributions at each timestamp while capturing temporal dependencies within the time-series data [28], [29]. However, these models cannot preserve important information present in the original data in a lower-dimensional space.

To compensate for this, the deep autoencoding Gaussian mixture model (DAGMM) combines an AE with GMM to learn the normal data distribution [21]. This method preserves important information in a lower-dimensional space by maintaining reduced dimensionality and reconstruction error characteristics. The unsupervised anomaly detection (USAD) model is composed of two AEs that utilize adversarial training to maximize the reconstruction error between normal and abnormal data [30]. However, these two methods do not account for the temporal dependency of the sequences.

SES-AD is another approach to project high-dimensional time-series into a low-dimensional embedding space [31]. This method employs a space-embedding strategy that first reduces the dimensionality of the time series and then calculates the dissimilarity between adjacent sub-sequences in this lower-dimensional space. The dissimilarity vector is subsequently processed by an LSTM-based model for signal reconstruction and abrupt change point identification, followed by a statistical method to detect abnormal sub-sequences. While SES-AD effectively reduces the dimensionality of multivariate time series to identify anomalies, it has limited interpretability, similar to the aforementioned models.

MTAD-GAT utilizes two graph layers, namely, the feature- and time-oriented layers. It captures temporal dependency within each time series by forecasting a single timestamp and reconstructing the entire time series [32]. The algorithm OmniAnomaly utilizes a stochastic RNN to detect anomalies in multivariate time-series data [33]. It focuses on learning robust representations by incorporating stochastic variable connections and planar normalizing flow techniques. MAD-GAN incorporate an LSTM-based GAN architecture to capture the temporal dependencies in time-series data [34]. This model uses an anomaly score that combines the losses of the generator and discriminator of GAN. CAE-M uses a characterization network and a memory network to consider spatial–temporal dependency in time-series data [35]. Although above methods capture the fundamental aspects of time-series data, they fail to consider the interactions between variables and their significance in multivariate time-series data. Additionally, their complex model structures lead to high computational costs during model training and inference. Furthermore, these methods struggle to provide intuitive explanations for anomaly detection results.

Recently, LRRDS utilized time series visualization techniques for anomaly detection [36]. LRRDS identifies discords in multivariate time series by generating a recurrence plot and detecting abrupt changes through local recurrence rates. It segments the time series at these change points and calculates the dissimilarity between sub-sequences to find discords. However, the algorithm for determining anomalies or discords relies on statistical features, which limits its flexibility. To address this issue and enhance both the interpretability and performance of the model, we propose a new approach that visualizes the time series and then applies a neural network-based anomaly detection algorithm.

III. PROPOSED ARCHITECTURE

A. PROBLEM DESCRIPTION

Univariate time-series data (τ) contain one variable value (x_t) at one time point (t):

$$\tau = \{x_1, x_2, \dots, x_t\}. \quad (1)$$

The primary objective of the univariate time-series analysis is to investigate the correlation, trend, and seasonality within the value based on chronological order. Unlike univariate time-series data, multivariate time-series data comprise multiple variables at each timestep, wherein each variable represents a distinct aspect and undergoes a change over time. Owing to the high-dimensionality of multivariate time-series data and the interactions and influences between multiple variables, a more intricate analysis is necessary. Therefore, multivariate time-series analysis requires an architecture capable of meticulously

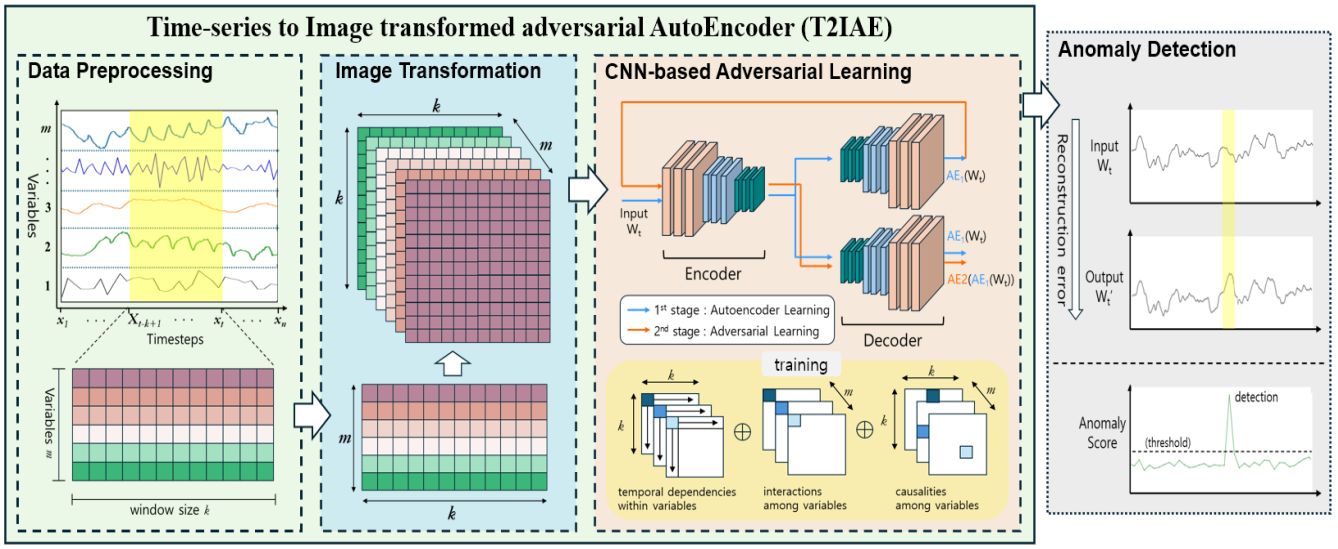

FIGURE 1. Overall architecture of the proposed time-series to image-transformed adversarial autoencoder (T2IAE), which comprises three main parts: data preprocessing, image transformation, and CNN-based adversarial learning.

exploring temporal dependencies, interactions, and causal relationships among multiple variables.

B. OVERVIEW OF THE PROPOSED MODEL

Fig. 1 depicts the overall architecture of the proposed model, which comprises three main parts: data preprocessing, image transformation, and CNN-based adversarial learning. First, the data are normalized and divided into fixed-size window sequences. Each window sequence is then transformed into an image, which is a variable containing temporal information; the window sequence is converted into a three-dimensional (3D) collection of these images. The image collection is trained in two stages using one encoder and two decoders. The first stage focuses on training the model to accurately reconstruct the original input, whereas the second stage trains the model to distinguish between the original input and the output of the first stage. These training techniques enable the identification of temporal associations, interactions, and causal relationships between variables. Finally, the reconstruction error between the original input and final output is calculated and used as an anomaly score for detection.

C. DATA PREPROCESSING

In the case of multivariate time-series data, machine learning models learn by extracting features from variables. Variables with different measurement scales could exhibit a disproportionate influence on the analysis, potentially introducing bias; this can be addressed using normalization. Normalization transforms the numerical variables into a common scale while maintaining their relative importance. This ensures that each variable exhibits an equal impact on the learning process. Normalization can lead to a more stable and efficient learning process for the models, resulting in better performance. In this study, we used the min–max normalization, which scales variables using their minimum and maximum values. All variables were transformed within a range of 0 to 1, where the minimum and maximum values of each variable were 0 and 1, respectively [36]. Normalization can be implemented using

$$\tilde{x_i} = \frac{(\mathrm{x}_i - min(X))}{max(X) - min(X)}, \tag{2}$$

where $x_i \in \mathbb{R}^T$ denotes the time series of the i-th value in a variable X; and $\tilde{x_i}$ denotes the normalized x_i.

Anomalies in the time-series data are highly correlated with neighboring time points [23]. Therefore, identifying local anomalies in the entire dataset can be difficult. To address this issue, the normalized data are divided into fixed-length window sequences using the sliding window algorithm [38]. A window sequence W_t of size k at time t can be defined as

$$W_t = \{\tilde{x}_{t-k+1}, \ldots, \tilde{x}_{t-1}, \tilde{x}_t\}. \tag{3}$$

The normalized data can be transformed into a window $W = \{W_1, W_2, \ldots, W_T\}$.

D. IMAGE TRANSFORMATION

We transformed the time-based window sequences generated during data preprocessing into images. Converting time-series data into images can highlight, capture, and compress local features that are dispersed over time [39].

Fig. 2 illustrates the process of transforming a single window sequence into images. A single window sequence comprises m variables over k consecutive time points. Each variable is transformed into an image of size $k \times k$, resulting in the transformation of a single window sequence into an image group of m images.

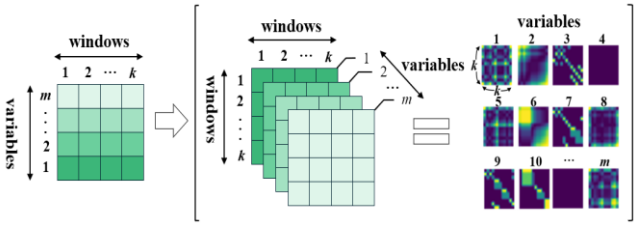

FIGURE 2. Image transformation of a single window sequence

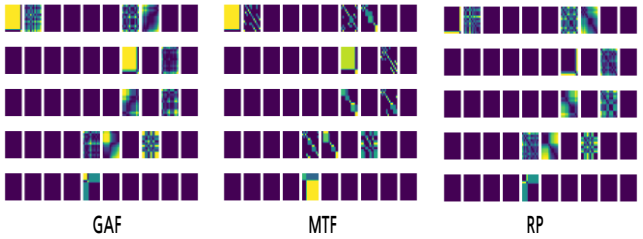

FIGURE 3. Image transformation results of the same time-series data

We performed a comparison by applying three image transformation techniques to the time-series data: Gramian angular field (GAF) [40], Markov transition field (MTF) [40], and Recurrence plot (RP) [41]. The GAF algorithm uses polar coordinates to represent temporal correlations between individual points within a time series. It retains temporal relationships when transforming time-series data into a visual image format owing to the incorporation of a polar coordinate-based matrix. The polar coordinates of the scaled time series can be calculated as follows:

$$\emptyset_i = arccos(x_i), \forall i \in \{1, 2, \dots, k\}. \tag{4}$$

The Gramian matrix is then calculated as the cosine of the sum of the angles, as indicated in (5).

$$GAF = \begin{bmatrix} \cos(\emptyset_1, \emptyset_1) & \cos(\emptyset_1, \emptyset_2) & \cdots & \cos(\emptyset_1, \emptyset_k) \\ \cos(\emptyset_2, \emptyset_1) & \cos(\emptyset_2, \emptyset_2) & \cdots & \cos(\emptyset_2, \emptyset_k) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\emptyset_k, \emptyset_1) & \cos(\emptyset_k, \emptyset_2) & \cdots & \cos(\emptyset_k, \emptyset_k) \end{bmatrix}. \tag{5}$$

MTF represents the transition probabilities of discretized time-series data. MTF is constructed by dividing a time-series dataset X into Q intervals based on its values. The time-series data value x_i is then assigned to the corresponding interval q_j ($j \in [1, Q]$). A weighted adjacency matrix W of size $Q \times Q$ can be constructed along the time axis using the first-order Markov chain method, where $w_{i,j}$ represents the frequency of transitioning from interval q_i to interval q_j. The Markov transition matrix is constructed by normalizing the sum of each column in matrix W to 1. During this process, the distribution of X and the time dependency are eliminated from W. To overcome this loss of information in W, MTF is defined by arranging each probability according to its corresponding timestep, as follows:

$$MTF = \begin{bmatrix} w_{ij}|x_1 \epsilon q_i, x_1 \epsilon q_j & \cdots & w_{ij}|x_1 \epsilon q_i, x_k \epsilon q_j \\ w_{ij}|x_2 \epsilon q_i, x_1 \epsilon q_j & \cdots & w_{ij}|x_2 \epsilon q_i, x_k \epsilon q_j \\ \vdots & \ddots & \vdots \\ w_{ij}|x_k \epsilon q_i, x_1 \epsilon q_j & \cdots & w_{ij}|x_k \epsilon q_i, x_k \epsilon q_j \end{bmatrix}. \tag{6}$$

The RP searches for the trajectory of an m-dimensional phase space by representing the recurrence of data values in a two-dimensional space. After obtaining the m-dimensional spatial trajectory of the time-series data, a distance matrix is constructed using the difference between the m-dimensional trajectory and the distance over time. The RP matrix refers to the record of the distance matrix for all combinations. The RP matrix $R_{i,j}$ is a vector composed of time pairs i and j, and can be defined as follows:

$$R_{i,j} = \theta\left(\varepsilon - \left\|\overrightarrow{x_i} - \overrightarrow{x_j}\right\|\right), \tag{7}$$

where θ denotes the Heaviside function; and ε represents the threshold value.

Fig. 3 depicts the results of the GAF, MTF, and RP transformations for the same time-series dataset. We observed that the same data were transformed into different forms of images depending on each image transformation technique. We aimed to analyze and compare the contributions of these three widely used image transformation techniques to multivariate time-series data.

E. CNN-BASED ADVERSARIAL LEARNING

The transformed 3D images (I) were fed into two-stage AEs that performed CNN-based adversarial learning. The two-stage AE comprised an encoder that condensed the input data into a latent vector and two decoders that reconstructed the data into a form similar to that of the original images. The encoder was designed to be shared between the two decoders [30]. The combinations of the encoder with the first and second decoders were referred to as AE_1 and AE_2, respectively.

In the first stage, both AE_1 and AE_2 performed traditional AE learning. The objective is to minimize the reconstruction error, which enables the model to generate an output (x') similar to the input (x). Reconstruction loss can be defined as follows:

$$\begin{aligned} \mathcal{L}_{AE_1} &= \|I - AE_1(I)\|_2; \\ \mathcal{L}_{AE_2} &= \|I - AE_2(I)\|_2. \end{aligned} \tag{8}$$

As AE-based anomaly detection is trained only on normal data, it tends to reconstruct anomalous data with a low reconstruction error when they closely resemble normal data. Therefore, detecting anomalous data using traditional AE-based models is difficult. To address this issue, we trained AE_2 to distinguish between the original input and the reconstructed output from AE_1 via adversarial learning in the second stage. In other words, AE_2 was trained to maximize the reconstruction error based on adversarial learning. The training objectives for each AE can be indicated as

$$\min_{AE_1} \max_{AE_2} \|I - AE_2(AE_1(I))\|_2. \quad (9)$$

The reconstruction loss for each AE in the second stage can be defined as

$$\begin{aligned} \mathcal{L}_{AE_1} &= +\|I - AE_2(AE_1(I))\|_2; \\ \mathcal{L}_{AE_2} &= -\|I - AE_2(AE_1(I))\|_2. \end{aligned} \quad (10)$$

We obtained the overall loss function for the model by combining (8) and (10) from the two stages, as follows:

$$\begin{aligned} \mathcal{L}_{AE_1} &= \frac{1}{n}\|I - AE_1(I)\|_2 + (1 - \frac{1}{n})\|I - AE_2(AE_1(I))\|_2; \\ \mathcal{L}_{AE_2} &= \frac{1}{n}\|I - AE_2(I)\|_2 - (1 - \frac{1}{n})\|I - AE_2(AE_1(I))\|_2, \end{aligned} \quad (11)$$

where n denotes the number of training epochs. By adding 1/n and (1−1/n) to the loss function, the model learned to focus on AE learning during the first few iterations of training and gradually shifted its focus to adversarial learning as the training progressed.

Based on the two trained AEs, the anomaly score for the test dataset ($\hat{I}$) can be defined as

$$\mathcal{S} = \alpha\left\|\hat{I} - AE_1(\hat{I})\right\|_2 + \beta\left\|\hat{I} - AE_2(AE_1(\hat{I}))\right\|_2, \quad (12)$$

where the coefficients α and β determine the sensitivity based on the proportion of reconstruction errors between AE_1 and AE_2. If the anomaly score exceeded a certain threshold, we considered the window sequence to be an anomaly. As the reconstruction error weight of AE_1 increased, the sensitivity of anomaly detection decreased. Consequently, the ratio of true positives (TPs) to false positives (FPs) decreased. By contrast, increasing β resulted in the model exhibiting high anomaly detection sensitivity, thereby increasing the number of both TPs and FPs. The sum of α and β was 1, and a tradeoff between FPs and TPs occurred according to the weights of the coefficients.

Fig. 4 illustrates the CNN-based architecture of the encoder used for processing the transformed 3D images. The decoders are structured in the reverse order of the encoder architecture. In this study, the CNN-based architecture utilized convolution layers, batch normalization, activation functions, max pooling, and dropout for image analysis. The arrangement of the layers exhibited a considerable impact on the accuracy and efficiency of the model. Batch normalization is a key technique that stabilizes the training process and improves accuracy. Batch normalization should be performed immediately after the convolution layer and before the activation function to achieve optimal results [42]. Max pooling emphasizes the features within a specific region via downsampling [43]. However, applying dropout before batch normalization can lead to an unstable analysis; therefore, dropouts should be performed after batch normalization [44]. In this study, we employed three CNNs to extract and analyze the key features of the images.

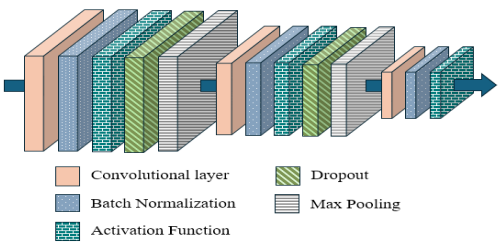

FIGURE 4. Convolutional neural network (CNN)-based architecture of the encoder for processing three-dimensional (3D) images

IV. EXPERIMENTAL ANALYSIS

A. DATASETS

We used five publicly available multivariate time-series datasets for our experiments. Table 1 lists the characteristics of the datasets.

- SWaT: The secure water treatment (SWaT) dataset is derived from an industrial water treatment plant testbed managed by Singapore's Public Utility Board, which represents a scaled-down version of a real-world facility [45]. The dataset was collected over 11 consecutive days, with seven days captured during normal operational conditions and four days recorded during simulated attack scenarios. Data were collected every second and contained 51 variables.
- WADI: The dataset for the water distribution (WADI) testbed, which is an extension of the SWaT testbed, spanned a period of 16 consecutive days, with 14 days of data collected during normal operation and 2 days recorded under attack scenarios [46]. Test data were identified based on an attack scenario. Data were collected every second and included 123 variables (excluding *null* variables).
- SMAP: The soil moisture active passive (SMAP) satellite dataset is a publicly available real-world dataset labeled by experts from the National Aeronautics and Space Administration (NASA) [17]. The dataset comprises 55 entities, each with 25 variables.
- MSL: The Mars science laboratory (MSL) dataset is also a real-world dataset collected by NASA [17]. This dataset comprises 27 entities, each with 55 variables.
- SMD: The server machine dataset (SMD) is a large-scale, multivariate time-series dataset collected from a real-world internet company [33]. This dataset comprises 28 entities, each with 38 variables.

TABLE 1. Characteristics of the datasets

Datasets	#Variables (#Entities)	#Train	#Test	Anomalies
SWaT	51 (1)	496 800	449 919	11.98%
WADI	123 (1)	1 048 571	172 801	5.99%
SMAP	25 (55)	135 183	427 617	12.79%
MSL	55 (27)	58 317	73 729	10.53%
SMD	38 (28)	708 405	708 420	4.16%

B. BASELINE MODELS

The effectiveness of the proposed model was evaluated by experimentally comparing its performance with the following state-of-the-art models in terms of multivariate time-series anomaly detection.

- An AE is a neural network trained to reconstruct its input. Here, anomaly detection is achieved by identifying data points with reconstruction errors that exceed a predefined threshold. [47].
- The IF model is an ensemble-based technique that uses multiple decision trees. It continuously splits the trees and identifies anomalies based on the isolation level of each data instance [24].
- LSTM-VAE is a reconstruction-based model that replaces the feedforward network of the existing variable AE with LSTM [29].
- DAGMM is a deep autoencoding Gaussian model that uses an AE for dimensionality reduction and a GMM for density estimation of complex input data [21].
- OmniAnomaly combines gated recurrent units with VAE and utilizes a stochastic RNN to focus on learning robust representations by incorporating planar normalizing flow techniques and probabilistic variable connections [33].
- USAD is an unsupervised method with two AEs that utilizes adversarial training to maximize the reconstruction error between normal and abnormal data [30].
- MTAD-GAT is a reconstruction-based model that learns the representation of each univariate time series by reconstructing the original input while capturing both temporal and spatial dependencies via two parallel GAT layers [32].
- CAE-M is a jointly optimized model that combines a convolutional AE for reconstruction with an attention-based bidirectional LSTM and an AR model for prediction [35].
- MAD-GAN is an LSTM-based GAN model that can capture temporal dependencies in time-series data. It employs an anomaly score derived from the combined losses of the generator and discriminator to detect anomalies [34].
- MSCRED is a model that extracts diverse features of system states by generating multi-scale signature matrices and processing them using a convolutional encoder and decoder [50].
- GDN is a model that generates a graph representing the relationships between sensors and extracts features from that graph using a graph neural network (GNN) [19].

C. EXPERIMENTAL SETTINGS

We implemented the proposed model and the baseline models in Python 3.8, PyTorch 2.0.1, and CUDA 12.2. The experimental setup utilized a server equipped with an Intel(R) Core (TM) i7-6700K CPU @ 4.00 GHz and an NVIDIA GeForce RTX 2080Ti graphics card. We used the Adam optimizer with a learning rate of 0.0001 and set the batch size to 32. The model was trained for 50 epochs and implemented early stopping with a patience value of 10. The input data were selected as a sequential subset using the sliding window algorithm with a window size of 12 for SWAT and WADI, and 6 for SMAP, MSL, and SMD datasets. Each kernel size of the three convolution layers was 3, the stride was 1, and the dropout rate for each CNN was 0.2. Table 2 presents the detailed architecture of the encoder and decoder.

D. EVALUATION METRICS

We used the precision (P), recall (R), and F1 score (F1) to evaluate the anomaly detection performance of the T2IAE.

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2PR}{P + R}, \qquad (13)$$

where TP denotes the correctly identified anomaly, FP indicates the incorrectly identified anomaly as normal, true negative (TN) represents the correctly identified normal data, and false negative (FN) denotes the incorrectly identified abnormal data as normal.

We evaluated the threshold that exhibited the highest performance for each model and considered the detection results as anomalies when they exceeded the corresponding threshold. As the inputs of the model were the images transformed from the window sequences, the

TABLE 2. Detailed architecture of the encoder and decoder (*v*: variables; *w*: window size; *k*: kernel size; *s*: stride; *p*: padding; c_{out}: output channels; *d*: dropout rate)

Module	Layer name	Layer
-	-	Input size= $v \times w \times w$
Encoder	Layer1	Conv2d ($k = 3$, $s = 1$, $p = 1$, $c_{out} = [v/2]$) BatchNorm2d ReLU Dropout ($d = 0.2$) Maxpool2d
	Layer2	Conv2d ($k = 3$, $s = 1$, $p = 1$, $c_{out} = [v/2^2]$) BatchNorm2d ReLU Dropout ($d = 0.2$) Maxpool2d
	Layer3	Conv2d ($k = 3$, $s = 1$, $p = 1$, $c_{out} = [v/2^3]$) BatchNorm2d ReLU
Decoder	Layer1	ConvTranspose2d ($k = 6$, $s = 1$, $p = 1$, $c_{out} = [v/2^2]$) BatchNorm2d ReLU Dropout ($d = 0.2$)
	Layer2	ConvTranspose2d ($k = 6$, $s = 1$, $p = 1$, $c_{out} = [v/2]$) BatchNorm2d ReLU Dropout ($d = 0.2$)
	Layer3	ConvTranspose2d ($k = 6$, $s = 1$, $p = 1$, $c_{out} = v$) Sigmoid

TABLE 3. Anomaly detection results including Precision (P), Recall (R), and F1 score. The best performances are highlighted in bold, and the second-best performances are underlined.

Methods	SWaT			WADI			SMAP		
	P	R	F1	P	R	F1	P	R	F1
AE	0.9801	0.6386	0.7733	0.9940	0.1573	0.2716	0.6790	0.9553	0.7894
IF	0.9542	0.5837	0.7242	0.3015	0.1688	0.2164	0.5014	0.5114	0.4685
LSTM-AVE	0.9897	0.6379	0.7758	0.9932	0.1311	0.2316	0.6607	0.8775	0.7682
DAGMM	0.4695	0.6659	0.5507	0.0750	0.9229	0.1387	0.5611	0.9060	0.7373
OmniAnomaly	0.9938	0.6463	0.7832	0.9947	0.1327	0.2342	0.6802	0.9381	0.8074
USAD	0.9965	0.6724	0.8030	0.5028	0.3016	0.3771	0.7317	0.9964	0.8159
MTAD-GAT	0.9704	0.6913	0.8074	0.2818	0.8012	0.4169	0.7777	0.9883	0.8505
CAE-M	0.9056	0.8093	0.8548	0.2782	0.7918	0.4117	0.7313	0.9832	0.8387
MAD-GAN	0.8843	0.7832	0.8307	0.2233	0.9124	0.3588	0.7736	0.9815	0.8491
MSCRED	0.9969	0.7765	0.8730	0.4503	0.3009	0.3607	0.7869	0.9798	0.8728
GDN	0.9052	0.8332	0.8677	0.4136	0.3009	0.3484	0.7729	0.9921	0.8689
T2IAE-GAF	0.9555	0.7611	0.8473	0.6391	0.6244	**0.6316**	0.7952	0.9916	0.8826
T2IAE-RP	0.9937	0.6784	0.8385	0.9532	0.2936	0.4489	0.7960	0.9916	0.8831
T2IAE-MTF	0.8913	0.8832	**0.8872**	0.8098	0.3650	0.5032	0.7981	0.9916	**0.8844**

Methods	MSL			SMD			Average		
	P	R	F1	P	R	F1	P	R	F1
AE	0.8319	0.9378	0.8817	0.8852	0.5833	0.7032	0.8728	0.6545	0.6838
IF	0.5481	0.6542	0.5965	0.5938	0.8532	0.5866	0.5660	0.5543	0.5184
LSTM-AVE	0.8086	0.9352	0.8673	0.8099	0.6483	0.7201	0.8589	0.6460	0.6726
DAGMM	0.7169	0.9515	0.8177	0.8557	0.7081	0.7749	0.5477	0.8309	0.6039
OmniAnomaly	0.9132	0.8791	0.8958	0.8507	0.8046	**0.8270**	0.8922	0.6802	0.7095
USAD	0.8794	0.9886	0.9308	0.8263	0.7190	0.7690	0.7792	0.7356	0.7392
MTAD-GAT	0.8878	0.9858	0.9242	0.8230	0.6938	0.7529	0.7481	0.8321	0.7504
CAE-M	0.8107	0.9858	0.8897	0.7990	0.7034	0.7481	0.7050	0.8547	0.7486
MAD-GAN	0.9026	0.9624	0.9268	0.8359	0.7144	0.7704	0.7239	0.8708	0.7472
MSCRED	0.8569	0.9858	0.9169	0.7896	0.6801	0.7308	0.7761	0.7446	0.7508
GDN	0.8752	0.9626	0.9168	0.8222	0.7199	0.7677	0.7578	0.7617	0.7539
T2IAE-GAF	0.9061	0.9892	0.9458	0.8249	0.7501	0.7857	0.7963	0.8233	**0.8186**
T2IAE-RP	0.9067	0.9892	**0.9462**	0.8267	0.7568	0.7903	0.8953	0.7419	0.7814
T2IAE-MTF	0.9043	0.9892	0.9449	0.8174	0.7394	0.7764	0.8405	0.7937	0.7992

evaluation metric was applied to each window sequence. If one or more anomalies existed in a window sequence, the window was considered anomalous.

V. RESULTS AND DISCUSSION

A. PRIMARY RESULTS

We evaluated the performance of the proposed T2IAE for anomaly detection using multivariate time-series data by comparing it with eleven other models. The proposed model used three approaches to convert time-series data into images, namely, GAF, MTF, and RP; the corresponding models were referred to as T2IAE-GAF, T2IAE-MTF, and T2IAE-RP, respectively. Table 3 presents the anomaly detection performance of T2IAE and the other models compared with respect to the SWaT, WADI, SMAP, MSL, and SMD datasets. The best F1 score is indicated in bold and the second-best F1 score is underlined for each dataset. We observed that the proposed model exhibited superior performance compared with most baseline models, thereby validating the effectiveness of the approach. T2IAE's performance is only slightly behind OmniAnomaly in SMD dataset.

The average F1 scores of the T2IAE models for the five datasets were 0.8186, 0.7814, and 0.7992 for T2IAE-GAF, T2IAE-RP, and T2IAE-MTF, respectively. These were the best performance scores compared with the average F1 scores of the other models. Our models achieved the results by capturing not only temporal dependencies but also interactions and causal relationships between variables via images that

preserved temporal information. Each variable of the multivariate time-series data was transformed into an image that preserved the temporal order, enabling the model to utilize temporal information. Furthermore, the collection of these images enabled the model to exploit spatial information. The model can identify the causal relationships between variables by integrating the spatial–temporal information. IF and DAGMM exhibited the weakest performances owing to their failure to incorporate the temporal information of the variables into their anomaly detection mechanisms. USAD showed limited performance despite employing adversarial learning due to its ignorance for spatial information. Recently developed models, such as MTAD-GAT (0.7504), CAE-M (0.7486), MAD-GAN (0.7472), MSCRED (0.7508), and GDN (0.7539), achieved better performance than other baseline models by capturing dependencies within each time series. However, their performance is approximately 10% lower than that of our models.

In the proposed models, the model with GAF showed the highest performance. The performance discrepancies among image transformation techniques can be attributed to their distinct methods of preserving temporal dependencies in time-series data. GAF effectively preserves temporal order by measuring the changes in the time-series data within a polar coordinate system. By contrast, MTF calculates the transition probabilities between discrete time-series data points, which could lead to some loss of temporal sequence information. RP measures the time required to return to a previously visited state, limiting its ability to capture long-term temporal dependencies.

Next, we compared the computational performance of T2IAE with these baseline models that showed comparable performances, we measured the time taken by each dataset per epoch. For the SMAP, MSL, and SMD datasets, which contained multiple entities, we used a single entity. Table 4 presents the results of the analysis. The proposed models require additional time for image transformation of the time-series data. However, these models exhibited significantly shorter training times than the other models due to their simple neural network architectures. In particular, T2IAE-RP reduced the training time by up to 50 times compared with MTAD-GAT with respect to the MSL dataset. MAD-GAN and

TABLE 4. Training time (in seconds) per epoch with respect to each dataset

Methods	SWaT	WADI	SMAP	MSL	SMD
MTAD-GAT	126.76	25.26	19.63	120.74	355.06
CAE-M	49.56	11.20	7.75	15.67	100.49
MAD-GAN	429.03	28.61	12.91	9.74	104.11
MSCRED	158.78	21.81	11.22	24.77	112.73
GDN	155.61	12.14	11.74	10.09	89.38
T2IAE-GAF	37.03	12.21	4.72	4.73	23.72
T2IAE-RP	20.08	9.49	2.10	2.41	13.13
T2IAE-MTF	27.09	11.29	3.61	3.96	17.55

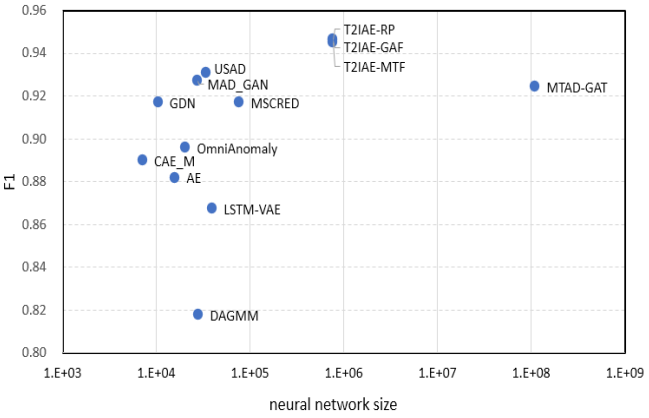

FIGURE 5. Comparison of the relationship between neural network size (# of trainable parameters) and performance (F1 score) of deep learning models in the MSL dataset

MSCRED employ more complex LSTM networks compared to basic neural network architectures, while GDN and MTAD-GAT represent time-series data as intricate graph structures. Consequently, these models demand a higher computational load. Among image transformation techniques, the results confirm that RP has lower computational complexity compared to MTF and GAF.

In addition, we investigated the relationship between neural network size and model performance. Generally, larger networks can potentially achieve higher performance according to scaling laws, as they can learn more data and express complex relationships. However, excessively large neural networks may suffer from overfitting on a limited amount of training data [51]. Fig. 5 illustrates the relationship between the number of trainable parameters and F1 scores for deep learning models excluding IF on the MSL dataset. This simultaneously demonstrated the performance improvement with increasing neural network size and the performance degradation due to overfitting in MTAD-GAT with excessively large neural network size. Our models achieved superior performance by effectively capturing time series characteristics through an appropriate increase in neural network size facilitated by the transformation of time-series data into images.

B. INTERPRETABLE ANOMALY DETECTION

Although transforming time-series data into images offers performance advantages, its primary strength lies in its superior intuitive interpretability compared with raw time-series data. This interpretability facilitates an in-depth analysis of the anomaly detection results.

Fig. 6 depicts the reconstruction errors detected for each variable to determine the anomaly in the SWaT dataset using the T2IAE-MTF model. In the case of normal data, the difference between the original and restored data is close to zero, whereas the difference is close to one in the case of anomalous data. We display 50 of the 51 variables in tabular form, where values closer to zero are indicated in

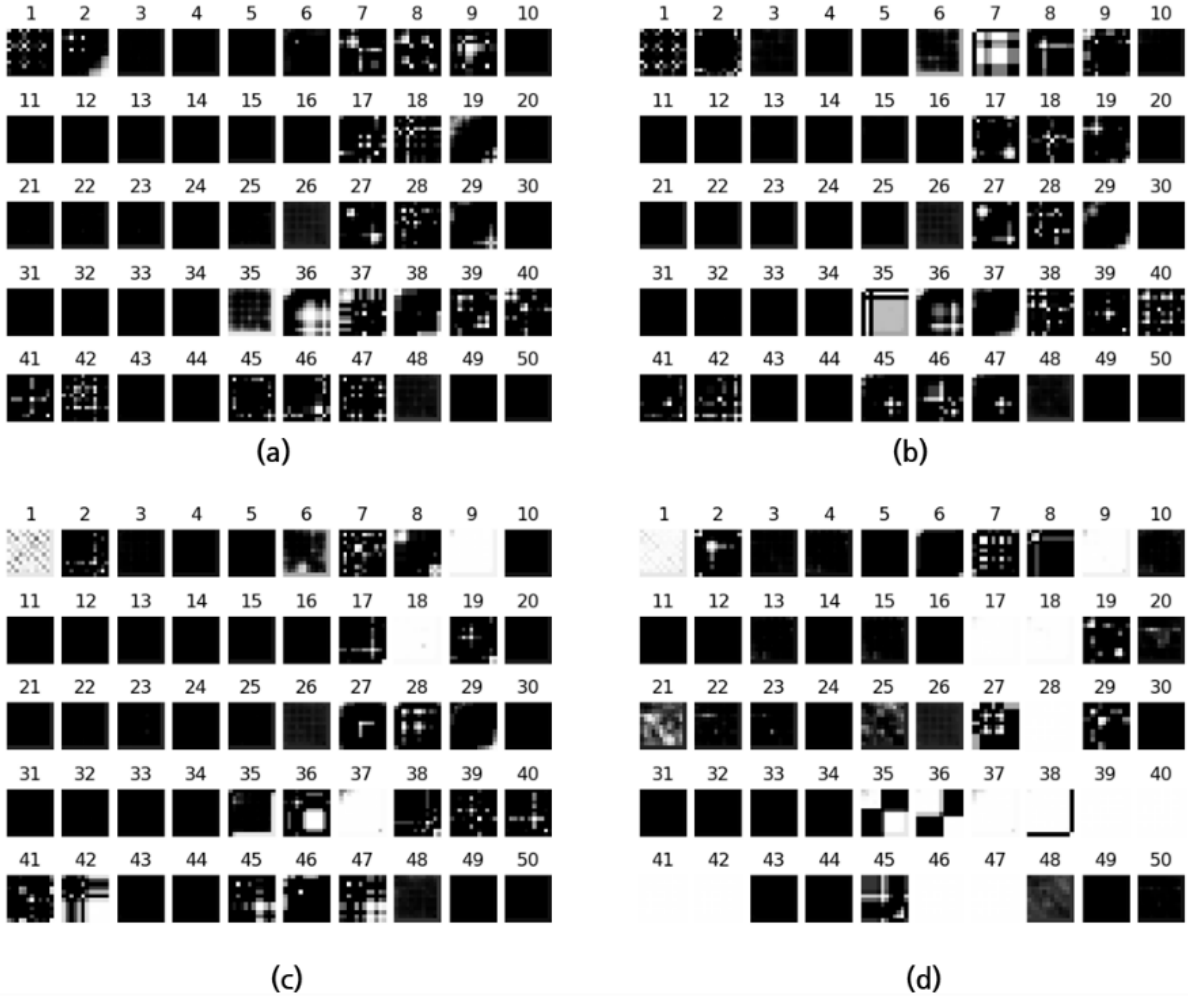

FIGURE 6. Reconstruction errors between original and restored data of the secure water treatment (SWaT) dataset with respect to variables. (a) and (b) represent the specific time points for the reconstruction errors of the normal data, (c) and (d) represent the specific time points for the reconstruction errors of the abnormal data.

a darker shade and those closer to one are denoted in a lighter shade. The name and number of each equipment (variable) and the 6-stage SWaT testbed processes are described in [48]. We used the same number for each piece of equipment as indicated in the figure.

Figs. 6(a) and 6(b) represent the specific time points for the reconstruction errors of the normal data, and Figs. 6(c) and 6(d) represent the specific time points for the reconstruction errors of the abnormal data[1]. Although a few white dots exist in Figs. 6(a) and 6(b), the anomaly score is sufficiently small and insignificant to avoid exceeding a certain threshold. By contrast, Figs. 6(c) and 6(d) contain several bright images, indicating the detection of anomalous data. We consider the details presented in [49] for the subsequent analysis here. According to [49], this time point is under a cyber-attack caused by Scenario 28. The attack involves closing the pump (P302) in the third stage to block the inflow to the first tank (T401) in the fourth stage. Fig. 6(c) depicts the state at the beginning of the anomaly, where the restoration errors of the 9th (FIT201) and 18th (FIT301) variables are remarkably large. Both variables were measured using flow meters, likely because the flow rates in stages 2 and 3 changed rapidly when the pump in stage 3 was closed. Fig. 6(d) illustrates the situation approximately 8 h after the occurrence of the scenario depicted in Fig. 6(c). Several additional variables exhibit significant restoration errors. In particular, the restoration errors of the sensors after stage 4 (Nos. 28, 37–42, 46, and 47) increase significantly. These sensors are flow meters or pressure meters that enable the identification of equipment malfunctions over time. Therefore, based on the T2IAE results, the correlation, causality, and other relationships between the variables within the anomalous section can be intuitively interpreted.

[1] The time points are as follows: (a) 2015-12-28 11:57:20; (b) 2015-12-28 13:44:00; (c) 2015-12-31 02:10:40; (d) 2015-12-31 10:10:40

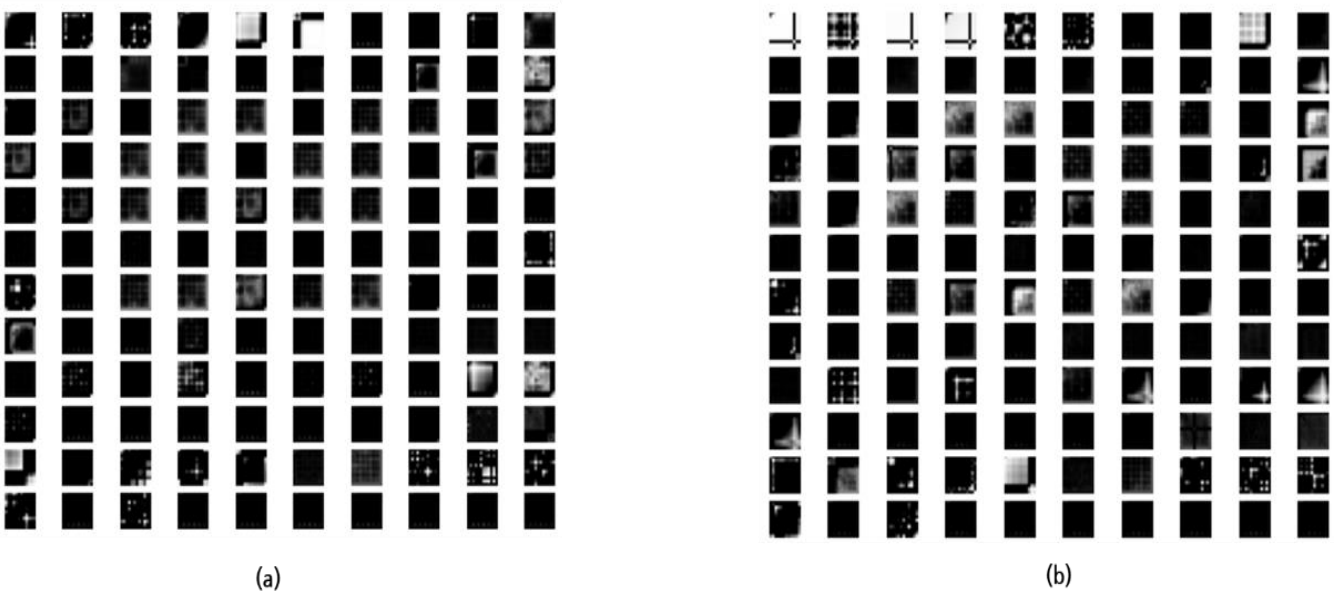

FIGURE 7. Reconstruction errors between original and restored data of the abnormal data in the water distribution (WADI) dataset with respect to variables

Figs. 7(a) and 7(b) depict the specific time points for the reconstruction errors of the abnormal data[2] in the WADI dataset. Similar to Fig. 6, the variables with anomalies are displayed brightly because of the significant difference between the original and reconstructed data. Fig. 7(a) indicates that when the attack turns off the 6th variable in the upper-left corner (1_FIT_001), which is a flow indication transmitter, the reconstruction error of that variable increases rapidly. This attack causes the chemical dosing pump to operate. In Fig. 7(b), the first, third, and fourth variables in the upper-left corner are highlighted. These variables are derived from sensors that analyze water quality (1_AIT_001, 1_AIT_003, and 1_AIT_004). Based on these findings, we inferred that an issue existed with the water quality. This implies that visualizing and reconstructing multivariate time-series data offers a detailed explanation and enables an intuitive understanding of the detected results.

C. HYPERPARAMETER SENSITIVITY ANALYSIS

We then investigated the effects of varying the T2IAE parameters on the model performance. Experiments were conducted using the T2IAE-MTF model with the SWaT dataset.

As indicated in (12), more weight is attached to the reconstruction of AE_1 for a larger sensitivity threshold α, and more weight is attached to AE_2 for a larger β value. Increasing α and decreasing β can reduce the number of FPs while minimizing the reduction in the number of TPs [30]. To compare the impact of sensitivity threshold variations on the proposed model, we performed anomaly detection using a single-trained T2IAE-MTF model while adjusting α and β in increments of 0.2 without re-training with the SWaT dataset. As indicated in Table 5, increasing the value of α leads to an increase in recall (R), and the number of FNs decreases more than the number of TPs. These findings are consistent with those of USAD [30]. As β increases, precision (P) also increases, and the number of TPs increases more than the number of FPs. This enables data handlers to prioritize either FN reduction or TP increase by adjusting the sensitivity according to their preferences.

Subsequently, we examined the performance of the proposed model with changes in the window size. Determining the optimal window size is crucial because it significantly affects the model performance. Fig. 8(a) illustrates the results of seven different window sizes, $W \in [6, 12, 24, 36, 48, 60, 72]$, indicating that the best F1 score is obtained for a window size of 12. The F1 score reduced continuously when the window size exceeded 12, indicating

TABLE 5. Results of anomaly detection obtained with different sensitivity thresholds for the SWaT dataset.

P	R	F1	α	β
0.9797	0.7653	0.8593	0.0	1.0
0.9786	0.7653	0.8589	0.2	0.8
0.9508	0.8004	0.8691	0.4	0.6
0.9530	0.8004	0.8702	0.6	0.4
0.9342	0.8226	0.8748	0.8	0.2
0.8913	0.8832	0.8872	1.0	0.0

[2] The time points are as follows: (a) 2017-10-17 10:26:00; (b) 2017-10-17 10:34:00; where the attack occurred for 9 m and 50 s from 2017-10-17 10:24:10

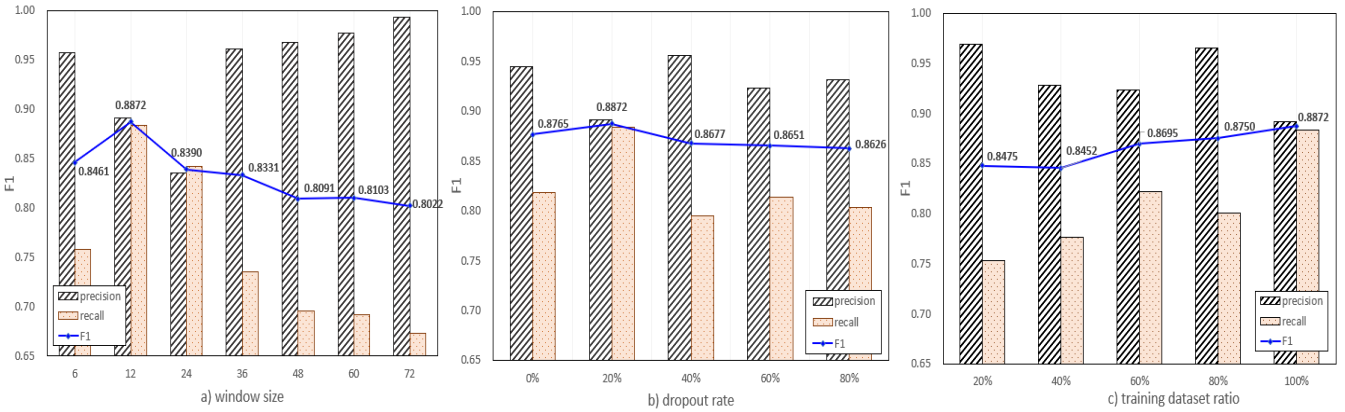

FIGURE 8. **F1 scores with respect to the a) window size, b) dropout rate, and c) training dataset ratio in the secure water treatment (SWaT) dataset**

a decline in model performance. Larger images were created using a larger window size to train the T2IAE model. Images with large window sizes tended to capture the correlation between adjacent time points inadequately because they had to capture the relationships between distant points within the image. However, the proposed model using images with a window size of 12 adequately captured important correlations between adjacent time points.

Dropout can prevent overfitting and improve performance by reducing the co-adaptation among pixels in the image data [52]. Therefore, we evaluated the performance of the proposed model using different dropout rates. Fig. 8(b) presents the results for five dropout rates, $D \in [0\%, 20\%, 40\%, 60\%, 80\%]$. The F1 score was the highest (0.8872) at a dropout rate of 20%. This indicates that the dropout layer improved the model performance. However, excessively high dropout rates can deteriorate the model performance to an even worse state than when no dropout layer is used. Therefore, determining the dropout rate that yields the best results is crucial because simply implementing a dropout layer does not ensure an improved performance.

Finally, we investigated the impact of the training dataset size on the detection performance. Fig. 8(c) illustrates the results for five different training dataset ratios, $T \in [20\%, 40\%, 60\%, 80\%, 100\%]$. Here, 100% of the training dataset refers to the entire training data existing in Table 1, whereas 80% represents the dataset that excludes the final 20% from the entire training dataset. We observed that the performance of the proposed model improved steadily as the amount of training data increased. In other words, the variance and bias decreased with the increase in the amount of data, thereby improving the model performance.

VI. CONCLUSION

In this study, we propose a novel time-series to image-transformed anomaly detection method that adopts three image transformation techniques and CNN-based adversarial learning. The proposed model facilitates the learning of correlations between adjacent time-series data variables by transforming multivariate time-series data into images. Additionally, adversarial learning performed using two AEs enables the effective learning of temporal characteristics in multivariate time-series data. We empirically analyzed five publicly available real-world datasets to evaluate the anomaly detection performance of the proposed model and determined that it outperformed other state-of-the-art methods. Furthermore, the proposed model enables humans to intuitively interpret detected results of multivariate time-series data, facilitating appropriate explanations of the detection results and enhancing the model's usability.

Image transformation for learning is a critical factor affecting both performance and interpretability. Therefore, in the future, we will seek ways to improve the image transformation techniques to further enhance the detection performance of T2IAE. We also aim to enhance our model with an attention mechanism for input images, thereby creating an optimized framework for multivariate time series anomaly detection.

ACKNOWLEDGMENT

Jiyoung Kang and Minseok Kim are co-first authors.

REFERENCES

[1] M. Ghobakhloo, “Industry 4.0, digitization, and opportunities for sustainability,” J. Clean. Prod., vol. 252, 119869, Apr. 2020. DOI: 10.1016/j.jclepro.2019.119869

[2] G. Y. Lee, M. Kim, Y. J. Quan, M.S. Kim, T. J. Kim, H. S. Yoon, S. Min, D. H. Kim, J. W. Mun, J. W. Oh, I. G. Choi, C. S. Kim, W. S. Chu, J. Yang, B. Bhandari, C. M. Lee, J.B. Ihn, and S.H. Ahn, “Machine health management in smart factory: A review,” J. Mech. Sci. Technol., vol. 32, pp. 987–1009, Mar. 2018. DOI: 10.1007/s12206-018-0201-1

[3] Y. Zhang, Y. Chen, and Z. Pan, “A deep temporal model for mental fatigue detection,” in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, IEEE, pp. 1879–1884, Oct. 2018. 10.1109/SMC.2018.00325

[4] D. Yankov, E. Keogh, U. Rebbapragada, “Disk aware discord discovery: Finding unusual time series in terabyte sized datasets,” Knowl. Inf. Syst., vol. 17, pp. 241–262, Nov. 2008. DOI: 10.1007/s10115-008-0131-9

[5] N. R. Pokhrel, H. Rodrigo, and C. P. Tsokos, “Cybersecurity: Time series predictive modeling of vulnerabilities of desktop operating system using linear and non-linear approach,” J. Inf. Secur., vol. 8, p. 362, Oct. 2017. DOI: 10.4236/jis.2017.84023

[6] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl, "Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition," in Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), Seattle, WA, USA, pp. 143–152, Oct. 2012. DOI: 10.1109/VAST.2012.6400557

[7] I. Zingman, B. Stierstorfer, C. Lempp, and F. Heinemann, "Learning image representations for anomaly detection: Application to discovery of histological alterations in drug development," Medical Image Analysis, vol. 92, Feb. 2024

[8] S. Bajpai, K. Sharma, and B. K. Chaurasia, "Intrusion detection framework in IoT networks," SN Computer Science, vol. 4, 350, 2023.

[9] S. Bajpai, K. Sharma, and B. K. Chaurasia, "A Hybrid Meta-Heuristics Algorithm: XGBoost-Based Approach for IDS in IoT," SN Comput. Sci., vol. 5, 537, 2024.

[10] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental local outlier detection for data streams," in Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, Honolulu, HI, USA, IEEE, pp. 504-515, Mar. 2007. DOI: 10.1109/CIDM.2007.368917

[11] V. Hautamaki, I. Karkkainen, and P. Franti, "Outlier detection using k-nearest neighbour graph," in Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, vol. 3, pp. 430–433, Aug. 2004. DOI: 10.1109/ICPR.2004.1334558

[12] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the 2nd International Conference on Knowledge Discovery & Data Mining, pp. 226–231, 1996.

[13] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104, May 2000. DOI: 10.1145/342009.335388

[14] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in GI/ITG Workshop MMBnet, vol. 7, no. 9, Sep. 2007.

[15] C. C. Aggarwal, "Outlier analysis," In Data mining, pp. 237–263. Springer, 2015.

[16] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," arXiv preprint arXiv:1704.02971, Apr. 2017.

[17] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in Proceedings of the International Conference on Knowledge Discovery & Data Mining, pp. 387–395, Jul. 2018. DOI: 10.1145/3219819.3219845

[18] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, H. Qiao, "Unsupervised anomaly detection via variational autoencoder for seasonal KPIs in web applications," in Proceedings of the 2018 World Wide Web Conference, pp. 187–196, Apr. 2018. DOI: 10.1145/3178876.3185996

[19] A. Deng, and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4027–4035, May 2021. DOI: 10.1609/aaai.v35i5.16523

[20] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," in 2020 IEEE International Conference on Big Data, Atlanta, GA, USA, pp. 30–43, Dec. 2020. DOI: 10.1109/BigData50022.2020.9378139

[21] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in International Conference on Learning Representations (ICLR), Feb. 2018.

[22] L. Shen, Z. Li and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," Adv. Neural Inf. Process. Syst., NeurIPS, vol. 33, pp. 13016–13026, 2020.

[23] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," arXiv preprint arXiv:2110.02642, Oct. 2021. DOI: 10.48550/arXiv.2110.026642

[24] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in Proceedings of the 2008 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy, pp. 413–422, Dec. 2008. DOI: 10.1109/ICDM.2008.17

[25] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Comput., vol. 13, no. 7, pp. 1443–1471, Jul. 2001. DOI: 10.1162/089976601750264965

[26] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," Int. J. Distrib. Sensor Netw., vol. 12, no. 1, p. 9653230, Jan. 2016. DOI: 10.1155/2016/9653230

[27] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 777–785, Jun. 2017. DOI: 10.1137/1.9781611974973.87

[28] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv:1607.00148, Jul. 2016. DOI: 10.48550/arXiv.1607.00148

[29] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," IEEE Robot. Autom. Lett., vol. 3, no. 3, pp. 1544–1551, Feb. 2018. DOI: 10.1109/LRA.2018.2801475

[30] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395–3404, Aug. 2020. DOI: 10.1145/3394486.3403392

[31] Z. Ji, Y. Wang, K. Yan, X. Xie, Y. Xiang, and J. Huang, "A space-embedding strategy for anomaly detection in multivariate time series," Expert Syst. Appl., vol. 206, 117892, 2022.

[32] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in Proceedings of the IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, pp. 841–850, Nov. 2020. DOI: 10.1109/ICDM50108.2020.00093

[33] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828–2837, Jul. 2019. DOI: 10.1145/3292500.3330672

[34] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in International Conference on Artificial Neural Networks, Springer, pp. 703–716, Sep. 2019. DOI: 10.1007/978-3-030-30490-4_56

[35] Y. Zhang, Y. Chen, J. Wang, Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," IEEE Trans. Knowl. Data Eng., vol. 35, no. 2, pp. 2118–2132, Aug. 2021. 10.1109/TKDE.2021.3102110

[36] M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, "A novel computational approach for discord search with local recurrence rates in multivariate time series," Inf. Sci., vol. 477, pp. 220-233, 2019.

[37] L. Al Shalabi, and Z. Shaaban, "Normalization as a preprocessing engine for data mining and the approach of preference matrix," in 2006 International Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX '06), Szklarska Poreba, Poland, pp. 207–214 May 2006. DOI: 10.1109/DEPCOS-RELCOMEX.2006.38

[38] Y. B. Yahmed, A. A. Bakar, A. RazakHamdan, A. Ahmed, and S. M. Syed Abdullah, "Adaptive sliding window algorithm for weather data segmentation," J. Theor. Appl. Inf. Technol., vol. 80, no. 2, pp. 322–333, Oct. 2015.

[39] G. R. Garcia, G. Michau, M. Ducoffe, J. S. Gupta, and O. Fink, "Time series to images: Monitoring the condition of industrial assets with deep learning image processing algorithms," arXiv preprint, arXiv:2005.07031, May 2020.

[40] Z. Wang, and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," In Proceedings of the AAAI Conference on Artificial Intelligence, Apr. 2015.

[41] J. P. Eckmann, S. O. Kamphorst and D Ruelle, “Recurrence Plots of Dynamical Systems,” World Sci. Ser. Nonlinear Sci. Ser. A, vol. 16, pp. 441–446, Sep. 1995.
[42] S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in International Conference on Machine Learning, pp. 448–456, Jun. 2015.
[43] Y. L. Boureau, J. Ponce, and Y. LeCun, “A theoretical analysis of feature pooling in visual recognition,” In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, pp. 111–118, Jun. 2010
[44] X. Li, S. Chen, X. Hu, and J. Yang, “Understanding the disharmony between dropout and batch normalization by variance shift.” In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2682–2690, 2019.
[45] A. P. Mathur, and N. O. Tippenhauer, “SWaT: A water treatment testbed for research and training on ICS security,” In 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater), Vienna, Austria, pp. 31–36, Apr. 2016. DOI: 10.1109/CySWater.2016.7469060
[46] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, “WADI: A water distribution testbed for research in the design of secure cyber physical systems,” In Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER ’17, 25–28. New York, NY, USA: Association for Computing Machinery. ISBN 9781450349758, pp. 25–28, Apr. 2017. DOI: 10.1145/3055366.3055375
[47] S. Agarwal, “Data mining: Data mining concepts and techniques,” in 2013 International Conference on Machine Intelligence and Research Advancement, Katra, India, pp. 203–207, Dec. 2013. DOI: 10.1109/ICMIRA.2013.45
[48] J. Goh, S. Adepu, K. N. Junejo, A. Mathur, “A dataset to support research in the design of secure water treatment systems,” in 11th International Conference on Critical Information Infrastructures Security, Paris, France, Springer, pp. 88–99, Oct. 2016. DOI: 10.1007/978-3-319-71368-7_8
[49] A. L. Gómez, L. F. Maimó, A. H. Celdrán, and F. J. Clemente, “SUSAN: A deep learning based anomaly detection framework for sustainable industry,” Sustain. Comput. Inform. Syst., vol. 37, 100842, Jan. 2023. DOI: 10.1016/j.suscom.2022.100842
[50] 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. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 1409–1416.
[51] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” 2020, arXiv preprint arXiv:2001.08361.
[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929-1958, Jan. 2014.

JIYOUNG KANG received the M.S degree in computer engineering from Yonsei University, Seoul, South Korea, in 2023. She is currently a staff engineer at Samsung Electronics, where she researched deep learning, data analysis and knowledge graph.

MINSEOK KIM received the B.S. degree in computer science and engineering from Pusan National University, Busan, South Korea, in 2002 and the M.S. degree in computer science and engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2004. He is currently pursuing the Ph.D. degree in computer science with Yonsei University, Seoul. He is currently a software architect at the Korea Financial Telecommunications & Clearings Institute (KFTC). His research interests include deep learning, anomaly detection, and neural networks.

JINUK PARK received the B.S. degree in statistics from University of Seoul in 2016 and the Ph.D. degree in computer science from Yonsei university, Seoul, South Korea, in 2022. His current research interests include time-series modeling, neural networks, quantization for neural models.

SANGHYUN PARK (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Seoul National University, in 1989 and 1991, respectively, and the Ph.D. degree from the Department of Computer Science, University of California at Los Angeles (UCLA), in 2001. He is currently a Professor with the Department of Computer Science, Yonsei University, Seoul, South Korea. His current research interests include databases, data mining, bioinformatics, and flash memory.