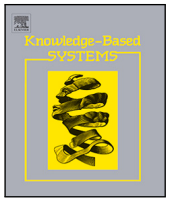
Knowledge-Based Systems 290 (2024) 111507

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism

Hyeongwon Kang, Pilsung Kang *

Department of Industrial & Management Engineering, Korea University, 126-16 Anam-dong 5-ga, Seongbuk-gu, Seoul, Republic of Korea

ARTICLE INFO

Keywords:
Multivariate time-series
Anomaly detection
Transformer
XAI
Attention mechanism

ABSTRACT

The primary objective of multivariate time-series anomaly detection is to spot deviations from regular patterns in time-series data compiled concurrently from various sensors and systems. This method finds application across diverse industries, aiding in system maintenance tasks. Capturing temporal dependencies and correlations between variables simultaneously is challenging due to the interconnectedness and mutual influence among variables in multivariate time-series. In this paper, we propose a unique method, the Variable Temporal Transformer (VTT), which utilizes the self-attention mechanism of transformers to effectively understand the temporal dependencies and relationships among variables. This proposed model performs anomaly detection by employing temporal self-attention to model temporal dependencies and variable self-attention to model variable correlations. We use a recently introduced evaluation metric after identifying potential overestimations in the performance of traditional time series anomaly detection methods using the point adjustment protocol evaluation metric. We confirm that our proposed method demonstrates cutting-edge performance through this new metric. Furthermore, we bring forth an anomaly interpretation module to shed light on anomalous data, which we verify using both synthetic and real-world industrial data.

1. Introduction

The advancements in industrial automation, sensor technology, Internet of Things (IoT) systems, and digital transformation have led to the generation, storage, and real-time processing of vast amounts of time-series data across various domains, including manufacturing, IT, and healthcare [1,2]. In this big data era, fast, accurate, and efficient anomaly detection is crucial for maintaining the integrity and security of diverse systems, preventing potential accidents and economic losses [1–3]. For instance, General Electric (GE) offers a solution for detecting anomalies in energy-related equipment by collecting and analyzing sensor data through its industrial internet platform, Predix. This platform facilitates efficient maintenance and operation of customers' systems while minimizing economic losses resulting from unexpected outages or accidents [4]. Similarly, Bosch has implemented a system in its manufacturing processes that employs anomaly detection algorithms to monitor time-series sensor data in real-time, enabling early anomaly detection to enhance efficiency and reliability of production processes while reducing unnecessary maintenance costs and production losses [5]. Owing to these benefits, numerous public and private companies operating continuous systems have expressed significant interest in adopting systems that effectively detect anomalies in time series data [6–8].

Multivariate time-series data consists of several variables tracked over regular intervals, recording a value for each variable at a specific time. This data type is defined by interrelations and interactions among variables [9]. The central goal of time-series anomaly detection is to pinpoint data points that substantially diverge from the common trends in the entire time-series dataset [10], with the objective of classifying a given data point as anomalous based on historical data. Identifying anomalies in multivariate time series data presents a significant challenge because it demands simultaneous comprehension of temporal dependencies and variable relationships [9]. Moreover, the complexity of data gathered from modern IT systems, attributable to the abundance of variables, intensifies the challenges in modeling anomaly detection [9]. Consequently, there is a demand for a method capable of detecting anomalies in high-dimensional multivariate time series data by effectively understanding the correlations between variables. Typically, anomalies are rare in conventional system time series data, which limits the precise labeling of all historical data as normal or anomalous. The rarity of anomalies also makes it impractical to collect data covering all potential system failures. As a result, most time-series anomaly detection methods have gravitated towards learning models that do not utilize anomaly labels, with notable techniques including the Local Outlier Factor [11], One-Class Support Vector Machine [12],

* Corresponding author.
E-mail addresses: hyeongwon_kang@korea.ac.kr (H. Kang), pilsung_kang@korea.ac.kr (P. Kang).

https://doi.org/10.1016/j.knosys.2024.111507
Received 11 June 2023; Received in revised form 16 January 2024; Accepted 11 February 2024
Available online 1 March 2024

and Isolation Forest [13]. To overcome these constraints, researchers have adopted deep neural networks, proven to be successful in fields like computer vision and natural language processing, for time-series anomaly detection [14]. These approaches have confirmed that the application of deep models via representation learning of neural networks allows for the consideration of temporal dependencies, leading to superior anomaly detection performance compared to traditional methods [10].

Deep neural network-based anomaly detection methods for time-series data typically fall into two primary classifications: prediction-based and reconstruction-based methods. Prediction-based methodologies identify anomalies by learning the normal data distribution via predicting future time-series data based on past data [15,16]. These methods consider prediction errors on new time-series data as signals of anomalies, under the premise that a greater divergence of the prediction model from the learned normal data distribution results in a larger difference between the actual and predicted values. Notable prediction-based methods include LSTM-AD [15], DeepAnT [16], FuseAD [17], and LSTM-NDT [18]. On the other hand, reconstruction-based methods detect anomalies by transforming input data into a latent space and then striving to reconstruct the input data from this space [2,10,15]. Noteworthy examples of these reconstruction-based methods include OmniAnomaly [19] and BeatGAN [20]. These deep neural network strategies for identifying anomalies in time-series data largely use convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to capture the temporal patterns in the data. However, given that CNNs inherently concentrate on local information during learning, and RNNs encounter challenges with long-term dependencies, both architectures necessitate support in assimilating more comprehensive information about the input sequence.

In anomaly detection tasks that require accounting for various patterns within time-series data, using the aforementioned CNN/RNN architecture as-is could result in significant performance degradation. Consequently, there is a growing need for models capable of learning global information from time-series data. In response to this challenge, recent studies have attempted to develop time-series anomaly detection techniques employing the Transformer [21] architecture. Although originally proposed for NLP, Transformers have been successfully applied to both machine vision and NLP due to their ability to effectively learn global dependencies in sequences through their self-attention mechanism [22]. In the context of time-series data, Transformers create a self-attention map that illustrates the temporal significance of each moment, represented by the distribution of association weights across all points in time. This benefit allows the association distribution at each moment to deliver a more insightful representation of the temporal context, accentuating dynamic patterns like cycles or trends in the time series. Consequently, various time-series anomaly detection models have been proposed that utilize Transformers as their core architecture, including TranAD [23], Anomaly Transformer [24], and STOC [25].

However, while the Transformer-based methodologies proposed so far have the advantage of considering the temporal association at each time point, they have the limitation of not considering the association between variables at the same time point or in a continuous time series. The reason is that a typical transformer is based on a self-attention mechanism that does not consider the variables at each time point individually, but organizes them into one input token and processes them simultaneously [8]. Therefore, it can learn the temporal dependence of each time point, but it is difficult to learn the temporal dependence between each variable. In this paper, we propose a self-attention mechanism that can simultaneously consider temporal associations and inter-variable dependencies in the transformer structure to propose a reconstruction-based unsupervised learning model, The Variable Temporal Transformer (VTT), introduced in this paper, facilitates effective time series anomaly detection and interpretation of anomalous causes. This proposed self-attention mechanism blends temporal and variable attention, allowing for consideration of variable correlations via the transposed matrix of the existing attention input, in addition to temporal attention. The advantage of employing such a self-attention structure is the capability to monitor changes in correlation weights between time and variables through a comparison of the attention maps of reconstructed normal time series and anomalous time series during inference. This empowers the creation of an interpretable time series anomaly detection model capable of estimating when an anomaly occurs and the variables causing it. To validate the efficiency of the Variable Temporal Transformer (VTT), we pit it against seven established unsupervised time-series anomaly detection models, using four real-world multivariate time-series datasets. Recognizing that the point-adjustment F1 score can exaggerate a model's performance, we evaluated our model using the Area Under Curve (AUC) metric in conjunction with the recently proposed $F1_{PA\%K}$ measure. Our findings indicate that the VTT outshines comparable models by at least 0.003 and by as much as 0.43, with an average AUC of 0.48. Furthermore, the ablation study highlighted the efficacy of our proposed structural design and confirmed the VTT's capability to interpret anomaly detection results, as validated through both synthetic and real industrial datasets. The contributions are summarized as follows:

- We introduce variable attention that considers variable correlations using the transpose matrix of the existing attention input.
- We enable pinpointing anomaly occurrences and estimating causal variables by tracking changes in association weights during inference.
- The VTT achieves state-of-the-art results, with ablation studies and real-world data confirming the effectiveness and interpretability of our approach.

The rest of this paper is arranged as follows: Section 2 provides a brief overview of existing time-series anomaly detection methods using artificial neural network models, research that employs Transformers, and Explainable AI (XAI) in the context of time-series. Section 3 offers a detailed description of the proposed methodology. Section 4 outlines the benchmark datasets, reference models, and the procedures followed during the experiments. Section 5 presents and discusses the experimental results, ablation studies, and also reviews qualitative outcomes through the lens of XAI. Lastly, Section 6 concludes the paper with a summary and discussion.

2. Related works

2.1. Deep learning-based time-series anomaly detection (without Transformer)

Time series anomaly detection models based on deep learning can be broadly divided into prediction-based approaches and reconstruction-based approaches. Prediction-based approaches use models that identify regular patterns in time series data and predict future data points based on these patterns. Large discrepancies (residuals) between predicted and actual observed values are flagged as anomalies. Primarily, RNNs and CNNs have been utilized as prediction models to capture the temporal dynamics in time-series data. LSTM-AD [15] introduced an anomaly detection method using stacked LSTMs [26], an advanced type of RNN, which identifies anomalies by estimating the multivariate Gaussian distribution of prediction errors in the training data and then comparing these errors in new data with this distribution. DeepAnT [16] developed a deep CNN model consisting of convolutional and pooling layers, trained to learn the normal pattern of the time-series. The anomaly score is determined by the Euclidean distance between the predicted and actual values. FuseAD [17] implemented a hybrid approach using the ARIMA statistical model and a deep CNN model. ARIMA forecasts the next value in the time series and feeds it into an input sequence for the deep CNN model. The anomaly score is

Table 1
Baseline summary table.

Category	Method	Contributions
Predict	LSTM-AE	Stacked LSTM networks Leverage the likelihood of prediction error to detect anomalies
	DeepAnT	Deep CNN models built by stacking CNN and pooling layers
	FuseAD	Fuse a ARIMA, with a Deep CNN
Reconst	EncDec-AD	Training a Normal Distribution Using Auto-Encoders
	LSTM-VAE	Restoring data using a sampled representation with a VAE
	DAGMM	Combine Auto-Encoders with Gaussian mixture models
	USAD	Auto-Encoders with one encoder two independent decoders
	OmniAnomaly	A model that combines SRAE and VAE Reconstruction and probabilistic structure learning
	BeatGAN	Model with Generative Adversarial Networks
	Anomaly Transformer	Transformer-based Detecting anomalies through association discrepancies

again computed as the Euclidean distance between predicted and actual values.

Reconstruction-based time series anomaly detection methodologies learn patterns in time-series data, reconstruct the data, and then use the difference between the original and reconstructed data to identify anomalies. This approach learns a low-dimensional representation that effectively captures the structure of the original time-series data and uses it to reconstruct the original data, with data points exhibiting significant reconstruction errors considered anomalies. EncDec-AD [27] learns the typical data distribution using an autoencoder structure that consists of an encoder, which condenses the input data into a low-dimensional latent space, and a decoder that restores the data from the compact representation. LSTM-VAE [28] restructures the training data using a Variational Auto-Encoder (VAE), where the encoder masters the probability distribution of the training data, and the decoder rebuilds the data using a representation sampled from the distribution that the encoder has learned. DAGMM [29] combines an auto-encoder with a Gaussian Mixture Model (GMM) to detect anomalies using the reconstruction error of the data and density estimation based on the GMM. USAD [9] employs an auto-encoder-based approach, but unlike traditional auto-encoders, it aims to minimize the difference between the original and reconstructed data using two independent auto-encoders. OmniAnomaly [19] uses a model that combines Stacked Recurrent Autoencoder (SRAE) and Variational Autoencoder (VAE) to detect anomalies by learning probabilistic structures concurrently with data reconstruction. In contrast, BeatGAN [20] utilizes a Generative Adversarial Networks (GANs) model, where the generator learns the normal pattern of the time-series data and attempts to deceive the discriminator by generating new normal data. The discriminator tries to distinguish the generator's fake data from the real data, prompting the generator to progressively learn the normal pattern of the time series data more effectively.

2.2. Transformer-based time-series anomaly detection

Transformer [21] is a technique originally designed for machine translation in Natural Language Processing (NLP), which facilitates learning of global dependencies within sequences via attention mechanisms. It has outperformed RNNs and CNNs, traditionally used for sequential data handling, in numerous NLP tasks including machine translation [30]. Moreover, its effectiveness extends beyond NLP to various other fields like speech processing and computer vision [31–33]. Its proficiency has also been demonstrated in the time-series domain, with multiple studies utilizing its structure. Notable studies leveraging the Transformer in the time series realm include Informer [34], Autoformer [35], and Fedformer [36]. The Anomaly Transformer [24], a specialized Transformer-based method for time-series anomaly detection, aims to leverage the inherent difference between normal and anomalous patterns in terms of association distribution. This is done via association discrepancy, where anomalies are rare and normal patterns prevalent. It is difficult for anomalies to form strong associations with the overall series, but due to their continuity, anomalies tend to show similar patterns at close time points (see Table 1).

3. Proposed method

In this paper, we propose a Variable Temporal Transformer (VTT) using the Variable Temporal Attention Mechanism to overcome the limitation of the Transformer, specifically its difficulty in learning temporal dependencies between variables, in order to enable effective anomaly detection and interpretation in multivariate time-series data.

Multivariate time-series data $\mathbf{T} = \mathbf{x}_1, \ldots, \mathbf{x}_T$ represents a set of sequential multivariate vectors in which two or more variables are recorded chronologically, with $\mathbf{x}_t \in \mathbb{R}^v$ denoting the observation at time t. In the context of multivariate time-series anomaly detection, unsupervised learning involves grasping the features of a given training dataset without specific labels indicating whether anomalies are present or not. Following this learning process, it decides if the observation $\mathbf{x}_r (r > T)$ at a future time point is considered an anomaly. This task requires identifying observations not seen during the training phase and is mainly achieved by quantitatively measuring the difference between the training data $\mathbf{T}$ and the new data $\mathbf{x}_r$. The proposed VTT is a reconstruction-based anomaly detection model, in which the difference between the reconstruction output of the model trained on the untrained sample $\mathbf{x}_r$ and the normal set $\mathbf{T}$ is measured as an anomaly score. This score is then compared to a threshold value to determine if it constitutes an anomaly.

3.1. Variable Temporal Transformer

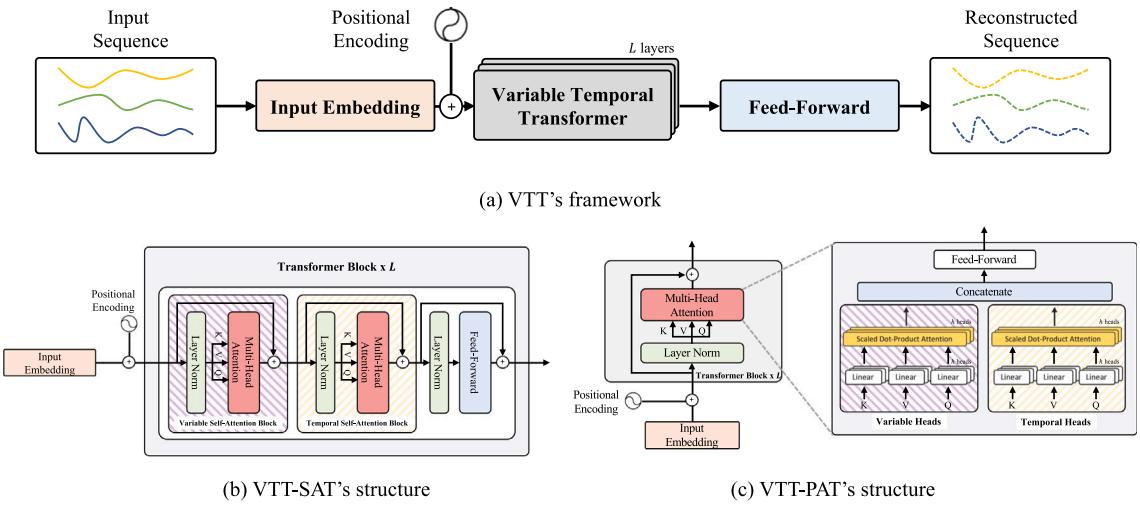
3.1.1. Overall architecture

The structure of the VTT proposed in this study is illustrated in Fig. 1. Rather than using a conventional self-attention module like the vanilla Transformer encoder, the VTT adopts an alternating pattern between variable temporal self-attention modules and feed-forward layers. The proposed VTT model applies L layers to an input time-series $X \in \mathbb{R}^{N \times V}$, where N signifies the length and V denotes the number of variables. The operations of the lth layer are represented by the following equations Eq. (1) and Eq. (2):

$$\mathbf{Z}^l = \text{Layer-Norm}\left(\text{Variable \& Temporal-Attention}\left(\mathbf{X}^{l-1}\right) + \mathbf{X}^{l-1}\right), \quad (1)$$

$$\mathbf{X}^l = \text{Layer-Norm}\left(\text{Feed-Forward}\left(\mathbf{Z}^l\right) + \mathbf{Z}^l\right), \quad (2)$$

where $\mathbf{X}^l \in \mathbb{R}^{N \times V \times d_{\text{model}}}, l \in \{1, \ldots, L\}$ represents the output of the lth layer with channel d_{model}. The initial input, $\mathbf{X} = \text{Embedding}(X)$, stands

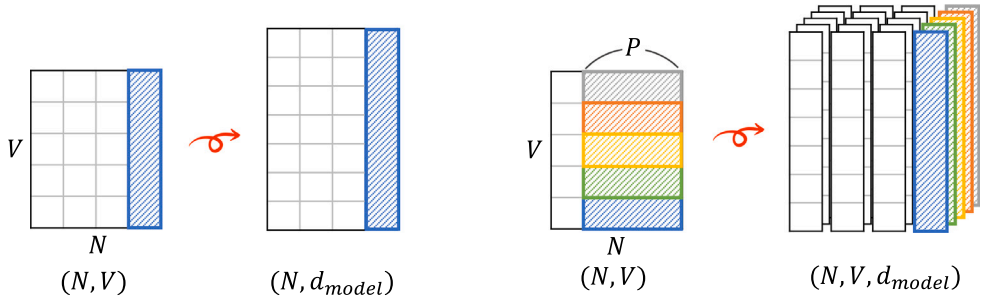


(a) VTT's framework

(b) VTT-SAT's structure

(c) VTT-PAT's structure

Fig. 1. Variable Temporal Transformer architecture.



(a) Vanilla Transformer's Embedding

(b) Variable Temporal Transformer's Embedding

Fig. 2. Difference between the embeddings in vanilla transformer and VTT.

for the raw embedded time-series, while $\mathbf{Z}^l \in \mathbb{R}^{N\times V\times d_{\text{model}}}$ is the hidden representation of the lth layer. Lastly, Variable & Temporal-Attention$(\cdot)$ is an attention module that calculates temporal dependencies among variables.

VTT first performs embedding to generate a multi-resolution representation while preserving variable information. Subsequently, it models the correlation and temporal dependence between variables through the Variable Temporal Transformer layer. Lastly, the feed-forward layer is employed to reconstruct the data, enabling anomaly detection through differences from the input data [27–29].

3.1.2. Model inputs and embedding

As shown in Fig. 2(a), the Vanilla Transformer embeds data into a single token for all variables at one time point. Using this conventional embedding method, the information of each variable is merged, so the information of individual variable from the original data cannot be identified. Consequently, the VTT proposed in this paper adopts a method to preserve the information of variables by independently embedding each variable. Instead of embedding all variables at one time point into one token, we embed the values of each variable from the current time point to a certain previous point P in time into one token, as depicted in Fig. 2(b). An embedding method that utilizes information up to the previous point P is to use an Dilated Causal Convolution layers, which is described in more detail in the following paragraphs. By applying this embedding method, we obtain a three-dimensional vector $\mathbf{X} \in \mathbb{R}^{N\times V\times d_{\text{model}}}$ that preserves information about all time points and variables, unlike the two-dimensional representation of the Vanilla Transformer's embedding method.

We adopted an embedding method using Dilated Causal Convolution layers with different kernel sizes and dilation factors to leverage multi-resolution while preserving variable information. Dilated Causal Convolution is a Causal Convolution method that considers only historical data up to time step t to generate an output at time step t and uses a dilation factor to adjust the convolutional kernel spacing for learning long-range dependencies [37]. As shown in Fig. 3, $X_t \in \mathbb{R}^{1\times V}$ contained in the input window, J layers of multiple dilated causal convolutions with different kernel sizes and scaling factors are applied in parallel to one variable in the input window to obtain their respective output values $X'_t \in \mathbb{R}^{1\times V\times J}$. These outputs are then concatenated to form a single output value, and a linear layer is used to compute the embedding vector $\mathbf{u}_t$ in the d_{model} dimension. This process is repeated for all points in the input window.

Next, we incorporate a time stamp embedding to include the order information of the input sequence. To integrate the input sequence order into the embedding vector $\mathbf{u} \in \mathbb{R}^{N\times V\times d_{model}}$, we add positional encoding in a manner similar to the Vanilla Transformer, thereby forming the final input $\mathbf{X}$ for the Variable Temporal Transformer encoder layer.

3.1.3. Variable & Temporal Attention

The VTT proposed in this paper uses two attention modules: Temporal self-attention and Variable self-attention. For Temporal self-attention, the Vanilla Transformer's self-attention is adopted to extract temporal semantic information. The three-dimensional vector $\mathbf{X}^l$ is converted into a two-dimensional vector $\mathbf{X}^l_{\mathcal{T}} \in \mathbb{R}^{N\times(V\times d_{\text{model}})}$ along the time axis to input the embedding X^l into Temporal self-attention. Conversely, Variable self-attention utilizes a transposed form of Temporal

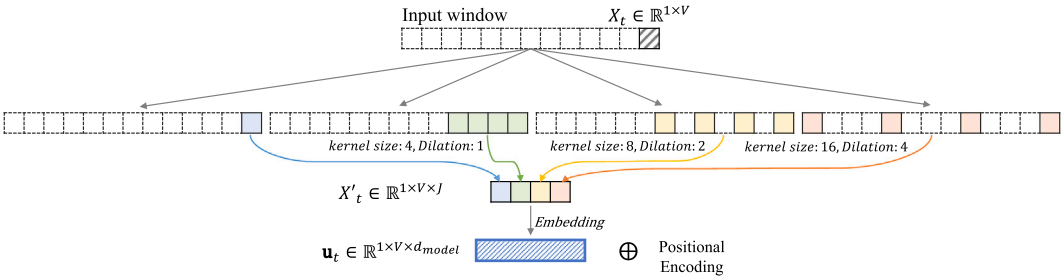
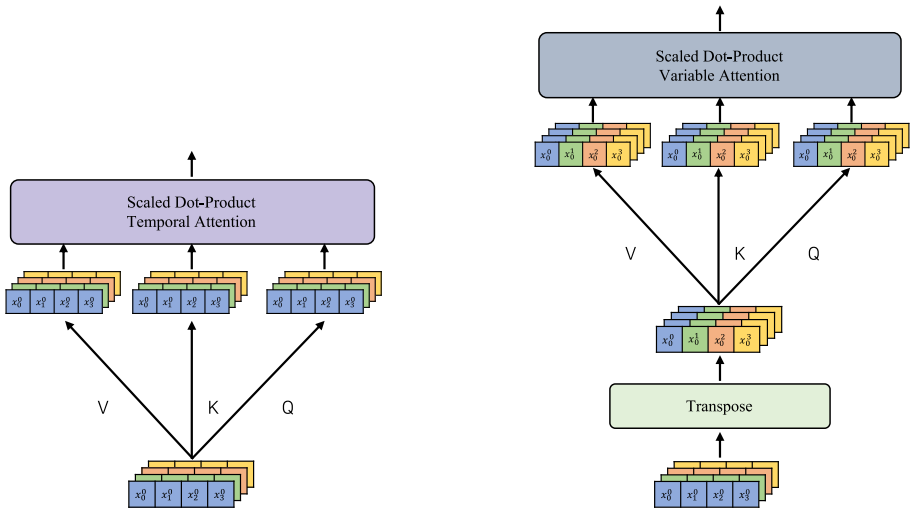


Fig. 3. Embedding process of VTT.



(a) Temporal Self-Attention (b) Variable Self-Attention

Fig. 4. Temporal Self-Attention and Variable Self-Attention.

self-attention to model the correlation between variables, converting the three-dimensional vector $\mathbf{X}^l$ into a two-dimensional vector $\mathbf{X}^l_{\mathcal{V}} \in \mathbb{R}^{V\times(N\times d_{\text{model}})}$ along the variable axis to input the embedding $\mathbf{X}^l$ into Temporal self-attention. The formulas for Temporal self-attention and Variable self-attention are shown in Eq. (3):

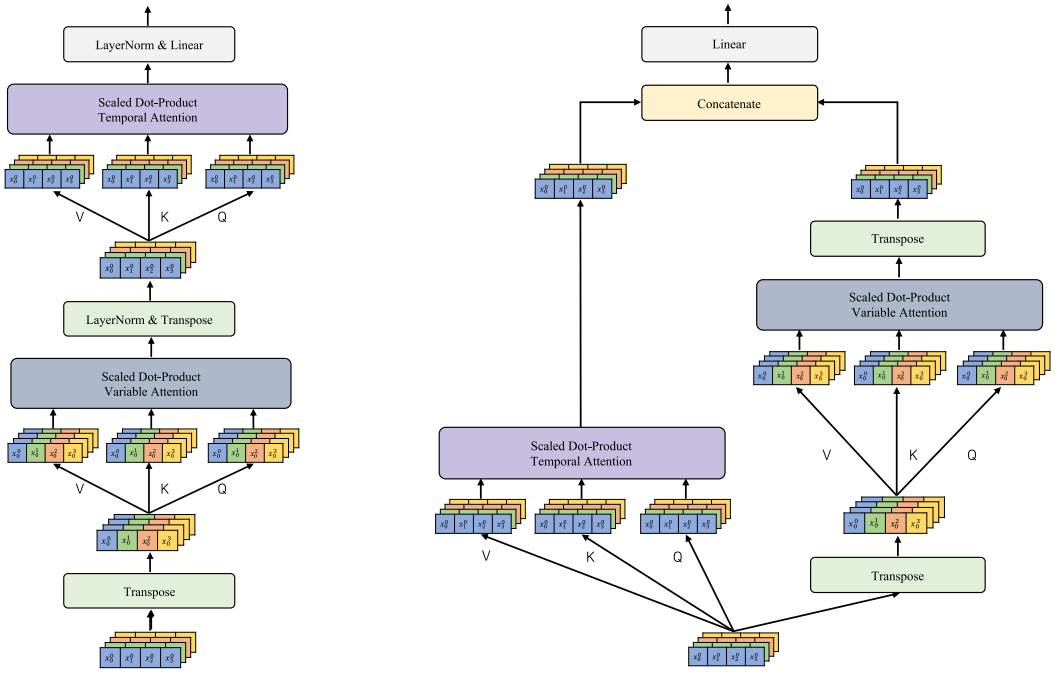
$$\begin{aligned}
\text{Initialization}: \quad & \begin{matrix} Q_{\mathcal{T}}, K_{\mathcal{T}}, V_{\mathcal{T}} = \mathbf{X}^{l-1}_{\mathcal{T}} W^l_{Q_{\mathcal{T}}}, \mathbf{X}^{l-1}_{\mathcal{T}} W^l_{K_{\mathcal{T}}}, \mathbf{X}^{l-1}_{\mathcal{T}} W^l_{K_{\mathcal{T}}} \\ Q_{\mathcal{V}}, K_{\mathcal{V}}, V_{\mathcal{V}} = \mathbf{X}^{l-1}_{\mathcal{V}} W^l_{Q_{\mathcal{V}}}, \mathbf{X}^{l-1}_{\mathcal{V}} W^l_{K_{\mathcal{V}}}, \mathbf{X}^{l-1}_{\mathcal{V}} W^l_{V_{\mathcal{V}}} \end{matrix}, \\
\text{Temporal self-attention}: \quad & \mathcal{T}^l = \text{Softmax}\left(\frac{Q_{\mathcal{T}} K^{\mathrm{T}}_{\mathcal{T}}}{\sqrt{V \times d_{\text{model}}}}\right), \\
\text{Variable self-attention}: \quad & \mathcal{V}^l = \text{Softmax}\left(\frac{Q_{\mathcal{V}} K^{\mathrm{T}}_{\mathcal{V}}}{\sqrt{N \times d_{\text{model}}}}\right), \\
\text{Temporal Reconstruction}: \quad & \hat{\mathbf{Z}}^l_{\mathcal{T}} = \mathcal{T}^l V_{\mathcal{T}}, \\
\text{Variable Reconstruction}: \quad & \hat{\mathbf{Z}}^l_{\mathcal{V}} = \mathcal{V}^l V_{\mathcal{V}},
\end{aligned} \tag{3}$$

where $Q_{\mathcal{T}}, K_{\mathcal{T}}, V_{\mathcal{T}} \in \mathbb{R}^{N\times(V\times d_{\text{model}})}$ are the queries, keys, and values of Temporal self-attention, and $Q_{\mathcal{V}}, K_{\mathcal{V}}, V_{\mathcal{V}} \in \mathbb{R}^{V\times(N\times d_{\text{model}})}$ are the queries, keys, and values of Variable self-attention. And, $W^l_{Q_{\mathcal{T}}}, W^l_{K_{\mathcal{T}}}, W^l_{V_{\mathcal{T}}} \in \mathbb{R}^{(V\times d_{\text{model}})\times(V\times d_{\text{model}})}$, $W^l_{Q_{\mathcal{V}}}, W^l_{K_{\mathcal{V}}}, W^l_{V_{\mathcal{V}}} \in \mathbb{R}^{(N\times d_{\text{model}})\times(N\times d_{\text{model}})}$ are the parameter matrices for Q, K, V of the lth layer of Temporal self-attention and Variable self-attention, respectively. To obtain temporal Attention scores and variable attention scores, we multiply each query by its key and divide by the square root of the $V \times d_{\text{model}}$ and $N \times d_{\text{model}}$ dimensions for gradient stabilization. The attention map is normalized via Softmax$(\cdot)$ to obtain the final self-attention scores. Finally, $\hat{\mathbf{Z}}^l_{\mathcal{T}} \in \mathbb{R}^{N\times(V\times d_{\text{model}})}$, $\hat{\mathbf{Z}}^l_{\mathcal{V}} \in \mathbb{R}^{V\times(N\times d_{\text{model}})}$ are the hidden representations of Temporal self-attention and Variable self-attention after the lth layer, respectively (see Fig. 4).

In the multi-head version with h heads, $Q_{m\mathcal{T}}, K_{m\mathcal{T}}$, and $V_{m\mathcal{T}}$ represent the query, key, and value of the mth head for temporal attention. Similarly, $Q_{m\mathcal{V}}, K_{m\mathcal{V}}, V_{m\mathcal{V}}$ represent the query, key, and value of the mth head for variable attention. Here, $Q_{m\mathcal{T}}, K_{m\mathcal{T}}, V_{m\mathcal{T}} \in \mathbb{R}^{N\times\frac{(V\times d_{\text{model}})}{h}}$, and $Q_{m\mathcal{V}}, K_{m\mathcal{V}}, V_{m\mathcal{V}} \in \mathbb{R}^{V\times\frac{(N\times d_{\text{model}})}{h}}$. We then concatenate the output values $\{\hat{\mathbf{Z}}^l_{m\mathcal{T}} \in \mathbb{R}^{N\times\frac{(V\times d_{\text{model}})}{h}}\}_{1\le m\le h}$, and $\{\hat{\mathbf{Z}}^l_{m\mathcal{V}} \in \mathbb{R}^{V\times\frac{(N\times d_{\text{model}})}{h}}\}_{1\le m\le h}$ from multiple heads to obtain the final results $\hat{\mathbf{Z}}^l_{\mathcal{T}}$, $\hat{\mathbf{Z}}^l_{\mathcal{V}}$.

3.1.4. Variable Temporal Transformer's structure

In the VTT proposed in this paper, we integrate the aforementioned Temporal self-attention and Variable self-attention to enable the model to learn both temporal and inter-variable correlations. As illustrated in Fig. 5, we propose two approaches depending on the combination: Variable Temporal Transformer-Serial Attention (VTT-SAT), a serial structure that learns temporal dependence after learning inter-variable correlations, and Variable Temporal Transformer-Parallel Attention (VTT-PAT), a parallel structure that simultaneously learns inter-variable correlations and temporal dependence. First, VTT-SAT converts the input sequence $\mathbf{X}^l$ into inputs $\mathbf{X}^l_{\mathcal{V}}$ for variables and performs variable self-attention. Subsequently, the obtained $\hat{\mathbf{Z}}^l_{\mathcal{V}}$ is converted into a temporal input $\mathbf{X}^l_{\mathcal{T}}$, and the final representation $\hat{\mathbf{Z}}^l \in \mathbb{R}^{N\times V\times d_{\text{model}}}$ is derived through temporal self-attention. Conversely, VTT-PAT converts the input sequence $\mathbf{X}^l$ into input $\mathbf{X}^l_{\mathcal{V}}$ for variables and input $\mathbf{X}^l_{\mathcal{T}}$ for time, and performs variable self-attention and temporal self-attention in parallel with both input values. To concatenate the



(a) VTT-SAT Attention module (b) VTT-PAT Attention module

Fig. 5. The attention module and structure of VTT-SAT and VTT-PAT.

respective outputs $\hat{\mathbf{Z}}_{\mathcal{V}}^{l}, \hat{\mathbf{Z}}_{\mathcal{T}}^{l}$ from variable and temporal self-attention, we reshape and concatenate them into three-dimensional vectors of size $N \times V \times d_{model}$ to derive one final representation $\hat{\mathbf{Z}}^{l} \in \mathbb{R}^{N \times V \times 2d_{\text{model}}}$. Both the VTT-SAT and VTT-PAT structures use a single feed-forward layer to derive the final reconstruction output $\hat{\mathbf{X}} \in \mathbb{R}^{N \times V}$ after running the aforementioned transformer layers.

3.1.5. Loss function

The loss function for the constructed reconstruction model is the average Mean Squared Error (MSE) between the actual values $\mathbf{X}_t = \mathbf{x}_{t-N+1}, \mathbf{x}_{t-N+2}, \ldots, \mathbf{x}_t$ and the reconstructed values $\hat{\mathbf{X}}_t = \hat{\mathbf{x}}_{t-N+1}, \hat{\mathbf{x}}_{t-N+2}, \ldots, \hat{\mathbf{x}}_t$, spread across the total count of sequences T in the training dataset. The loss function for the input time series $\mathbf{X} \in \mathbb{R}^{N \times V}$ is expressed by Eq. (4):

$$L_{\text{Total}}(\hat{\mathbf{x}}; \mathbf{x}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 . \tag{4}$$

As shown, a reconstruction model, trained to diminish the difference between the rebuilt and true values, can extract a compact representation from the training data. This procedure accurately grasps the structure of the initial time-series data, generating reconstructed values that align with the distribution of normal data.

3.2. Anomaly scoring and detecting

In this paper, we utilize the reconstruction error at each moment in time as an anomaly score, marking occurrences where this value surpasses a certain threshold as anomalies. We employ Mean Squared Error (MSE) to calculate the reconstruction error. The formula for the anomaly score is given in equation 5:

$$\mathcal{A}(\hat{\mathbf{x}}; \mathbf{x}) = |\mathbf{x} - \hat{\mathbf{x}}|_2^2 , \tag{5}$$

where $\mathcal{A}(\hat{\mathbf{x}}; \mathbf{x}) \in \mathbb{R}^{N \times 1}$ symbolizes the anomaly score at each individual time point.

Time-series anomaly detection models that leverage unsupervised learning can generate an anomaly score for the input data at any given time. However, without labels to provide correct answers, establishing a threshold value to discern anomalies is a challenging task. In this paper, we follow the protocol in [38] to set the threshold that yields the best F1. This method allows us to evaluate the inherent anomaly detection ability of different methods without letting the thresholding technique affect their performance. We used the threshold that yielded the best F1 score as the final threshold to obtain the performance of the proposed model and the comparison model, and then performed a comparative analysis between the models.

3.3. Anomaly Interpretation

In this paper, we introduce an interpretable module for abnormal time-series data detected by the proposed VTT. As shown in Fig. 6, the proposed method estimates the cause of the anomaly by comparing the attention map extracted from the original data containing the anomaly and the attention map extracted from the reconstructed data. When the original data with anomalies is used as input to VTT, it can be assumed that a different attention map will be generated compared to when normal data is used. Since the output time-series of the VTT is reconstructed with normal data similar to the input time-series but without anomalies, the output time-series can be used again as input to the VTT to extract the attention map for reconstruction. The difference between the attention map required to reconstruct an anomaly time-series with normal data and the attention map required to reconstruct a similar normal time-series provides information about which variables have significantly changed their weights and which variables have altered their correlations. This method can be applied not only to variable attention but also to temporal attention, and the interpretability of the proposed method is verified in Section 5.4, Anomaly Interpretation Analysis.

4. Experimental setting

4.1. Datasets

In this paper, we demonstrate the effectiveness of the proposed VTT by utilizing four benchmark datasets widely used in multivariate time

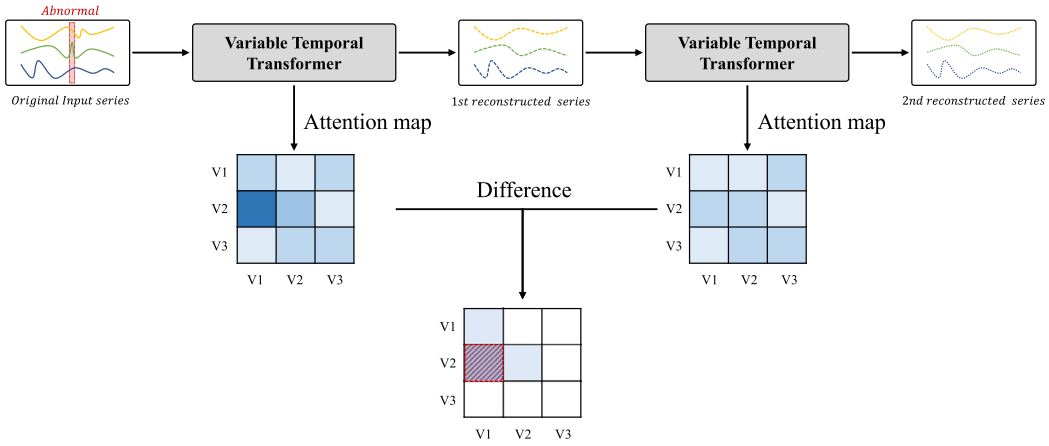


Fig. 6. Anomaly interpretation process.

series anomaly detection research: SWaT, SMD, SMAP, and MSL. (1) SWaT (Secure Water Treatment) [39] comprises water quality-related time-series data collected from 51 sensors over 11 days, with 36 instances of anomalies occurring in the final four days of the 11-day span. The dataset also includes a detailed account of the attack. We utilize this description in our paper to ascertain if the Variable Temporal Transformer can offer a suitable interpretation in the Anomaly Interpretation analysis in Section 5.4.1. (2) SMD (Server Machine Dataset) [19] is a 5-week, 38-dimensional dataset collected from a large Internet company, encompassing 38 metrics obtained from 28 servers over the course of five weeks. (3) MSL (Mars Science Laboratory rover) and SMAPE (Soil Moisture Active Passive satellite) [18] are datasets released by NASA, featuring 55 and 25 dimensions respectively, and containing telemetry anomaly data extracted from Incident Surprise Anomaly (ISA) reports from the spacecraft monitoring system. Among the datasets we used, SMD, MSL, and SMAP consist of multiple sub-datasets, and we crafted models for each sub-dataset individually. Table 2 provides a summary of the number of variables, observations, composition of training/validation/test data, and the percentage of true anomalies in the test data for the datasets used in this paper.

Table 2
Details of benchmarks.

	SWaT	SMD	MSL	SMAP
Variables	51	38	55	25
Number of entities	1	28	27	54
#Train (0.8)	396,000	566,724	46,653	110,403
#Valid (0.2)	99,000	141,681	11,664	27,601
#Test (labeled)	449,919	708,420	73,729	427,617
Anomaly (%)	11.98	4.16	10.72	13.13

4.2. Baselines and evaluation metrics

In this paper, we selected seven reconstruction-based anomaly detection method models for comparison to demonstrate the effectiveness of the proposed VTT: EncDec-AD [27], LSTM-VAE [28], BeatGAN [20], USAD [9], DAGMM [29], OmniAnomaly [19], and Anomaly Transformer [24].

The Point Adjustment F1 score, a variation of the standard F1 score, is currently the most widely-used performance assessment measure in the realm of time-series anomaly detection. This metric refines the traditional F1 score to better align with the specific characteristics of anomaly detection. As displayed in the given Eq. (6), the usual F1 score is calculated as the harmonic mean of Precision and Recall, which helps evaluate the equilibrium between these two metrics:

$$F1 - score = 2 \cdot \frac{precision \cdot Recall}{precision + Recall}. \tag{6}$$

Anomalies often continue for a duration, forming a continuous anomaly period. It is assumed that sounding an alarm for every point within this anomaly period is unnecessary once an anomaly is detected. Instead, a single notification can trigger system recovery actions. Reflecting this, the point adjustment strategy treats all points within the anomaly period as correctly identified if at least one observation in the ground-truth continuous anomaly period is accurately identified. However, recent studies have both theoretically and empirically shown that the Point Adjustment F1 score method can lead to an overestimation of the model's performance [38,40]. One experiment even demonstrated that the Point Adjustment F1 score calculated with random anomaly scores surpassed most of the current leading-edge methods [38,40]. To evaluate the performance of each method more accurately, we employ a recently proposed metric, $F1_{PA\%K}$ [38], as our assessment measure.

$F1_{PA\%K}$ is a method that applies Point Adjustment to an anomaly only if the percentage of correctly detected anomalies in the anomaly exceeds the PA%K threshold, K. The adjusted label is equal to Eq. (7):

$$\hat{y}_t = \begin{cases} 1, & \text{if } \mathcal{A}(\hat{\mathbf{x}}; \mathbf{x}) > \delta \text{ or } t \in S_m \text{ and } \frac{|\{t' | t' \in S_m, \mathcal{A}(\hat{\mathbf{x}}; \mathbf{x}) > \delta\}|}{|S_m|}, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\hat{y}_t$ is the $F1_{PA\%K}$ scaled label for time t, $|\cdot|$ denotes the size of S_m (anomaly segment), and K can be manually selected between 0 and 100. In this paper, the $F1_{PA\%K}$ area under the curve (AUC) obtained by increasing K from 0 to 100 is evaluated as the final performance. Following the protocol in [38], we set 16 values of K.

4.3. Implementation details

The VTT was developed using Pytorch [41] version 1.7.1 and trained on a server equipped with an Intel (R) Xeon (R) CPU E5-2698 v4 @ 2.20 GHz and two Tesla V100 PCIe 32 GB GPUs. The ADAM [42] optimizer was used for training with an initial learning rate of 10^{-4}. The Variable Temporal Transformer consisted of three layers, and the number of channels in the hidden states d_{model} was set to 128 with the number of heads h at 8. It included three dilated causal convolution layers for embedding. For the kernel size and dilation factor of the convolution layers, we set them to (4, 1), (8, 2), and (16, 3), respectively, through a tuning process that allowed us to progressively utilize information from past points in time. For all datasets, we set the sliding window size to 100 and the step size to 50, following the settings used by Anomaly Transformer and other existing anomaly detection studies [1]. The training process used a batch size of 32, and learning was stopped early if the validation dataset loss did not decrease for 10 successive epochs.

Table 3
Multivariate time series anomaly detection results on four datasets.

Dataset	SWaT			SMD		
Metric	$F1_{PA}$ (K=0)	F1 (K=100)	AUC	$F1_{PA}$ (K=0)	F1 (K=100)	AUC
LSTM-AE	86.27	77.13	82.27	**86.89**	46.09	56.48
LSTM-VAE	86.46	76.57	81.38	86.67	45.54	55.71
BeatGAN	85.40	71.57	79.75	82.55	28.94	34.71
USAD	87.02	76.57	81.38	86.72	46.20	56.61
DAGMM	85.55	76.57	81.35	86.67	45.87	56.14
OmniAnomaly	87.32	76.57	81.39	86.68	45.85	56.10
Anomaly Transformer	**96.35**	2.18	3.62	76.94	2.50	4.23
VTT-SAT	84.65	78.00	83.28	86.65	**47.63**	**57.91**
VTT-PAT	84.64	**78.29**	**83.52**	86.68	46.68	56.75
Dataset	MSL			SMAP		
Metric	$F1_{PA}$ (K=0)	F1 (K=100)	AUC	$F1_{PA}$ (K=0)	F1 (K=100)	AUC
LSTM-AE	86.65	20.01	28.58	79.25	13.30	19.33
LSTM-VAE	86.12	19.86	28.33	81.43	13.11	19.27
BeatGAN	**89.97**	5.94	7.34	**86.57**	7.52	10.19
USAD	87.00	19.84	28.23	81.21	13.29	20.00
DAGMM	85.93	19.69	28.43	81.86	**16.56**	**24.46**
OmniAnomaly	85.73	19.76	28.41	81.93	16.08	23.71
Anomaly Transformer	83.72	3.18	4.90	81.60	3.89	6.26
VTT-SAT	85.97	18.94	27.85	81.16	15.02	22.01
VTT-PAT	87.15	**20.24**	**29.24**	81.13	14.86	22.00

5. Experimental results

5.1. Main results

In Section 5.1, we compare the performance of VTT with the seven reconstruction-based methods selected in Section 4.2 to validate the effectiveness of VTT in terms of three quantitative metrics: Point Adjusted F1 score, F1 score, and $F1_{PA\%K}$ AUC. Table 3 shows the experimental results of all methods on the four datasets, with bold and underlined values indicating the best performance. All performance measurements were taken on the test set. VTT achieved the best results on three datasets (SWaT, SMD, and MSL) based on AUC. While other reconstruction-based methods do not explicitly model the correlation between variables, the excellent performance of the proposed methodology in this paper confirms the effectiveness of explicitly modeling the correlation between variables for anomaly detection in multivariate time series. On the SWaT and MSL datasets, VTT-PAT performed well considering variables and temporals in parallel, while VTT-SAT performed better on the SMD dataset considering variables and temporals in series. However, the performance difference between VTT-SAT and VTT-PAT on the SMD dataset was very close. Depending on which dataset you use, VTT-SAT and VTT-PAT can be used optionally, but VTT-PAT seems to be the better choice. Interestingly, among USAD, OmniAnomaly, and Anomaly Transformer, which have recently shown to perform very well in time-series anomaly detection, Anomaly Transformer exhibits a very high Point Adjusted F1 score, but a sharp drop in F1 and $F1_{PA\%K}$ AUC performance. For the Anomaly Transformer, the anomaly score tended to repeatedly increase and decrease, and setting the threshold using a point adjustment strategy resulted in an overestimation of detected anomalies because it was able to detect all anomaly segments despite many false alarms. This indicates that the anomaly transformer performs poorly on a particular metric, the $F1_{PA\%K}$ AUC, which can be interpreted as an overestimate as argued in [38]. In contrast, the proposed VTT outperforms in all three evaluation metrics used in the experiment: F1 score, and $F1_{PA\%K}$ AUC, indicating that the proposed VTT can not only detect more anomalous segments but also generate more true alarms within anomalous segments, proving robust to changes in K%, the hyperparameter of Point Adjustment, compared to other methods.

Fig. 7 shows a graph of $F1_{PA\%K}$ as a function of the K value. We ran the experiment for 16 values of K, with increments of 1 from 0 to 5 and increments of 10 from 10 to 100, and recorded the $F1_{PA\%K}$ score for each K. The larger the K value, the larger the percentage of anomalies that should be detected within the anomaly segment, which is a more stringent evaluation of the scoring strategy. In the SWaT dataset, the $F1_{PA\%K}$ score decreases less as the value of K increases, while in the other datasets, the $F1_{PA\%K}$ score decreases significantly as the value of K increases. This suggests that the SWaT dataset can detect many anomalies within the anomaly segment. In contrast, the other datasets have fewer anomaly points detected as anomalies within the anomaly segment. Each methodology's $F1_{PA\%K}$ score graphs each methodology, we can see that for all datasets, Anomaly Transformer and BeatGAN have a $F1_{PA\%K}$ score decreases sharply. Anomaly Transformer showed a tendency to repeatedly increase and decrease anomaly scores as mentioned earlier, and BeatGAN underperformed the reconstruction compared to the other baseline models. We can see that the detected outliers are overestimated when using the point adjustment strategy. VTT's $F1_{PA\%K}$ score graph for VTT, which is suggested by most datasets, has a larger area at the bottom, indicating that more anomalies are detected within the actual anomaly segment than the other methodologies.

5.2. Ablation study

An ablation study was conducted to determine the importance of correlations between variables and how using the Dilated Causal Convolution layer for embedding, retaining variable information and utilizing multi resolution affects the VTT model. As shown in Table 4, VTT was compared to Vanilla Transformer and VTT with embeddings removed. Vanilla Transformer used only Temporal Attention with Variable Attention removed and used vanilla embeddings. Since the VTT without embedding cannot be represented by the three-dimensional vector $\mathbf{X}_0$, it uses a two-dimensional vector as the input vector and does not apply the vector transformation process performed by the VTT. First, comparing the proposed VTT with the version without Variable Attention shows that the VTT with Variable Attention, which models correlations between variables, outperforms the Vanilla Transformer, which only considers time dependence, on all benchmark datasets by AUC. This confirms the importance of considering the correlation between variables in multivariate time series anomaly detection and indicates that the proposed VTT effectively captures the correlation between variables. Furthermore, by comparing the proposed VTT with a de-embedded version, we can see that while removing the embedding

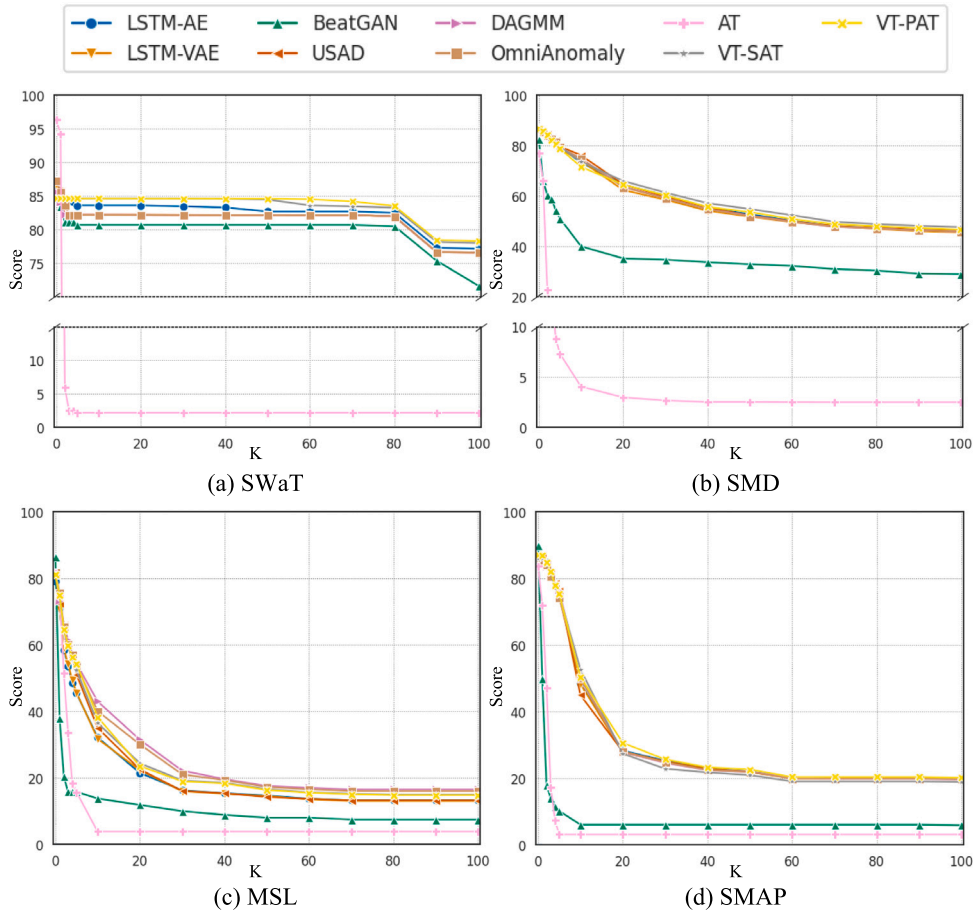


Fig. 7. Graph of AUC as K varies.

Table 4
Ablation study of variable attention and embedding methods.

Variable attention		Embedding		SWaT	SMD	MSL	SMAP
Serial	Parallel	Vanilla	Dilated causal	AUC	AUC	AUC	AUC
×	×	O	×	82.23	55.90	28.25	21.56
O	×	×	×	81.26	55.90	28.49	19.61
×	O	×	×	81.44	55.88	**29.43**	18.89
O	×	×	O	83.28	**57.91**	27.85	**22.01**
×	O	×	O	**83.52**	56.75	29.24	22.00

Table 5
Ablation study of embedding methods with different numbers of layer.

# Layer	Metric	$F1_{PA}$ (K=0)	F1 (K=100)	AUC
1	VTT-SAT	84.68	77.98	83.29
	VTT-PAT	84.62	78.11	83.38
2	VTT-SAT	**84.95**	77.83	83.15
	VTT-PAT	84.65	78.12	83.43
3	VTT-SAT	84.65	78.00	83.28
	VTT-PAT	84.64	**78.29**	**83.52**

reduces the complexity of the model, it also decreases its performance. This demonstrates that the embedding method using the Dilated Causal Convolution layer can help learn a representation of time series data while additionally utilizing multi resolution information.

5.3. Parameter sensitivity

Table 5 presents the results of VTT's experiments with varying the number of layers on the SWaT dataset. The number of layers was tested from 1 to 3. As seen in Table 5, VTT's performance achieves the highest AUC value when the number of layers is 3. However, the difference in performance is not significant when the number of layers is 1 or 2. From this result, we can see that the performance is not very sensitive to the number of layers and remains stable between 1 and 3 layers.

5.4. Anomaly interpretation analysis

We validate VTT's ability to interpret anomalous time series data with the anomaly interpretation module proposed in Section 3.3. For validation, we use the SWaT benchmark dataset, which is both synthetic and real-world industrial data, to estimate the cause of anomalies by comparing the attention map extracted from the original data and the attention map extracted from the reconstructed data.

5.4.1. Synthetic data

To validate the interpretative ability of the VTT model for anomalous time-series data, we generated synthetic data and conducted experiments. The synthetic data consisted of five multivariate time-series, including sinusoidal and linear patterns with some noise. To establish

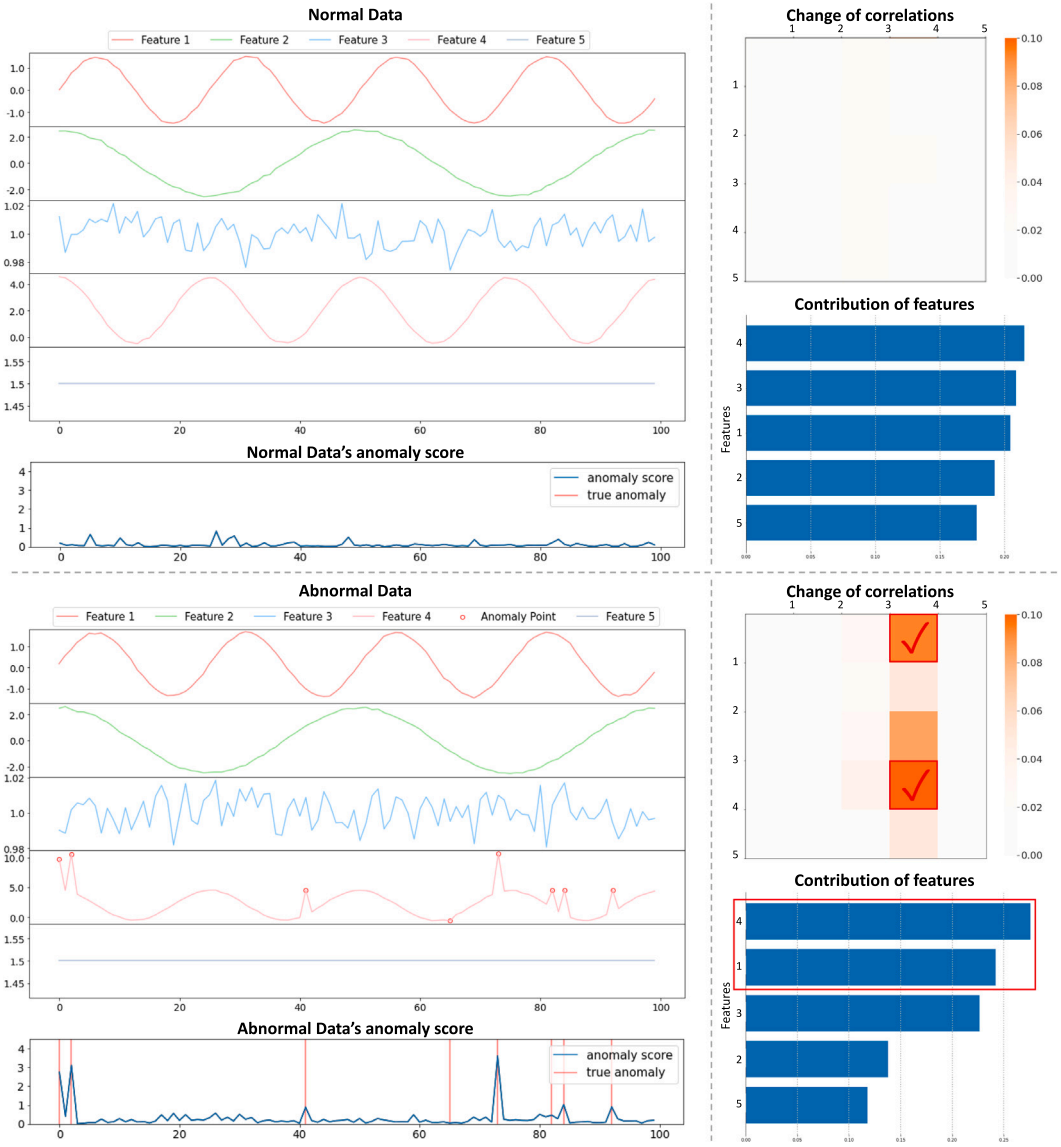


Fig. 8. Visualize training and test data results from synthetic datasets.

a correlation between the variables, Feature 1 and Feature 4 have sine and cosine waves with the same period but different amplitudes, while the remaining variables are linear with cosine waves of different periods. The training data consists of only normal data without anomalies, and only the test data contains anomalies. We added a point global anomaly to Feature 4 only for the anomalies. Fig. 8 displays the raw time-series graph and anomaly score graph for the training and test data, a heatmap showing the change in correlation, and the contribution of the most influential variables to the anomalies. In the anomaly score graph of the training data, we can see that the overall score is low, and in the test data, the score is high in the anomaly segment. Examining the change in correlation, the change in correlation between the input data and the reconstructed data was minimal for the training data. Conversely, for the test data, the correlation of Feature 4 for all features changed the most, and the change in correlation can be observed in Feature 1, which shares the same period as the anomalous Feature 4.

5.4.2. SWaT data

The SWaT benchmark dataset provides a description of the attack, which can be used to validate its interpretation on real-world industrial time-series datasets. The SWaT dataset is a multivariate time-series of flow rates and water levels in a hydropower system. As illustrated in Fig. 9, when an anomaly occurs in one of the water valves of a hydropower system, there is typically a cascade of pressure, flow, and velocity changes in the upstream and downstream pipes. Fig. 10 presents an interpretation of the normal and anomalous sections of the SWaT data. The change in correlation in the normal segment is minimal, but the change in correlation in the anomaly segment is significant. The anomaly shown in Fig. 10 was expected to be a reverse osmosis (RO) shutdown due to a change in the AIT-504 value, based on the description provided by the SWaT dataset. Even though Reverse Osmosis (RO) was not terminated, we can assume that it would have been affected by the change in AIT-504, which in turn

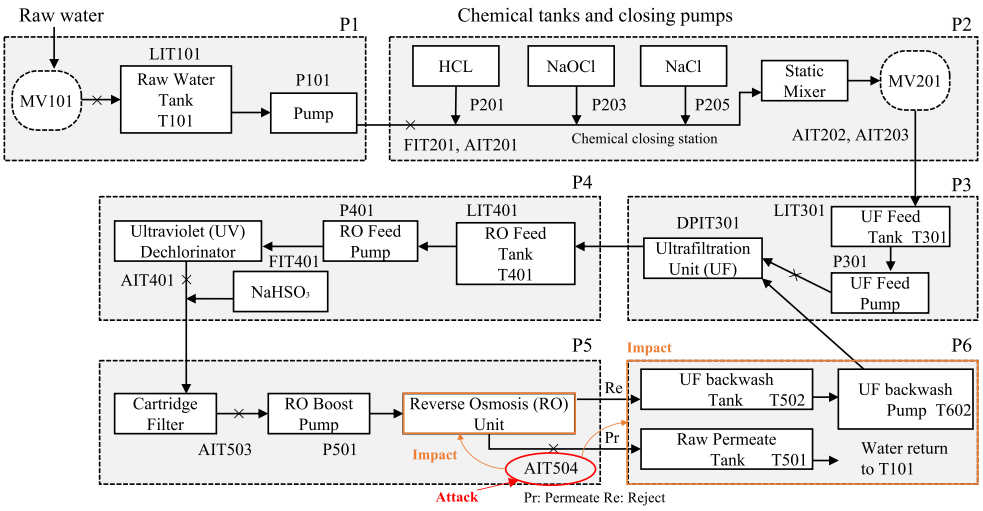


Fig. 9. Physical water treatment process in SWaT.

Fig. 10. Visualize training and test data results from SWaT data set.

would have affected the variables in P6. By examining the change in correlation, we can see that PIT501 and PIT502, belonging to P6, were the variables that changed significantly in correlation with AIT504, which was directly attacked. Additionally, we can observe that the correlation of AIT504 and PIT502 for all features changed the most. This demonstrates that it is possible to interpret anomalies using the proposed anomaly interpretation method.

6. Conclusion

Time-series anomaly detection is crucial for maintaining the integrity and security of time-series data from sensors and systems, preventing potential accidents and economic losses. Anomaly detection in multivariate systems is a challenging task, particularly for systems dealing with multivariate time-series data, as it requires considering temporal dependencies and relationships between numerous variables. Thus, it is essential to account for the correlations between variables to better detect anomalies in multivariate time series data. However, existing time series anomaly detection studies learn temporal dependencies that are independent of the variables. In this study, we aimed to perform anomaly detection that can simultaneously consider temporal dependencies and interrelationships between variables and propose an interpretation module for anomalies through changes in correlations between variables.

In this paper, we propose the Variable Temporal Transformer (VTT), an anomaly detection method for multivariate time-series that effectively captures both temporal dependencies and inter-variable relationships. The VTT utilizes a newly designed Variable self-attention mechanism, an alteration of the existing Temporal self-attention, to grasp temporal dependencies, along with a novel Variable self-attention to explicitly map the correlations among variables. Functioning as an unsupervised learning method, it does not employ anomalies in data preprocessing and learning, but instead calculates reconstruction values via a model that learns the normal distribution of the training data,

labeling data points with substantial reconstruction errors as anomalies. To circumvent the issue of inflated performance estimation using the Point Adjustment F1 score, we employ the recently introduced $F1_{PA\%K}$ to compare VTT with seven established anomaly detection methods across four publicly available datasets. Experimental results show that our proposed VTT outperforms baseline methods that do not consider the correlation between variables. We also confirm that VTT performs better in an ablation study using a vanilla transformer that does not use variable self-attention. This confirms that considering the correlation between variables in a multivariate outlier detection task plays a key role in better detecting anomalies. Moreover, we propose a methodology to deduce the origins of anomalies by comparing attention maps extracted from original data with anomalies and those derived from the reconstructed data. This approach allows identification of variables that have undergone significant weight changes and those that have altered their correlations. We illustrate this interpretability capacity qualitatively using both synthetic and real-world industrial data. In both experiments, anomalies in one variable affected other highly correlated variables, and the effects were qualitatively confirmed by attention scores.

Transformers' self-attention mechanism computes the relationship between each pair of input tokens. Since the proposed VTT utilizes two attention modules, the complexity of the model is affected by the number of variables and the size of the sliding window: for an input sequence of length N and number of variables V, it incurs $O(N^2+V^2)$ complexity. This makes training and inference on very long sequences computationally expensive and memory intensive. In the future, we plan to work on more efficient ways to better capture temporal dependencies and correlations between variables while addressing complexity issues by modifying input shapes like patch.

CRediT authorship contribution statement

Hyeongwon Kang: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Pilsung Kang:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A2C2005455). This work was also supported by Institute of Information and communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00471, Development of Autonomous Control Technology for Error-Free Information Infrastructure Based on Modeling and Optimization).

References

[1] L. Marti, N. Sanchez-Pi, J.M. Molina, A.C.B. Garcia, Anomaly detection based on sensor data in petroleum industry applications, Sensors 15 (2) (2015) 2774–2797, http://dx.doi.org/10.3390/s150202774, URL https://www.mdpi.com/1424-8220/15/2/2774.

[2] L.Y. Li, J.C. Yan, H.Y. Wang, Y.H. Jin, Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder, IEEE Trans Neural Netw. Learn. Syst. 32 (3) (2021) 1177–1191, http://dx.doi.org/10.1109/Tnnls.2020.2980749, URL https://ieeexplore.ieee.org/abstract/document/9064715.

[3] Z.J. Niu, K. Yu, X.F. Wu, LSTM-based VAE-GAN for time-series anomaly detection, Sensors 20 (13) (2020) http://dx.doi.org/10.3390/s20133738, URL https://www.mdpi.com/1424-8220/20/13/3738.

[4] G. Electric, GE predix: The industrial internet platform, GE Predix (2016) URL https://www.ge.com/digital/iiot-platform.

[5] Bosch, Anomaly detection in time series sensor data for predictive maintenance, 2016, URL https://www.bosch.com/stories/anomaly-detection-predictive-maintenance/.

[6] L.D. Xu, W. He, S.C. Li, Internet of things in industries: A survey, IEEE Trans. Ind. Inform. 10 (4) (2014) 2233–2243, http://dx.doi.org/10.1109/Tii.2014.2300753, URL https://ieeexplore.ieee.org/abstract/document/6714496.

[7] J. Gao, X. Song, Q. Wen, P. Wang, L. Sun, H. Xu, Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks, 2020, arXiv preprint arXiv:2002.09545.

[8] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I.A.T. Hashem, A. Siddiqa, I. Yaqoob, Big IoT data analytics: Architecture, opportunities, and open research challenges, IEEE Access 5 (2017) 5247–5261, http://dx.doi.org/10.1109/Access.2017.2689040, URL https://ieeexplore.ieee.org/document/7888916.

[9] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M.A. Zuluaga, USAD : UnSupervised anomaly detection on multivariate time series, in: Kdd '20: Proceedings of the 26th Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404, http://dx.doi.org/10.1145/3394486.3403392, URL https://dl.acm.org/doi/10.1145/3394486.3403392.

[10] A. Geiger, D.Y. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, TadGAN: Time series anomaly detection using generative adversarial networks, in: 2020 IEEE International Conference on Big Data, Big Data, 2020, pp. 33–43, http://dx.doi.org/10.1109/BigData50022.2020.9378139, URL https://ieeexplore.ieee.org/abstract/document/9378139.

[11] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, LOF: Identifying density-based local outliers, SIGMOD Rec. 29 (2) (2000) 93–104, http://dx.doi.org/10.1145/335191.335388, URL https://dl.acm.org/doi/10.1145/335191.335388.

[12] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, vol. 12, 2000, pp. 582–588, URL https://proceedings.neurips.cc/paper_files/paper/1999/hash/8725fb777f25776ffa9076e44fcfd776-Abstract.html.

[13] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation forest, in: Icdm 2008: Eighth Ieee International Conference on Data Mining, Proceedings, 2008, pp. 413–+, http://dx.doi.org/10.1109/Icdm.2008.17, URL https://ieeexplore.ieee.org/abstract/document/4781136.

[14] M. Braei, S. Wagner, Anomaly detection in univariate time-series: A survey on the state-of-the-art, 2020, arXiv preprint arXiv:2004.00433.

[15] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, Long short term memory networks for anomaly detection in time series, in: ESANN, Vol. 2015, 2015, p. 89.

[16] M. Munir, S.A. Siddiqui, A. Dengel, S. Ahmed, DeepAnT: A deep learning approach for unsupervised anomaly detection in time series, IEEE Access 7 (2019) 1991–2005, http://dx.doi.org/10.1109/Access.2018.2886457, URL https://ieeexplore.ieee.org/abstract/document/8581424.

[17] M. Munir, S.A. Siddiqui, M.A. Chattha, A. Dengel, S. Ahmed, FuseAD: Unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models, Sensors 19 (11) (2019) http://dx.doi.org/10.3390/s19112451, URL https://www.mdpi.com/1424-8220/19/11/2451.

[18] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, in: Kdd'18: Proceedings of the 24th Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, 2018, pp. 387–395, http://dx.doi.org/10.1145/3219819.3219845, URL https://dl.acm.org/doi/abs/10.1145/3219819.3219845.

[19] Y. Su, Y.J. Zhao, C.H. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Kdd'19: Proceedings of the 25th Acm Sigkdd International Conferencce on Knowledge Discovery and Data Mining, 2019, pp. 2828–2837, http://dx.doi.org/10.1145/3292500.3330672, URL https://dl.acm.org/doi/abs/10.1145/3292500.3330672.

[20] B. Z, S.H. Liu, B. Hooi, X.Q. Cheng, J. Ye, BeatGAN: Anomalous rhythm detection using adversarially generated time series, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 4433–4439, URL https://dl.acm.org/doi/abs/10.5555/3367471.3367658.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, Nips 2017, 2017, URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[22] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, C. Schmid, ViViT: A video vision transformer, in: 2021 IEEE/Cvf International Conference on Computer Vision, ICCV 2021, 2021, pp. 6816–6826, http://dx.doi.org/10.1109/Iccv48922.2021.00676, URL https://openaccess.thecvf.com/content/ICCV2021/html/Arnab_ViViT_A_Video_Vision_Transformer_ICCV_2021_paper.html?ref=https://githubhelp.com.

[23] S. Tuli, G. Casale, N.R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, 2022, arXiv preprint arXiv:2201.07284.

[24] J. Xu, H. Wu, J. Wang, M. Long, Anomaly transformer: Time series anomaly detection with association discrepancy, 2021, arXiv preprint arXiv:2110.02642.

[25] J. Kim, H. Kang, P. Kang, Time-series anomaly detection with stacked transformer representations and 1D convolutional network, Eng. Appl. Artif. Intell. 120 (2023) http://dx.doi.org/10.1016/j.engappai.2023.105964, URL https://www.sciencedirect.com/science/article/pii/S0952197623001483.

[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735, URL https://www.bioinf.jku.at/publications/older/2604.pdf.

[27] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, 2016, arXiv preprint arXiv:1607.00148.

[28] D. Park, Y. Hoshi, C.C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robot. Autom. Lett. 3 (3) (2018) 1544–1551.

[29] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y.Q. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) URL https://dl.acm.org/doi/abs/10.5555/3455716.3455856.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[33] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, Conformer: Convolution-augmented transformer for speech recognition, 2020, arXiv preprint arXiv:2005.08100.

[34] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2020, pp. 11106–11115.

[35] H.X. Wu, J.H. Xu, J.M. Wang, M.S. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, in: Advances in Neural Information Processing Systems, Vol. 34, Neurips 2021, 2021, URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

[36] T. Zhou, Z.Q. Ma, Q.S. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International Conference on Machine Learning, Vol. 162, 2022, URL https://proceedings.mlr.press/v162/zhou22g.html.

[37] A.v.d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, 2016, arXiv preprint arXiv:1609.03499.

[38] S. Kim, K. Choi, H.S. Choi, B. Lee, S. Yoon, Towards a rigorous evaluation of time-series anomaly detection, in: Thirty-Sixth AAAI Conference on Artificial Intelligence / Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence / Twelveth Symposium on Educational Advances in Artificial Intelligence, 2022, pp. 7194–7201, URL https://ojs.aaai.org/index.php/AAAI/article/view/20680.

[39] A.P. Mathur, 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, 2016, pp. 31–36, http://dx.doi.org/10.1109/cyswater.2016.7469060, URL https://ieeexplore.ieee.org/abstract/document/7469060.

[40] K. Doshi, S. Abudalou, Y. Yilmaz, Reward once, penalize once: Rectifying time series anomaly detection, in: 2022 International Joint Conference on Neural Networks, Ijcnn, 2022, http://dx.doi.org/10.1109/Ijcnn55064.2022.9891913, URL https://ieeexplore.ieee.org/abstract/document/9891913.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z.M. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J.J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, Vol. 32, Nips 2019, 2019, URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

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