Representation Learning Using a Multi-Branch Transformer for Industrial Time Series Anomaly Detection

Ruichuan Zhang*
University of Illinois
Urbana-Champaign
United States
rzhang65@illinois.edu

Fangzhou Cheng
AWS AI Labs
United States
fzc@amazon.com

Aparna Pandey
AWS AI Labs
United States
apapan@amazon.com

ABSTRACT

In recent years, due to the rapid expansion of the Industrial Internet of Things (IIoT), substantial amounts of high-dimensional industrial time series data have been generated. Anomaly detection in such industrial time series data is a challenging task due to complex temporal dynamics. In this paper, we propose multi-branch transformer with Gaussian mixture model (MBTGMM), a novel transformer-based framework to address some of these challenges. In the framework, normal representations are learned with a multi-branch transformer architecture that comprises of a convolution branch and a multi-head attention branch in order to learn both short- and long-term temporal dependencies in the time series data. These representations are then fed into a Gaussian mixture model for density estimation and anomaly detection task. Experimental results on public industrial datasets show the effectiveness of our proposed framework, and the ablation studies clearly demonstrate the efficacy of our design choices.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection; Unsupervised learning**.

KEYWORDS

Anomaly detection, Unsupervised learning, Time series, Transformer, Representation learning, Deep learning

ACM Reference Format:
Ruichuan Zhang, Fangzhou Cheng, and Aparna Pandey. 2022. Representation Learning Using a Multi-Branch Transformer for Industrial Time Series Anomaly Detection. In *MiLeTS '22: 8th SIGKDD Workshop on Mining and Learning from Time Series (MiLeTS '22).* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

The rapid advancements of Internet of Things technologies have enabled the collection of a vast amount of industrial time series data. This leads to a need for mining this data for valuable downstream tasks such as detecting anomalies in a timely fashion from these data to prevent catastrophic failures and reduce unplanned system downtime, especially for critical industrial systems, such as manufacturing plants [1], industrial robots [2], spacecrafts [3], and water systems [4, 5].

Anomaly detection for industrial time series data aims to find the individual sample or a sequence of samples that deviates from normal samples or sequences. It is challenging to develop an effective anomaly detection method for industrial systems due to the lack of labeled anomalies and poor data quality. Therefore, supervised methods are, in general, difficult to apply to this task. Classical unsupervised approaches such as principal component analysis [6], k-means [7], and one-class support vector machines [8], have been applied to deal with this task. However, these methods cannot capture temporal and multi-variate dependencies simultaneously without using custom hand-engineered features, which make them less effective. Another line of work for anomaly detection focuses on density estimation methods [9, 10]. These methods model the distribution of normal samples during training and flag samples located in low-density regions as anomalies. More recently, normalizing flow [11] has been proposed for anomaly detection by producing tractable distributions.

In recent years, deep learning-based methods have demonstrated promising results in this area. These methods can mainly be divided into two categories: reconstruction-based [12, 13] methods and prediction-based methods [3, 14]. Reconstruction-based methods aim to learn the representations of normal data and detect anomalies based on the reconstruction error. On the other hand, prediction-based methods are good at capturing temporal information by using historical data to predict future values. The error between real and predicted value is used to detect anomalies. More recently, transformer-based architecture [15] has enjoyed widespread success in natural language processing [16], computer vision [17], and audio signal processing [18]. Transformer-based models are usually trained in a self-supervised fashion on unlabeled datasets, and then the trained models are used to generate representations for downstream tasks or fine tuned on a typically much smaller, task-specific dataset [16–18]. The multi-head self-attention mechanism provides the ability to capture long-term temporal dependencies and improves computational efficiency and scalability compared to models based on recurrent neural networks. Recently, transformer-based methods are also adopted for anomaly detection tasks [19, 20].

*Work done during an Amazon internship.

MiLeTS '22, August 15th, 2022, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

Motivated by the success of transformer in other domains and availability of largely untapped, unlabelled data in industrial domain, we developed MBTGMM, a novel multi-branch transformer-based framework for anomaly detection of time series data. In the framework, a multi-branch transformer encoder block is designed to capture both long- and short-term temporal dependencies, where a 1-D convolution branch aims to model the short-term local temporal dependencies, and a multi-head self-attention branch aims to focus on the long-term dependencies. The framework uses masked learning strategies to learn representations of normal time series data, and the learned representations are then fed into a Gaussian mixture model (GMM) for learning the distribution of normal representations. The anomalies can then be detected if their representations fall into low density regions in the distribution.

To summarize, the main contributions of our work are:

- We propose MBTGMM, a novel multi-branch transformer-based framework for industrial time series anomaly detection.
- In MBTGMM, a 1-D convolutional branch and a multi-head self-attention branch are designed to capture short-term and long-term temporal dependencies respectively.
- We conduct experiments on three publicly available industrial time series datasets and the results show that our proposed framework is competitive with state-of-the-art methods.
- We design ablation studies that demonstrate the efficacy of our design choices.

2 BACKGROUND

2.1 Unsupervised Anomaly Detection of Time Series Data

Traditional unsupervised methods for time series anomaly detection can mainly be divided into prediction-based and reconstruction-based methods. Prediction-based methods use historical data to predict future values based on the learned temporal dependencies. Anomalies are associated with high prediction errors. Examples of prediction-based methods include the auto regressive integrated moving average (ARIMA) [14] and long-short term memory (LSTM) recurrent neural networks [3]. LSTM models require sequential propagation of all samples through the network in order to incorporate long-term temporal dependencies. This results in computational inefficiencies and can lead to exploding gradients [21]. Another category of methods uses an encoder-decoder architecture where the encoder learns a low-dimensional representation of the time series data and the decoder reconstructs the input using the low-dimensional representation. Anomalies are associated with high reconstruction errors. Examples of reconstruction-based methods include LSTM autoencoder (LSTM-AE) [22], convolutional LSTM-AE [23], VAE [24, 25], and LSTM-VAE [13]. OmniAnomaly [26] proposed a stochastic recurrent neural network with a planar normalizing flow to enhance the capability of modeling multivariate dependencies. USAD [27] is recently proposed to reconstruct the normal samples in the training set by incorporating adversarial training with autoencoders.

Recent methods have further extended their abilities to capture the temporal dependencies between samples and multi-variate dependencies among features. For example, [28] extends the one-class classification objective that considers multiple hyperspheres obtained from a hierarchical process to capture multi-scale temporal dynamics. [29] and [30] use graph neural networks to learn relationships between features and then use the learned structure and latent representations for anomaly detection task. [31] introduces the cycle-consistent generative adversarial network (GAN) architecture in which generators are used for time series reconstruction, and then the anomaly scores are calculated with both the reconstruction errors from the generators and the prediction errors from the discriminator.

2.2 Transformer-based Anomaly Detection

Transformer architecture utilizes the attention mechanism in an encoder-decoder structure [15] foregoing recurrence and convolutions and thereby, resulting in a more efficient implementation. In natural language processing, [16] employs masked language modeling and next sentence prediction as the two self-supervised tasks for pretraining BERT. Variants of BERT, such as XLNet [32], RoBERTa [33], ELECTRA [34] and ALBERT [35], learn the representations with encoders, while other models learn that based on decoders, such as GPT [36] and BART [37]. In computer vision, pre-trained vision transformers (ViT) [17] treats an image as a sequence of patches and processes it with a standard transformer encoder. Other examples include Video ViT [38], Swin Transformer [39], Focal Transformer [40], and Pyramid ViT [41]. In audio signal processing, [18] uses masked modeling techniques with contrastive learning to learn representations at individual audio signal level.

Recently, transformer-based methods have been adopted for anomaly detection tasks for many modalities of data including images, videos [42–46] and time series [19, 20, 47, 48]. The representations learned by transformer-based models can be directly used in downstream anomaly detection task [44]. They can also be used to reconstruct the input data [20, 46–48] or predict the future data [42], and compute the reconstruction or prediction error as the anomaly scores. Several transformer-based models are designed for time series data, which combine transformers with VAEs [47, 49], GANs [20], and graph-based learning architecture [46], for better representation learning and anomaly detection.

3 METHODOLOGY

3.1 Problem Formulation

We consider the problem of anomaly detection for a time series, denoted as $\mathbf{X} = [x_1, x_2, ..., x_t, ..., x_L]$, where L is the total length of the time series. At timestamp t, $x_t \in \mathbb{R}^m, \forall t$, where $m = 1$ for univariate time series data and $m > 1$ for multivariate time series data.

Detecting anomalous timestamps: Given an input time series $\mathbf{X}$, at timestamp t, we aim to predict an anomaly score $s(t)$. If $s(t)$ exceeds a predetermined threshold T, then the data point is considered anomalous.

Detecting anomalous time series: Given an input time series $\mathbf{X}$, we aim to predict an anomaly score S for the entire time series, based

on $s(t)$ of individual time stamps. If S exceeds a predetermined threshold T, then the time series is detected as anomaly.

3.2 Overall Architecture of Proposed MBTGMM

To solve the anomaly detection task illustrated above, we propose MBTGMM, an anomaly detection framework using the representations learned by a multi-branch transformer with a GMM for density estimation. Fig. 1 shows the overall architecture of the proposed MBTGMM. Firstly, a multi-branch transformer-based model is used to learn the representations of input time series. The multi-branch transformer encoder is designed to capture both short-term and long-term temporal dependencies in the time series while learning the representations. Secondly, the learned representations are fed into a GMM, which detects anomalies by assigning higher scores to samples that in low-density regions. GMM is an efficient and flexible model that helps approximate mixture distributions [9]. Thus, it is suitable to be deployed in systems with multiple underlying sub-populations that require fast processing such as industrial applications. In the following subsections, we will introduce each part of our proposed MBTGMM in detail.

3.3 Multi-branch Transformer for Time Series Representation Learning

MBTGMM comprises of an encoder-only structure ϕ_e, with stacked multi-branch transformer blocks, that learns the representation of a time series. Given an input window $\mathbf{w}$ which consists of l timestamps, the input linear projection layer linearly projects the input window to a feature space of the same dimension as the transformer encoder. To make transformer encoder aware of the order of the input data to better capture temporal dependencies, we add positional encodings to the output of the input layer. We use the sinusoidal encodings [15] as the initial positional encodings, while making the encodings learnable [17].

The encoder of the proposed MBTGMM consists of a stack of N identical blocks, each with a multi-head self-attention branch in parallel with a 1-D convolution branch, and a fully connected feed-forward layer. A residual connection and a layer normalization are applied to each layer. The design of a multi branch transformer encoder can encourage the multi-head attention to focus on the global, long-term dependencies and the 1-D convolution to focus on the local, short-term dependencies [50]. The input to the multi-branch transformer encoder of dimension d is split into two equal-size parts, with each passing through the 1-D convolution branch and the multi-head self-attention branch separately. This can save the computational cost by 2×. Let the input to the multi-head self-attention branch be denoted as $e^1 \in \mathbb{R}^{l \times d/2}$. The multi-head self-attention branch first projects the input to three intermediate representations, query (Q), key (K), and value (V) of d_k, d_k, and d_v dimensions respectively:

$$Q_i = e_i^1 W_i^Q, K_i = e_i^1 W_i^K, V_i = e_i^1 W_i^V \tag{1}$$

where e_i^1 is the input to i-th head; $W_i^Q \in \mathbb{R}^{d/2 \times d_k}$, $W_i^K \in \mathbb{R}^{d/2 \times d_k}$, and $W_i^V \in \mathbb{R}^{d/2 \times d_v}$ are the trainable weight matrices of i-th head. The self-attention captures the dependencies amongst these positions, through a scaled-dot product attention of Q_i, K_i, and V_i:

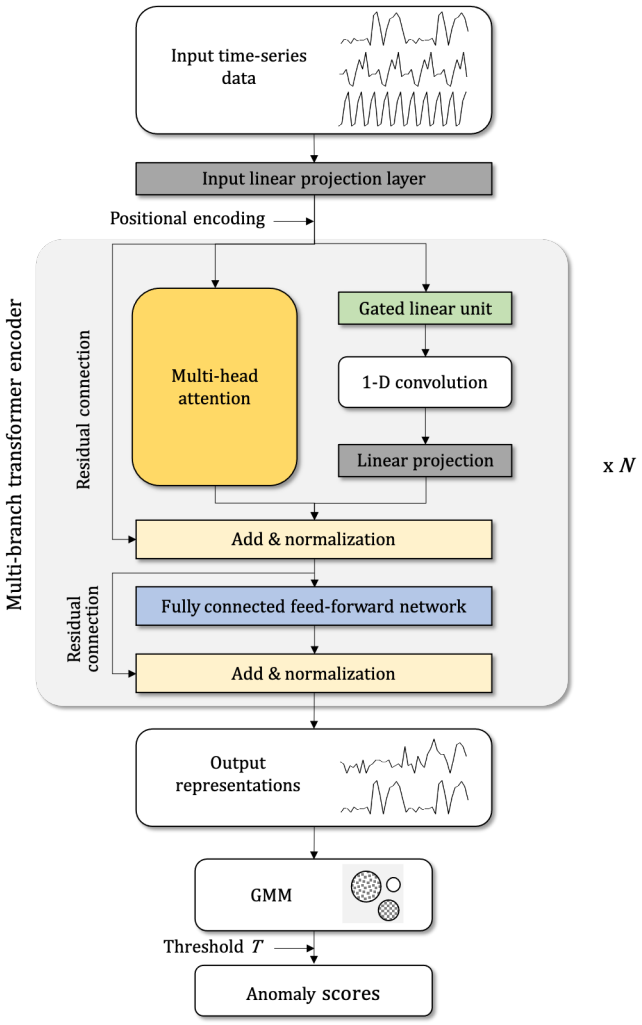


Figure 1: Overall architecture of proposed MBTGMM for anomaly detection.

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i \tag{2}$$

The multi-head design further enables computing such representations and extracting information from different feature spaces jointly:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^M \tag{3}$$

where $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$, h is the total number of heads, and $W^M \in \mathbb{R}^{hd_v \times d/2}$.

A 1-D convolution branch is placed in parallel with the multi-head self-attention branch to enhance the capability of the model to capture the short-term temporal dependencies. The second half of input to the encoder, $e^2 \in \mathbb{R}^{l \times d/2}$, is fed into this branch. A gated linear unit is first applied followed by a 1-D convolution layer and a linear projection layer. We denote the output of this branch

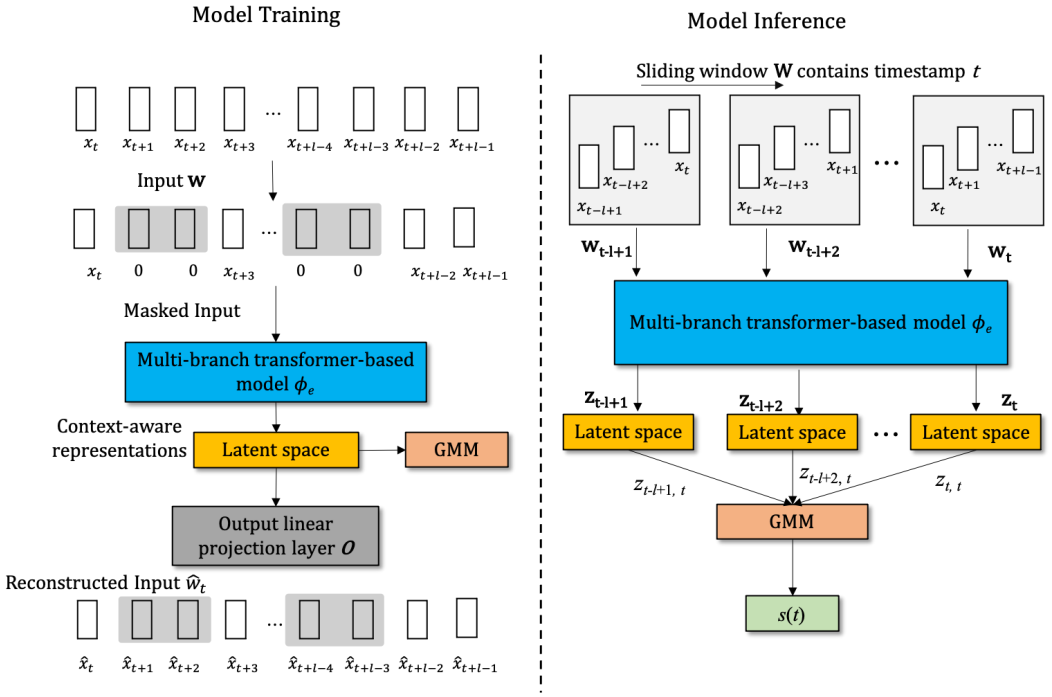


Figure 2: Training and inference process of MBTGMM using a masked representation learning strategy.

as Conv1D. The two branches merge together by the following fully connected feed-forward layer, which consists of two linear transformations and a ReLU activation.

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{4}$$

where $x = \mathrm{Concat}(\mathrm{MultiHead}, \mathrm{Conv1D})$.

With the designed multi-branch transformer, our proposed MBTGMM is able to learn the context-aware representations from both short-term and long-term temporal dependencies during training. The learned representations are then fed into a GMM for the downstream anomaly detection task.

3.4 Model Training and Inference

We train the proposed MBTGMM to learn context-aware representations of the input time series using a masking strategy, as shown in Fig. 2.

We randomly select input time stamps and mask them to be all zeros by applying element-wise multiplication with a binary mask [16, 18]. The mask is created independently for each training window and epoch. On average, a proportion r of the entire time-series data are masked, and we chose $r = 0.15$ empirically for the experiments. Each masked segment has a length that follows a geometric distribution with a mean length l_m and is succeeded by an unmasked segment of mean length l_u [51], which is calculated by:

$$l_u = \frac{1-r}{r} l_m. \tag{5}$$

Instead of using multi-branch transformer decoder, we use an output linear projection layer O to project the representations $z \in \mathbb{R}^{l \times d}$ generated by multi-branch transformer encoder to the reconstructed input as $\hat{x}$:

$$\hat{x} = zW_o + b_o \tag{6}$$

where $W_o \in \mathbb{R}^{d \times m}$ and $b_o \in \mathbb{R}^{l \times m}$. We used the Adam optimizer with a warm-up schedule [15] during training. We minimize the mean square error (MSE) loss $\mathcal{L}$ between the reconstructed and the original samples, and only the samples at masked timestamps $t \in M$ are considered when calculating $\mathcal{L}$.

$$\mathcal{L} = \frac{1}{|M|} \sum_{t \in M} (x_t - \hat{x}_t)^2 \tag{7}$$

where |M| denotes the number of masked timestamps in M. Once we obtain the learned representations z using the trained multi-branch transformer, GMM is applied to fit the distribution of the representations. Assuming there are K mixture components in GMM, the mixture probability p_i, mean μ_i, and variance σ_i of a mixture component i can be estimated using the expectation–maximization (EM) algorithm, as per Eqs. 8 and 9.

$$p_i = \frac{\sum_{j=1}^{l} q_{j,i}}{l}, \forall i \in [K] \tag{8}$$

$$\mu_i, \sigma_i = f_{EM}([z_j, q_{j,i}]_{j=1}^{l}), \forall i \in [K] \tag{9}$$

where $q_{j,i}$ is the probability that representation point j belongs to the mixture component i, and f_{EM} is the EM estimator. In our experiments, multiple GMMs with different number of components

K are estimated, and the model with the lowest Bayesian information criterion (BIC) is selected for the anomaly detection task. The entire training procedure is summarized in **Algorithm 1**.

Algorithm 1 Training of MBTGMM for time series anomaly detection

Require:
Training time series data X;
Multi-branch transformer-based model ϕ_e;
Output linear projection layer O;
Iteration limit E;
Gaussian mixture model G

Initialize weights of ϕ_e, O, G
$n \leftarrow 0$
while $n <$ E
    Create masked time-series data X' from X with masked time stamps M;
    $\hat{X} = O(\phi_e(X'))$
    $\mathcal{L} = \frac{1}{|M|} \sum_{t \in M} (x_t - \hat{x}_t)^2$;
    Update parameters of ϕ_e, O using $\mathcal{L}$
$z = \phi_e(X)$
Fit multiple GMMs with different number of components using z, and select the best G with the lowest BIC.

During model inference, to get the anomaly score for timestamp t, we first find all the windows $\mathbf{W} = [\mathbf{w_{t-l+1}}, \mathbf{w_{t-l+2}}, ..., \mathbf{w_j}, ..., \mathbf{w_t}]$ that contain timestamp t, where $\mathbf{w_j} = [x_j, x_{j+1}, ..., x_{j+l-1}]$. Then we generate the representations $\mathbf{Z} = [\mathbf{z_{t-l+1}}, \mathbf{z_{t-l+2}}, ..., \mathbf{z_j}, ..., \mathbf{z_t}]$ using the multi-branch transformer-based model ϕ_e, where $\mathbf{z_j} = [z_{j,j}, z_{j,j+1}, ..., z_{j,t}, ..., z_{j,j+l-1}]$; and $z_{j,t}$ is the learned representation at timestamp t in $\mathbf{z_j}$.

$$\mathbf{z_j} = \phi_e(\mathbf{w_j}); \tag{10}$$

The fitted GMM G is used to calculate negative log likelihood for $z_{j,t}$ in each representation $\mathbf{z_j} \in \mathbf{Z}$ to get the anomaly score for sample at timestamp t :

$$G(z_{j,t}|\mu_i, \sigma_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-(z_{j,t}-\mu_i)^2/2\sigma_i^2} \tag{11}$$

$$s(t) = \frac{-1}{|\mathbf{Z}|} \sum_{z_{j,t} \in \mathbf{Z}} log(\sum_{i=1}^{K} p_i G(z_{j,t}|\mu_i, \sigma_i)) \tag{12}$$

where $|Z|$ denotes the number of learned representations in $\mathbf{Z}$. To detect anomalous time series, we further compute the median anomaly score of all samples in the time series as its anomaly score S.

4 EXPERIMENTS

4.1 Experimental Setups

Datasets: Three industrial datasets are used in the experiments to evaluate our proposed MBTGMM. Mars Science Laboratory (MSL) and Soil Moisture Active Passive (SMAP) are two datasets that contain telemetry anomaly data provided by NASA [3]. The anomaly data in these two datasets record the unexpected events during post launch operations of spacecrafts. Another dataset was recently released by the Prognostics and Health Management Society data challenge 2021 (PHM21) [52], which contains both normal and anomalous time series in a fuse quality control pipeline used for industrial manufacturing production lines. Each time series contains several steps needed to accomplish the fuse quality examination task. Anomalies are associated with one or more steps in the anomalous time series. For each dataset, we normalize both the training and test data using the mean and standard deviation of the training data. A sliding window is applied on the normalized data to generate the input data to train the models. For MSL and SMAP, we detect anomalous timestamps. For PHM21 dataset, we detect anomalous time series. A summary of the three aforementioned datasets are provided in Table 1.

Table 1: Dataset description.

Dataset	SMAP	MSL	PHM21
Number of subsets	55	26	99
Number of features	25	55	247
Size of training data	2556	2160	367920
Size of testing data	8071	2731	579850
Anomaly %	12.8	10.5	29.3
Anomaly type	time stamp	time stamp	time series

Baseline methods: We compare the anomaly detection results of MBTGMM with popular and state-of-the-art methods, including isolation forests (IF) [53], autoencoders (AE), MAD-GAN [54], LSTM [55], LSTM-AE [22], LSTM-VAE [13], DAGMM [12], MTAD-GAT [56], OmniAnomaly [26], USAD [27], and GTA [30]. We also compare with the winning teams of PHM21 challenge who use supervised learning-based (HIRUTEK) [57] and rule-based methods (LDM) [58] on PHM21 dataset. In our implementation of MBTGMM, we use input window size l of 50; the training batch size is 32; the number of heads h is 4; the kernel size of 1-D convolutional branch is chosen as 3; and the input dimension d of multi-branch transformer model ϕ_e is selected as 16 for SMAP and MSL, and 128 for PHM21 dataset. Our proposed MBTGMM and some of the baselines, including LSTM, LSTM-AE, are implemented using Pytorch version 1.6.0 with CUDA 10.1, and we directly take the results of the rest of the baselines from literature [27, 30]. The experiments are performed on an Amazon p3.2xlarge EC2 with one NVIDIA Tesla P100 GPU.

4.2 Evaluation Metrics

We adopt the standard Precision, Recall and F1-Score (F1) as the evaluation metrics to evaluate and compare the performance of MBTGMM and other baseline methods. For a fair comparison with state-of-the-art methods on SMAP and MSL, we adopted the point-adjusted approach proposed in [25] to calculate the performance metrics for those two datasets. In this metric, all the samples in one anomalous event are adjusted to true positives, if at least one sample in that event is originally correctly detected as true positive. The negative samples remain the same and no adjustment is applied to them. After that, the point-adjusted precision, recall, and F1 are calculated based on the adjusted predictions. We search over all possible thresholds for the theoretically best F1 during test, and

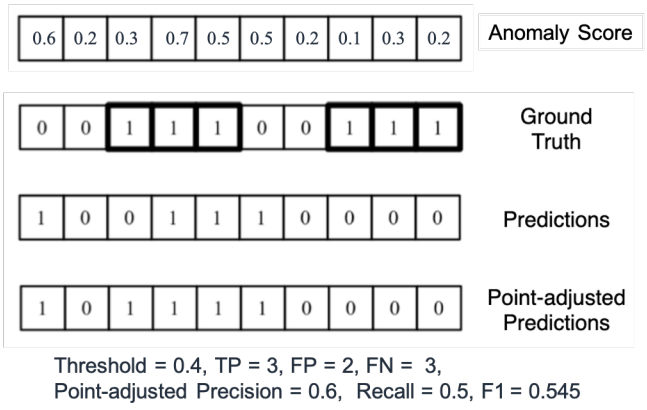


Figure 3: An illustrative example of calculating point-adjusted metrics

Table 2: Experimental results on SMAP and MSL datasets.

	SMAP			MSL		
Method	Prec.	Rec.	F_{best}	Prec.	Rec.	F_{best}
IF	0.442	0.511	0.474	0.568	0.674	0.617
AE	0.722	0.980	0.831	0.854	0.975	0.910
MAD-GAN	0.805	0.821	0.813	0.852	0.899	0.875
LSTM	0.777	0.997	0.873	0.852	0.977	0.910
LSTM-AE	0.795	0.991	0.882	0.853	0.977	0.911
LSTM-VAE	0.716	0.988	0.830	0.860	0.976	0.914
DAGMM	0.633	0.988	0.775	0.756	0.980	0.854
MTAD-GAT	0.891	0.912	0.901	0.875	0.944	0.908
OmniAnomaly	0.759	0.976	0.854	0.914	0.889	0.901
USAD	0.770	0.983	0.864	0.881	0.979	0.927
GTA	0.891	0.918	0.904	0.910	0.912	0.911
MBTGMM	0.916	1.000	**0.931**	0.911	0.990	**0.942**

Table 3: Anomaly detection results of proposed MBTGMM and baselines on PHM21.

	Methods	Precision	Recall	F_{best}
Unsupervised	LSTM	0.434	0.793	0.561
	LSTM-AE	0.434	0.793	0.561
	MBTGMM	1.000	0.793	0.885
Supervised	HIRUTEK	1.000	0.897	0.945
	LDM	0.862	0.893	0.877

denote it as F_{best}. We also report the corresponding precision and recall for the F_{best}. An illustrative example of how to calculate point-adjusted metrics is provided in Fig. 3.

4.3 Experimental Results

Table 2 shows the results of our proposed methods and baselines on SMAP and MSL datasets. The table clearly shows MBTGMM achieves the highest F_{best} among all baseline methods. More concretely, our proposed method improves the F_{best} by 2.7% and 1.5% on SMAP and MSL respectively, compared to the second best in the table. Moreover, our method achieves 1 and 0.99 recall on SMAP and MSL respectively, which are also the highest among all baselines. This indicates MBTGMM is able to capture most of the anomalous events, which is critical for industrial anomaly detection tasks. These results demonstrate the effectiveness of our proposed method compared with other methods.

IF [53], MAD-GAN [54], and DAGMM [12] have the lowest F_{best} on both datasets. The reason is that they mainly model the dependencies in between features while are weak at modeling temporal dependencies [26]. In our proposed MBTGMM, the design of multi-branch transformer is able to capture both short- and long-term temporal dependencies, which makes our methods better in detecting temporal anomalies.

LSTM [55] is a prediction-based model and detects anomalies based on the residuals of predictions and actual values. However, for industrial datasets, there are many extraneous factors that make the datasets unpredictable, which making LSTM less effective [3]. In MBTGMM, the combination of GMM and multi-branch transformer are able to learn the normal patterns of the representations, as well as capture short- and long-term temporal dependencies.

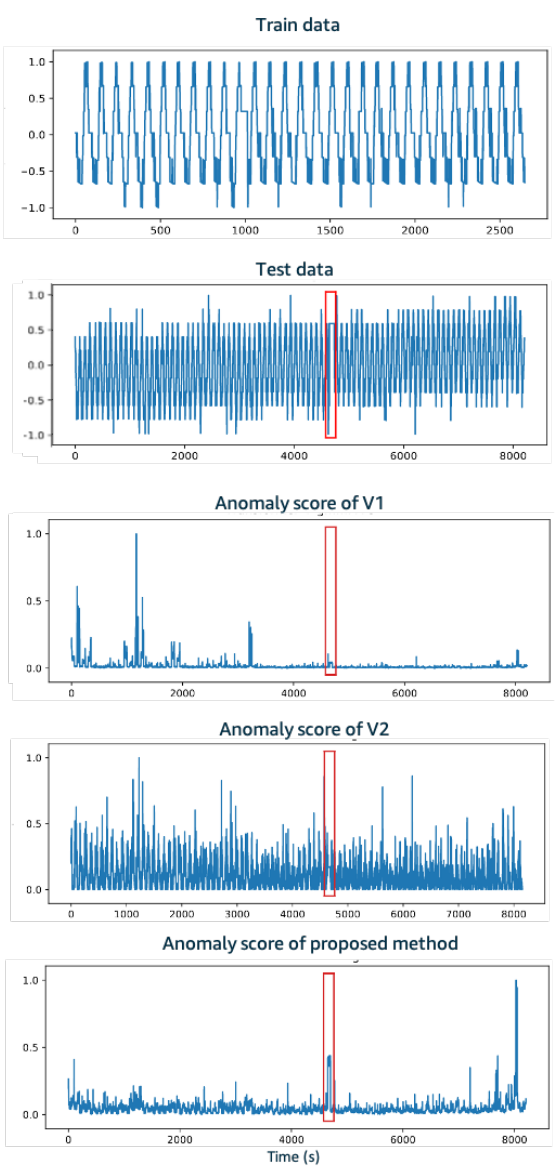
LSTM-VAE [13] utilizes a VAE to learn low-dimensional representations by extracting local information from input window, and then uses LSTM for sequential modeling. However, LSTMs are often slow and inefficient in learning long-term temporal dependencies, especially when the data is noisy [56]. OmniAnomaly [26] solves this problem by using gated recurrent unit and stochastic variable connection. However, the sequential learning mechanism of recurrent network limits the long-term sequence modeling capability of the model. Transformer architecture with multi-head self-attention enables MBTGMM to capture long-term temporal dependencies in an efficient manner.

Recent models such as MTAD-GAT and GTA learn the graph structure of variables to model relationships along feature dimension, and achieve the anomaly detection task with transformer-based models. MBTGMM outperforms those methods because convolution branch in the transformer encoding block is able to capture short-term temporal dependencies, and GMM in the framework can learn the normal distribution of learned representations.

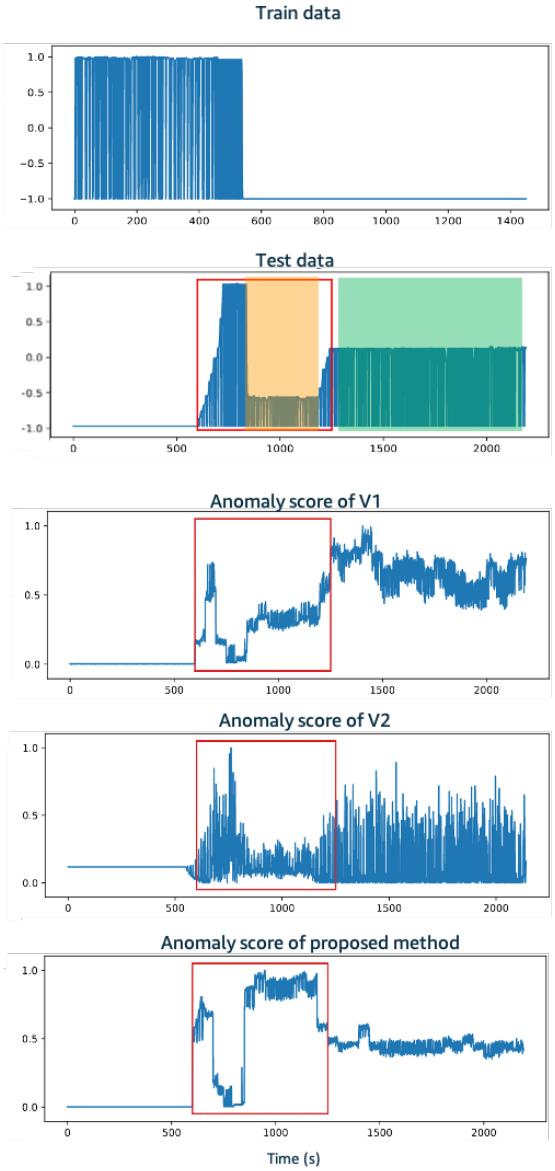
We also conduct experiments on PHM21 dataset and compare our results with the winning teams of the challenge. For this dataset, we detect anomalous time series. Therefore, we split each of the 70 normal time series into two parts without overlap, one of which is used for training and the other for testing. Thus, we have 70 normal time series for training and 99 time series for testing. During inference, we first calculate the anomaly score for each sample in the time series using the trained model, and then use the median of all anomaly scores in that time series as the final anomaly score. Since the labels are provided in the challenge, top teams mainly used supervised-based and rule-based methods with custom feature engineering and trained the model with both normal and anomalous data. For example, team HIRUTEK [57], who ranks at 1^{st} place in the challenge, utilized a combination of decision tree algorithms and a propagation system by adding a Kalman filter to

Table 4: Ablation studies on variants of our proposed methods on three datasets.

	SMAP			MSL			PHM21		
Method	Prec.	Rec.	F_{best}	Prec.	Rec.	F_{best}	Prec.	Rec.	F_{best}
MBTGMM	0.916	1.000	**0.931**	0.911	0.990	**0.942**	1.000	0.793	**0.885**
V1	0.916	0.993	0.928	0.908	0.980	0.924	0.857	0.828	0.842
V2	0.895	0.994	0.912	0.872	0.975	0.902	0.478	0.759	0.587



(a) SMAP A-3



(b) SMAP D-16

Figure 4: Case studies on SMAP dataset to show the effectiveness of proposed method.

update the probabilities. Team LDM [58] who ranks at 3^{rd} place in the challenge designed a rule-based method by comparing the characteristics of normal and abnormal data in the training set. The results of PHM21 are shown in Table 3. Note that neither approaches use deep learning methods. MBTGMM has the highest score among the unsupervised methods, and is able to improve F_{best} by 59% compared to LSTM and LSTM-AE. Moreover, our results are higher than the supervised rule-based method developed by team LDM, which shows the competitiveness of our approach compared to supervised methods.

4.4 Ablation Studies

In this section, we conduct ablation studies to demonstrate the efficacy of the design of each component in MBTGMM. Two variants of MBTGMM are created by excluding one major component in the framework and comparing the performance in terms of the F_{best}. To study the effectiveness of multi-branch transformer encoder block, we create variant **V1** of our framework in which the 1-D convolutional branch is excluded from the encoder blocks. To show the effectiveness of detecting anomalies with learned representations, we design a variant **V2** of our framework, in which we directly use the reconstruction error between the reconstructed and the original data as the anomaly scores, instead of estimating density of learned representations and calculating anomaly scores using GMM.

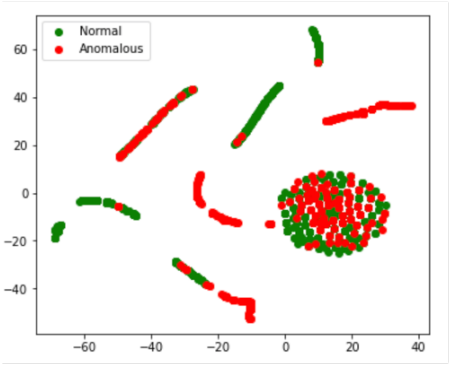
The results of the variants are shown in Table 4. From the table, our proposed MBTGMM achieves the highest F_{best} over all three datasets. By adding the 1-D convolutional branch in the transformer encoder, MBTGMM can improve average F_{best} by 2.3 % over three datasets. By using representation with GMM for anomaly detection, our MBTGMM can achieve an average of 14.9 % improvement compared with variant V2 which directly uses transformer for prediction.

Two examples are provided in Fig. 4 to further demonstrate the advantage of the designs of our framework. Fig. 4 (a) and (b) show the normalized training data, test data, and anomaly scores of MBTGMM and its variants for two subsets in SMAP dataset.

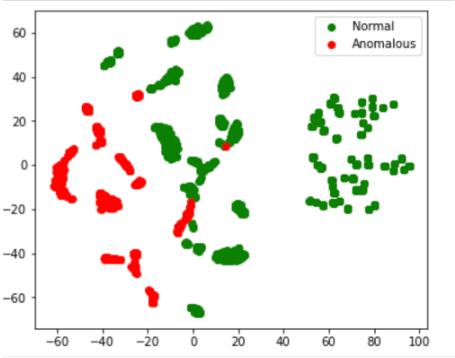
For subset A-3, the normal data has a periodic pattern. The anomaly event highlighted in red rectangle has a constant value at around 0.6. From the plots of anomaly scores, it is clear that our proposed method can assign higher anomaly scores by capturing this short-term temporal dependency change, while its variants fail to capture this. For data subset D-16, there are two modes in the normal training data, viz., a) the switching mode during which the values switch between 1 and -1, and b) the constant mode where the value is constant at -1. During inference, the value switches between -1 and other values. Our proposed method consistently allots high anomaly score within the anomalous region, unlike its variants. Compared to the anomaly scores of V1, MBTGMM can assign higher score for points in orange area compared to the green area, because the 1-D convolutional branch in MBTGMM is able to capture the short-term dependencies, where the anomalous samples in the orange area are further away from the cluster containing switching mode than the samples in the green area. Thus, the anomaly scores are higher for samples in the orange region. For both case studies, V2 cannot assign higher anomaly scores for anomalous events, which demonstrates the importance of using learned representation for anomaly detection task. The results from two case studies above clearly demonstrate the effectiveness of each component in MBTGMM.

To get a more intuitive understanding of the advantage of using MBTGMM's learned representations for anomaly detection, we use the t-distributed stochastic neighbor embedding (t-SNE) [59] to visualize the features into a two-dimension (2-D) map. The mapped features of input and output of multi-branch transformer model ϕ_e from SMAP D-16 subset are shown in Fig. 5. From the plot, the 2-D feature representations of normal and anomalous input samples of ϕ_e are not separable, while the learned representations are much more separable. These results further indicate that the multi-branch transformer of the proposed MBTGMM is learning separable representations of the input features that is necessary for the downstream anomaly detection task.

In summary, based on the ablation studies, we can conclude the following: (1) There is a considerable gap between MBTGMM and the variant V2 without using representation learning and GMM, which demonstrates the effectiveness of using representation learning for time series anomaly detection. (2) The 1-D convolutional branch in transformer encoder helps capture short-term dependencies in the input data, and can detect anomalies that violate short-term dependencies. These results again support that every



(a) t-SNE of input of ϕ_e



(b) t-SNE of output of ϕ_e

Figure 5: t-SNE visualization of input and output of ϕ_e for SMAP D-16 subset.

component in MBTGMM is valuable for anomaly detection task for industrial time series.

5 CONCLUSIONS

In this paper, we proposed MBTGMM, a multi-branch transformer-based framework for detecting anomalies from time series data by learning normal representations. A 1-D convolutional branch was added in the transformer encoder block to capture short-term temporal dependencies. GMM is utilized in the framework to estimate density of learned normal representations and detect anomalies in low-density regions. Results from extensive experiments on three industrial datasets show the efficacy of the proposed MBTGMM compared to the state-of-the-art methods on anomaly detection task. We also provided ablation and case studies to explain how each component in MBTGMM helps with detecting temporal anomalies. Future improvements will include exploring advanced architecture to learn better representations of such time series.

REFERENCES

[1] Pavol Tanuska, Lukas Spendla, Michal Kebisek, Rastislav Duris, and Maximilian Stremy. Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0. *Sensors*, 21(7):2376, 2021.

[2] Fangzhou Cheng, Ajay Raghavan, Deokwoo Jung, Yukinori Sasaki, and Yosuke Tajika. High-accuracy unsupervised fault detection of industrial robots using current signal analysis. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2019.

[3] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

[4] Aditya P. Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ics security. *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, 2016.

[5] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya 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*, pages 25–28, 2017.

[6] Randy Paffenroth, Philip Du Toit, Ryan Nong, Louis Scharf, Anura P Jayasumana, and Vidarshana Bandara. Space-time signal processing for distributed pattern detection in sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):38–49, 2013.

[7] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer, 2007.

[8] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[9] Nicola Acito, Marco Diani, and Giovanni Corsini. Gaussian mixture model based approach to anomaly detection in multi/hyperspectral images. In *Image and Signal Processing for Remote Sensing XI*, volume 5982, page 59820O. International Society for Optics and Photonics, 2005.

[10] Ritwik Giri, Fangzhou Cheng, Karim Helwani, Srikanth V Tenneti, Umut Isik, and Arvindh Krishnaswamy. Group masked autoencoder based density estimator for audio anomaly detection. *Proc. DCASE*, pages 51–55, 2020.

[11] Masataka Yamaguchi, Yuma Koizumi, and Noboru Harada. Adaflow: Domain-adaptive density estimator with application to anomaly detection and unpaired cross-domain translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3647–3651. IEEE, 2019.

[12] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[13] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

[14] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[18] Ruixiong Zhang, Haiwei Wu, Wubo Li, Dongwei Jiang, Wei Zou, and Xiangang Li. Transformer based unsupervised pre-training for acoustic representation learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6933–6937. IEEE, 2021.

[19] Hengyu Meng, Yuxuan Zhang, Yuanxiang Li, and Honghua Zhao. Spacecraft anomaly detection via transformer reconstruction error. In *International Conference on Aerospace System Science and Engineering*, pages 351–362. Springer, 2019.

[20] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*, 2022.

[21] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[22] André Gensler, Janosch Henze, Bernhard Sick, and Nils Raabe. Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 002858–002865. IEEE, 2016.

[23] Shahroz Tariq, Sangyup Lee, Youjin Shin, Myeong Shin Lee, Okchul Jung, Daewon Chung, and Simon S Woo. Detecting anomalies in space using multivariate convolutional lstm with mixtures of probabilistic pca. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2123–2133, 2019.

[24] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[25] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196, 2018.

[26] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan 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*, pages 2828–2837, 2019.

[27] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404, 2020.

[28] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33:13016–13026, 2020.

[29] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(5), pages 4027–4035, 2021.

[30] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. Learning graph structures with transformer for multivariate time series anomaly detection in iot. *IEEE Internet of Things Journal*, 2021.

[31] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 33–43. IEEE, 2020.

[32] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[34] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[35] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[37] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[38] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF*

International Conference on Computer Vision, pages 6836–6846, 2021.

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[40] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

[41] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[42] Xinyang Feng, Dongjin Song, Yuncong Chen, Zhengzhang Chen, Jingchao Ni, and Haifeng Chen. Convolutional transformer based dual discriminator generative adversarial networks for video anomaly detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5546–5554, 2021.

[43] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021.

[44] Walter Hugo Lopez Pinaya, Petru-Daniel Tudosiu, Robert Gray, Geraint Rees, Parashkev Nachev, Sébastien Ourselin, and M Jorge Cardoso. Unsupervised brain anomaly detection and segmentation with transformers. *arXiv preprint arXiv:2102.11650*, 2021.

[45] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. *arXiv preprint arXiv:2104.13897*, 2021.

[46] Liyang Chen, Zhiyuan You, Nian Zhang, Juntong Xi, and Xinyi Le. Utrad: Anomaly detection and localization with u-transformer. *Neural Networks*, 147:53–62, 2022.

[47] Xixuan Wang, Dechang Pi, Xiangyan Zhang, Hao Liu, and Chang Guo. Variational transformer-based anomaly detection approach for multivariate time series. *Measurement*, page 110791, 2022.

[48] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

[49] Hongwei Zhang, Yuanqing Xia, Tijin Yan, and Guiyang Liu. Unsupervised anomaly detection in multivariate time series through transformer-based variational autoencoder. In *2021 33rd Chinese Control and Decision Conference (CCDC)*, pages 281–286. IEEE, 2021.

[50] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.

[51] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

[52] Prognostics and Health Management Society. Data challenge, 2021.

[53] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[54] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 703–716. Springer, 2019.

[55] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[56] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 841–850. IEEE, 2020.

[57] Kerman López de Calle-Etxabe, Meritxell Gómez-Omella, and Eider Garate-Perez. Divide, propagate and conquer: Splitting a complex diagnosis problem for early detection of faults in a manufacturing production line. In *PHM Society European Conference*, volume 6(1), pages 9–9, 2021.

[58] Osarenren Kennedy Aimiyekagbon, Lars Muth, Meike Wohlleben, Amelie Bender, and Walter Sextro. Rule-based diagnostics of a production line. In *PHM Society European Conference*, volume 6(1), pages 10–10, 2021.

[59] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.