DuoGAT: Dual Time-oriented Graph Attention Networks for Accurate, Efficient and Explainable Anomaly Detection on Time-series

Jongsoo Lee*
Byeongtae Park*
Hanyang University
Seoul, South Korea
{leejongsoo,byeongtae}@hanyang.ac.kr

Dong-Kyu Chae
Hanyang University
Seoul, South Korea
dongkyu@hanyang.ac.kr

ABSTRACT

Recently, Graph Neural Networks (GNNs) have achieved state-of-the-art performance on the multivariate time-series anomaly detection task by learning relationships between variables (sensors). However, they show limitations in capturing temporal dependencies due to lack of sufficient consideration on the characteristics of time to their graph structure. Several studies constructed a time-oriented graph, where each node represents a timestamp within a certain sliding window, to model temporal dependencies, but they failed to learn the trend of changes in time-series. This paper proposes **Du**al time-**o**riented **G**raph **AT**tention networks (**DuoGAT**) that resolves the aforementioned problems. Unlike previous work that uses the simple complete undirected structure for time-oriented graphs, our work models directed graphs with weighted edges that only connect from prior events to posterior events, and the edges that connect nearby events are given higher weights. In addition, another time-oriented graph is used to model time series stationary via differencing, which especially focuses on capturing the series of changes. Empirically, our method outperformed the existing state-of-the-art work with the highest F1-score for the four real-world dataset while maintaining low training cost. We also proposed a novel explanation method for anomaly detection using DuoGAT, which provides time-oriented reasoning via hierarchically tracking time points critical in a specific anomaly detection. Our code is available at: https://github.com/ByeongtaePark/DuoGAT

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; **Neural networks**.

KEYWORDS

Multivariate time-series, Anomaly detection, Explainable AI, Graph neural networks

ACM Reference Format:
Jongsoo Lee, Byeongtae Park, and Dong-Kyu Chae. 2023. DuoGAT: Dual Time-oriented Graph Attention Networks for Accurate, Efficient and Explainable Anomaly Detection on Time-series. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614857

1 INTRODUCTION

In recent years, there has been an explosion in time-series data generation as a result of the widespread adoption of sensors and Internet of Things (IoT) devices in various fields such as healthcare, smart factories, or cybersecurity [28, 44]. Research on multivariate time-series analysis has gained great attention from researchers and practitioners in these fields. Anomaly detection on such a data stream, which aims to identify a certain period of time generating data significantly different from the overall series, is one of the most important tasks; a rapid and accurate anomaly detection can support diagnosis and maintenance of a system and prevent any potential problems that may have led to critical damage [7, 10, 16].

Anomaly detection can be performed in either supervised or unsupervised manner. Following recent trend, this paper focuses on the unsupervised approach where all the training examples are assumed to be normal. In this case, anomalies are detected based on a notion of residual, which indicates how each data point is significantly different from normal patterns learned from training data. Traditional methods in this area are based on the unsupervised models such as *Principal Component Analysis* (*PCA*) [40], *K-nearest neighbors* (*KNN*) [3] and *Feature Bagging* (*FB*) [20]. Recent detectors are mostly based on deep neural networks which can be roughly categorized by *AutoEncoder* based methods (e.g., USAD [5] and DAGMM [49]), *Convolutional Neural Networks* (*CNN*) based methods (e.g., SR-CNN [34] and MSCRED [46]), (3) *Recurrent Neural Networks* (*RNN*) based methods (e.g., LSTM-AD [25], LSTM-ED [24], LSTM-VAE [33], OmniAnomaly [41] and THOC [39]) and (4) *Generative Adversarial Networks* (*GAN*) [15] based methods (e.g., TAnoGAN [6] and MAD-GAN [22]). Very recently, several methods are designed based on *Graph Neural Networks* (GNN) [35] to explicitly model relationships between data from sensors and their latent interactions, where notable examples include GDN [11], MTAD-GAT [48] and GTA [8], achieving state-of-the-art performance on the multivariate time-series anomaly detection.

Among them, our work is motivated MTAD-GAT [48], which is a framework learning from both the feature-oriented graph and *time-oriented graph* to capture the complex dependencies in both

*Both authors contributed equally to this research.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614857

feature and temporal dimensions. Here, the time-oriented graph considers all the timestamps within a sliding window as a complete graph where each node represents a feature vector on the corresponding timestamp. This structure helps in modeling the temporal dependencies within each time-series, although there is still room for improvement.

However, we argue that existing GNN-based detectors still have limitations in capturing temporal dependencies. In terms of their graph structures, the feature-oriented graphs have intrinsic difficulty in modeling relationships between data near times. Even though the time-oriented graph suggested by MTAD-GAT [48] targets modeling such dependencies, it may not work well due to its undirected, unweighted and complete graph structure that cannot model both the direction and weight according to the time flow but increases computational complexity. In addition, time-series anomalies would be more likely to occur as the change in sensor values is sudden and large [4, 32]. However, existing GNN-based work does not try to specifically model the changes in features.

In this paper, we present **DuoGAT**: **Du**al time-**o**riented **G**raph **AT**tention networks towards accurate, efficient and explainable detection of time-series anomalies. Instead of the undirected, unweighted and complete time-oriented graph used by the previous work, we build a directed and weighted graph where each directed edge starts from prior time point to posterior time and higher weights are given to the edges connecting nearby times. Our design prevents the unrealistic modeling that (1) a data point at a posterior time affects prior time and (2) equal weight is assigned to both edges connecting near data points and far data points in time. In addition, we transform the input time-series via *differencing* [29], i.e., computing the differences between consecutive observations, and model this additional data with another time-oriented graph, focusing on understanding the patterns of change in features. Dual graph attention networks are then trained on top of the two graphs, each aims to model temporal dependencies between time points and the changes in sensor values over time, respectively. Here, the multi-dimensional attention mechanism [43] is employed to reflect various information obtained from latent spaces with multiple dimensions to the output attention matrix.

Our contributions can be structured in three folds:

- **DuoGAT**: It adopts the directed and weighted graph structure to model a given time-series data for better capturing temporal dependencies inside. It also takes additional input data generated via differencing and models it through another time-oriented graph so that it learns to be more attentive to changes over time. Its dual graph attention networks are based on the multi-dimensional attention mechanism, which benefits from rich information in multiple embedding spaces.
- **Accurate and efficient:** Empirically, DuoGAT achieves a state-of-the-art performance in four real-world dataset (SWaT, WADI, SMAP and MSL) while maintaining low computational complexity. Our ablation study confirms the effectiveness of each aforementioned idea of DuoGAT.
- **Explainable:** Additionally, we provide an explanation method for anomaly detection working on top of DuoGAT. Existing explanatory methods mostly use the attention scores to point out important sensors on a specific anomaly detection, which lacks the temporal relationship between detection of abnormal time period and data points near the time period. Our explanation covers the gap via hierarchically tracking critical time points to provide time-oriented reasoning.

2 RELATED WORK

This section summarizes the existing literature on unsupervised time-series anomaly detection in terms of the two data types: univariate time-series and multivariate time-series [10].

2.1 Univariate Time-Series Anomaly Detection.

Univariate time series is like data collected from a single sensor, where each instance consists of single attribute. The anomaly detectors in the early stage focused on this type of data. For example, *PCA* [40], one of a dimensionality reduction methods, detects anomalies through the principal component classifier that finds observations extreme and out of a normal correlation structure. *KNN* [3] uses the distance between data points to find observations far from majority of data. *ARIMA* [47] is a statistical regression analysis model that predicts future values based on prior values, where anomalies are identified based on the residual errors.

Recently, deep learning-based approaches have been applied to the univariate time-series anomaly detection. *SR-CNN* [34] generates spectral residual inspired by the visual saliency domain and applies CNN on the residual to detect anomalies. *DeepAnt* [30] consists of a time-series predictor based on CNN to forecast future values and an anomaly detector module working based on the predicted values. *TAnoGAN* [6] is a GAN-based detector that models the normal data's distribution via the adversarial training process [15] and computes reconstruction loss between an observation sequence and its generated version from the learned distribution given its latent code.

2.2 Multivariate Time-Series Anomaly Detection.

Multivariate time-series data is typically collected from a set of multiple sensors, where each instance in dataset consists of multiple attributes. *FB* [20] is a bagging-based method that aggregates the scores of the detectors using a meta-estimator that fits multiple detectors trained with various small subsets of features. *AE* [1] is an anomaly detector using the Autoencoder structure, where the anomaly score is calculated based on the reconstruction error: the difference between the input raw data and its reconstructed version. *KitNET* [27] uses an ensemble of AEs, each calculating the reconstruction error to collectively detect anomalies. *LSTM-VAE* [33] consists of an encoder that projects observations and their temporal dependencies into latent space using a series of connected LSTM and VAE (Variational AutoEncoder) layers, and a decoder that estimates the expected distribution of the observations from the latent space representation. *LSTM-NDT* [18] considers an error set, which includes the differences between the predicted and actual values. This set is smoothed to mitigate spikes in errors that often occur in LSTM-based predictions. *DAGMM* [49] consists of a compression network that generates the low-dimensional representation via an AutoEncoder and an estimation network that accepts the generated representation to process density estimation based on

GMM for input data. *GAN-AD* [21] adopts LSTM-RNN style generator and discriminator to capture the distribution of time-series data. *MAD-GAN* [22] designs the generator and discriminator structure based on LSTM-RNN to capture temporal correlation of time-series distributions and detects anomalies through reconstruction and discrimination errors. *OmniAnomaly* [41] is based on a stochastic RNN that combines VAE [19] and Gated Recurrent Unit (GRU) [9]. It learns temporal dependence between stochastic variables with a stochastic variable connection and a planar normalizing flow, and uses reconstruction probabilities to detect anomalies. *USAD* [5] is also a GAN-inspired model with an adversarial learning autoencoder architecture that learns to amplify reconstruction errors while achieving good stability.

Very recently, *Graph Neural Networks* (*GNN*) [35] have been actively adopted in this task to compensate for the limitations of existing methods that cannot explicitly learn the relationships among features. Especially, the *Graph Attention Networks* (*GAT*) structure has been commonly chosen to focus on the most important signals. *GDN* [11] is a GAT-based model that learns relationships between sensors by using the feature-oriented graph. It selects the top-k similar sensors based on the learned feature representations of a given sensor, forecasts its future behavior, and identifies deviations from the learned sensor relationships. *MTAD-GAT* [48] designs a feature-oriented graph attention layer to capture relationships between multiple features and a time-oriented graph attention layer to understand temporal dependencies. The two GAT layers are jointly optimized with the forecasting loss and the reconstruction loss functions. *MST-GAT* [12] consists of a multimodal graph attention network (M-GAT), which includes a multi-head attention module and two relational attention modules, and a temporal convolution network to capture temporal dependencies. It detects anomalies by simultaneously optimizing the reconstruction and prediction modules. *GTA* [8] consists of the three encoder layers and one decoder layer, including a context encoding block composed of l levels multiscale dilated convolution and graph convolution pairs. Multi-branch self-attention is used to solve the quadratic complexity challenge of the original multi-head attention mechanism.

3 PROPOSED METHOD

Time-series is a sequence of data points collected over time, generally with equal time intervals. Multivariate time-series can be formally defined as $x = \{x_1, .., x_n\}$ where n is the maximum length of timestamps and each $x_t \in R^m$ is an m-dimensional vector at time t with m sensors. The goal of the anomaly detection is to detect whether an observation at time t is an anomaly. A common approach to multivariate time-series anomaly detection is a forecasting-based approach that uses observations $x = \{x_{t-k}, ..., x_{t-1}\}$ belonging to a window of size k to determine whether there is an anomaly at time t. This approach then calculates anomaly score through the difference between actual and predicted values at x_t and judges anomaly when the score exceeds a pre-defined threshold. This task is performed in an unsupervised manner; our training data is assumed to be containing only normal data and we determine whether each given time-series window of test data is an anomaly or not based on the computed anomaly scores. Following prior work [8, 48], we applied the min-max normalization to both training and test data before putting them to our model.

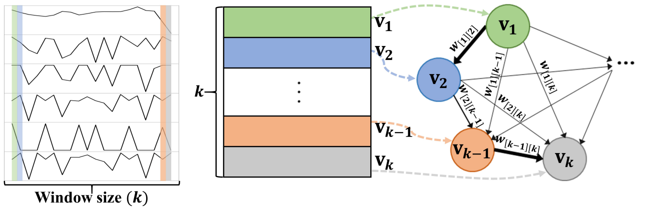

Figure 1: Our graph configuration in an arbitrary window with size k. In the graph, the node $\mathbf{v}_k$ represents the time point k and has sensor values as feature vectors. An edge with a weight $w_{[1][k]}$ has a direction from 1 to k, where k is a posterior time point, and represents influence of $\mathbf{v}_1$ on $\mathbf{v}_k$. The thickness of each edge represents the magnitude of the weight values.

3.1 Overview

Figure 2 shows the overview of DuoGAT. It consists of the following four main components:

- **Graph configuration.** We define a directed and weighted time-oriented graph to model temporal characteristics, which can be simply represented by the upper triangular adjacency matrix format.
- **Differencing of time-series.** We generate additional input data obtained through differencing [29]. We then obtain the differencing-based attention score that captures the amount of changes in sensor values over time.
- **Dual GAT layers based on multi-dimensional attention.** DuoGAT has two GAT layers: T-GAT layer aligned with the original time-series input and the D-GAT layer that deals with the additional input produced by differencing. For each layer, multi-dimensional attention mechanism is adopted to benefit from various perspectives of multiple embedding spaces.
- **Anomaly score computation.** Anomaly score is calculated by sequentially passing the hidden states derived from the two GAT layers to the Gated Recurrent Unit (GRU) and the Fully-Connected layer (FC).

The following subsections introduce the aforementioned components in detail.

3.2 Graph Construction

As previously mentioned, the graph structures of the existing GNN-based detectors cannot properly model the characteristics of time. Our graph structure resolves the problem by the structure of a directed and weighted graph, illustrated in Figure 1.

Formally, each node in our graph represents a time point, and has the feature vector representing the values of sensors at the corresponding time. Considering the window size k, a set of node features within a window can be denoted as $V = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k], V \in R^{k \times m}$ where m is dimension of feature (i.e., the number of sensors)[1].

[1] If a data point is mapped to a node, we use notation $\mathbf{v}$ instead of x [48].

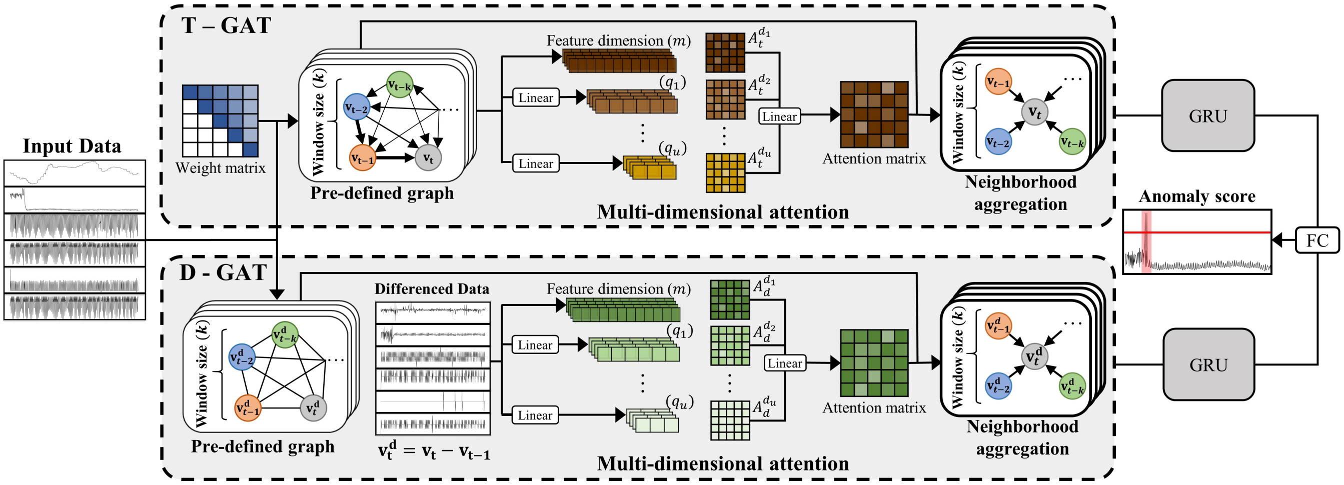

Figure 2: Overview of our DuoGAT with two GAT layers (T-GAT and D-GAT layers) including multi-dimensional attention. Each A_t^d indicates an attention matrix. The outputs of each GAT layer is fed to each GRU layer. The outputs from both GRU layers are concatenated and fed to the final fully connected layer to forecast the values of next time point.

For defining the edge directions, we set each element a_{ij} of the adjacency matrix $A = \{a_{11}, a_{12}, ..., a_{ij}\} \in R^{k \times k}$ as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } i \leq j \\ 0, & \text{otherwise} \end{cases}$$

where $a_{ij} = 1$ indicates that node i points to node j and $a_{ij} = 0$ represents that there is no connection among i and j.

Finally, we construct a weighted adjacency matrix where the weights are differently assigned so that closer time points have more influence. For the weighted adjacency matrix $A^w = \{w_{11}, w_{12}, ..., w_{ij}\}$, $A^w \in R^{k \times k}$, we define each element w_{ij} as follows:

$$w_{ij} = \begin{cases} \log_k(k - (j - i) + 1), & i < j \\ 1, & i = j \\ 0. & \text{otherwise} \end{cases}$$

3.3 Dual GAT Layers

Here, two GAT layers are designed to learn characteristics of multivariate time-series modeled via our graph structure.

3.3.1 T-GAT: the time-oriented GAT layer. The goal of the T-GAT layer is to learn the relationships between time points to reflect the sequential characteristic of time-series. The T-GAT layer is trained with the graph we introduced in the previous subsection as an input. An attention score α_{ij} of T-GAT, which represents the importance between the time points (nodes) i and j, is calculated by:

$$e_{ij} = \omega_t^T \cdot LeakyReLU(W_t \cdot [\mathbf{v}_i \,\|\, \mathbf{v}_j]) \tag{1}$$

$$\alpha_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{n \in N(i)} exp(e_{in})} \tag{2}$$

where $W_t \in R^{2m \times 2m}$ is a learnable parameter matrix, $[\quad \| \quad]$ indicates the concatenation operation, and $\omega_t \in R^{2m}$ is a learnable parameter vector for the attention mechanism. e_{ij} is the attention coefficient of node i to j, and normalized with the Softmax function. $N(i) = \{j \mid w_{ij} > 0\}$ is the set of neighbors of node i.

3.3.2 D-GAT: the differencing-based GAT layer. Another GAT layer, named as D-GAT, is designed to capture sudden changes that can cause anomalies by attentively learning how much the sensor values change over time. To this end, we first produce the data with differencing [29] of input time-series, which is known to remove trends and seasonality in time-series by computing the differences between consecutive data points (i.e., $x_t^{\prime} = x_t - x_{t-1}$). The D-GAT layer then applies the attention mechanism to the differenced data in order to output a differencing-based attention score that focuses on the feature differences over time. Formally, a differencing-based attention score α_{ij}^d between node i and node j is calculated by:

$$\mathbf{v}_t^d = \mathbf{v}_t - \mathbf{v}_{t-1} \tag{3}$$

$$e_{ij}^d = \omega_d^T \cdot LeakyReLU(W_d \cdot [\mathbf{v}_i^d \,\|\, \mathbf{v}_j^d]) \tag{4}$$

$$\alpha_{ij}^d = softmax(e_{ij}^d) = \frac{exp(e_{ij}^d)}{\sum_{n \in N(i)} exp(e_{in}^d)} \tag{5}$$

where $W_d \in R^{2m \times 2m}$ is a learnable parameter matrix and $\omega_d \in R^{2m}$ is a learnable parameter vector for the attention mechanism. $\mathbf{v}_t^d$ is the node feature vector at the time point t that was applied differencing. $N(i) = \{j \mid i \geq j\}$ is the set of neighbor nodes of i.

3.3.3 Multi-dimensional attention for the Dual GAT layers. This subsection introduces the details of the multi-dimensional attention equipped with the two GAT layers. The typical choice in the literature has been the multi-head attention approach, where an attention score is calculated through parallel computation for the same dimension of embedding space.

However, we argue that it is insufficient to capture various information in the same dimensions for learning the relationships between nodes. To overcome the limitation, each feature vector $\mathbf{v} \in R^m$ should be embedded in multiple dimensions of latent spaces to calculate the attention score while reflecting various aspects:

$$V^{\prime} = [\mathbf{v}_1^{\prime}, \mathbf{v}_2^{\prime}, \ldots, \mathbf{v}_k^{\prime}] \in R^{k \times |q|}$$

where $q \in \{q_1, q_2, ..., q_u\}$ denotes the set of embedding dimension values and each $\mathbf{v}'$ denotes the embedding of the corresponding node feature. To calculate the attention score, we performed the attention mechanism parallel to each embedding node feature vector $\mathbf{v}'$. The attention scores from each embedding dimension are passed through a linear function to derive a multi-dimensional attention score. Formally, the multi-dimensional attention score s_{ij} between the node pair $\mathbf{v}_i$ and $\mathbf{v}_j$ is derived as follows:

$$s_{ij} = LeakyReLU(W_t^{multi} \cdot [||_{c=1}^{u}(\alpha_{ij})^{q_c} \,||\, \alpha_{ij}]) \tag{6}$$

where $(\alpha_{ij})^{q_c}$, from $c = 1$ to u, are the attention scores for embedding vectors $\mathbf{v}_i' \in R^{q_c}$ and $\mathbf{v}_j' \in R^{q_c}$ of nodes i and j, respectively, and (α_{ij}) is the attention score for the original m-dimensional feature vectors. All these are concatenated to contain various information from the multi-dimensional attention. $W_t^{multi} \in R^{u \times 1}$ is a learnable parameter vector. Applying the multi-dimensional attention to our T-GAT layer helps understand relationships between time points from various perspectives.

The output of T-GAT layer $\mathbf{h}_i$ is the hidden state of node i by aggregating neighborhood nodes by reflecting the multi-dimensional attention score s_{ij}^t and the weighted adjacency matrix A^w. Formally, $\mathbf{h}_i$ is calculated by:

$$\mathbf{h}_i = \sigma(\sum_{j \in N(i)} s_{ij} w_{ij} \mathbf{v}_j) \tag{7}$$

where σ represents the sigmoid activation function. Note that $\mathbf{h}_i \in R^m$ has the same dimension with the input x.

The multi-dimensional approach is also applied to the D-GAT layer to benefit from various perspectives. The multi-dimensional differencing-based attention score s_{ij}^d between node pairs $\mathbf{v}_i^d$ and $\mathbf{v}_j^d$ is calculated as follows:

$$s_{ij}^d = LeakyReLU(W_d^{multi} \cdot [||_{c=1}^{u}(\alpha_{ij}^d)^{q_c} \,||\, \alpha_{ij}^d]) \tag{8}$$

where $(\alpha_{ij}^d)^{q_c}$ is the differencing-based attention score for embedding vectors $\mathbf{v}_i^{d'} \in R^{q_c}$ and $\mathbf{v}_j^{d'} \in R^{q_c}$ of node i and node j. $W_d^{multi} \in R^{u \times 1}$ is a learnable parameter vector. The output of D-GAT layer, $\mathbf{h}_i^d$, is also the hidden state of node i updated by aggregating neighborhood nodes, each with the score s_{ij}^d. Here, unlike the T-GAT layer, our D-GAT does not apply the weighted adjacency matrix A^w because the differencing of time-series weakens the temporal characteristics [14], so we allow all the interactions among time points within a window. As a result, the output $\mathbf{h}_i^d$ is calculated by:

$$\mathbf{h}_i^d = \sigma(\sum_{j \in N(i)} s_{ij}^d \mathbf{v}_j) \tag{9}$$

Again, $\mathbf{h}_i^d \in R^m$ has the same dimension with the input x.

3.4 Training Objective

The outputs of our T-GAT and D-GAT layers are fed to each GRU (Gated Recurrent Unit) layer. Formally, its operations can be denoted as follows:

$$r_t = \sigma(W_r \mathbf{h}_i + U_r g_{t-1} + b_r) \tag{10}$$

$$z_t = \sigma(W_z \mathbf{h}_i + U_z g_{t-1} + b_z) \tag{11}$$

$$\tilde{g_t} = tanh(W\mathbf{h}_i + U(r_t \odot g_{t-1}) + b) \tag{12}$$

$$g_t = (1 - z_t) \odot g_{t-1} + z_t \odot \tilde{g_t} \tag{13}$$

where z and r are the update and reset gates, respectively. W_z, W_r, W, b_z, b_r and b are the parameters of the GRU layer. $\mathbf{h}_i$ and g_t are respectively the input and output of it. $\odot$ denotes element-wise multiplication.

Finally, the outputs of the two GRU layers are concatenated and fed to the Fully-Connected (FC) layer, which then forecasts the values of the sensors at the following time point, i.e., $\hat{x_t}$, within a window. Its prediction error is measured by Root Mean Square Error (RMSE), which is used as a loss function for our model training:

$$L_{RMSE} = \sqrt{\frac{\sum_{t=k+1}^{T_{train}} (\hat{x_t} - x_t)^2}{T_{train} - k}} \tag{14}$$

where T_{train} indicates the last time point(instance) of the training dataset.

3.5 Anomaly Detection

After the model training is complete, we compute the anomaly scores by measuring the difference between the actual and predicted values of the sensors at a given time. The anomaly score of sensor o at time point t is obtained as follows:

$$Anomaly\ score_t^o = |\hat{x}_t^o - x_t^o| \tag{15}$$

Since each sensor data has various characteristics, the anomaly score for each sensor has various scales. As a result, there is a risk of relying on a particular sensor with a big scale to detect anomalies. To mitigate this issue, we standardized the calculated anomaly scores as follows:

$$Anomaly\ score_t^o = \frac{Anomaly\ score_t^o - \mu_o}{\sigma_o} \tag{16}$$

where μ_o and σ_o denotes the average and standard deviation of anomaly scores for each sensor. Then, the final anomaly score is derived using the maximum value of the anomaly scores at the given time point:

$$A(t) = \max_o(Anomaly\ score_t^o) \tag{17}$$

Finally, the model detects an anomaly when the score exceeds a pre-defined threshold, which is selected by the grid search on a validation set.

4 EXPERIMENTAL SETTINGS

We conducted extensive experiments to evaluate our DuoGAT. This section summarizes our experimental environment.

Dataset	# features	# train	# test	Anomaly rate(%)
SWaT	51	496,800	449,919	12.14
WADI	127	1,209,601	172,801	5.99
SMAP	25	135,183	427,617	13.13
MSL	55	58,317	73,729	10.72

Table 1: A summary statistics of the dataset.

4.1 Dataset

We employed the four real-world publicly available dataset: SWaT (Secure Water Treatment) [26], WADI (Water Distribution) [2], SMAP (Soil Moisture Active Passive satellite) [31] and MSL (Mars Science Laboratory rover) [38]. SWaT and WADI are water treatment physical testbed systems available from iTrust.[23] Specifically, SWaT testbed is an industrial control system (ICS) for the purpose of security research and was launched on March 15, 2015. It was collected from 51 sensors and actuators that operated continuously for 11 days, including 7 days of normal operation and 4 days with an attack scenario. WADI is an extension of SWaT and was launched on July 26, 2016. It was collected from 127 sensors and actuators that operated for 16 consecutive days, including 14 days of normal operation and 2 days with an attack scenario.

SMAP and MSL are spacecraft dataset collected from NASA, available at the public storage[4]. SMAP is one of the first Earth observation satellites developed by NASA. The orbiting observatory measures surface soil conditions everywhere on Earth every 2 ~ 3 days, distinguishing between frozen and thawed land. On land not frozen or covered in water, SMAP uses this information to create a global map of soil moisture, measuring how much water is in the top layer of soil. MSL gathers imaging, spectroscopy, composition data, and other measurements for selected Martian soils, rocks, and the atmosphere. Table 1 summarizes the statistics of each data.

4.2 Evaluation Metrics

We evaluated the anomaly detection performance with Precision, Recall and F1-score, which are the most widely-used metrics. Each can be computed by:

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

where TP, FP and FN stand for true positive, false positive and false negative, respectively. Following prior work [8], we applied the point-adjust approach [42] to compute the evaluation metrics. This is because anomalous observations often occur in the form of contiguous anomaly segments. In this approach, even if only one observation is detected as an anomaly in an anomalous segment, we assume that all other observations in that anomalous segment are detected as anomalies and consider the classification to be correct.

4.3 Competitors

We compared DuoGAT with a wide range of anomaly detector families: **PCA** [40], **KNN** [3] and **FB** [20] from the *traditional detectors*, **AE** [1], **KitNET** [27], **DAGMM** [49], **OmniAnomaly** [41] and **USAD** [5] which are the *AutoEncoder-based methods*, **LSTM-NDT** [18] and **LSTM-VAE** [33] within the *LSTM-based models*, **GAN-AD** [21] and **MAD-GAN** [22] from the *GAN-based detectors*, and the *GNN-based state-of-the-arts* including **GDN** [11], **MTAD-GAT** [48], **MST-GAT** [12] and **GTA** [8]. Most of these competitors employed are explained in the Related Work section.

4.4 Implementation Details

We implement our model using PyTorch version 1.7.1 with CUDA 11.0 and PyTorch Geometric Library [13] version 1.5.0. We conducted all our experiments on a server equipped with the 11^{th} Generation Intel CPU, 3.50GHz i9-11900KF, and the NVIDIA GeForce RTX 3090 GPU. We adopted the Adam optimizer with learning rate $1e^{-3}$. We trained the model with over 30 epochs and used early stopping with the patience value of 15. We set the each embedding dimension of set q to 50% and 25%, and the hidden dimension size of GRU layers to 150. For each dataset, we empirically chose the sliding window size among {5, 50, 150, 30} and the hidden dimension size of the FC layers among {150, 150, 100, 150} based on the validation set. More details can be found in our code.

5 RESULTS AND ANALYSES

This section reports the results of our extensive experiments.

5.1 Performance Comparisons

Tables 2 and 3 report the multivariate time-series anomaly detection accuracy of each method in terms of Precision, Recall and F1-score. Our DuoGAT exhibited state-of-the art performance for all the dataset used. Specifically, in term of F1-score, DuoGAT achieved 5.49%, 10.71%, 2.50% and 3.20% relative improvement over the best competitor, GTA, on SWaT, WADI, SMAP and MSL dataset, respectively.

We also observed that GNN-based methods tend to show higher accuracy than other categories. This result implies the importance of learning the relationships between nodes (i.e., sensors or time points) through a graph structure. However, the existing GNN-based models do not fully reflect temporal characteristics in time-series to their graph structures. For example, GDN does not capture temporal characteristics as it focuses on learning relationships between sensors with the feature-oriented graph structure. MTAD-GAT used the undirected, unweighted and complete time-oriented graph, which models a contradiction like posterior data points affects prior data points in time. GTA adopts GCN which includes dilated convolution layers for temporal data, but it still learns based on the sensor-based graph structure. Unlike the existing work, we carefully designed the graph structure and input data configurations, which plays a key role in achieving such high performance.

5.2 Ablation Study

Our ablation study aims to verify that each idea applied to our DuoGAT really helps in improving the detection performance. To this

[2] https://itrust.sutd.edu.sg/testbeds/secure-water-treatment-swat/

[3] https://itrust.sutd.edu.sg/testbeds/water-distribution-wadi/

[4] https://s3-us-west-2.amazonaws.com/telemanom/data.zip

Method	SWaT			WADI		
	Precision	Recall	F1-score	Precision	Recall	F1-score
PCA◆[40]	0.2492	0.2163	0.23	0.3953	0.0563	0.10
KNN◆[3]	0.0783	0.0783	0.08	0.0776	0.0775	0.08
FB◆[20]	0.1017	0.1017	0.10	0.0860	0.0860	0.09
DAGMM◆[49]	0.2746	0.6952	0.39	0.5444	0.2699	0.36
AE◆[1]	0.7263	0.5263	0.61	0.3435	0.3435	0.34
LSTM-VAE◆[33]	0.9624	0.5991	0.74	0.8779	0.1445	0.25
MAD-GAN◆[22]	0.9897	0.6374	0.77	0.4144	0.3392	0.37
OmniAnomaly◇[41]	0.9825	0.6497	0.78	0.9947	0.1298	0.23
USAD◇[5]	0.9851	0.6618	0.79	0.6451	0.3220	0.43
GDN◆[11]	0.9935	0.6812	0.81	0.9750	0.4019	0.57
MST-GAT◇[12]	0.9873	0.7241	0.84	0.9824	0.4351	0.60
GTA◆[8]	0.9483	0.8810	0.91	0.8391	0.8361	0.84
DuoGAT	0.9712	0.9428	**0.96** (5.49%)	0.8942	0.9797	**0.93** (10.71%)

Table 2: Performance comparison of DuoGAT and the competitors on SWaT and WADI. The best performance by F1-score is shown in bold. The percentage next to DuoGAT's F1-score represents improvement over the second best performer whose score is underlined. ◆ : Results from GTA [8]. ◇ : Results from MST-GAT [12].

Method	SMAP			MSL		
	Precision	Recall	F1-score	Precision	Recall	F1-score
PCA◇[40]	0.2884	0.1993	0.2357	0.2937	0.2414	0.2650
DAGMM◆[49]	0.5845	0.9058	0.7105	0.5412	0.9934	0.7007
AE◇[1]	0.7216	0.7995	0.7586	0.7166	0.5008	0.5896
LSTM-VAE◆[33]	0.8551	0.6366	0.7298	0.5257	0.9546	0.6780
GAN-AD◆[21]	0.6710	0.8706	0.7579	0.7102	0.8706	0.7823
KitNet◆[27]	0.7725	0.8327	0.8014	0.6312	0.7936	0.7031
MAD-GAN◆[22]	0.8049	0.8214	0.8131	0.8517	0.8991	0.8747
OmniAnomaly◆[41]	0.7416	0.9776	0.8434	0.8867	0.9117	0.8989
USAD◇[5]	0.9096	0.8529	0.8803	0.9308	0.8917	0.9108
LSTM-NDT◆[18]	0.8965	0.8846	0.8905	0.5934	0.5374	0.5640
GDN◇[11]	0.8932	0.8872	0.8902	0.9135	0.8612	0.8866
MTAD-GAT◆[48]	0.8906	0.9123	0.9013	0.8754	0.9440	0.9084
GTA◆[8]	0.8911	0.9176	0.9041	0.9104	0.9117	0.9111
MST-GAT◇[12]	0.9126	0.8983	0.9054	0.9506	0.8910	0.9198
DuoGAT	0.8634	1.000	**0.9267** (2.35%)	0.9271	0.9538	**0.9403** (2.23%)

Table 3: Performance comparison of DuoGAT and the competitors on SMAP and MSL. The best performance by F1-score is shown in bold. The percentage next to DuoGAT's F1-score represents improvement over the second best performer whose score is underlined. ◆ : Results from GTA [8]. ◇ : Results from MST-GAT [12].

end, we made the following ablations: DuoGAT *without the D-GAT layer* that learns from the additional data produced by differencing (denoted as -w/o D-GAT), DuoGAT *without the multi-dimensional attention* for either or both the T-GAT and D-GAT layers (denoted as -w/o T&D-multi, -w/o T-multi, and -w/o D-multi), DuoGAT *without our weighted and directed graph*, which uses the undirected, unweighted and complete graph structure instead (denoted as -w/o weight), and embedding dimension experiments. The results are summarized in Tables 4, 5 and 6.

From the results, we have the following observations:

- We observed a performance decrease when DuoGAT neglects the differenced time-series data in all dataset. This result sheds light on the importance of additionally learning data generated from differencing, which highlights the changes in sensor values,

Method	SWaT			WADI		
	Prec	Rec	F1	Prec	Rec	F1
DuoGAT	0.971	0.943	0.96	0.894	0.978	0.93
- w/o D-GAT	0.971	0.922	0.946	0.822	1.00	0.902
- w/o T&D-multi	0.940	0.927	0.934	0.774	0.980	0.865
– w/o T-multi	0.935	0.953	0.943	0.789	0.980	0.874
– w/o D-multi	0.918	0.960	0.938	0.864	0.980	0.918
- w/o weight	0.975	0.911	0.942	0.649	0.916	0.760

Table 4: Ablation study on SWaT and WADI.

Method	SMAP			MSL		
	Prec	Rec	F1	Prec	Rec	F1
DuoGAT	0.863	1.000	0.927	0.927	0.954	0.940
- w/o D-GAT	0.841	0.858	0.849	0.878	0.939	0.907
- w/o T&D-multi	0.783	0.926	0.849	0.859	0.939	0.898
– w/o T-multi	0.813	0.738	0.774	0.857	0.935	0.894
– w/o D-multi	0.877	0.857	0.867	0.880	0.939	0.909
- w/o weight	0.827	0.867	0.846	0.927	0.903	0.915

Table 5: Ablation study on SMAP and MSL.

in the anomaly detection task. This especially helps capture sudden changes of features and thereby improves the accurate of anomaly detection.

- It was also observed that the multi-dimensional attention approach is very effective in anomaly detection: we can see the degraded performance in all cases where the multi-dimensional attention is not applied. This results confirm that considering various perspectives from multiple dimensions of embedding spaces helps clearly learn the relationships between time points in time-series data.
- We also confirm that our graph structure, which is directed and weighted, can properly reflect the characteristics of time whereas the previous undirected and unweighted graph: we observed that the detection performance decreased when DuoGAT adopted the undirected and unweighted version.
- To investigate the impact of embedding dimensions on detection performance, we performed additional experiments with the various embedding dimension sets q: we observed that using two embedding dimensions ({25%, 50%}) together with the original embedding size (100%) shows the best performance. We also confirmed the superiority of our method compared to the original multi-head attention mechanism using the same dimensions.

We believe that the results of our ablation studies shed light on the importance of reflecting temporal characteristics to graph structures, learning from data of differenced time-series, and the usage of multi-dimentional attention, and suggest potential research directions towards developing more effective detector of time-series anomalies.

5.3 Training Time Comparison

The experiments in this subsection aims to verify the training efficiency of our DuoGAT. We compared the training time cost of DuoGAT with several GNN-based methods. For fair comparison, we set all the environments related to the model training such as the window size and the batch size (these are set to 10 and 256, respectively) to be identical. We also used the same computer for running all the compared methods. We measured the average time cost per epoch, aggregated from running 10 epochs.

Embedding dimensions (q)	SWaT	WADI	SMAP	MSL
{25%}	0.938	0.917	0.831	0.909
{50%}	0.937	0.924	0.841	0.913
{25%, 50%}	**0.957**	**0.935**	**0.927**	**0.940**
{25%, 50%, 75%}	0.930	0.910	0.811	0.925
Multi-head	0.941	0.912	0.852	0.913

Table 6: Results on different sets of embedding dimensions.

Methods	SWaT	WADI	SMAP	MSL
GDN [11]	38.42s	254.11s	8.03s	4.74s
MTAD-GAT [48]	57.01s	653.98s	**6.43s**	7.56s
GTA [8]	351.26s	846.13s	157.24s	46.55s
DuoGAT	**33.50s**	**88.08s**	8.70s	**3.87s**

Table 7: Average training time per epoch (in seconds).

Table 7 reports the result. DuoGAT is generally performed most efficiently compared to the other GNN-based methods. One notable point is that our DuoGAT can be trained much faster than GAT, almost 10 ~ 20 times faster, which is one of the strongest competitors in terms of the anomaly detection accuracy. GTA involves quite a lot components such as dilated convolution, graph convolution, the three encoder layers and one decoder layer. On the contrary, DuoGAT consists of fewer components than GTA, such as T-GAT and D-GAT layers and the GRU layer, but exhibits higher accuracy while consuming much less training time.

6 EXPLAINABILITY

Recently, various approaches have been studied to explain deep learning models, and a number of explanation methods for GNN have been studied [17, 23, 36, 37, 45]. In particular, in the field of multivariate time-series anomaly detection, explanation is mainly attempted through the attention mechanism: it provides an explanation via comparing the attention scores of normal and anomaly to specify the certain sensor causing the anomaly [11, 48].

However, such explanations lack the continual temporal dependencies between the anomaly detection and its previous data stream. For providing explanations that is more suitable for time-series data, we suggest a novel explanation method that follows the idea of perturbation. Unlike the existing methods that identify a specific sensor, our explanation can recursively point out time points that lead to anomaly detection based on the influence between time points.

As previously mentioned our graph structure reflects the characteristics of time that an event occurring at a certain time point is influenced by the behavior of the adjacent previous time points. In order to quantitatively measure the influence, we apply the masking scheme to each of the adjacent previous time points, in order to derive the importance score of each time point on causing the detected anomaly. In addition, this process can be recursively performed to grasp the cascading between time points, which also reflects the sequential characteristic of time-series. As a result, the

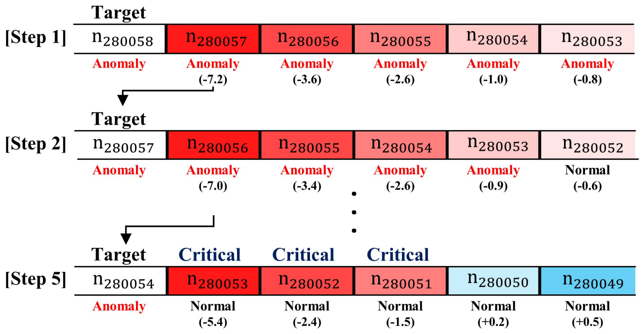

Figure 3: The process of tracking critical time points for detected anomaly on SWaT. n_{280058} is the target index number on SWaT test set. The intensity of the color is determined in proportion to the importance score and is indicated by a number below the label.

critical time points can be recursively tracked, starting from a detected anomaly to the time point when the importance score is the highest, which will be repeated a predetermined number of times. Finally the overview of the chain of these critical time points can provide an explanation. Since anomalous observations often occur in the form of contiguous anomaly segments, which means that the first anomaly in a segment can trigger other anomalies in the same segment, our explanations have ability to highlight this pattern.

In summary, the process of our proposed explanation method is as follows:

(i) A specific anomaly detected by our DuoGAT is selected as the explanation target.
(ii) Edge masking is performed on the nodes (i.e. timestamps in a window) in our two graphs.
(iii) The importance score of each time point is calculated by the difference in anomaly score before and after masking the corresponding edge. After then, the time point with the highest importance score is selected as the *next explanation target.*
(iv) The process from (ii) to (iii) is repeated *n* times.

The importance score calculated from (ii) is the difference between the values before and after masking. It is categorized into negative, positive, and zero. A negative number represents the fact that our model predicted the explanation target a little closer to the anomaly due to the masked time point, which had a positive effect on detecting the anomaly by making it easier to detect. A positive number represents the fact that our model predicted the explanation target slightly closer to normal due to the masked time point, which had a negative effect on detecting the anomaly by making it difficult to detect. Zero represents that the masked time point had no effect on detecting the anomaly. In this way the suggested explanation method can provide an easily understandable explanations for anomaly occurrence.

As a working example, Figure 3 illustrates a real case obtained from a detection of abnormal time n_{280058} in the test set of SWaT. From the anomaly detected by our DuoGAT, we computed the importance scores of each time point and found out that n_{280053}, n_{280052}, and n_{280051} are critical time points for detected anomaly. From these points, we recursively traced the cause of anomaly occurrence along the time steps, assuming that it was triggered by a cascade of events rather than a single time point.

Discussion: However, there may be a sudden spike in the time series due to various unpredictable fluctuations such as external shocks to sensors, etc. It is indeed an anomaly, but the previous time points may not have influenced it. However, in such scenarios, our explanation method may try to recursively trace the cause of the anomaly. Therefore, our explanation method may not be suitable for environments where such sudden spikes occur frequently. In addition, our explanation works on a time-oriented graph. If it is used for GNN-based detectors that operate on feature-oriented graphs, it will not work. Our future work is to design an anomaly detector that uses both time-oriented and feature-oriented graphs like MTAD-GAT [48], and to develop an explanation method that can track both the time points and the variables that affect the occurrence of anomalies.

7 CONCLUSIONS

In this paper, we proposed DuoGAT, an accurate, efficient and explainable anomaly detector for multivariate time-series data. Our work starts with a careful design of graph structure that can properly model a given time-series. Our model consists of two GAT layers that aim to capture the characteristics of time-series: a Time-oriented Graph Attention (T-GAT) layer that learns relationships between time points and a Differencing-based Graph Attention (D-GAT) layer that attentively learns changes in sensor values over time. We also adopted the multi-dimensional attention in the dual GAT layers for further improvements. We conducted extensive experiments using four real-world dataset (SWaT, WADI, SMAP and MSL). Empirically, DuoGAT outperformed other state-of-the-arts, especially improved GTA, the state-of-the-art in this area, by a wide margin. Furthermore, the training time of DuoGAT is shown to be faster than the compared GNN-based methods, which means that our framework enabled faster deployment at model serving level for anomaly detection tasks in the real-world environment with significantly less number of parameters compared to the other GNN-based models. Our ablation study confirms that each idea of DuoGAT is really effective in improving the detection performance. On top of DuoGAT, we present an explanation method based on a masking approach. Decision making can be assisted by our proposed explanation with visualized importance scores through highlighting possible causes of an anomaly.

ACKNOWLEDGMENTS

This work was partly supported by (1) commissioned research project supported by the affiliated institute of ETRI[2022-062], (2) the DGIST R&D program of the Ministry of Science and ICT of KOREA (23-IT-10-03), (3) Samsung Electronics Co., Ltd. and (4) Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373,Artificial Intelligence Graduate School Program (Hanyang University)).

REFERENCES

[1] Charu C Aggarwal and Charu C Aggarwal. 2017. *An introduction to outlier analysis.* Springer. 236–265 pages.

[2] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. 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.* 25–28.

[3] Fabrizio Angiulli and Clara Pizzuti. 2002. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery.* Springer, 15–27.

[4] Elena-Simona Apostol, Ciprian-Octavian Truică, Florin Pop, and Christian Esposito. 2021. Change point enhanced anomaly detection for IoT time series data. *Water* 13, 12 (2021), 1633.

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

[6] Md Abul Bashar and Richi Nayak. 2020. TAnoGAN: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 1778–1785.

[7] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

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

[9] Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*.

[10] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. 2019. Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal* 7, 7 (2019), 6481–6494.

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

[12] Chaoyue Ding, Shiliang Sun, and Jing Zhao. 2023. MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection. *Information Fusion* 89 (2023), 527–536.

[13] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[14] Wayne A Fuller. 2009. *Introduction to statistical time series.* John Wiley & Sons.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[16] Mahmudul Hasan, Md Milon Islam, Md Ishrak Islam Zarif, and MMA Hashem. 2019. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things* 7 (2019), 100059.

[17] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. 2022. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering* (2022).

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

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* 157–166.

[21] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. 2018. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758* (2018).

[22] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks.* Springer, 703–716.

[23] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems* 33 (2020), 19620–19631.

[24] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).

[25] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series.. In *ESANN*, Vol. 2015. 89.

[26] Aditya P Mathur and Nils Ole Tippenhauer. 2016. 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).* IEEE, 31–36.

[27] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089* (2018).

[28] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. 2018. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 2923–2960.

[29] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. 2015. *Introduction to time series analysis and forecasting.* John Wiley & Sons.

[30] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access* 7 (2018), 1991–2005.

[31] Peggy O'Neill, Dara Entekhabi, Eni Njoku, and Kent Kellogg. 2010. The NASA soil moisture active passive (SMAP) mission: Overview. In *2010 IEEE International Geoscience and Remote Sensing Symposium.* IEEE, 3236–3239.

[32] Jesus Pacheco and Salim Hariri. 2018. Anomaly behavior analysis for IoT sensors. *Transactions on Emerging Telecommunications Technologies* 29, 4 (2018), e3188.

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

[34] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining.* 3009–3017.

[35] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[36] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2018. Interpreting graph neural networks for nlp with differentiable edge masking. In *Proceedings of the international conference on learning representations.*

[37] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schütt, Klaus-Robert Müller, and Grégoire Montavon. 2021. Higher-order explanations of graph neural networks via relevant walks. *IEEE transactions on pattern analysis and machine intelligence* 44, 11 (2021), 7581–7596.

[38] Anita Sengupta, Adam Steltzner, Al Witkowski, and Jerry Rowan. 2007. An overview of the Mars Science Laboratory parachute decelerator system. In *2007 IEEE Aerospace Conference.* IEEE, 1–8.

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

[40] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 2003. *A novel anomaly detection scheme based on principal component classifier.* Technical Report. Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering.

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

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

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[44] Martin Wollschlaeger, Thilo Sauter, and Juergen Jasperneite. 2017. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE industrial electronics magazine* 11, 1 (2017), 17–27.

[45] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019).

[46] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1409–1416.

[47] Yin Zhang, Zihui Ge, Albert Greenberg, and Matthew Roughan. 2005. Network anomography. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement.* 30–30.

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

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