Online Anomaly Detection in Industrial IoT Networks Using a Supervised Contrastive Learning-Based Spatiotemporal Variational Autoencoder

Lun Tang, Ruiyu Wei, Bingsen Xia, Yuanchun Tang, Weili Wang, Qiong Huang, and Qianbin Chen, *Senior Member, IEEE*

***Abstract*—As industrial IoT networks evolve, they become increasingly vulnerable to cyberattacks, such as Denial of Service and backdoor attacks, which lead to anomalies in data streams (e.g., unusual spikes or drops in traffic, sudden changes in device behavior, or irregular communication patterns). To address the challenge of detecting these anomalies amidst dynamic data distributions and diverse abnormal patterns, this article proposes a supervised contrastive learning-based spatiotemporal variational autoencoder (SC-STVAE) for anomaly detection in online data streams. A multihead graph attention network (MD-GAT) is utilized to capture feature correlations, while a temporal convolution network serves as the hidden layer in the variational autoencoder. This enables SC-STVAE to learn both feature correlations and temporal dependencies. To resolve the issue of ambiguous positive and negative boundaries, supervised contrastive learning is introduced within the STVAE, improving boundary distinction and detection accuracy. To mitigate performance degradation due to data drift, an event-triggered elastic weight consolidation algorithm is introduced, which updates model parameters based on reliability thresholds. Additionally, a fuzzy entropy-weighted anomaly score, which measures the error between reconstructed data and original inputs by computing the weighted sum of the mean squared error across each dimension, is introduced. Experimental results demonstrate superior performance in terms of accuracy, recall, and F1 score compared to benchmark algorithms.**

***Index Terms*—Anomaly detection, industrial Internet of Things (IIoT), lifelong learning, supervised contrastive learning, variational autoencoder (VAE).**

I. Introduction

THE INDUSTRIAL Internet of Things (IIoT) merges information technology with industrial manufacturing, leveraging Internet of Things (IoT) technologies to connect equipment, sensors, systems, and personnel in production and operations [1]. Despite its anticipated rapid growth, the widespread deployment of IIoT devices in insecure environments, combined with diverse communication protocols and constrained device resources, presents significant security challenges. Network packets transmitted by IIoT devices and processed through edge nodes are particularly vulnerable to network-layer attacks, such as distributed Denial of Service (DDoS) attacks. These attacks, among the most prevalent in IIoT environments, severely disrupt services and result in considerable economic losses [2].

Anomaly detection methods seek to uncover and recognize unusual activities within a system to prevent significant failures. These anomalies are often the result of noise, hacking tools, or other external factors [3]. Specifically, anomalies could include unexpected traffic spikes due to DDoS attacks, unauthorized access attempts identified through unusual login patterns, or abnormal communication behaviors caused by malware or hacking tools. Such abnormal activities induced by attackers typically leave identifiable traces in infected networks, including altered traffic flow, unusual data packet structures, or irregular access times, which can be leveraged to detect both known and unknown attacks. Reconstruction-based anomaly detection methods monitor network traffic in real time by learning the distribution of normal data generated by legitimate devices. If the reconstruction of network data exhibits significant deviations, it indicates unfamiliar data that may suggest the presence of an intruder [4].

Building on this, in time series anomaly detection, integrating feature correlations, such as spatial dependencies, significantly improves model performance by capturing complex interdependencies among features. Real-world data often exhibit intricate relationships between features, especially in dynamic environments where both spatial and temporal factors influence the data. By considering both dimensions, models can more effectively capture these dependencies, leading to enhanced detection accuracy.

Received 13 September 2024; revised 26 December 2024; accepted 26 January 2025. Date of publication 13 February 2025; date of current version 23 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62071078 and Grant 62401091, and in part by the China Postdoctoral Science Foundation (CPSF) under Grant 2024MD754041. *(Corresponding author: Bingsen Xia.)*

Lun Tang, Ruiyu Wei, Weili Wang, Qiong Huang, and Qianbin Chen are with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the Chongqing Key Laboratory of Mobile Communications Technology, Chongqing 400065, China (e-mail: tangluncq@163.com; s220132176@stu.cqupt.edu.cn; wangwl@cqupt.edu.cn; huangqiong@cqupt.edu.cn; chenqb@cqupt.edu.cn).

Bingsen Xia and Yuanchun Tang are with the State Grid Fujian Electric Power Company Ltd., Economic and Technical Research Institute, Fuzhou 350013, China (e-mail: xiabingsen3@126.com; 13220325314@163.com).

Digital Object Identifier 10.1109/JIOT.2025.3537864

Reconstruction models aim to replicate data distribution rather than detect anomalies. Prior studies [5] focused on enhancing reconstruction ability but overlooked distinguishing anomalous data. When anomalies closely resemble normal data, these methods struggle to identify them [6], resulting in detection errors.

Thus, this article tackles the following issues. 1) How to improve the model's capability to accurately distinguish between normal and abnormal instances within the latent space? 2) How to address the varying importance of different features for anomaly detection? 3) How to mitigate the decline in detection accuracy due to data drift? To tackle these challenges, an online anomaly detection method for IIoT network data streams is proposed, leveraging a spatiotemporal variational autoencoder variational autoencoder (VAE) combined with supervised contrastive learning. The main contributions of this article are as follows.

1) To improve latent space representations for anomaly detection, a supervised contrastive learning-based spatiotemporal variational autoencoder (SC-STVAE) is proposed. The model integrates MD-GAT to capture feature correlations and a temporal convolutional network (TCN) to extract temporal patterns. Supervised contrastive learning ensures latent space boundaries by bringing positive samples (from the same batch) closer and separating negative samples (generated through random perturbations). This approach enhances the model's ability to distinguish normal from anomalous instances, significantly boosting detection accuracy.
2) To address the varying contributions of features to anomaly detection, a fuzzy entropy-based weighted anomaly score method is proposed. By calculating fuzzy entropy, feature importance is measured, and corresponding weights are assigned, ensuring each feature's impact is accurately reflected in the anomaly detection process.
3) To mitigate performance degradation caused by data drift, an event-triggered elastic weight consolidation (EWC) method is introduced. Model reliability is monitored using the Hoeffding inequality, and when reliability drops below a threshold, parameter updates are triggered. Regularization is added to preserve essential parameters from prior tasks, preventing catastrophic forgetting and enabling rapid adaptation to new data.

The structure of this article is as follows. Section II reviews the related work, while Section III outlines the problem, the system model, and details the offline training and online anomaly detection phases of the algorithm. Section IV presents the simulation results, and Section V concludes this article.

II. Related Works

A. Anomaly Detection

Anomaly detection methods can be broadly grouped into two main types: 1) those that are supervised and 2) those that are unsupervised. Given the scarcity of labeled anomalies in practical applications, the use of supervised methods is restricted. Unsupervised methods typically detect anomalies based on prediction or reconstruction errors. For instance, In [7], two parallel graph attention layers were employed to capture intricate causal relationships among data. It optimizes both prediction and reconstruction models to enhance representations, and uses the forecasting errors and reconstruction probabilities to calculate inference scores for anomaly detection. In [8], long short-term memory (LSTM) networks were combined with a VAE, using an LSTM-based encoder to simultaneously encode multivariate time series data along with the sequential dependencies into a latent representation, capturing both variable and temporal information to accomplish anomaly detection tasks. While these methods demonstrated effective representation learning, in [9], a more efficient gated recurrent unit (GRU) network replaced LSTM, and smoothing regularization was added to enhance the model's robustness, allowing it to handle partial anomalies in the input data. In [10], a model for detecting and diagnosing anomalies (TranAD) was proposed, built on deep Transformer networks, leveraging an attention-driven sequence encoder to effectively extract details from time-based patterns. It further implemented a dynamic focal score technique to capture resilient multifaceted features and used adversarial learning to guarantee model consistency. Expanding the scope to include spatiotemporal anomaly detection, in [11], a multichannel semi-supervised anomaly detection algorithm (MCBiWGAN-GTN) is proposed, leveraging BiWGAN-GTN for spatiotemporal feature extraction and CEEMDAN for reducing data complexity in KPI data from cloud data centers. In [12], a stochastic recurrent neural network (OmniAnomaly) was proposed, which modeled temporal dependencies and stochasticity in the data through stochastic variable connections and planar normalizing flows. In [13], an integrated deep learning model combining LSTM and autoencoder (AE) architectures was proposed to address the imbalance in datasets. The model effectively learns normal data patterns using LSTM and reduces data dimensionality by identifying key features with AE, achieving better performance compared to traditional methods. Further advancing transfer learning applications in anomaly detection, in [14], a transformer-based transfer learning model (IDS-INT) was proposed. By leveraging SMOTE to balance traffic and employing a CNN-LSTM hybrid model to detect attacks, the approach effectively handles feature complexity and data imbalance, achieving better performance compared to traditional methods. In [15], the anomaly Transformer explored the self-attention weights to detecting anomalies and introduced a min-max training method to enhance the separation between normal and abnormal data.

B. Contrastive Learning

Contrastive learning seeks to bring together alike instances within the representation space while pushing apart distinct instances within the same area. For example, in [16], autoregressive models were used to forecast upcoming the latent space values, and the InfoNCE loss function was introduced to enhance the model's ability to produce strong data representations. Building on this approach, in [17], tackled the

issue of acquiring negative samples by introducing a dynamic dictionary that retrieves pairs of data points for comparison in contrastive learning. In [18], the critical importance of expanding the dataset in contrastive learning was emphasized, introducing a learnable nonlinear mapping approach designed to boost the effectiveness of the produced contrastive feature vectors. Furthermore, in [19], a hierarchical contrastive strategy was introduced to capture multilevel contextual insights from time series data. In [20], contrastive learning was employed for time series data, introducing an unsupervised architecture that utilizes diverse contrasting approaches to extract meaningful representations from time series data. To introduce more information, In [21], the CAE-AD introduced a multigranular contrastive approach to extract normal data patterns. In [22], a new self-adversarial variational AE incorporating contrastive learning was introduced. This model treats reconstructions of normal samples as reference samples and reconstructions of anomalous samples as contrastive samples, allowing it to use contrastive learning to effectively distinguish between them. This approach measures the similarity between inputs and positive instances, in addition to the difference between inputs and negative instances. Furthermore, in [23] proposed an adversarial contrastive AE (ACAE), where merging and breaking down features are used as surrogate tasks for contrastive learning, and adversarial training is introduced to learn invariant representations for multivariate time series, resulting in robust representations. Contrastive learning often utilizes data transformation methods to produce various perspectives of the same instance, training the model by reducing the disparity between these perspectives. However, this may lead the model to ignore the relationships between the instance and other similar instances within the same category, potentially impeding its capacity to learn intraclass coherence and making it difficult to differentiate between distinct categories in the feature space.

C. Lifelong Learning

Lifelong learning, or continual learning, is an effective approach to enabling opaque models to continuously learn and adapt over time. It emphasizes learning from new data while integrating new knowledge without catastrophic forgetting. As such, the lifelong learning theory is particularly well-suited for scenarios involving online data streams, where anomaly detection models continually discover new information from real-time sensor data streams and adaptively update themselves. Several studies have emerged in this area. For instance, in [24], an innovative lifelong approach was introduced. This method integrates continuous changepoint detection with effective model updating techniques, leveraging mechanisms, such as experience replay and a structured memory system, which is sustained through processes like integration and abstraction. In [25], a biologically inspired lifelong anomaly detection approach was introduced, which adjusts to changing conditions and efficiently retrieves past data from its memory repository for model updates. In [26], a real-time learning and diagnostic technique utilizing flexible neural networks was introduced. In this method, internal nodes and output units are dynamically incorporated when new process faults are identified, allowing the system to adjust to emerging fault types. In [27], an advanced AE with incremental learning features was presented. This approach stacks multiple AEs horizontally to enable incremental fault detection (IFD). To tackle the challenge of diagnosing bearing faults with evolving fault modes, to tackle more complex challenges, in [28], a lifelong learning technique leveraging generative feature replay was introduced. This approach utilizes a replay alignment loss to integrate features generated by the model with those from the current task, successfully addressing issues of class imbalance and minimizing the risk of catastrophic forgetting. Finally, in [29], a model for continuous learning that relies on meta-representation in weight space was presented. This approach maintains diagnostic accuracy efficiently by deducing and upholding the distribution of meta-representations, thus ensuring low-cost performance preservation. In the implementation of lifelong learning, the most common update trigger mechanism is periodic time-based triggers. This approach activates model updates at fixed time intervals offering operational simplicity but also presenting the risk of unnecessary resource consumption.

III. An Online Anomaly Detection Method for Data Streams in Industrial IoT Networks Based on Supervised Contrastive Learning and Spatiotemporal Variational Autoencoders

A. Problem Description

In IIoT environments, devices, sensors, and edge nodes are deployed across large-scale, distributed networks that monitor and control critical industrial processes, including manufacturing, energy management, and transportation systems. These networks are inherently susceptible to various network-layer attacks, such as DDoS attacks, unauthorized access attempts, and data manipulation, which can significantly disrupt operations and undermine system integrity. The dynamic nature of IIoT systems, where devices continuously generate and transmit data in real time, further complicates anomaly detection. The data distribution within these systems is highly variable, with traffic patterns fluctuating frequently due to factors like operational conditions, device behavior, and external influences. In multivariate time series anomaly detection, the primary objective is to identify unusual patterns at the entity level, rather than focusing solely on individual variables [12]. Anomaly detection in this context involves identifying samples in real-world datasets that deviate significantly from other observations. The anomaly detection problem can be formulated as follows: during the training phase, it is generally assumed that no anomalies are present, and the model is trained using data collected from IIoT devices. The training data consists of N univariate time series with T timestamps, denoted as $X = [x_1, x_2, \ldots, x_T] \in R^{N*T}$, $x_t = [x_t^1, x_t^2, x_t^i, \ldots, x_t^n] \in R^N, i = 1, 2, \ldots, n$ at t timestamp.where x_t^i represents the value observed by the ith sensor at time t. During the detection phase, the objective is to identify anomalies in the data stream. The model's final output

is $y_i \in \{0, 1\}$, where $y_i = 1$ signifies that the observation at time t is identified as anomalous.

B. Method Description

In the context of IIoT networks, online anomaly detection in data streams presents significant complexity and variability. Anomalous behavior is often determined by a combination of multiple spatial features, making single-feature analysis insufficient for accurate anomaly detection. Therefore, the detection process must consider the interrelationships among various spatial features.

Initially, the raw data is normalized to ensure uniform feature scaling, thereby eliminating dimensional disparities. Each feature is treated as an independent node to construct a graph structure. The fast dynamic time warping (Fast-DTW) algorithm is then employed to compute the correlations between features, producing an adjacency matrix representing the feature relationships. This adjacency matrix, along with the preprocessed data, is subsequently input into the MD-GAT network to generate outputs $\tilde{X}$ that incorporate spatial information.

Next, the processed data is passed through a VAE with TCNs to extract temporal information and generate latent representations. To enhance the model's discriminative capability, supervised contrastive learning is employed to constrain the generated latent representations, thus facilitating a clearer separation between normal and anomalous samples.

In the online anomaly detection process, the model undergoes initial training in an offline batch mode before transitioning to the online detection phase. Given the dynamic nature of data distribution, the model's anomaly detection performance may degrade over time. To maintain adaptability and stability, the model is updated using the EWC method, which is triggered by events. The process begins with anomaly detection in the data stream using the offline-trained model. When the model's reliability, assessed via the Hoeffding inequality, falls below a predefined threshold, the model parameters are updated through EWC. Following this update, the model rebuilds the data using its updated parameters.

To evaluate the contribution of each feature to anomaly detection, fuzzy entropy is utilized, and the errors across various dimensions are weighted and combined to produce an anomaly score. The generated score is then assessed against a predefined threshold, where values exceeding the threshold are identified as anomalies. The overall structure of the proposed method is illustrated in Fig. 1 below.

C. Offline Training Phase

1) STVAE: Analyzing individual features in multidimensional data is often insufficient for accurate anomaly detection. Therefore, identifying the relationships between features in multivariate time series is crucial, and this should be achieved without relying on predefined assumptions. In this model, each feature is represented as an individual node, with edges representing the relationships among features. We employ an undirected graph based on a dynamic adjacency matrix, denoted as $A = (V, E)$, where $V = [v_1, v_2, v_3, \ldots, v_N]$ denotes the collection of nodes (features) and $E = [e_{ij}] \in R^{N*N}$ represents the set of edges. Specifically,where e_{ij} indicates a connection between node v_i to node v_j, denoting a relationship between these two features. DTW is a commonly utilized method for assessing the similarity between time series. It leverages dynamic programming to compute the alignment between two time series, allowing for nonlinear mapping between their time points. However, DTW encounters significant computational challenges, rendering it less practical for large-scale datasets or real-time use cases. To address this issue, Fast-DTW was developed, which improves upon DTW by reducing the search space and shortening the sequence length, significantly improving efficiency. Fast-DTW operates by initially coarsening the original sequences through iterative data extraction, where multiple fine-grained data points are averaged to generate a coarser representation. The DTW algorithm is then applied to these coarse-grained sequences. After the DTW computation on the reduced data, the coarse sequences are reverted to their original granularity, yielding the similarity distance between the two time series

$$D_{i,j} = \text{Fast\_DTW}\left(x_i^T, x_j^T\right). \tag{1}$$

The notation $D_{i,j}$ denotes the measure of dissimilarity between the time series of variables i and j, computed using the Fast-DTW algorithm. A larger distance $D_{i,j}$ indicates a weaker correlation between variables i and j. A correlation threshold ε, is set to determine the relationship between features. If the DTW distance between two features exceeds this threshold, i.e., $|D_{i,j}| \geq \varepsilon$, no connection is established between variables i and j. Otherwise, a link is formed between the two variables, represented as follows:

$$e_{ij} = \begin{cases} 1, & D_{i,j} > \varepsilon \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

To manage the high computational and memory demands of MD-GAT's parallel operations, the graph's sparsity is regulated by modifying the threshold ε. This approach ensures that computational efficiency and resource usage remain optimized without compromising the accuracy of anomaly detection After the dynamic generation of the graph structure, the connections between adjacent nodes are modeled using the graph attention layer (GAT). The GAT can capture the relationships between nodes in various graph configurations. For a graph with N nodes, denoted as $x_1, x_2, \ldots, x_N$, where h_i signifies the feature vector for each node i, the GAT layer computes each node's output representation through the following process:

$$h_i = \sigma\left(\sum_{j=1}^{L} \alpha_{ij} x_j\right) \tag{3}$$

where h_i represents the output representation of node i, σ denotes the sigmoid activation function, and α_{ij} denotes the attention score, which quantifies the influence of node j on node i, with j being one of the neighboring nodes of i, where L indicates the total count of neighboring nodes around i. The attention score α_{ij} is computed using the following formula:

$$e_{ij} = \text{LeakyReLU}\left(\omega^T \cdot \left(x_i \oplus x_j\right)\right) \tag{4}$$

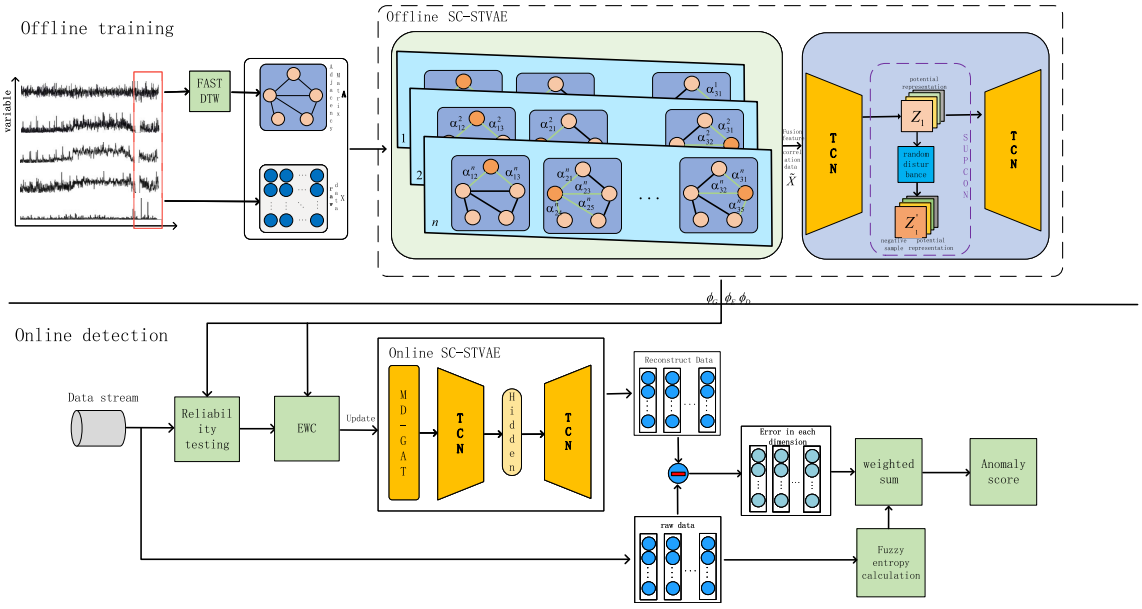

Fig. 1. System architecture.

$$\alpha_{ij} = \mathrm{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{j=1}^{L} \exp(e_{ij'})} \tag{5}$$

where $\oplus$ indicates the concatenation of the feature vectors from two nodes, and ω represents a trainable parameter vector, with $\omega \in \mathbb{R}^{2m}$, where m is the dimensionality of the node's feature vector. The activation function used is LeakyReLU, a type of nonlinear function [30].

The multihead attention mechanism is applied. Once the K-head attention mechanism processes and combines the feature vectors, the resulting output representation for the feature vector is

$$h_i' = \prod_{k=1}^{K} \sigma\left(\sum_{j' \in \mathcal{N}(i)} \alpha_{ij'}^{k} w^{k} h_{j'} \right) \tag{6}$$

where $\prod$ stands for the concatenation of vectors, while $\alpha_{ij'}^{k}$ indicates the attention score derived from the kth attention head, and w^{k} refers to the weight matrix used for linearly transforming the input vectors. As highlighted in [31], when multihead attention (MDA) is used in the final layer of the network, concatenation tends to deliver subpar results in this layer's output. Therefore, an averaging technique is utilized, and the resulting output from the final layer, computed through the MDA approach, is determined as follows:

$$h_i' = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j' \in \mathcal{N}(i)} \alpha_{ij'}^{k} w^{k} h_{j'} \right). \tag{7}$$

In the output of the MD-GAT mechanism, richer spatial correlation information is extracted by integrating the relationships with other nodes. By feeding x and A into the graph attention network with multiple heads, the resulting data $\tilde{X}$ is produced, incorporating the correlations between the features

$$\tilde{X} = G(X, A; [W_G, b_G]) \tag{8}$$

where W_G and b_G represent the weights and biases of MD-GAT, respectively, which together form the parameters ϕ_G of MD-GAT.

The data, after fusing feature correlations, is passed as input to the TCN, replacing the hidden layer of the VAE. The VAE, a robust deep generative model, employs variational Bayesian inference to capture the latent probability distribution of data.

The VAE model operates by encoding input data x into a latent space, generating a mean and variance for the latent variable distribution z [32]. The latent vector is then resampled, and the decoder reconstructs the data based on this distribution. The goal of the encoder is to maximize the likelihood P_X of the original data, though direct calculation of P_X is computationally prohibitive. To approximate the true posterior probability $p_\varphi(x|z)$, the encoder computes $q_\theta(z|x)$, which is used by the decoder to reconstruct the data. The reparameterization trick, $z = \mu + \varepsilon * \sigma$, is employed to simplify gradient calculation, with μ and σ drawn from a normal distribution. The loss function is given as

$$L(\theta, \varphi) = E_{z\sim q}\left[\log\left(p_\varphi(x|z)\right)\right] - D_{\mathrm{KL}}\left[q_\theta(z|x)||p_\varphi(z)\right] \tag{9}$$

where θ refers to the encoder parameters, while φ pertains to those of the decoder, respectively. The first component represents the likelihood of reconstruction, which measures the loss incurred during data reconstruction. The second component involves the KL divergence, which assesses how closely the distributions align.

TCN is an 1-D convolutional neural network specifically designed for handling temporal data. Compared to various Recurrent Neural Network (RNN) structures, TCN has been found to achieve or even surpass RNN performance on many tasks. TCN has a longer receptive field and successfully identifies long-term relationships. Compared to RNN or LSTM, TCNs are adept at managing extended time series data with enhanced stability and efficiency. By employing causal convolutions, the output at each time step is influenced solely

by the current and preceding inputs. Preserving the natural order of time series data. Therefore, using TCN as the hidden layer in a VAE can help the VAE better handle time series data and adapt more effectively to data patterns during online detection. This combination results in an STVAE.

To Construct the Encoder: Map the fused feature correlations $\tilde{X}$ to a lower-dimensional representation Z

$$Z = \mathrm{TCN}\big(\tilde{X}; [W_{ET}, \mathrm{b}_{ET}]\big). \tag{10}$$

The TCN layer receives the input to produce a reduced-dimensional representation denoted as Z. The parameters of this layer include W_{ET}, the weight matrix, and b_{ET}, the bias vector. Together, these elements constitute the encoder's parameter set, ϕ_E.

To Construct the Decoder: This involves creating a network that mirrors the structure of the encoder in reverse. Specifically, the sequence of layers in the decoder mirrors that of the encoder E. Once the latent variables Z are obtained from the distribution P_z, they are fed into the decoder's TCN layer. This process reconstructs the original data, resulting in an output represented as $\hat{X}$

$$\hat{X} = \mathrm{TCN}(Z; [W_{DT}, \mathrm{b}_{DT}]). \tag{11}$$

W_{DT} and b_{DT} denote the weights and biases associated with the TCN layer of the generator, collectively constituting the parameters of the decoder, denoted as ϕ_D.

2) Spatiotemporal Variational AE With Supervised Contrastive Learning: Due to the issue of ambiguous boundaries between normal and anomalous samples in the latent space, the SC-STVAE model is introduced by incorporating supervised contrastive learning into the original STVAE framework. This modification aims to enhance the model's ability to distinctly separate normal and abnormal data points in the latent space, thereby improving the overall anomaly detection performance. By leveraging supervised contrastive learning, the SC-STVAE model constrains the latent representations, ensuring a clearer delineation between positive and negative samples, which is crucial for accurate anomaly detection in complex data streams.

Contrastive learning is a type of self-supervised learning approach that operates without the need for manually annotated category labels. Rather than relying on external labels, it leverages the inherent structure of the data itself to guide the learning process. This method aims to develop feature representations that are useful for subsequent tasks by comparing and contrasting samples. The fundamental principle behind contrastive learning is to focus on learning feature representations that are effective at distinguishing between different samples, rather than meticulously examining every detail of the data. The general framework of this approach can be outlined as follows:

$$\mathrm{score}\big(x, x^{+}\big) \gg \mathrm{score}\big(x, x^{-}\big) \tag{12}$$

where x^{+} denotes a positive sample that resembles x, while x^{-} represents a negative sample that is dissimilar to x. The score function quantifies the degree of similarity between these samples. The aim of contrastive learning is to train an encoder to ensure that the representation of x is closely aligned with positive examples and distinctly separated from negative examples.

Conventional contrastive learning methods generate pairs of similar and dissimilar samples by determining if the samples are derived from the same source or different ones. This approach does not account for correlations between different images belonging to the same class, which can lead to issues where samples from the same class might still be far apart. Therefore, [33] proposed supervised contrastive learning, where instances of the same category are treated as positive pairs and differentiated from the remaining instances in the batch, which are regarded as negative pairs. This modification has been shown to encourage the model to better capture intraclass similarity. This approach motivates the encoder to produce highly similar features for all items belonging to the same class, enhancing the effectiveness of clustering within the resulting representation space as compared to conventional contrastive learning methods. Given that anomaly detection can be framed as a binary classification task, leveraging supervised contrastive learning can refine the latent features produced by the VAE encoder.

For each anchor, the SupCon loss first calculates the similarity scores between it and all other samples of the same class, then performs a weighted sum of these scores.

Since anomaly detection is typically an unsupervised learning task with no available label information, pseudo-labeling is applied to the latent representations. Specifically, latent representations of time series from the same batch generated by the VAE encoder are labeled as positive samples, and noise is randomly added to the original generated latent representations, which are then treated as negative samples. The supervised contrastive loss

$$L_{\mathrm{sup}} = \frac{-1}{|p_i|} \sum_{p \in P_i} \log \frac{\exp\big(Z_i \cdot Z_p/\tau\big)}{\sum_{k \in K_i} \exp(Z_i \cdot Z_k/\tau)} \tag{13}$$

where Z_i represents the latent representation generated at the anchor point, Z_p represents the latent representations of other samples in the same batch excluding the anchor point, and Z_k represents the latent representations after random perturbation. p_i identifies the group of examples that share the same label as the one currently being analyzed in the batch. The term $|p_i|$ denotes the total count of items within this group, while K_i signifies the collection of negative pairings that are distinct from Z_k. τ is a temperature coefficient, a tunable parameter that, when smaller, puts more emphasis on distinguishing the sample from the most similar other samples.

Consequently, the total loss function comprises three components: the reconstruction error, the KL divergence penalty, and the supervised contrastive loss. The reconstruction error quantifies the discrepancy between the original data and the generated data, and is calculated using the mean-squared error (MSE) as follows:

$$L_{\mathrm{recon}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t}^{T} \left(\hat{x}_t^{(i)} - x_t^{(i)}\right)^2. \tag{14}$$

Here, N denotes the number of dimensions, while T refers to the entire time duration. $\hat{x}_t^{(i)}$ corresponds to the reconstructed data at time t, and $x_t^{(i)}$ refers to the original data at the same time step.

The KL divergence term quantifies the difference between the Gaussian distribution inferred by the encoder and the actual Gaussian distribution. The formula for the KL divergence loss is:

$$L_{\mathrm{KL}} = D_{\mathrm{KL}}\left(\mathcal{N}\left(z;\mu,\sigma^2\right)\|\mathcal{N}\left(z;\mu_a,\sigma_a^2\right)\right) = \log\frac{\sigma_a}{\sigma} + \frac{\sigma^2+(\mu-\mu_a)^2}{2\sigma_a^2} - \frac{1}{2}. \tag{15}$$

The total loss function for the offline training process is given by the following formula:

$$L_{\mathrm{off}} = L_{\mathrm{mse}} + L_{\mathrm{KL}} + \lambda L_{\mathrm{sup}} \tag{16}$$

where λ serves as a hyperparameter that controls the weight of each regularization term in the loss function. The model is trained using the Adam optimizer for efficient parameter optimization. The offline training procedure is depicted in Fig. 2.

Incorporating additional data into the training process, rather than solely relying on normal instances, is beneficial for enhancing the model. While negative examples may not always represent actual anomalies, the encoder's ability to distinguish them plays a crucial role. This restriction helps foster more meaningful and separable latent representations, leading to better quality in the reconstructed outputs. As a result, the model becomes more adept at distinguishing between normal and abnormal inputs during inference.

The offline training algorithm is shown in Algorithm 1. Algorithm 1 Offline Training Algorithm for the Spatiotemporal Variational AE Based on Supervised Contrastive Learning.

Algorithm 1 Offline Training Algorithm

1: **Input**: Training batch data X_{train}, batch size m, initialized MD-GAT network parameters ϕ_G, initialized encoder parameter ϕ_E, initialized decoder parameters ϕ_D, hyperparameters λ,temperature coefficient τ.
2: **Output**: Optimized MD-GAT network parameters ϕ_G, encoder parameters ϕ_E, decoder parameters ϕ_D.
3: **For** each batch **do** :
4: Randomly generate a batch of size m from the training data X_{train}.
5: Calculate the adjacency matrix according to equation (1).
6: Generate the spatially fused data $\tilde{X}$ through MD-GAT.
7: Generate latent representations Z according to equation (10).
8: Generate reconstructed data $\hat{X}$ according to equation (11).
9: Label the original latent representation Z as positive samples, and generate negative latent representations Z_k via random perturbations.
10: Compute the supervised contrastive loss according to equation (13).
11: Compute the reconstruction error loss according to equation (14).
12: Compute the KL divergence loss according to equation (15).
13: Calculate the total loss according to equation (16).
14: Update MD-GAT parameters ϕ_G, encoder parameters ϕ_E, and decoder parameters ϕ_D using the Adam optimizer.
15: **End For**

D. Online Anomaly Detection Phase

Once the original model is trained on historical data, it is deployed to identify anomalies in real-time data streams. The online data stream may experience data drift, which can cause the offline model's anomaly detection capability to gradually weaken. This article uses an event-triggered EWC method to continuously update the historical models during the detection process. By reconstructing the data with the new model and comparing it with the original data, errors across dimensions are obtained. Fuzzy entropy is subsequently employed to evaluate the impact of various features on detecting anomalies. Errors across different dimensions are aggregated and weighted to produce an anomaly metric. This metric is then assessed against a set threshold; if it surpasses this limit, it is flagged as an anomaly.

1) Event-Triggered EWC Model Update Method: When concept drift happens, the distribution of data within the model becomes misaligned with that of the incoming data, resulting in decreased effectiveness of the anomaly detection system. A key indicator of the model's effectiveness is the rate at which detection errors occur. However, the detection error rate is usually assessed in a supervised setting, which often requires known data labels or additional labeling efforts. In unsupervised online anomaly detection, true data labels are not available, making this approach ineffective. Therefore, an event-triggered EWC method is proposed for online model updating. By employing Hoeffding's inequality to assess the reliability of the model, the original model parameters are retained when the model demonstrates high reliability, thereby maintaining its anomaly detection capability. When the model's reliability falls below a predefined threshold, the Fisher matrix is utilized to measure the importance of model parameters, followed by parameter updates to adapt to new knowledge while preserving historical knowledge. The workflow of the proposed method is illustrated in the following Fig. 3.

Hoeffding's inequality has been extensively utilized in statistical approaches for identifying points of concept drift [34] and has proven to be effective. It is employed to assess the model's reliability. Denote the anomaly scores from the current batch as $S(B_{\mathrm{cur}})$, and those from the previous batch as $S(B_{\mathrm{pre}})$. The model's reliability can be quantified using the bound provided by Hoeffding's inequality. Considering that the anomaly scores S_{cur} and S_{pre} are drawn from $S(B_{\mathrm{cur}})$ and $S(B_{\mathrm{pre}})$, respectively, the bound is $[S_{\min}, S_{\max}]$. The model's

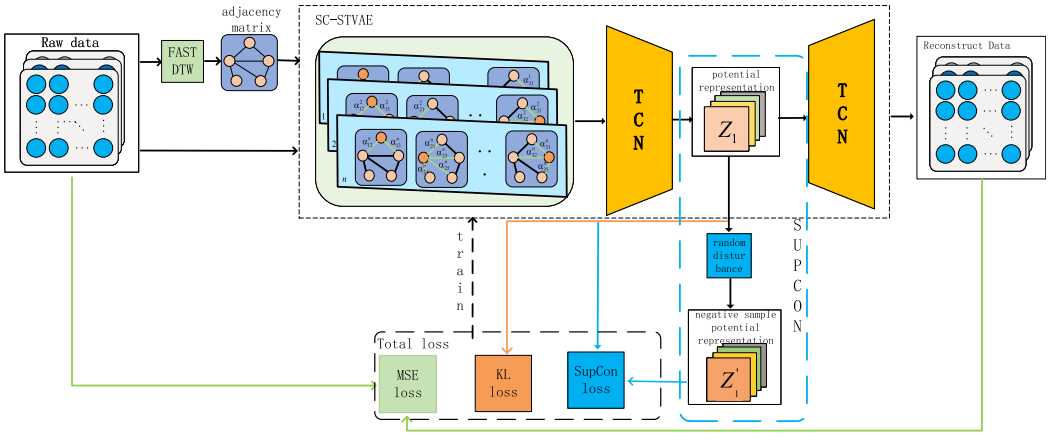

Fig. 2. Offline training process.

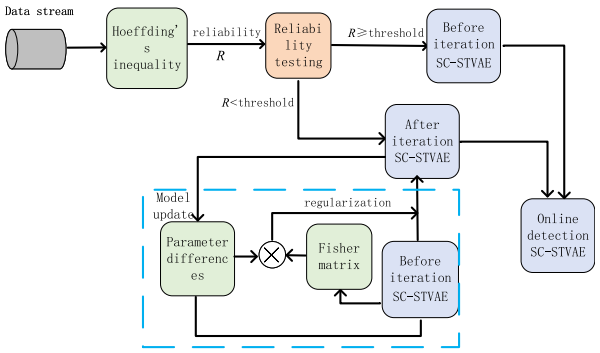

Fig. 3. Flowchart of elastic weight integration method based on event triggering.

reliability can be computed as follows:

$$pr\left\{\left|\overline{S}_{\text{cur}}-\overline{S}_{\text{pre}}\right|\geq\varepsilon\right\}\leq e^{\frac{-b\varepsilon^2}{(S_{\max}-S_{\min})^2}} \tag{17}$$

where $\overline{S}_{\text{cur}} = (1/b)\sum_{i=1}^{b} S_{\text{cur}}^{b}$, $\overline{S}_{\text{pre}} = (1/b)\sum_{i=1}^{b} S_{\text{pre}}^{b}$, $b = |B_{\text{cur}}| = |B_{\text{pre}}|$, $\varepsilon = |\text{avg}(S(B_{\text{cru}})) - \text{avg}(S(B_{\text{pre}}))|$, $S_{\min} = \min(\min(S(B_{\text{cur}})), \min(S(B_{\text{pre}})))$, and $S_{\max} = \max(\max(S(B_{\text{cur}})), \max(S(B_{\text{pre}})))$. the model's reliability R is given by the following formula:

$$R = e^{\frac{-b\varepsilon^2}{(S_{\max}-S_{\min})^2}}. \tag{18}$$

The model's reliability reflects the extent of data drift, and when the model's performance on a new data batch falls below a threshold α, model updates are triggered. During model updates, the importance of previous tasks is preserved by applying different constraints through regularization to mitigate the problem of catastrophic forgetting. Parameters that are more important for previous tasks are protected or frozen during the update process, while less important parameters are given greater plasticity and can be modified more extensively. A quadratic penalty is incorporated into the loss function to address the discrepancies between the previous model's parameters and those of the updated model, thereby minimizing adjustments to the task-specific weights that capture previously acquired knowledge.

The EWC mechanism primarily involves computing the Fisher information matrix and regularization. The Fisher information matrix quantifies the impact of parameters on the gradient of the objective function and can be used to quantify the importance of parameters in previous tasks. Regularization is calculated based on the Fisher information matrix and the model's prediction results, and these results are then added to the loss function to impose constraints on parameters highly relevant to previous tasks. These constraints determine the direction of model updates

$$\log p(X|q) = \log p(X_{\text{new}}|q) + \log p(q|X_{\text{old}}) - \log p(X_{\text{new}}) \tag{19}$$

where X_{old} represents the historical data, X_{new} represents the new data used for updates, and X is the set of both current and historical data. The goal of EWC is to maximize the posterior probability represented by the above formula. The intermediate term in this formula is difficult to compute explicitly, so EWC uses a Laplace approximation to handle this term, transforming it into a problem of solving the Fisher information matrix. By determining the second-order partial derivatives of the loss function for each model parameter, the Fisher information matrix F_i associated with θ_i is derived, as detailed below

$$F_i = E\left[\left(\frac{\partial^2 L_{\text{old}}(\theta_i|x_i)}{\partial\theta\partial\theta^T}\right)\right] \tag{20}$$

where $L_{\text{old}}(X_{\text{old}};\theta_i)$ represents the loss function corresponding to the historical data X_{old}, and x_i denotes the time series data within the historical data portion. Using the Fisher information matrix, a regularization component is introduced into the loss function to constrain the parameters. The lifelong learning loss of the model is then given by the following formula:

$$L'(\theta) = L_{\text{off}}(\theta) + \sum_i \frac{\eta}{2} F_i\left(\theta_i^* - \theta_i\right)^2 \tag{21}$$

where $L(\theta)$ represents the loss for the anomaly detection task, θ_i^* is the updated model parameters, and θ_i are the parameters learned from previous tasks. The hyperparameter η controls the strength of the regularization. Minimizing the loss function allows the model to preserve previously acquired knowledge while learning new tasks. This approach helps to reduce the risk of catastrophic forgetting and supports the

Algorithm 2 Event-Triggered Dynamic Weight Incremental Update Algorithm

Input: Data stream X_{new},Historical MD-GAT network parameters ϕ_G,Historical encoder parameters ϕ_E,Historical decoder parameters ϕ_D,Hyperparameters λ,Reliability threshold α.
Output: Updated MD-GAT network parameters ϕ_G^*,Updated encoder parameters ϕ_E^*,Updated decoder parameters ϕ_D^*.
Compute the model reliability $R$for the current batch using Equation (18).
if $R < \alpha$ **then** : // Reliability exceeds the threshold
    Calculate the Fisher Information Matrix using Equation (20).
    Calculate the online loss function using Equation (21).
    Update MD-GAT, encoder, and decoder parameters based on the online loss function using Equation (21).
    Return the updated MD-GAT network parameters ϕ_G^*, encoder parameters ϕ_E^*, and decoder parameters ϕ_D^*.
else:
    Return Return the historical MD-GAT network parameters ϕ_G, encoder parameters ϕ_E, and decoder parameters ϕ_D.
End if

ability of the model to continuously learn over time. The event-triggered dynamic weight increment update algorithm is shown in Algorithm 2.

2) Generation of Anomaly Scores Based on Fuzzy Entropy: In the anomaly detection stage, the input data is processed through the encoder, which transforms it into a latent space representation. This latent representation is then passed to the generator, which uses it to produce the reconstructed data. The anomaly score is determined by evaluating the discrepancy between the reconstructed output and the original input. Generally, this score is calculated by aggregating the discrepancies across all features, where each feature is given the same level of importance. This approach is clearly not ideal, as in some extreme cases, the anomaly of a sample can be determined by the reconstruction error of just a few categorical features. In real-world scenarios, various features usually have varying levels of impact on the overall anomaly detection in the dataset. Each feature may play a different role in highlighting the anomalies, leading to differences in how their contributions affect the overall detection outcome. Therefore, this article introduces anomaly score generation based on fuzzy entropy weighting.

Fuzzy entropy, a concept outlined in [35], is utilized to quantify aspects of uncertainty like ambiguity and inconsistency. In the context of anomaly detection, where the objective is to pinpoint infrequent instances exhibiting unusual behavior or properties, such uncertainty itself can be indicative of anomalies. Consequently, fuzzy entropy serves as a valuable tool for calculating the anomaly score by reflecting these uncertain features.

Fuzzy entropy is calculated as follows.

1) Given a time series $x_1, x_2, \ldots, x_T$ of length T, one can specify the embedding dimension l (where $l \leq T-2$) and determine the acceptable similarity margin r. Using these parameters, the phase space is then reconstructed for further analysis

$$X_i = \{x_i, x_{i+1}, \ldots, x_{i+l-1}\} - u_i \quad i = 1, 2, \ldots, T-l+1 \tag{22}$$

where

$$u(i) = \frac{1}{l}\sum_{k=0}^{l-1} x_{i+k}. \tag{23}$$

2) The distance between two vectors X_i and X_j can be described as the largest absolute difference between their respective elements. This can be expressed mathematically as

$$\text{Dist}_{ij}^m = \max_{k=0,1,\ldots,m-1}\left\{\left|(x_{i+k}-u_i)-\left(x_i-u_j\right)\right|\right\} \quad 1 \leq i,j \leq N-l+1 \quad (i \neq j). \tag{24}$$

3) A fuzzy membership function is employed to quantify the similarity between the vectors $X(i)$ and $X(j)$. It is expressed as follows:

$$A_{ij}^l = \begin{cases} 1, \text{Dist}_{ij}^l = 0 \\ \exp\left[-ln2\left(\frac{\text{Dist}_{ij}^l}{r}\right)^2\right], \text{Dist}_{ij}^l > 0. \end{cases} \tag{25}$$

4) Define the function $\phi^l(r)$, which represents the relationship dimension in l dimensions, given by

$$\phi^n(r) = \frac{1}{N-l+1}\sum_{i=1}^{N-l+1}\left(\frac{1}{N-l}\sum_{j=1,j\neq i}^{N-l+1} A_{ij}^l\right). \tag{26}$$

5) The final expression for fuzzy entropy is then given by

$$\text{FuzzyEn}(l, r, N) = \ln\frac{\phi^l(r)}{\phi^{l+1}(r)}. \tag{27}$$

Fuzzy entropy assesses how the values are distributed across different attributes. When each attribute's value range is uniformly distributed, those with greater uncertainty will exhibit more distinctive anomalies. Thus, attributes with higher uncertainty should be emphasized more and given greater significance in the anomaly score computation. Furthermore, the weights for each attribute are normalized within the range [0, 1] to ensure that their total sums up to 1. This normalization process is represented by the following formula:

$$W\left(x^{(i)}\right) = \frac{FE\left(x^{(i)}\right)}{\sum_{i=1}^{N}\left(FE\left(x^{(i)}\right)\right)} \in [0, 1] \tag{28}$$

where $FE(x^{(i)})$ represents the fuzzy entropy of the ith dimension data. The fuzzy entropy weights for different features are normalized to reflect the varying role of individual variables in detecting anomalies. By weighting the errors on each feature dimension according to these fuzzy entropy weights and summing them, the anomaly score formula is given

$$\text{Score}(x_i) = \sum_{j}^{M} W\left(x^{(j)}\right)\left(\hat{x}_t^{(j)} - x_t^{(j)}\right)^2 \tag{29}$$

Algorithm 3 Online Anomaly Detection Algorithm

Input: Multivariate time-series stream data $\tilde{X}_i$, hyperparameters λ and Reliability threshold α.
Output:Anomaly status (anomalous or normal)
Retrieve the historical MD-GAT network parameters ϕ_G, encoder parameters ϕ_E, and decoder parameters ϕ_D using Algorithm 1.
for each data point in the stream **do**: // Reliability exceeds the threshold
  Use the updated MD-GAT network parameters ϕ_G^*, encoder parameters ϕ_E^*, and decoder parameters ϕ_D^* to reconstruct the data.
  Calculate the fuzzy entropy of each feature using Equation (27).
  Calculate the normalization factor for the fuzzy entropy weights using Equation (28).
  Calculate the anomaly score utilizing the weighted fuzzy entropy as outlined in Equation (29).
  if the anomaly score exceeds the predefined threshold **then**:
    label the data as anomalous and set $y_i = 1$.
  else:
    label the data as normal and set $y_i = 0$.
  end if
  Store the data,$i \leftarrow i + 1$
  while $i > m$ **do**
    Use Algorithm 2 to update MD-GAT parameters ϕ_G^*, encoder parameters ϕ_E^*, and decoder parameters ϕ_D^*.
  Set the data in the batch to zero,$i \leftarrow 0$
  End while
End for

where $\hat{x}_t^{(j)}$ represents the reconstructed data of the jth dimension at time t, and $x_t^{(j)}$ represents the original data of the jth dimension at time t.

In the anomaly detection phase, the guideline is as follows: if the score indicating an anomaly at a specific time exceeds the set threshold, that timestamp is categorized as anomalous; otherwise, it is classified as normal. The streaming peak over the threshold (DSPOT) method [36] is used for setting the anomaly threshold for streaming data. This method, based on extreme value theory, automatically selects an appropriate threshold for time series.

The online anomaly detection algorithm is outlined in Algorithm 3.

IV. Simulation Evaluation

A. Experimental Setup

Table I shows the parameters used in the experiments. Default values will be used unless otherwise specified.

The anomaly detection model was implemented using the PyTorch package in Python on an AMD Ryzen 7-5800H CPU with 16 GB of RAM.

1) Dataset: To validate the feasibility of the proposed model, experiments were conducted using two open-source datasets. One of these is WUSTL-IIOT-2021[37]. The dataset

TABLE I
Simulation Parameter Settings

parameter	default
Batch size	128
Sequence length	64
Number of attention heads	6
Temperature coefficient	0.07
Regularization strength	10
optimizer	Adam
learning rate	1e-3

comprises data from industrial IoT networks, designed to support cybersecurity studies by emulating real-world industrial environments. This dataset features four attack categories: 1) command injection; 2) Denial of Service (DoS); 3) reconnaissance; and 4) backdoor. The dataset contains 1 194 464 records, with an anomaly rate of approximately 7.2%. The other dataset is UNSW-NB15 [38], a network traffic dataset based on the IoT, created by the Canberra Network Range Lab at the University of New South Wales using the IXIA PerfectStorm tool. It provides a contemporary depiction of network traffic, including both routine operational behavior and nine distinct modern attack types, such as DoS, backdoor, and reconnaissance. This dataset combines real-world normal activities with synthetic contemporary attack behaviors. UNSW-NB15 consists of 2 227 000 data instances, with an anomaly rate of approximately 12.1%, offering a comprehensive representation of network traffic under various conditions. The initial baseline model is trained using the first 10% of the dataset, while the remaining 90% is employed for real-time evaluation of the data stream.

Due to the varying dimensions or units of the features, which can differ by orders of magnitude, normalization is essential to ensure that some metrics are not overlooked due to their minimal impact on performance. Therefore, time series data undergo min–max normalization. During the real-time data processing phase, the minimum and maximum values are continuously updated to reflect the smallest and largest values observed among the incoming samples. With each new sample, these extreme values are recalibrated to align with the latest data.

2) Evaluation Metrics: To assess the effectiveness of the proposed model for anomaly detection, three key evaluation metrics are employed: 1) precision; 2) recall; and 3) F1 score. These metrics are utilized to gauge how well the model performs in identifying anomalies, providing a comprehensive view of its accuracy, completeness, and overall effectiveness in detecting abnormal events. The calculation formulas are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{30}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{31}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{32}$$

In this scenario, TP stands for the count of truly positive instances that the model has correctly identified as positive. FP

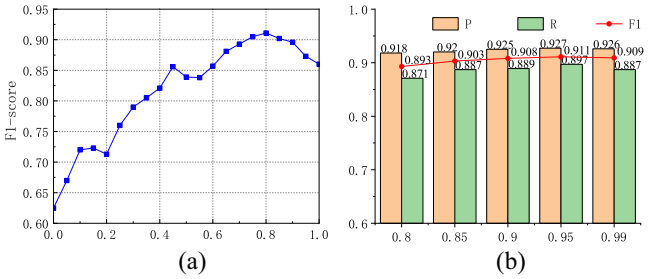

Fig. 4. Impact of the weight λ and reliability threshold α of supervised contrast loss on the anomaly detection performance of the model. (a) λ. (b) α

TABLE II
EFFECTIVENESS OF VARIOUS ALGORITHMS IN DETECTING ANOMALIES

Dataset	Algorithm	Precision	Recall	F1
WUSTL-IIOT-2021	MTAD-GAT	0.894	0.847	0.869
	TFAD	0.876	0.865	0.870
	TranAD	0.902	0.884	0.892
	SC-STVAE	**0.927**	**0.897**	**0.911**
UNSW-NB15	MTAD-GAT	0.852	0.870	0.860
	TFAD	0.844	0.868	0.855
	TranAD	0.873	0.891	0.881
	SC-STVAE	**0.901**	**0.913**	**0.906**

denotes the count of negative instances mistakenly identified as positive. TN represents the number of truly negative instances that the model has accurately labeled as negative, while FN refers to the number of actual positive cases that have been incorrectly identified as negative.

3) Comparison of Algorithms: The anomaly detection performance of the model is compared against four recent state-of-the-art algorithms as follows.

1) *MTAD-GAT [7]:* This model incorporates two layers of graph attention to model intricate causal dependencies within multivariate time series data. It concurrently refines both predictive and reconstructive models to enhance feature representation, leveraging prediction inaccuracies and reconstruction likelihoods to derive inference metrics for detecting anomalies.
2) *TranAD [15]:* This model investigated the self-attention weights for detecting anomalies and proposed a min-max training approach to enhance the distinction between normal and anomalous data.
3) *TFAD [39]:* This model for anomaly detection in time series utilizes a time-frequency analysis approach, incorporating both temporal and spectral information to boost effectiveness. The time-frequency framework combines methods for decomposing time series with techniques for augmenting data, aiming to enhance both the model's accuracy and its interpretability.

B. Results and Analysis

1) Performance Evaluation: The impact of various parameters on model performance was analyzed. All experiments were conducted using the WUSTL-IIOT-2021 dataset. Specifically, the effect of the weight of the supervised contrastive loss λ and the reliability threshold α on anomaly detection performance was evaluated.

As illustrated in Fig. 4, varying the values of λ and α has a substantial effect on the model's ability to detect anomalies. By fixing λ and α in turn, the F1 score on the WUSTL-IIOT-2021 dataset is evaluated using various λ and α values.

For the WUSTL-IIOT-2021 dataset, both the F1 score and model performance increase as λ and α rise, reaching their peak when $\lambda = 0.8$ and $\alpha = 0.95$. However, further increases in λ and α lead to a decline in the F1 score. The optimal anomaly detection performance for the WUSTL-IIOT-2021 dataset is achieved at $\lambda = 0.8$ and $\alpha = 0.95$, while for the UNSW-NB15 dataset, the best performance occurs at $\lambda = 0.75$ and $\alpha = 0.95$.

The anomaly detection performance comparison among various algorithms is shown in Table II. In this comparison, the top-performing results of the three baseline models (MTAD-GAT, TFAD, and TranAD) are marked with an underline, whereas the highest overall performance is indicated in bold. From the information in Table II, we can derive the following insights.

1) For the WUSTL-IIOT-2021 dataset, the proposed model outperforms MTAD-GAT, TFAD, and TranAD by 0.033, 0.051, and 0.025 in accuracy, respectively, and by 0.05, 0.032, and 0.013 in recall. It achieves a high accuracy of 0.927 and a high recall of 0.897, resulting in the highest F1 score of 0.911.
2) For the UNSW-NB15 dataset, the proposed model exceeds MTAD-GAT, TFAD, and TranAD by 0.049, 0.057, and 0.028 in accuracy, respectively, and by 0.043, 0.045, and 0.022 in recall. It achieves an accuracy of 0.901 and a recall of 0.913, yielding the highest F1 score of 0.906.

Compared to TFAD and MTAD-GAT, both TranAD and the proposed model leverage deep learning models tailored for handling temporal data as both encoders and decoders. Rather than extracting temporal features independently and then applying encoding-decoding procedures, these models more effectively maintain the temporal relationships within the data. This enhancement strengthens the feature extraction and reconstruction capabilities for time-series data, significantly improving the models' anomaly detection performance.

2) Stability Experiments: In anomaly detection, model performance stability is critical. Fig. 5 illustrates the precision, recall, and F1 scores of five algorithms tested over ten iterations. As illustrated in the figure, the remaining three network models exhibit more consistent performance across all three metrics compared to TFAD, indicating improved stability. This stability can be linked to the adoption of encoder–decoder frameworks in these models, which facilitate the extraction of latent features and the reconstruction of data, thereby enhancing the learning of the underlying data patterns. Additionally, the proposed model employs supervised contrastive learning to constrain the latent representations, further enhancing its stability.

3) Ablation Studies: To assess the role of each model component in achieving precise anomaly detection, ablation studies were performed. Various model configurations were

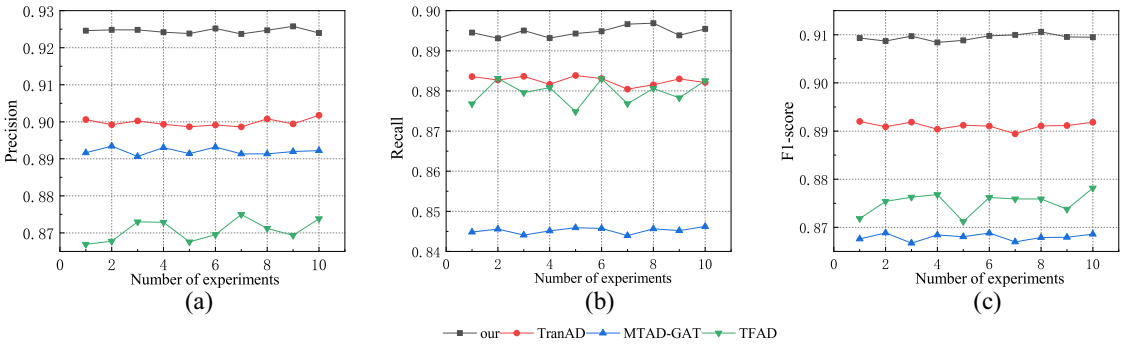

Fig. 5. Impact of the weight λ and reliability threshold α of supervised contrast loss on the anomaly detection performance of the model. (a) Precision stability. (b) Recall stability. (c) F1-score stability.

TABLE III
ABLATION EXPERIMENTS WERE PERFORMED ON THE MODEL ON TWO DATA SETS, USING F1-SCORE AS THE EVALUATION INDEX

model	Supcon	Weighted Fuzzy	E-EWC	Wustl-iiot-2021	UNSW-NB15
SC-STVAE*		✓	✓	0.883	0.869
SC-STVAE**	✓		✓	<u>0.903</u>	<u>0.895</u>
SC-STVAE***	✓	✓		0.896	0.883
SC-STVAE	✓	✓	✓	**0.911**	**0.906**

created to evaluate how different elements influence the model's effectiveness in detecting anomalies.

1) *SC-STVAE*:* This variant uses an online-updated spatiotemporal variational AE (STVAE) and employs a fuzzy entropy-weighted anomaly score to detect anomalies.
2) *SC-STVAE**:* This variant incorporates supervised contrastive learning into the online-updated STVAE to detect anomalies.
3) *SC-STVAE***:* This variant does not perform online updates and relies on the offline-generated model to detect anomalies.

These experiments aim to isolate the effects of online updates, supervised contrastive learning, and fuzzy entropy weighting on the model's anomaly detection performance.

The ablation experiment results are shown in Table III, with the top-performing results marked in bold and the runner-up results underscored. The following sections explain how each component influences performance.

Impact of Supervised Contrastive Learning: The objective of integrating supervised contrastive learning is to regulate the latent representations of input data. As demonstrated in Table III, the performance of the model on both datasets exceeds that of the version without supervised contrastive learning, which indicates that refining the latent space through this technique significantly improves anomaly detection.

Impact of Fuzzy Entropy Weighting: The role of fuzzy entropy is to evaluate how much various features contribute. Thereby enhancing the significance of key features and improving detection accuracy. Table III reveals that the model incorporating fuzzy entropy weighting outperforms the variant without it, suggesting that fuzzy entropy weighting significantly contributes to improving anomaly detection performance.

Impact of Online Model Updates: The model employs online updates to mitigate the effects of data drift, which can degrade detection performance over time. Table III demonstrates that the model with online updates outperforms the one without. This indicates that online updates are crucial for maintaining the model's effectiveness in detecting anomalies amidst evolving data distributions.

By implementing these approaches, the model becomes more adept at understanding the spatial and temporal relationships between different elements, simplifying the data, and mitigating the effects of outliers during training. This results in improved accuracy and dependability for detecting anomalies in industrial IoT settings.

V. CONCLUSION

This article investigated the challenge of online detecting anomalies in real-time data streams within industrial IoT networks and introduces an innovative approach built on a spatiotemporal variational AE (STVAE) with supervised contrastive learning. The method addresses the inability of traditional reconstruction-based models to clearly differentiate between normal and abnormal samples in latent space by incorporating supervised contrastive learning. Latent representations of positive samples are contrasted with those of negative samples generated through random perturbations, improving the model's boundary constraints and detection capabilities.

Additionally, a fuzzy entropy-weighted anomaly scoring method is introduced to account for the varying contributions of different features to anomaly detection. This approach weights MSEs by the contribution of each feature as determined by fuzzy entropy. To combat data drift and its detrimental effects on model performance, an event-triggered

EWC algorithm is employed to incrementally update the model.

Despite the promising performance achieved by the proposed method in experiments, several limitations remain. First, due to the complexity and multimodal characteristics of data in industrial IoT environments, this study primarily focuses on a single data source and specific scenarios. Future research could explore multimodal data fusion to further enhance anomaly detection performance. Second, the event-triggering mechanism proposed in this study relies on predefined threshold values. Further investigation is required to dynamically optimize these thresholds in complex and dynamic environments. Finally, when processing high-frequency real-time data streams, the model may face constraints related to computational resources. Future work could focus on designing lightweight models or leveraging distributed computing to improve the real-time performance and adaptability of the algorithm.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] S. Soltan, P. Mittal, and H. V. Poor, "{BlackIoT}:{IoT} botnet of high wattage devices can disrupt the power grid," in *Proc. 27th USENIX Secur. Symp. (USENIX Secur.)*, 2018, pp. 15–32.

[3] J. Hong, C.-C. Liu, and M. Govindarasu, "Integrated anomaly detection for cyber security of the substations," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1643–1653, Jul. 2014.

[4] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," *J. Netw. Comput. Appl.*, vol. 77, pp. 18–47, Jan. 2017.

[5] X. Chen et al., "DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, 2021, pp. 2225–2230.

[6] D. Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1705–1714.

[7] H. Zhao et al., "Multivariate time-series anomaly detection via graph attention network," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2020, pp. 841–850.

[8] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018.

[9] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1177–1191, Mar. 2021.

[10] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," 2022, *arXiv:2201.07284*.

[11] L. Tang et al., "Swarm learning anomaly detection framework for cloud data center using multichannel BiWGAN-GTN and CEEMDAN," *Digit. Commun. Netw.*, to be published used.

[12] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Min.*, 2019, pp. 2828–2837.

[13] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in industrial Internet of Things," *Digit. Commun. Netw.*, vol. 9, no. 1, pp. 101–110, 2023.

[14] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digit. Commun. Netw.*, vol. 10, no. 1, pp. 190–204, 2024.

[15] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," 2022, *arXiv:2110.02642*.

[16] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019, *arXiv:1807.03748*.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[19] Z. Yue et al., "TS2Vec: Toward universal representation of time series," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8980–8987.

[20] E. Eldele et al., "Time-series representation learning via temporal and contextual contrasting," 2021, *arXiv:2106.14112*.

[21] H. Zhou, K. Yu, X. Zhang, G. Wu, and A. Yazidi, "Contrastive autoencoder for anomaly detection in multivariate time series," *Inf. Sci.*, vol. 610, pp. 266–280, Sep. 2022.

[22] X. Zhang, S. Shi, H. Sun, D. Chen, G. Wang, and K. Wu, "ACVAE: A novel self-adversarial variational auto-encoder combined with contrast learning for time series anomaly detection," *Neural Netw.*, vol. 171, pp. 383–395, Mar. 2024.

[23] J. Yu et al., "An adversarial contrastive autoencoder for robust multivariate time series anomaly detection," *Expert Syst. Appl.*, vol. 245, Jul. 2024, Art. no. 123010.

[24] K. Faber, R. Corizzo, B. Sniezynski, and N. Japkowicz, "VLAD: Task-agnostic VAE-based lifelong anomaly detection," *Neural Netw.*, vol. 165, pp. 248–273, Aug. 2023.

[25] R. Corizzo, M. Baron, and N. Japkowicz, "CPDGA: Change point driven growing auto-encoder for lifelong anomaly detection," *Knowl.-Based Syst.*, vol. 247, Jul. 2022, Art. no. 108756.

[26] J. B. Gomm, "Process fault diagnosis using a self-adaptive neural network with on-line learning capabilities," *IFAC Proc. Vol.*, vol. 28, no. 12, pp. 69–74, 1995.

[27] Y. Fu, H. Cao, X. Chen, and J. Ding, "Broad auto-encoder for machinery intelligent fault diagnosis with incremental fault samples and fault modes," *Mech. Syst. Signal Process.*, vol. 178, Oct. 2022, Art. no. 109353.

[28] Y. Liu, B. Chen, D. Wang, L. Kong, J. Shi, and C. Shen, "A lifelong learning method based on generative feature replay for bearing diagnosis with incremental fault types," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, Apr. 2023.

[29] S. Liu, J. Huang, J. Ma, and J. Luo, "Class-incremental continual learning model for plunger pump faults based on weight space meta-representation," *Mech. Syst. Signal Process.*, vol. 196, Aug. 2023, Art. no. 110309.

[30] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.

[31] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2018, *arXiv:1710.10903*.

[32] C. Sun, Z. He, H. Lin, L. Cai, H. Cai, and M. Gao, "Anomaly detection of power battery pack using gated recurrent units based variational autoencoder," *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109903.

[33] P. Khosla et al., "Supervised contrastive learning," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, 2020, pp. 18661–18673.

[34] S. Yoon, Y. Lee, J.-G. Lee, and B. S. Lee, "Adaptive model pooling for online deep anomaly detection from a complex evolving data stream," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2022, pp. 2347–2357.

[35] Q. Hu, D. Yu, Z. Xie, and J. Liu, "Fuzzy probabilistic approximation spaces and their information measures," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, pp. 191–201, Apr. 2006.

[36] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 1067–1075.

[37] M. Zolanvari, M. Teixeira, L. Gupta, K. Khan, and R. Jain, *WUSTL-IIoT-2021 Dataset for IIoT Cybersecurity Research*, Washington Univ. St. Louis, St. Louis, MO, USA, 2021.

[38] R. Moustafa and J. Slay, "A comprehensive data set for network intrusion detection systems," School Eng. Inf. Technol., Univ. New South Wales Australian Def. Force Acad., Canberra, ACT, Australia, Rep. UNSW-NB15, 2015.

[39] I. Haloui, J. S. Gupta, and V. Feuillard, "Anomaly detection with Wasserstein GAN," 2018, *arXiv:1812.02463*.

Lun Tang received the Ph.D. degree in communication and information system from Chongqing University, Chongqing, China.

He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing. His current research interests include 5G cellular networks, interference management, and small cell networks.

Ruiyu Wei received the B.S. degree in electronic information engineering from Tangshan University, Tangshan, China, in 2022. He is currently pursuing the M.S. degree in information and communication engineering with the Key Laboratory of Mobile Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.

His research interests include anomaly detection, industrial Internet of Things, and lifelong learning.

Bingsen Xia received the M.S. degree from North China Electric Power University, Jilin City, China, in 2018.

He is currently a Senior Engineer of Electrical Engineering. He works with the State Grid Fujian Economic Research Institute, Fuzhou, China. His current research interests include electronic communication planning and design, and designing of information and communication for electric Internet of Things.

Yuanchun Tang received the B.S. degree in communication engineering from Northeast Electric Power University, Jilin City, China, in 1998.

He is currently a Senior Engineer of Electrical Engineering. He works with the State Grid Fujian Economic Research Institute, Fuzhou, China. His current research interests include electronic communication planning and design, and designing of information and communication for electric Internet of Things.

Weili Wang received the M.E. and Ph.D. degrees in information and communication engineering from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2018 and 2023, respectively.

She was with Carleton University, Ottawa, ON, Canada, as a Visiting Researcher form December 2021 to January 2023. She is currently a Postdoctoral Researcher with the Cyber Security and Information Law Research Center, Chongqing. Her current research interests include intelligent network management and self-healing techniques in 6G.

Qiong Huang received the M.S. degree from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2005.

She is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications. Her current research interests include wireless transmission technology and network function virtualization.

Qianbin Chen (Senior Member, IEEE) received the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2002.

He is currently a Professor with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China, and the Director of Chongqing Key Laboratory of Mobile Communication Technology, Chongqing. He has authored or co-authored over 100 papers in journals and peer-reviewed conference proceedings, and has co-authored seven books. He holds 47 granted national patents.