IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 July 2024, accepted 15 August 2024, date of publication 19 August 2024, date of current version 27 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3445533

RESEARCH ARTICLE

Graph Learning Framework for Data Link Anomaly Detection

CHANG YANG[1], LISHA WU[1], JING XU[1], YINGJIE REN[1], BO TIAN[2], AND ZHENHUA WEI[2]
[1]Big Data Center of State Grid Corporation of China, Beijing 100052, China
[2]School of Control and Computer Engineering, North China Electric Power University, Beijing 100026, China

Corresponding author: Bo Tian (1397494259@qq.com)

This work was supported by the Science and Technology Project from Big Data Center of State Grid Corporation of China under Contract SGSJ0000YYJS2400049.

ABSTRACT The anomaly detection in data links aims to identify the state of the link during data transmission, which is a critical task for ensuring information transmission security. Most anomaly detection methods focus solely on individual link characteristics, disregarding the inter-link structural information, thus hindering effective generalization to graph-structured data. In this study, we introduce a Graph Learning-based Data Link Anomaly Detection model (GLDAE) that considers both the link features and the communication network structure. Specifically, GLDAE comprises a graph enhancement module, a link feature autoencoder, a structure autoencoder, and a discriminator, enabling simultaneous learning of edge features and the latent representation of the graph structure. Moreover, to enhance the model's generalization capability, we employ contrastive learning between the original graph and its enhanced version. Additionally, to achieve joint learning of edge features and graph structure, we integrate edge feature embeddings and structure embeddings as inputs to the decoder. Finally, utilizing the well-trained encoder to encode link features and derive a new feature representation, we feed it into an MLP classifier to determine the link's status. Experimental evaluations were conducted on four authentic datasets (NF-UNSW-NB15, NF-UNSW-NB15-v2, NF-ToN-IoT, NF-ToN-IoT-v2), comparing our model against state-of-the-art baseline models, showcasing the substantial potential of our approach.

INDEX TERMS Anomaly detection, contrastive learning, data links, graph neural networks.

I. INTRODUCTION

The proliferation of communication networks has led to a significant increase in the volume of communication data, resulting in a rapid expansion of data flow and connections. This surge in data transmission can give rise to various issues [1], including link disruptions and data loss, which can lead to incomplete and erroneous data transfer, ultimately impacting the overall data analysis and application efficacy. Consequently, ensuring the stability and reliability of information transmission within data links is paramount for transmission operations, with timely detection of anomalies in data links emerging as a key area of concern.

Early research efforts focused on utilizing traditional machine learning techniques for anomaly detection in data links. For instance, Lawal et al. [2] employed XGBoost to compare the accuracy of signature-based and anomaly-based binary and multi-class tasks in IoT data link analysis. Sarhan et al. [3] assessed classification performance on an enhanced link dataset using an Extra Tree ensemble classifier comprising more than 50 random decision trees. Wang et al. [4] introduced a novel clustering method (CCAD) tailored for collective anomaly detection in network traffic. Xiao et al. [5] adopted a strategy of transforming link data into bipartite graphs and hypergraphs, leveraging graph embedding features and original link attributes for anomaly detection. The intricate nature and variability of data links pose challenges for traditional methods in accurately identifying abnormal link states and pinpointing the specific links affected by

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

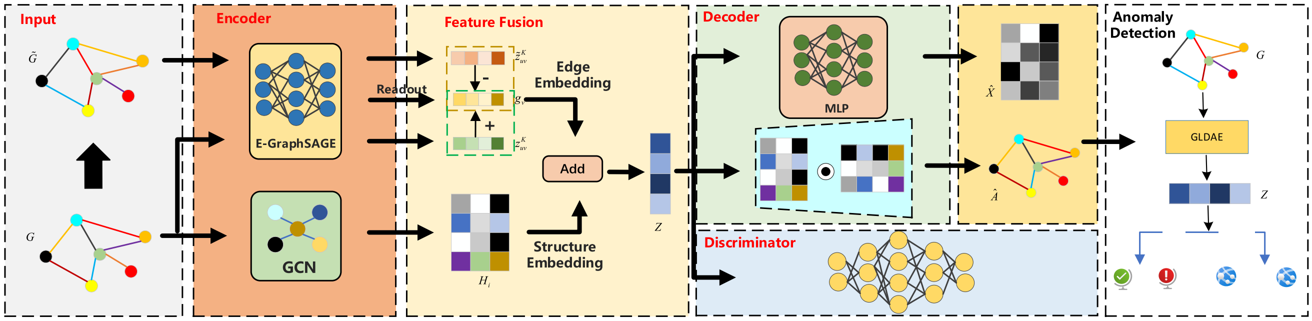

FIGURE 1. The proposed framework of GLDAE. First, the embedded representation of the graph is obtained by using an encoder, which generates positive and negative samples of the graph's edges. Further learning is conducted through mutual information comparison. The learned edge features are then fused with the structural features, and these fused features are used to reconstruct the graph. Finally, the trained model is employed to re-encode the test data to obtain the embedded representation, which is utilized to complete the downstream anomaly detection task.

anomalies, thereby impeding the maintenance efficiency of data links.

As deep learning has advanced, researchers have increasingly explored the application of neural networks for anomaly detection in data links. Liu et al. [6] utilized Convolutional Neural Networks (CNN) to extract statistical features from network traffic for anomaly identification. Lunardi et al. [7] employed autoencoders to detect network anomalies by analyzing the initial data packets of network flows, incorporating adversarial training techniques.

Graph structures possess a robust capability to represent non-Euclidean data. This area of research is gaining significant traction, with ongoing advancements and a continually solidifying theoretical foundation. Zhuo et al. [8] developed an end-to-end detection model using Graph Convolutional Neural Networks (GCN) to identify zombie network nodes based on topological data. Lo et al. [9] directly classified link features using E-GraphSAGE. Caville et al. [10] introduced perturbed graphs based on E-GraphSAGE, derived feature embeddings of links through mutual information between edges and graphs, and applied anomaly detection algorithms for classification. Xu et al. [11] integrated attention mechanisms into encoders to generate graph embeddings, sampled node subgraphs, employed positive and negative subgraphs for contrastive learning, and conducted self-supervised detection of network flows. Wang et al. [12] introduced the Edge Feature Attention Network (EGAT), which integrates edge and attention mechanisms. In EGAT, the computation of messages and attention weights encompasses all features of nodes and edges. Chang et al. [13] enhanced Graph Neural Networks (GNNs) by incorporating residual learning into E-GraphSAGE and EGAT, thereby maximizing the utilization of available graph information. Zhang et al. [14] proposed the Graph Inference and Distribution (GID) framework, which combines network structure learning with GNN parameter optimization in a two-layer learning framework. GID employs multi-head cosine similarity and reconstruction techniques to achieve improved graph structure representation. Altaf et al. [15] developed a model that constructs the data link as a multi-edge graph structure. This model merges the strengths of spatial and spectral GNNs, making it suitable for complex multi-graph structures and capable of handling multi-edge and multi-dimensional edge features.

Detecting anomalies in data links plays a crucial role in safeguarding the security of information transmission. Most existing studies concentrate on a single aspect of the graph, such as edge features or structural properties. However, the state of a link is intrinsically linked to both aspects. To improve detection results, this study introduces a novel graph-based model, named GLDAE (Graph Learning Framework for Data Link Anomaly Detection).The key contributions of this research are outlined below:

1) We propose a novel model for data link state anomaly detection based on graph learning. This model employs a dual-channel network to extract features from both edge attributes and structural characteristics of the data link. By integrating these two types of features, the model significantly enhances its expressive capability. For learning link features, we utilize graph enhancement techniques to generate positive and negative examples from the original data, introducing graph contrastive learning. The approach maximizes the mutual information between edge and graph representations to achieve more stable anomaly detection for links.
2) We have developed a feature fusion module designed to integrate edge features and topological structure features learned by the dual-channel network. This integration enhances the model's capacity to understand and interpret the data link comprehensively.
3) By combining the autoencoder architecture with the graph neural network, we effectively leverage the autoencoder's nonlinear feature learning and data dimensionality reduction capabilities alongside the graph neural network's feature extraction strengths. This synergy results in richer and more meaningful feature embeddings.
4) Both contrastive learning and autoencoders are well-suited for unsupervised learning scenarios. Our model, which integrates these two techniques,

is particularly applicable in situations with limited labeled data. It offers an unsupervised method for learning low-dimensional representations of graph data.

5) The effectiveness of our approach, GLDAE, was evaluated on four network datasets, demonstrating superior performance compared to existing technologies.

II. METHODOLOGY

This section presents a comprehensive overview of the anomaly detection framework devised for data links. Illustrated in Fig. 1, the framework comprises a graph enhancement module, edge feature and graph structure encoders, decoders, feature fusion module, and discriminator. Initially, the original graph undergoes transformation into an enhanced graph through the graph enhancement module. Subsequently, the encoders process both graphs, utilizing edge feature embedding and mutual information from positive and negative graph pairs to compute the discrimination score. The edge features and graph structure information are then integrated, followed by the individual reconstruction of edge features and graph topology. Ultimately, anomaly detection for data links is executed using the amalgamated embedding features. The detailed operational steps are outlined in Algorithm 1.

A. GRAPH AUGMENTATION

To enhance the model's generalization capability, we randomly shuffle the edges within the original link graph to create augmented graphs for comparison with the initial graph. The alteration involves solely rearranging the order of edge features, resulting in $\tilde{X} \neq X$, while maintaining the adjacency matrix and edge count unchanged, ensuring $\tilde{A} = A$.

Algorithm 1 Training Process of GLDAE

```
Input:
  Graph $G(v, e, x)$;
  Edge features $e$;
  Node features $x$;
  Depth $K$;
  Differentiable aggregator functions AGG;
Output:
  Embeddings $z_{uv}$ and Optimized E-GraphSAGE and GCN encoder $f$
Initialize the parameters $\theta$ and $w$ for the encoder f, the parameters $\alpha$ and $\beta$,and the discriminator D;
for $epoch \leftarrow 1$ to $T$ do
    $z_v^K, z_{uv}^K = f_1(G, \theta)$
    $\tilde{z}_{uv}^K = f_1(\tilde{G}, \theta)$
    $g_v = Readout(z_v^K) = \sum_{k=1}^{n} \frac{z_v^k}{n}$
    $D(z_{uv}^K, e_v) = sigmoid(z_{uv}^K \cdot w \cdot g_v)$
    $D(\tilde{z}_{uv}^K, e_v) = sigmoid(\tilde{z}_{uv}^K \cdot w \cdot g_v)$
    $L_{con} = -\frac{1}{2n} \sum_{i=1}^{n} (\mathbb{E}_G logD(z_{uv}^K, g_v) + \mathbb{E}_{\tilde{G}} log(1 - D(\tilde{z}_{uv}^K, g_v)))$
    $H_i^l = f_2(G, \theta)$
    $Z = \alpha z_{uv} \oplus (1 - \alpha) H_i$
    $\widehat{A} = sigmoid(Z(Z)^T)$
    $\widehat{X} = Z_j^l = ReLU(MLP(Z_j^{l-1} | W_j^l, b_j^l))$
    $L_{str} = \left\| A - \widehat{A} \right\|$
    $L_{fea} = \left\| X - \widehat{X} \right\|$
    $L = \beta L_{con} + (1 - \beta)(L_{str} + L_{fea})$
    $\theta, w \leftarrow Adam(L)$
end for
```

B. ENCODER

The model adeptly integrates structural and edge feature information to holistically learn the graph's features. By jointly acquiring the graph's structural details and edge features, the model ensures comprehensive learning. Given that data link features are embedded within the graph's edge structure, k-hop depth sampling and aggregation techniques are employed to capture edge feature information. The E-GraphSAGE algorithm [9] is utilized for precise extraction of edge features.

The algorithm takes as input the edge features $\{e_{uv}, uv \in \varepsilon\}$. Given that network flows solely consist of edge features without node features, the node features are initialized as $x_v = \{1, 1, \ldots, 1\}$, aligning the node feature vector's dimension with that of the edge feature vector. At the k-th layer, the neighbor aggregation function combines the edge features of sampled neighbors:

$$h_{N(v)}^k = AGG_k(\{e_{uv}^{k-1}, \forall u \in N(v), uv \in \varepsilon\}) \tag{1}$$

Here, e_{uv}^{k-1} denotes the edge feature of uv within the sampled neighbors $N(v)$ of node v at the $(k-1)$-th layer, where $\{e_{uv}^{k-1}, \forall u \in N(v), uv \in \varepsilon\}$ represents the edges in $N(v)$. The aggregation function AGG can take various forms, such as mean, pooling, or LSTM. For experimental simplicity, this study opts for the mean aggregation approach, computing the average of edge features within the node's sampled neighbors. Subsequently, the aggregated information $h_{N(v)}^k$ is concatenated with the node embedding h_v^{k-1} from the previous layer:

$$h_v^k = \sigma(W^k \cdot CONCAT(h_v^{k-1}, h_{N(v)}^k)) \tag{2}$$

Here, W^k denotes the weight parameter, $CONCAT$ signifies the concatenation function, and σ indicates a nonlinear activation function, which could be ReLU, Tanh, Sigmoid, or another function. After these operations, the final node embedding h_v^k is obtained. The node embedding at the k-th layer is then expressed as:

$$z_v^K = h_v^K \tag{3}$$

The ultimate edge embedding z_{uv}^K is derived by concatenating the embeddings of nodes u and v:

$$z_{uv}^K = CONCAT(z_u^K, z_v^K), uv \in \varepsilon \tag{4}$$

Likewise, the edge embedding of the augmented graph $\tilde{z}_{uv}^K$ can be acquired, and the comprehensive graph information is gathered through the Readout function:

$$g_v = Readout(z_v^K) = \sum_{k=1}^{n} \frac{z_v^k}{n} \tag{5}$$

TABLE 1. Statistics of datasets used in our experiments.

Dataset	Attribute	Classes	Samples	
			Version1	Version2
NF-UNSW-NB15 [16] NF-UNSW-NB15-v2 [17]	Label	Normal	1550712	2295222
		Attack	72406	95053
	Attack	Benign	1550712	2295222
		Fuzzers	19463	22310
		Analysis	1995	2299
		Backdoor	1782	2169
		DoS	5051	5794
		Exploits	24736	31551
		Generic	5570	16560
		Reconnaissance	12291	12779
		Shellcode	1365	1427
		Worms	153	164
NF-ToN-IoT [16] NF-ToN-IoT-v2 [17]	Label	Normal	270279	6099469
		Attack	1108995	10841027
	Atttack	Benign	270279	6099469
		Backdoor	17247	16809
		DoS	17717	712609
		DDoS	326345	2026234
		Injection	468539	684465
		MITM	1295	7723
		Password	156299	1153323
		Ransomware	142	3425
		Scanning	21467	3781419
		XSS	99944	2455020

The local edge embeddings from both graphs and the global graph information are inputted into the bilinear discriminator for comparison, enabling the calculation of corresponding scores:

$$D(z_{uv}^K, e_v) = sigmoid(z_{uv}^K \cdot w \cdot g_v) \tag{6}$$

$$D(\widetilde{z}_{uv}^K, e_v) = sigmoid(\widetilde{z}_{uv}^K \cdot w \cdot g_v) \tag{7}$$

w represents a trainable scoring matrix. The objective is to enhance the mutual information between the edge embeddings of the original graph and the augmented graph, while reducing the mutual information between the edge embeddings of the augmented graph and the original graph. The binary cross-entropy (BCE) loss function is employed to compute the loss for the current training iteration:

$$L_{con} = -\frac{1}{2n}\sum_{i=1}^{n}(\mathbb{E}_G logD(z_{uv}^K, g_v) + \mathbb{E}_{\widetilde{G}} log(1 - D(\widetilde{z}_{uv}^K, g_v))) \tag{8}$$

where $\mathbb{E}$ represents entropy, $\mathbb{E}_G$ denotes the data from the original graph, and $\mathbb{E}_{\widetilde{G}}$ denotes the data from the enhanced graph. The first term represents the entropy of the edge features from the original graph as well as the mutual information of the entire graph passing through the discriminator. Similarly, the second term represents the entropy of the edge features from the enhanced graph and the mutual information of the entire graph passing through the discriminator. The objective of the discriminator is to maximize the first term to 1 and minimize the second term to 0. In order to capture structural information, Graph Convolutional Network (GCN) is employed to project the initial network graph into a lower-dimensional embedding space. The input comprises the adjacency matrix A and the node feature matrix $H_i^{(0)} = x$:

$$H_i^l = GCN(x, A|W) = ReLU(\widetilde{D}_i^{-\frac{1}{2}}\widetilde{A}_i\widetilde{D}_i^{-\frac{1}{2}}H_i^{l-1}W_i) \tag{9}$$

Here, $\widetilde{A}_i$ represents the adjacency matrix with self-loops, $\widetilde{D}_i$ denotes the degree matrix of $\widetilde{A}_i$, W_i signifies the network parameter of GCN, and $ReLU(\cdot)$ denotes the activation function.

C. FEATURE FUSION

Enhancing detection accuracy requires leveraging both edge features and graph structural information effectively. The element-wise addition operation $\oplus$ is employed to merge these components, with $\alpha \in (0, 1)$ controlling the weight distribution for improved anomaly detection in data links:

$$Z = \alpha z_{uv} \oplus (1 - \alpha)H_i \tag{10}$$

D. FEATURE FUSION

The structural decoder utilizes matrix inner product to fuse feature Z as input for reconstructing the edge features of the original network graph:

$$\hat{A} = sigmoid(Z(Z)^T) \tag{11}$$

Minimize the structural reconstruction error through training with the Mean Squared Error (MSE) loss function:

$$L_{str} = \left\| A - \hat{A} \right\| \tag{12}$$

Employing a Multilayer Perceptron (MLP) to reconstruct the edge feature matrix of the graph, the edge feature matrix

TABLE 2. Comparison of main parameter settings.

Model	Parameter settings
E-GraphSAGE	2-Layer E-GraphSAGE, Hidden Layer Dimension: 256
Anomaly-E	1-Layer E-GraphSAGE, Hidden Layer Dimension: 256
NEGSC	1-Layer NEGAT, Attention heads: 3
GLDAE	1-Layer E-GraphSAGE, 1-Layer GCN, 1-Layer MLP, Hidden Layer Dimension: 256

is reconstructed using the fusion feature Z as input:

$$\widehat{X} = Z_j^l = ReLU(MLP(Z_j^{l-1}|W_j^l, b_j^l)) \tag{13}$$

Minimize the error in edge features by utilizing the Mean Squared Error (MSE) loss function during training:

$$L_{fea} = \left\|X - \widehat{X}\right\| \tag{14}$$

E. OBJECTIVE FUNCTION AND ANOMALY DETECTION

The objective function is defined as follows:

$$L = \beta L_{con} + (1 - \beta)(L_{str} + L_{fea}) \tag{15}$$

The objective function comprises several terms: L_{con} denotes the contrastive loss, L_{str} signifies the reconstruction error of the graph structure, and L_{fea} indicates the reconstruction error of the edge features. The parameter $\beta \in (0, 1)$ is utilized to adjust the weight between mutual information contrast loss and reconstruction error.

Upon convergence of the objective function, the trained encoder is employed to encode the link features, producing edge embeddings. These embeddings are subsequently inputted into an MLP classifier to perform anomaly detection.

F. TIME COMPLEXITY ANALYSIS

In this section, we analyze the computational complexity of GLDAE. For graph composition, constructing an enhanced graph requires randomly rearranging the edges of the original graph to achieve perturbation. Assuming there are V vertices and E edges, the time complexity of this process is O$(V+E)$.

Next, the two graphs are encoded by the encoder. Assuming that GCN and E-GraphSAGE have L layers and each layer has F filters, the time complexity of the encoder is O$(LF(V+E))$.

For the feature fusion module, most operations are linear, so the time complexity can be reduced to O$(V+E)$.

In the reconstruction stage of the graph, the edge features are processed using an MLP. Assuming there are L layers, each with F neurons, and the input dimension is X, the time complexity is O(LFX).

The adjacency matrix reconstruction uses the matrix inner product operation. Assuming the matrix dimension is $M \times N$, where M is the number of rows and N is the number of columns, and considering the matrix can be calculated in parallel, the time complexity of this operation is approximately O(N).

For anomaly detection, which also uses an MLP, the time complexity is O(LFX).

TABLE 3. Hyperparameter values used in GLDAE.

Hyperparameter	Values
No. Layers	1
No. Hidden	256
Epoch	100
Learning Rate	[1e-4,1e-3]
Activation Func	ReLU
Loss Func	BCE/CE/MSE
Optimiser	Adam

In summary, the total time complexity of the proposed model can be approximated as O$(2V + 2E + N + LF(2X + V + E))$.

III. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents a comprehensive evaluation of our proposed method's performance in detecting anomalies in data transmission links and identifying the types of abnormal attacks on data links. Furthermore, a series of ablation experiments were performed to validate the efficacy of each component within the model.

A. DATASETS

For the experiments, we utilized a dataset comprising universal NetFlow features [16] developed by Sarhan et al. [17], [18]. Specifically, we selected four datasets: NF-UNSW-NB15 [18], NF-UNSW-NB15-v2 [17], NF-ToN-IoT [18], and NF-ToN-IoT-v2 [17]. Given the datasets' extensive size, a subset of the data was chosen for experimentation, with abnormal samples representing approximately 4% of the normal samples. Detailed information about the datasets, supporting both binary and multi-class anomaly detection tasks in data transmission, is provided in Table 1.

NF-UNSW-NB15 is derived from the UNSW-NB15 dataset [19] and comprises 8 features. Within the dataset's 1,623,118 network flow records, there are 1,550,712 benign samples and 72,406 abnormal samples, encompassing nine types of attacks. NF-UNSW-NB15-v2 extends NF-UNSW-NB15 by incorporating 39 features, resulting in an expanded dataset of 2,390,275 samples, including 2,295,222 benign samples and 95,053 abnormal samples.

NF-ToN-IoT is derived from the ToN-IoT dataset [20] and comprises 8 features. Within the dataset's 1,379,274 network flow records, there are 270,279 benign samples and 1,108,995 abnormal samples, encompassing nine types of attacks. NF-ToN-IoT-v2 extends NF-ToN-IoT by incorporating 39 features, resulting in an expanded dataset of 16,940,496 samples, including 6,099,469 benign samples and 10,841,027 abnormal samples.

B. BASELINES

The model is compared against three baseline methods for binary and multi-class tasks in detecting anomalies in link transmission and identifying abnormal link attacks.

E-GraphSAGE [9] is a Graph Neural Network (GNN) technique that enhances the message passing mechanism of the original GraphSAGE to effectively learn edge features.

TABLE 4. Binary classification results.

Dataset	Metric	E-GraphSAGE [9]	Anomaly-E [10]	GLDAE(Ours)
NF-UNSW-NB15-v2	Acc	98.68%	96.36%	**99.09%**
	Precision	98.69%	95.83%	**99.25%**
	Recall	98.68%	96.36%	**99.09%**
	F1	98.68%	94.91%	**99.13%**
NF-ToN-IoT-v2	Acc	95.99%	95.12%	**96.52%**
	Precision	**96.44%**	91.90%	96.00%
	Recall	95.99%	95.12%	**96.52%**
	F1	**96.18%**	93.40%	95.68%
NF-UNSW-NB15	Acc	97.54%	95.35%	**97.67%**
	Precision	98.09%	92.58%	**98.21%**
	Recall	97.54%	95.35%	**97.67%**
	F1	97.72%	93.32%	**97.84%**
NF-ToN-IoT	Acc	96.80%	95.42%	**98.84%**
	Precision	96.42%	92.40%	**98.80%**
	Recall	96.80%	95.42%	**98.84%**
	F1	95.90%	93.84%	**98.78%**

TABLE 5. Multiclass classification results.

Dataset	Metric	E-GraphSAGE [9]	Anomaly-E [10]	NEGSC [11]	GLDAE(Ours)
NF-UNSW-NB15-v2	Acc	**97.29%**	96.15%	97.03%	97.13%
	Precision	**97.31%**	94.86%	96.57%	96.76%
	Recall	**97.29%**	96.15%	97.03%	97.13%
	F1	**97.17%**	95.47%	96.70%	96.85%
NF-ToN-IoT-v2	Acc	94.74%	**95.36%**	94.53%	94.22%
	Precision	**94.53%**	93.89%	94.43%	92.76%
	Recall	94.74%	**95.36%**	94.53%	94.22%
	F1	93.19%	**93.80%**	**93.80%**	92.62%
NF-UNSW-NB15	Acc	95.91%	95.53%	95.28%	**96.31%**
	Precision	94.41%	91.90%	93.70%	**95.14%**
	Recall	95.91%	95.53%	95.28%	**96.31%**
	F1	94.95%	93.59%	94.39%	**95.59%**
NF-ToN-IoT	Acc	90.07%	94.91%	89.66%	**95.23%**
	Precision	85.52%	91.92%	83.61%	**94.52%**
	Recall	90.07%	94.91%	89.66%	**95.23%**
	F1	87.62%	93.31%	86.49%	**94.60%**

Anomaly-E [10] is a Graph Neural Network (GNN)-based self-supervised approach that integrates E-GraphSAGE and DGI, leveraging the maximization of local-global mutual information through graph perturbation. Post-training, the encoded edge features are fed into four anomaly detection algorithms for classification, with the top-performing Isolation Forest (IF) detection algorithm chosen as a baseline method.

NEGSC [11] is a Graph Neural Network (GNN)-based self- supervised technique that employs an encoder featuring a graph attention mechanism to acquire edge features. Subsequently, node sampling is utilized to create subgraphs and generate positive and negative samples for contrastive learning.

C. EVALUATION METRICS

To assess the model's performance effectively, we employ four evaluation metrics: Accuracy, Precision, Recall, and F1-score. These metrics are commonly utilized in various studies [21] and are well-suited for evaluating model performance in both binary and multi-class tasks.

D. EVALUATION METRICS

The three baseline methods utilized the hyperparameter configurations specified by the authors. Our model was implemented using Python, PyTorch, and DGL. The experimental setup was conducted on a platform with the following specifications: 20 vCPU Intel(R) Xeon(R) Platinum 8457C, 100GB of memory, and an L20 GPU with 48GB. The weight hyperparameters α and β were optimized within the range of [0.1,0.9] through grid search, while the other hyperparameters are outlined in Table 3. The training phase utilized 70% of the data, with the remaining 30% of samples reserved for testing and performance evaluation.

E. DATASETS PREPROCESSING

Before training the model, we preprocessed the dataset. To facilitate subsequent operations, the numbers in the IP and Port columns were converted to string types. Given that the range of some feature values in the dataset is relatively wide and the comparability between features is low, directly using them without processing would lead to unstable calculations. Therefore, we normalized each column of numerical features, scaling the feature values to the range [0, 1]. The "Attack"

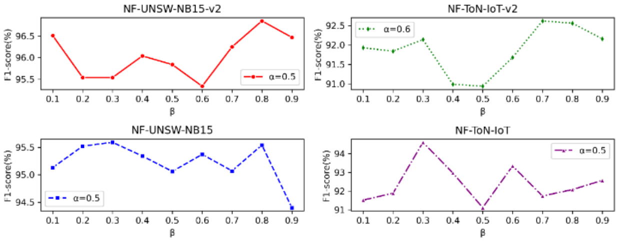

FIGURE 2. The results of different values of β on four datasets.

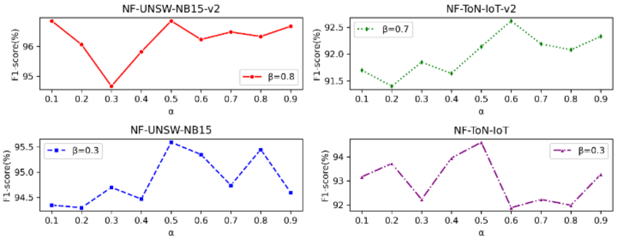

FIGURE 3. The results of different values of α on four datasets.

column, which is a categorical feature, cannot be used directly as a label for multi-classification task training. Thus, it was converted to a numerical feature using digital encoding based on its categories.

F. EXPERIMENTAL RESULTS

This section presents the experimental results of our proposed method. We commenced with binary classification experiments to differentiate between normal and abnormal network flows, followed by detailed multi-class experiments to evaluate the model's detection performance across different network anomalies. Additionally, ablation experiments were conducted to assess the effectiveness of each component in the model.

To ensure the scientific rigor of our experimental results, we conducted each experiment five times and then averaged the results. To determine whether the differences in these results are statistically significant, we performed a t-test on the outcomes of these five experiments, assuming the data follows a normal distribution. First, we established a null hypothesis: there is no significant difference in the mean results of the four evaluation indicators (Accuracy, Precision, Recall, and F1) between GLDAE and the baseline method. We then calculated the mean and standard deviation of the results for both GLDAE and the baseline method. Using these statistics, we derived the p-values. The results show that the p-values for all evaluation indicators are less than the commonly accepted significance level of 0.05. Consequently, we reject the null hypothesis, concluding that there is a significant difference between the evaluation indicators of our proposed method and the baseline method. This difference is not due to chance.

1) BINARY CLASSIFICATION RESULTS

In the binary classification experiments, GLDAE was compared with E-GraphSAGE and Anomaly-E. Table 4 presents the binary classification results of GLDAE and other baseline methods across the NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-UNSW-NB15, and NF-ToN-IoT datasets in terms of Accuracy, Precision, Recall, and F1-score. GLDAE outperformed the other baseline methods in all four metrics for NF-UNSW-NB15-v2, NF-UNSW-NB15, and NF-ToN-IoT. Specifically, compared to the best baseline method, GLDAE showed improvements in Accuracy by 0.41%, 0.13%, and 2.04%, Precision by 0.56%, 0.12%, and 2.38%, Recall by 0.41%, 0.13%, and 2.04%, and F1-score by 0.45%, 0.12%, and 2.88% respectively. However, in NF-ToN-IoT-v2, the Precision and F1-score of GLDAE were lower than the best baseline method, possibly due to the dataset's significantly higher number of abnormal samples compared to normal samples, leading to insufficient model learning and misclassification during classification.

2) MULTICLASS CLASSIFICATION RESULTS

In the multi-classification experiments, GLDAE was compared with E-GraphSAGE, Anomaly-E, and NEGSC. Table 5 displays the multi-classification results of GLDAE and other baseline methods in terms of Accuracy (Acc), Precision, Recall, and F1 score on the NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-UNSW-NB15, and NF-ToN-IoT datasets. Across these datasets, GLDAE outperformed the other baseline methods in all four metrics in NF-UNSW- NB15 and NF-ToN-IoT.Specifically, compared to the best baseline method, GLDAE achieved improvements of 0.4% and 0.32% in Accuracy, 0.64% and 1.29% in F1 score, and 0.4% and 0.32% in Recall. The Precision metric exhibited a more substantial increase, with gains of 0.76% and 2.6%, indicating that GLDAE can more accurately distinguish between normal and abnormal network types in these two datasets.

By analyzing the primary parameters of the experimental models in Table 2, it is evident that GLDAE exhibits greater diversity in network selection compared to the baseline models during the graph embedding stage. This diversity allows GLDAE to leverage the unique characteristics of multiple networks, whereas the three baseline models rely on a single network to extract edge or topological structure features, resulting in limited learned information. GLDAE combines the strengths of E-GraphSAGE, which efficiently aggregates edge features, and GCN, which captures comprehensive graph structure information, through a dual-channel network. This approach not only enhances the richness of the extracted features but also introduces variability in parameter settings, enabling the formation of different model combinations by adjusting the parameters. Furthermore, by integrating graph neural networks into the autoencoder architecture and utilizing MLP and matrix inner product for graph reconstruction, GLDAE ensures robust graph embedding representation during the encoding stage. Consequently, it learns richer and more meaningful features compared to the baseline methods, thereby enhancing overall model performance.

TABLE 6. Results of ablation experiments.

Model	NF-UNSW-NB15-v2		NF-UNSW-NB15		NF-ToN-IoT-v2		NF-ToN-IoT	
	Recall	F1	Recall	F1	Recall	F1	Recall	F1
GLDAE	**97.13%**	**96.85%**	**96.31%**	95.59%	**94.22%**	**92.62%**	**95.23%**	**94.60%**
w/o-str	97.06%	96.58%	96.19%	**95.94%**	93.80%	91.92%	92.84%	91.80%
w/o-ref	96.82%	96.69%	95.88%	95.11%	94.18%	92.10%	93.95%	93.20%
w/o-aug	96.86%	96.62%	96.22%	95.20%	93.72%	91.61%	92.99%	91.90%
w/o weight α	96.71%	96.30%	95.78%	95.41%	93.76%	91.63%	93.34%	92.53%
w/o weight β	96.69%	96.17%	95.89%	94.46%	94.19%	92.46%	93.84%	93.17%

In the NF-UNSW-NB15-v2 and NF-ToN-IoT-v2 datasets, GLDAE exhibited a slight decrease in all four metrics compared to the baseline methods, with variances ranging from approximately 0.15% to 1.14%. Upon examination, it was observed that the NF-UNSW-NB15-v2 and NF-ToN-IoT-v2 datasets consist of 39 features, while the NF-UNSW-NB15 and NF-ToN-IoT datasets contain only 8 features. This observation suggests that GLDAE may be better suited for scenarios with fewer feature dimensions, resulting in relatively strong detection performance. Conversely, in scenarios with higher feature dimensions, GLDAE may be susceptible to overfitting due to the impact of feature dimensions, leading to a marginal decline in detection performance.

3) PARAMETER SENSITIVE ANALYSIS

This section primarily investigated the influence of hyperparameters α (the fusion weight of edge features and graph structural information) and β (the trade-off between contrastive loss and reconstruction loss) on the multi-class detection performance of GLDAE across the NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-UNSW-NB15, and NF-ToN-IoT datasets. The experimental findings are depicted in Fig.2 and Fig.3. During the experiments, leveraging the optimal α and β values of the model, one parameter was held constant while the other was systematically varied within the range of (0,1) to assess the F1 score across different combinations of α and β.

Across the four datasets, GLDAE demonstrated consistent performance under varying values of α, with fluctuation ranges of 2.18%, 1.22%, 1.29%, and 2.71% for the metrics. This stability can be attributed to the feature fusion module, which adeptly integrates edge features and graph structural information to generate the final embedding features, thereby bolstering the algorithm's stability and efficacy. Similarly, when considering different values of β, the model exhibited stable performance across the four datasets, with fluctuation ranges of 1.51%, 1.68%, 1.19%, and 3.48%. This suggests that GLDAE possesses the advantage of being robust to parameter variations.

4) ABLATION EXPERIMENTS

To assess the efficacy of each module in GLDAE, we conducted a series of ablation experiments. In this context, "w/o-str" signifies the exclusion of GCN for graph structural feature learning, elucidating the significance of learning graph structural information. "w/o-ref" indicates the omission of the decoder module for graph reconstruction, confirming the decoder's effectiveness. "w/o-aug" denotes the absence of the graph augmentation module, highlighting the impact of graph augmentation on the model. "w/o weight α" signifies the elimination of the weight parameter for feature fusion, while "w/o weight β" represents the removal of the weight parameters for contrastive loss and reconstruction loss, underscoring the importance of the hyperparameters α and β.

Table 6 illustrates the efficacy of each module in GLDAE. The experimental findings reveal that the complete GLDAE exhibited the most superior overall performance across the four datasets, with only a slightly lower F1 score in NF-UNSW-NB15, potentially attributed to the imbalance in abnormal sample type distribution within the dataset. The outcomes of "w/o-str" and "w/o-ref" indicate that learning edge features or structural information features independently did not yield as strong results as when both were jointly learned, underscoring the importance of considering both aspects for network integrity. The results of "w/o-aug" demonstrate that training solely on the original network graph is less effective than incorporating augmented graphs, highlighting how contrastive learning of graphs can significantly enhance the model's generalization capability and detection performance. Moreover, the outcomes of "w/o weight α" and "w/o weight β" suggest that appropriately adjusting the loss function and feature fusion weights in the model can effectively boost its performance. Overall, the analysis indicates that all modules and hyperparameters of GLDAE have positively contributed to its performance.

IV. CONCLUSION

This study explores anomaly detection in data links, introducing a graph-based approach named Graph Learning Framework for Data Link Anomaly Detection (GLDAE). GLDAE leverages edge features and graph structural information to learn the latent feature representation of links for anomaly detection in data links. Experimental assessments were conducted on various netflow-based datasets, providing quantitative and qualitative evidence of the efficacy of the proposed methodology.

Although GLDAE is designed for anomaly detection in data links, its capability extends to processing information from both nodes and edges, rendering it suitable for various other scenarios. Future research endeavors will explore the temporal dependencies within data traffic and anomaly categories in data links. Given the dynamic nature of real data links, integrating temporal information is crucial for enhancing prediction accuracy.

REFERENCES

[1] J. Piet, A. Sharma, V. Paxson, and D. Wagner, ''Network detection of interactive SSH impostors using deep learning,'' in *Proc. 32nd USENIX Security Symp. (USENIX Secur.)*, 2023, pp. 4283–4300.

[2] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, ''An anomaly mitigation framework for IoT using fog computing,'' *Electronics*, vol. 9, no. 10, p. 1565, Sep. 2020, doi: 10.3390/electronics9101565.

[3] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, ''NetFlow datasets for machine learning-based network intrusion detection systems,'' 2020, *arXiv:2011.09144*.

[4] C. Wang, H. Zhou, Z. Hao, S. Hu, J. Li, X. Zhang, B. Jiang, and X. Chen, ''Network traffic analysis over clustering-based collective anomaly detection,'' *Comput. Netw.*, vol. 205, Mar. 2022, Art. no. 108760, doi: 10.1016/j.comnet.2022.108760.

[5] Q. Xiao, J. Liu, Q. Wang, Z. Jiang, X. Wang, and Y. Yao, ''Towards network anomaly detection using graph embedding,'' in *Proc. Int. Conf. Comput. Sci. (ICCS)*, in Lecture Notes in Computer Science, vol. 12140, V. V. Krzhizhanovskaya et al., Eds., Cham, Switzerland: Springer, 2020, pp. 156–169, doi: 10.1007/978-3-030-50423-6_12.

[6] H. Liu and H. Wang, ''Real-time anomaly detection of network traffic based on CNN,'' *Symmetry*, vol. 15, no. 6, p. 1205, Jun. 2023, doi: 10.3390/sym15061205.

[7] W. T. Lunardi, M. A. Lopez, and J.-P. Giacalone, ''ARCADE: Adversarially regularized convolutional autoencoder for network anomaly detection,'' *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1305–1318, Jun. 2023.

[8] J. Zhou, Z. Xu, A. M. Rush, and M. Yu, ''Automating botnet detection with graph neural networks,'' 2020, *arXiv:2003.06344*.

[9] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, ''E-GraphSAGE: A graph neural network based intrusion detection system for IoT,'' in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2022, pp. 1–9.

[10] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, ''Anomal-E: A self-supervised network intrusion detection system based on graph neural networks,'' *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 110030, doi: 10.1016/j.knosys.2022.110030.

[11] R. Xu, G. Wu, W. Wang, X. Gao, A. He, and Z. Zhang, ''Applying self-supervised learning to network intrusion detection for network flows with graph neural network,'' 2024, *arXiv:2403.01501*.

[12] Z. Wang, J. Chen, and H. Chen, ''EGAT: Edge-featured graph attention network,'' in *Proc. 30th Int. Conf. Artif. Neural Netw. Mach. Learn. (ICANN)*, Bratislava, Slovakia. Berlin, Germany: Springer, Sep. 2021, pp. 253–264.

[13] L. Chang and P. Branco, ''Graph-based solutions with residuals for intrusion detection: The modified E-GraphSAGE and E-ResGAT algorithms,'' 2021, *arXiv:2111.13597*.

[14] Y. Zhang, C. Yang, K. Huang, and Y. Li, ''Intrusion detection of industrial Internet-of-Things based on reconstructed graph neural networks,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2894–2905, Sep. 2023, doi: 10.1109/TNSE.2022.3184975.

[15] T. Altaf, X. Wang, W. Ni, G. Yu, R. Liu, and R. Braun, ''A new concatenated multigraph neural network for IoT intrusion detection,'' *Internet Things*, vol. 22, Jul. 2023, Art. no. 100818, doi: 10.1016/j.iot.2023.100818.

[16] B. Claise, *Cisco Systems NetFlow Services Export Version 9*, document RFC 3954, 2004, doi: 10.17487/rfc3954.

[17] M. Sarhan, S. Layeghy, and M. Portmann, ''Towards a standard feature set for network intrusion detection system datasets,'' *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 357–370, Feb. 2022, doi: 10.1007/s11036-021-01843-0.

[18] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, ''NetFlow datasets for machine learning-based network intrusion detection systems,'' in *Proc. Big Data Technol. Appl.*, 2020, pp. 117–135.

[19] N. Moustafa and J. Slay, ''UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),'' in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[20] N. Moustafa, May 2019, ''ToN_IoT datasets,'' *IEEE Dataport*, doi: 10.21227/fesz-dm97.

[21] M. Sarhan, S. Layeghy, and M. Portmann, ''Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection,'' *Big Data Res.*, vol. 30, Nov. 2022, Art. no. 100359.

CHANG YANG received the Ph.D. degree from the University of Chinese Academy of Sciences, China, in 2020. He is currently a Senior Engineer with the Big Data Center of State Grid Corporation of China. His current research interests include electric power big data and graph neural networks.

LISHA WU received the B.S. and M.S. degrees in software engineering from Beihang University, China, in 2017 and 2020, respectively. She is currently with the Big Data Center of State Grid Corporation of China, working in the big data analysis field. Her current research interests include data link quality, graph computing, and graph neural networks.

JING XU received the B.S. degree in economics from China Agricultural University, China, in 2021, and the M.S. degree in statistics practice from Boston University, USA, in 2023. He is currently with the Big Data Center of State Grid Corporation of China, working in metadata management and big data analysis. His current research interests include artificial intelligence, data transmission, and data management.

YINGJIE REN received the B.S. and M.S. degrees in communication engineering from Beijing Jiaotong University, in 2005 and 2008, respectively. He is currently with the Big Data Center of State Grid Corporation of China, working in grid digitization, including key technologies research, architecture design, and development. His current research interests include data aggregation, data integration, and data link monitoring.

BO TIAN is currently pursuing the M.S. degree in computer technology with North China Electric Power University. His current research interests include artificial intelligence, graph neural networks, and data transmission.

ZHENHUA WEI received the Ph.D. degree in computer application technology from Harbin Institute of Technology, Harbin, China, in 2005. In 2001, he joined Harbin Institute of Technology, as a Lecturer. In 2006, he joined North China Electric Power University, Beijing, China, as a Lecturer, where he is currently an Associate Professor with the College of Control and Computer Engineering. His current research interests include artificial intelligence, data analysis, and computer vision.