Computers & Security 128 (2023) 103131

Contents lists available at ScienceDirect

Computers & Security

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

# Robust unsupervised network intrusion detection with self-supervised masked context reconstruction☆

Wei Wang, Songlei Jian*, Yusong Tan*, Qingbo Wu, Chenlin Huang

*The Colledge of Computer, National University of Defense Technology, Changsha, China*

ARTICLE INFO

*Article history:*
Received 7 November 2022
Revised 27 December 2022
Accepted 2 February 2023
Available online 4 February 2023

*Keywords:*
Network intrusion detection
Unsupervised learning
Self-supervised learning
Temporal context
Anomaly detection

ABSTRACT

Modern network intrusion detection systems always utilize deep learning to improve their intelligence and feature learning abilities. To overcome the difficulties of accessing a large amount of labeled data and achieve early warning, lots of intrusion detection systems focus on unsupervised anomaly detection methods. However, most unsupervised anomaly detection methods ignore the temporal context and anomaly contamination in network intrusion data, which leads to suboptimal detection results. By considering the above practical problems, we propose a robust unsupervised intrusion detection system, i.e, RUIDS, by introducing a masked context reconstruction module into a transformer-based self-supervised learning scheme. The self-supervised learning scheme is designed to learn the intrinsic relationship within temporal contexts. And the masked context reconstruction module can learn more robust representations which are less sensitive to anomaly contamination. Extensive experiments on four intrusion datasets are conducted to show the effectiveness and robustness of RUIDS. Specifically, RUIDS achieves 9.04% and 9.58% improvements over the second-best method on the UNSW-NB15 and CICIDS-WED datasets in terms of AUC value respectively. We also test the robustness of our method with different anomaly contamination ratios, and our algorithm's performance has hardly decreased. The ablation study confirmed the effectiveness of the self-supervised learning scheme and the masked context reconstruction module.



## 1. Introduction

With the widespread use of the Internet, the importance of cyber security is increasing. The network intrusion detection system (IDS) is an effective technology to detect malicious network activity and enhance cyber security. With the powerful representation capability of deep learning, great progress on supervised network intrusion detection has been made Buczak and Guven (2015); Javaid et al. (2016); Wang et al. (2017, 2021). However, supervised deep learning-based intrusion detection methods require a large amount of labeled data for training while manually labeling data is expensive and difficult, especially for some zero-day intrusions or attacks.

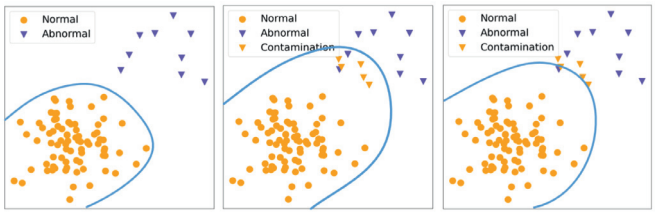
In order to avoid using a large amount of labeled data and achieve early detection of unseen intrusions, a lot of works focus on unsupervised intrusion detection Falco et al. (2019); Nisioti et al. (2018) which organizes the network data packets as normal tabular data and then conducts clustering Zong et al. (2018), reconstruction Alom and Taha (2017) or one-class classification Bergman and Hoshen (2020) to separate the normal data and abnormal data. However, the existing unsupervised intrusion detection methods fail to capture the real data characteristics of intrusion data, i.e., *temporal context* and *anomaly contamination*. Different from the general tabular data, which assumes that the objects follow the independent and identical distribution, the intrusion dataset has temporal context characteristics Also, the temporal context of the intrusion dataset is different from the time series data, such as smart meter and stock price datasets, in which time series exist in the entire dataset. It means from the first data object to the last data object is in line with the time series relationship. Intrusion detection systems typically examine all data packets entering and leaving of network for signs of intrusion. Some intrusion behaviors can only be established under certain contexts. The temporal contexts we proposed are specific in the intrusion datasets which emphasize the time order within

*Abbreviations:* RUIDS, the robust unsupervised intrusion detection system.

☆ This document is the results of the research project funded by the National Natural Science Foundation of China (No. 62002371, No. U19A2060, No. 62172431), the Foundation of PDL (No. WDZC20205250104), the Foundation of National University of Defense Technology (No. ZK21-17).

* Corresponding authors.
*E-mail addresses:* wangwei_09@nudt.edu.cn (W. Wang), jiansonglei@nudt.edu.cn (S. Jian), tanyusong@kylinos.cn (Y. Tan), wuqingbo@kylinos.cn (Q. Wu), clhuang@nudt.edu.cn (C. Huang).

https://doi.org/10.1016/j.cose.2023.103131
0167-4048/




(a) IDS on clean data (b) Common IDS on contaminated data (c) RUIDS on contaminated data

**Fig. 1.** The illustration of intrusion detection on clean data and contaminated data.

the contexts. Take a DOS (Denial of Service) attack as an example, which continuously sends a large number of data packets to the target host, delays the processing speed of the target host, and prevents the processing of normal tasks. The temporal order only exists when the attack begins. And there are no order relationships between different attacks.

Another important characteristic of intrusion data is anomaly contamination. As shown in Fig. 1(a), intrusion detection on clean data can be conducted with the common unsupervised anomaly detection methods which assume that all unlabeled data are normal ones. However, the practical intrusion situation is usually contaminated with unknown anomaly data which may cause a biased decision boundary learned by common IDS as shown in Fig. 1(b). Specifically, if there are outliers in the reconstruction-based training data, the model will learn some information about the outlier data, which will cause the decision boundary to be biased. Most one-class classification methods ignore the anomaly contamination and assume that all the training data is normal in order to learn a hypersphere or distance boundary to classify all positive samples into one class. And clustering-based detection methods are easily susceptible to anomaly contamination which makes their performance unstable.

To capture the temporal context and alleviate the anomaly contamination problem, we propose a robust unsupervised intrusion detection system, i.e., RUIDS, by introducing masked context reconstruction into a transformer-based self-supervised learning scheme. Different from the common self-supervised learning scheme, we adopt a transformer module to represent the temporal contexts instead of transforming individual objects with a multilayer perceptron. In this way, the temporal contexts and the sequential information inherent in the contexts are both preserved. With the transformer-based self-supervised learning scheme, the abnormal behavior can be fully exposed by transforming the original feature space into multiple representation spaces. Moreover, we propose the masked context reconstruction module which reconstructs the masked data by the unmasked data in a context. The masked context reconstruction process is conducted on each transformer-based data transformation. In this way, the self-supervised learning scheme becomes less insensitive to anomaly contamination and the contextual relationships can be learned in the transformed representation as shown in Fig. 1(c). The contributions of our work are summarized as follows:

- We propose a robust unsupervised intrusion detection system, i.e., RUIDS, with a novel self-supervised masked context reconstruction scheme, which simultaneously achieves accurate and robust intrusion detection without any labeled data.
- We propose the transformer-based self-supervised learning scheme for temporal context which is capable of learning the intrinsic relationships within temporal contexts.
- We propose the masked context reconstruction module inserted into the self-supervised learning scheme to learn more discriminative representations which magnify the abnormal intrusion behaviors and achieve greater tolerance for anomaly contamination.

Extensive experiments show that (1) the RUIDS outperforms the state-of-the-art unsupervised methods in terms of accuracy, F1 score, and the AUC (Area Under Curve) value on four real-world datasets; (2) comparing with other unsupervised methods, RUIDS is more robust with contamination data, and its detection accuracy hardly drops when the contaminated ratio less than 30%; (3) the transformer-based self-supervised learning scheme and mask context reconstruction module both make contributions to the intrusion detection with a thorough ablation study.

## 2. Related work

### 2.1. Unsupervised anomaly detection

In recent years, in order to detect network attacks with the absence of labels, researchers have proposed a large number of unsupervised intrusion detection methods. Unsupervised deep learning methods have become mainstream because of their ability to detect new types of attacks. We categorize unsupervised IDS into the reconstruction-based method, clustering-based method, and one-class classification method.

#### 2.1.1. Reconstruction-based methods

Reconstruction-based methods assume that outliers cannot be efficiently compressed or reconstructed from a low-dimensional mapping space. Compared to normal data, outliers have a high cost of reconstruction. The representative method of the reconstruction-based method is Principal Component Analysis (PCA) by using linear projection Tran and Tran (2018). The improved method O'Reilly et al. (2016) cuts down the sensitivity of data by forcing reduced dimensions of data. Autoencoder Sadaf and Sultana (2020); Xu and Fan (2022) is a commonly reconstruction-based neural network. It is based on the backpropagation algorithm and optimization method (such as the gradient descent method). Autoencoder methods use the input data as supervision to guide the neural network to learn a better representation. Abnormal detection is determined by the difference between the reconstructed output and the input. Generative Adversarial Network (GAN) Beula Rani and Sumathi M. E (2020) detects anomalies by modeling normal behavior through an adversarial training process and determines anomalies based on anomaly scores. However, reconstruction-based methods do not consider the loss of effective information caused by compressing data. Moreover, such methods usually cannot effectively reconstruct the original data from the low-dimensional projection of the data if there exists anomaly contamination.

#### 2.1.2. Clustering-based methods

The clustering-based method is another popular anomaly detection pattern. The deep clustering method formed by the combination of deep learning and clustering improves the clustering effect. This method usually uses a deep neural network to extract features and cluster the features to obtain the detection results. K-means algorithm Ma et al. (2019); Yang et al. (2016) realizes the division of samples by minimizing the mean square error within the class, but this algorithm is greatly affected by initialization, and cannot handle data with non-convex cluster shapes. The spectral clustering algorithm Shaham et al. (2018) transforms the clustering problem into an undirected graph multiplexing problem. By explicitly solving the feature map, a batch training strategy can be used to improve the scalability of large-scale data. But this explicitly solved feature map is not guaranteed to be globally optimal. Subspace clustering Ji et al. (2017) assumes that the data

of the same class are distributed in the same subspace, and the data of different classes are in different subspaces. The deep subspace clustering algorithm Zhang et al. (2019) can make full use of the powerful feature extraction ability of neural networks and use more discriminative features to find more accurate subspaces. Compared with other deep clustering algorithms, it has a better effect on processing high-dimensional data. Gaussian Mixture Models (GMM) assume that the samples of each class obey a separate Gaussian distribution, and the overall data obey a mixture of multiple Gaussian distributions. DAGMM Zong et al. (2018)combines the dimensionality reduction process and the density estimation process for end-to-end joint training. Compared with other algorithms, DAGMM has a large performance improvement. Mutual information Zhao et al. (2020) and KL divergence Xie et al. (2015) can also be used as clustering metrics. However, the learning models of deep clustering still cannot learn discriminative abnormal features, and these methods usually only use the statistical features of network traffic for intrusion detection, ignoring the time-series features of network traffic, and the detection accuracy is low.

#### 2.1.3. One-class classification methods

One-class methods assume that only normal data exists, or that the amount of abnormal data relative to normal data is minimal. OCSVM Schölkopf et al. (1999) attempts to construct a hyperplane that separates all data points from zeros in the feature space, maximizing the distance from the separating hyperplane to the zeros. Deep-SVDD Ruff et al. (2018)adopts a hypersphere rather than a hyperplane to determine outliers. The algorithm obtains a spherical boundary around the data in feature space, and the volume of this hypersphere is minimized, thereby minimizing the effect of outliers. This method makes decisions by hyperplane or hypersphere and judges the outer samples. Because kernel function calculation is time-consuming, it limits its application in massive data scenarios. Classical AD methods such as the One-Class SVM often fail in a high-dimensional and deep method that exits the hypersphere collapse

### 2.2. Self-supervised anomaly detection

Self-supervised learning is a type of unsupervised learning. Self-supervised learning mainly uses pretext tasks to mine its own supervision information from large-scale data. The neural network is trained by structured supervision information so that it can learn valuable representations for downstream tasks. The self-supervised anomaly detection models based on their pretext task into self-predictive methods and contrastive methods.

Self-predictive methods usually create the pretext task for every individual sample. It learns data representation by predicting the data transformation type or reconstructing original samples. Liron Bergman and Hoshen (2020) proposes the GOAD algorithm which constructs a classification task through geometric transformation for anomaly detection. GOAD trains a neural network to map the transformed data to a new sample space, and maps each transformed subspace to a hypersphere under the idea of One-class classification. Wang et al proposed the SLA$^2$P Wang et al. (2021) framework for anomaly detection. SLA$^2$P applies random projections to the embeddings by multiplicating matrices sampled randomly from a standard normal distribution. Transformations by different matrices give rise to pseudo labels, on which a DNN classifier is trained. The anomaly scores are generated leveraging the predictive uncertainty estimates of the network on the perturbed transformed features.

Contrastive methods mainly construct positive and negative samples through a pretext task. Representation is learned by comparing the distance difference between positive and negative samples. Inspired by the self-supervised learning of images, Chen et al Qiu et al. (2021) borrowed learnable transformations to deal with anomaly detection and embedded the transformed data into a semantic space where the transformed data representations are similar to the original data representations. The transformation spaces are easy to distinguish from each other. A new contrastive learning method for enhanced distribution is proposed by Kihyuk Sohn et al. (2020). This method expands the distribution of training data through data enhancement and uses the transformation data as different samples for self-supervised learning. Self-supervised learning tends to make the distance between different samples pull away, so in addition to pulling away from the original different samples, the transformation samples are also pulled away from the original samples. The sample distribution of the original data set does not obey the uniform distribution, and it is easy to separate normal samples from abnormal samples. Finding suitable pretext tasks is the most in-demand problem for self-supervised learning and the representations learned by self-supervised methods need to meet the optimization goals of anomaly detection.

### 2.3. Time series anomaly detection

The time series anomaly detection method based on deep learning performs well in high-dimensional data. Recurrent neural network (RNN) can capture the time series characteristics and remember the sequence data well, so it has the most advantage in dealing with sequence data. LSTM is a variant of RNN and performs better in solving long-distance memory problems. Malhotra et al Malhotra et al. (2015) proposed a time series anomaly detection algorithm based on LSTM (Stack-LSTM). The model first trains the model on the non-abnormal time series in an unsupervised way and then takes the trained model as a predictor to obtain the prediction error. The prediction error is input into the multivariate Gaussian distribution for anomaly detection. Lin et al Lin et al. (2020) proposed a time series anomaly detection model based on the VAE-LSTM hybrid model. The model uses the VAE module to form robust reconstruction features on the short window, then uses LSTM to estimate the long-term correlation of the series on the basis of VAE reconstruction features. The anomaly is distinguished according to the reconstruction error and threshold. Although these methods can achieve good performance in modeling the short-term timing information in the sequence, they are difficult to capture the long-term timing dependence in the data. At the same time, we notice that the attention mechanism can allow the information of the time step at any distance from the current time step to flow to the current time step, which makes the time series modeling method based on the attention mechanism can provide the model with long-term timing dependence capture ability.

The transformer is an encoder-decoder structure based on an attention-based mechanism. Yang Yang et al. (2022) proposes an intrusion detection model based on an improved vision transformer (ViT). A sliding window mechanism is presented to improve the capability of modeling local features for ViT. Ho Ho et al. (2022) uses image conversion from network data flow to produce an RGB image that can be classified using advanced deep-learning models. A Vision Transformer is used as a classifier to classify the resulting image. Shreshth Tuli et al. (2022) proposes TranAD, which is an anomaly detection and diagnosis model based on the deep transformer. TranAD uses attention-based sequence encoders to quickly perform reasoning and understand broader time trends in data. TranAD uses self-tuning based on focus scores to achieve powerful multimodal feature extraction and confrontation training to achieve stability. These methods directly process the intrusion data or convert it into image data, without taking into account the temporal context of the intrusion data.

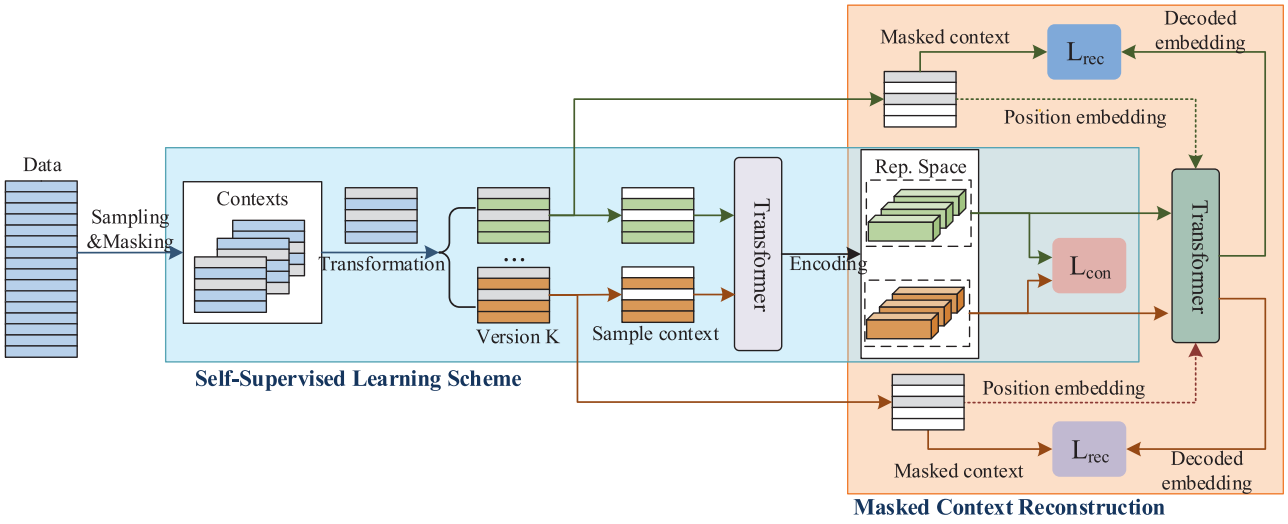



**Fig. 2.** The overview of RUIDS.

## 3. Method

We develop a robust unsupervised intrusion detection system (RUIDS), a deep intrusion detection method based on a transformer-based self-supervised learning scheme. A masked context reconstruction module is added to this scheme which is used in the temporal contexts to enhance the robustness of RUIDS. The scheme and the module are joint training together in the training process and we define an intrusion score to measure the abnormity of test data.

Fig. 2 shows the overview architecture of our RUIDS. It contains two main parts,i.e., the self-supervised learning scheme and the masked context reconstruction module. Intrusion data sets are generally expressed as feature information extracted from data packets which are shown as tabular data. To better extract the time series information of the data, we slice the data and set $N$ data objects to act as a context according to the time stamps. In the self-supervised learning scheme, we randomly sample multiple data objects from the context and mask the rest objects. We design a series of learnable transformations to transform the context into different latent spaces. A transformer module is added to these transformations to extract the sequential information of the retained data objects. The transformation spaces provide similar semantic information to the original features, and different transformations have different views from each other. The contrastive loss between the transformation version and the original version is calculated to optimize the neural network. The masked samples can be reconstructed by retaining samples according to the characteristic of the temporal context. In the masked context reconstruction module, We add a transformer module as the decoder to reconstruct masked samples. The input of this decoder is the transformer encode embedding of the retaining data and the position embedding of the masked data. The reconstruction loss only considers the mean square difference of masked data objects.

### 3.1. Transformer-based self-supervised learning scheme

In unsupervised intrusion detection tasks, we cannot directly train neural networks without labels. Self-supervised learning methods use certain characteristics of the data itself to learn data representations. For image or video data, the method of constructing positive and negative pairs of data through image enhancement (such as cropping, flipping, and color transformation), and performing constructive learning to obtain data representation has achieved good results Feng et al. (2021); Li et al. (2021). For tabular data, the NeuTral AD method proposed by Qiu et al Qiu et al. (2021) to construct the transformation version of context is state-of-the-art and we learn from this method in our work. The data of the intrusion dataset has a contextual correlation and has hardly been considered in previous work. In order to extract this sequential characteristic, we design a self-supervised learning scheme based on the transformer structure Vaswani et al. (2017).

Assume that a network behavior dataset $X$ has $n$ objects, that is, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and there is no label for every object. For every data object $x \in X$, the dimension of $x$ is $I$. Inspired by the MAE He et al. (2021) model, we process the data in context and take $C$ data objects as a data context without repetition according to the original order and the number of contexts is $H = \lfloor (n/C) \rfloor$. The $h$ data context is represented as $X_h = \{\mathbf{x}_{C(h-1)+1}, \mathbf{x}_{C(h-1)+2}, \ldots, \mathbf{x}_{C*h}\}$. For each data context, we randomly sample $S$ data objects and mask the remaining data samples. This sample operation is not repeated and the sample order of different data contexts is not the same as each other. For the $h$ data context, we mark the sample data set as $X_h^S$ and the masked data set as $X_h^M$.

Consider a set of transformations $T$, $T = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_K\}$ and we assume that these transformations are learnable. That is, these transformations can be expressed by a series of combinations of functions that can be optimized by gradient descent. These transformations encourage the transformed version $x^m = T_m(x)$ to be similar to the original version $x$ while encouraging it different from other transformation versions $x^n = T_n(x)$ if $m \neq n$. We define the score of two different transformed versions as

$$s(x^m, x^n) = \exp\left(\mathrm{sim}\left(f_\phi(T_m(x)), f_\phi(T_n(x))\right)/\tau\right) \tag{1}$$

where $\tau$ is an adjustable temperature parameter and is designed to get a better result. The similarity is defined as the cosine similarity $\mathrm{sim}(a, b) = a \cdot b / \|a\| \|b\|$ in the embedding space. The $f_\phi$ function is designed to extract the inherent feature. We deploy a transformer module as this $f_\phi$ function to extract sequential information in the embedding space of transformation versions.

In our self-supervised scheme, the data object in the context $X_H$ is divided into the sampled dataset $X_H^S$ and masked dataset $X_H^M$ after the transformation operation. Only sampled dataset $X_H^S$ in the context is used to extract the temporal context and expose the abnormal information through the transformer module. All objects of the context $X$ are mapped to an embedding space through a linear

layer to get the input of the transformer encoder $\mathbf{z}_0$.

$$\mathbf{z}_0 = X\mathbf{E} + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{I\times D}, \mathbf{E}_{pos} \in \mathbb{R}^{C\times D} \tag{2}$$

The transformer encoder uses constant latent dimension $D$ through all of its layers, so we map the data object to $D$ dimensions with a trainable linear projection embedding $E$ (Eq. 2). Position embeddings $\mathbf{E}_{pos}$ are added to the data embeddings to retain the position information. The transformer encoder consists of self-attention(SA) and MLP block. Layernorm (LN) is applied before every block and residual connections after every block. We put the embedding of $X^S$ dataset $\mathbf{z}_0^S \in \mathbf{z}_0$ into the transformer module to get the encoder embedding as Eq. 3.

$$\begin{aligned} \left(\mathbf{z}_{en}^S\right)' &= \mathrm{SA}_{\mathrm{encoder}}\left(\mathrm{LN}\left(\mathbf{z}_0^S\right)\right) + \mathbf{z}_0^S \\ \mathbf{z}_{en}^S &= \mathrm{MLP}_{\mathrm{encoder}}\left(\mathrm{LN}\left(\left(\mathbf{z}_{en}^S\right)'\right)\right) + \left(\mathbf{z}_{en}^S\right)' \end{aligned} \tag{3}$$

As we can see, the transformer encoder module only deals with the linear embedding of the sampled dataset. In our scheme, we apply a one-layer transformer module as the $f_\phi$ function in the Eq. 1 to obtain the embedding of the transformed version of $T_i(x)$. The contrastive loss is designed to guide optimizing the network.

$$L_{con} = -\sum_{k=1}^{K} \log \frac{s\left(x^k, x\right)}{s\left(x^k, x\right) + \sum_{l\neq k} s\left(x^k, x^l\right)} \tag{4}$$

The contrastive loss encourages the transformation version $x^k$ to be similar to $x$ in the embedding space and different transformation versions are different from each other. Contexts are transformed by a set of learnable transformations and then the sampled dataset transformations are mapped into latent space by the transformer module. The transformations and the transformer encoder are trained jointly by the contrastive loss.

### 3.2. Masked context reconstruction

In the transformer-based self-supervised learning scheme, only sampled objects are used to extract the temporal context feature. We get the embedding of sampled objects $\mathbf{z}_{en}^S$ through the one-layer transformer encoder. For masked objects, we propose a masked context reconstruction module adding to the self-supervised scheme to enhance the robustness of RUIDS.

The masked context is reconstructed through a one-layer transformer structure which is used as a decoder. The input of the decoder is consist of the embedding of the encoder $\mathbf{z}_{en}^S$ and mask tokens $\mathbf{E}_{token}$, which is a zero vector that indicates the presence of the masking part. The input of the decoder $\mathbf{z}_1$ is calculated as

$$\mathbf{z}_1 = \left[\mathbf{z}_{en}^S; \mathbf{E}_{token}\right] + \mathbf{E}_{pos}, \mathbf{E}_{token} \in \mathbb{R}^{(K-L)\times D}, \mathbf{E}_{pos} \in \mathbb{R}^{K\times D} \tag{5}$$

The splicing of $\mathbf{z}_{en}^S$ and $\mathbf{E}_{token}$ is set to the previous position as the original order before masking. Position embedding is added to the set of splices to indicate the position of all embedding. The structure of the decoder is similar to the encoder. The transformer decoder also includes a self-attention(SA) and an MLP block. We get the reconstruction of the masked samples $y$ by the decoder module which is consisted of a one-layer transformer module and a linear layer.

$$\begin{aligned} \mathbf{z}'_{de} &= \mathrm{SA}_{\mathrm{decoder}}(\mathrm{LN}(\mathbf{z}_1)) + \mathbf{z}_1 \\ \mathbf{z}_{de} &= \mathrm{MLP}_{\mathrm{decoder}}\left(\mathrm{LN}\left(\mathbf{z}'_{de}\right)\right) + \mathbf{z}'_{de} \\ \mathbf{y} &= \mathrm{LN}(\mathbf{z}_{de}) \end{aligned} \tag{6}$$

The last linear layer LN changes the output of the transformer to the original dimension. The output of the decoder module $\mathbf{y}$ has the same dimension as the input data. We calculate the mean squared error (MSE) between the output of the decoder module $Y_h^M$ and the original data $X_h^M$ of masked data as the reconstruction loss.

$$L_{\mathrm{rec}} = \frac{1}{M*H}\sum_{h=1}^{H}\left(Y_h^M - X_h^M\right)^2 \tag{7}$$

We calculate the average reconstruction loss of objects which belong to context-masked sets $X_h^M$.

### 3.3. Learning intrusion score

Normal objects can be reconstructed easily by the autoencoder module and intrusion objects have difficulty reconstructing to original data for intrusion objects are far from normal objects in embedding space. It means, that the average value of reconstruction loss for intrusion objects is bigger than normal objects and reconstruction loss can be used to detect intrusion objects. The numerator of the contrastive loss encourages the transformed samples to similar to the original objects while the denominator pushes the transformer version apart from each other in embedding space.

During training, the objects of intrusion datasets are divided into two datasets: the sampling objects dataset and the masking objects dataset. Context objects are transformed by a set of learnable transformations and the transformed version of the sampling dataset is embedded into a latent space by a one-layer transformer module. Masking objects are reconstructed through the transformer encoder and transformer decoder modules. The training loss of the RUIDS algorithm is as:

$$L = L_{con} + \alpha L_{rec} \tag{8}$$

The training loss is consist of the contrastive loss of the sampling context and the reconstruction loss of the masking context. The contribution of two loss is determined by the $\alpha$ factor and we will test parameter sensitivity in terms of the value of $\alpha$ in the next section. The detail of the training process is shown in Algorithm 1

**Algorithm 1** Learning Process for RUIDS.

**Require:** $X$ -the intrusion dataset, $H$-the number of contexts, $K$-the number of transformations, $\alpha$- the proportion of loss
**Ensure:** $\Theta$ - the parameters of the network

```
Generate context set{X_1, X_2, ..., X_H}
for i = 1 to epoch do
    for h = 1 to H do
        for k = 1 to K do
            X_h^k ← T_k(X_h)
        end for
        X_h ← [X_h; X_h^1; ... x_h^K]
        Generate the masking context X_h^M and sampling context X_h^S
        z_en_h^S ← Transformer_encoder(X_h^S) ref. to Eq. (2) and Eq.(3)
        Calculate the L_con ref. to Eq.(4)
        z_in_h ← [z_en_h^S; E_token] + E_pos
        z_de_h ← Transformer_decoder(z_in_h) ref. to Eq. (6)
        Calculate the L_rec ref. to Eq.(7)
        L = L_con + α L_rec
        Θ ← Θ − Adam[∇_Θ L]
    end for
end for
```

The detection process is different from the training process. There are no data in the testing dataset which is masking to generate a masking dataset. During the testing phase, we learn an intrusion score to judge the degree of abnormality of each data. We use

the value of the contrastive loss function of the data as the intrusion score. During the training process, the masking dataset does not go through the transformer-encoder structure, and the contras loss value cannot be obtained. So the data was not masked during the testing phase. We define the intrusion score $S(x)$ as

$$S(x) = L_{con}(x) \tag{9}$$

The process of calculating the intrusion score is shown in Algorithm 2. The test objects are sorted according to the descending order of the intrusion score and the part of the objects with the highest exception score are outliers.

**Algorithm 2** Detection Process for RUIDS.

**Require:** $X$ -the intrusion dataset, $H$-the number of contexts, $K$-the number of transformations, $\Theta$ - the parameters of the network

**Ensure:** $S$ - the intrusion score

```
Generate context set{X_1, X_2, ..., X_H}
for i = 1 to epoch do
    for h = 1 to H do
        for k = 1 to K do
            X_h^k ← T_k(X_h)
        end for
        X_h ← [X_h; X_h^1; ...x_h^K]
        z_en_h ← Transformer_encoder(X_h) ref. to Eq. (2) and Eq.(3)
        Calculate the S ref. to Eq.(9)
    end for
end for
```

## 4. Experiments

In this section, we first introduce a detailed experiment setup which includes the evaluation datasets, data preprocessing, comparison methods, and evaluation metrics. Then we conduct the experiments to demonstrate the effectiveness of the proposed RUIDS, we evaluate the RUIDS and the state-of-the-art unsupervised anomaly detection methods on 4 datasets. Also, to show the robustness of our method, We test the performance of the algorithm when the training data had different levels of contamination. And we do an ablation study to reveal the contributions of the two-loss function in RUIDS. Finally, we conduct the parameter sensitivity test to show the effect of different hyperparameters with respect to the performance.

### 4.1. Experiment setup

#### 4.1.1. Datasets and evaluation metrics

In our experiment, we deploy 4 datasets: KDDCUP, UNSW-NB15, CICIDS17-Friday, and CICIDS17-Wednesday.

- KDDCUP: KDDCUP Chen et al. (2005) was extracted from 9 weeks of network connectivity data collected in a network environment established by Lincoln Laboratory to simulate the United States Air Force LAN. This dataset contains 41 feature vectors and the label value. In addition to normal data, there are 4 kinds of attack data that are marked as abnormal. This resulted in the abnormal rate high to 81%. In the experiments, we treat normal data as 'anomaly' and attack data as 'normal'.
- UNSW-NB15: The UNSW-NB15 dataset Moustafa and Slay (2015) was created through the IXIA PerfectStorm tool by the Network Scope Lab of the Australian Centre for Cyber Security. This dataset contains 9 attack types. In the experiments, we mark all attack data samples as 'anomalies'.
- CICIDS17: The CICIDS17 dataset Sharafaldin et al. (2018) contains benign and most common attacks, similar to real-world data. Different attacks were implemented in different time periods in a specific way using network configuration files. This dataset contains multiple files, and in our experiments, we adopted the CICIDS17-Wed dataset and the CICIDS17-Fri dataset which are collected on Wednesday and Friday. These two files contain portscan and dos hulk attacks respectively. We mark data other than benign data as 'anomaly'.

**Table 1**
Statistics of the public benchmark datasets. Dim. refer to the dimension of the features.

| Dataset | Normal | Abnormal | Anomaly Ratio % | Dim. |
|---|---|---|---|---|
| **KDDCUP** | 396,743 | 97,278 | 19% | 41 |
| **UNSW-NB15** | 56,000 | 119,341 | 68% | 48 |
| **CICIDS17-FRI** | 127,538 | 158,930 | 55% | 80 |
| **CICIDS17-WED** | 440,031 | 252,672 | 36% | 80 |

Detailed information about these datasets is shown in Table.1. The division of data in the experiment refers to the method of DAGMM Zong et al. (2018). We train baseline deep-learning methods with 50% of normal data, and the remaining normal data and abnormal data are used as the test dataset to test the detection effect. Precision, recall, and F1-score are usually selected as the evaluation metrics for unsupervised anomaly detection. In our experiment, we use the true number of anomalies of the test data as the threshold for the algorithm to determine anomalies. This will result in these three evaluation metrics of the algorithm being very close. Considering the extremely unbalanced traffic categories in real network scenarios, we add AUC (area under the receiver operating characteristic curve) as an experimental evaluation metric. Finally, we use accuracy, F1-score, and AUC as evaluation metrics.

#### 4.1.2. Comparison methods

We consider traditional and deep learning anomaly detection methods and also compare them with state-of-the-art self-supervised methods.

- OC-SVM: By mapping the data to the feature space corresponding to the kernel, OC-SVM constructs a hyperplane between the data and the origin, which maximizes the distance from the hyperplane to zero.
- LOF: LOF(local outlier factor) Breunig et al. (2000) is based on density to determine outliers. By assigning an outlier factor LOF that depends on the density of the neighborhood to each data point, it is then judged whether the data point is an outlier.
- IF: Isolation Forest (iForest) Liu et al. (2012) is a fast outlier detection method based on ensemble, which utilizes a binary search tree structure called isolation tree (iTree) to isolate samples. Due to the small number of outliers and their alienation from most of the samples, outliers will be isolated earlier.
- DAGMM: DAGMM organically combines the dimensionality reduction process and the density estimation process for end-to-end joint training.
- Deep-SVDD: DEEP-SVDD uses a neural network to extract data features, and shrinks normal samples within the hypersphere (the center is $C$, the radius is $R$, and the center needs to be determined in advance). Abnormal samples are far away from the hypersphere and fall outside the sphere.
- GOAD: GOAD projects the data to different regions through some geometric transformations, and map these transformed data to a new sample space by training a neural network. Under the idea of one-class classification, each geometric transformation subspace is mapped into a sphere.
- NeuTraL AD: NeuTraL AD implements end-to-end anomaly detection using learnable transformations, which embed the transformed data into a semantic space where the transformed

data representation is similar to the original data representation, and different transformations are easy to distinguish.

Among the above several baseline methods, OC-SVM, LOF, and IF methods are traditional machine learning methods. A testing set is used for model training in these three experiments since there are no anomalies in the training set. To get a better detection result, we set the parameters of an algorithm according to the true proportion of the abnormal data. The last four baseline methods are deep-learning-based algorithms. In the DAGMM algorithm, the encoder module uses three hidden layers to compress data samples into 10 dimensions. The estimation network is a two-layer neural network and the output is the probability of belonging to normal and abnormal classes. The DEEP-SVDD algorithm compresses data samples into a low-dimensional sphere, and the compressed space of the KDDCUP dataset, UNSW-NB15 dataset, and CICIDS17 datasets are 30, 40, and 20 dimensions, respectively. The GOAD algorithm uses a 1-D convolution neural network to get different transformation versions. The transformed data are mapped into a 40-dimensional feature space for anomaly detection. The parameters of the NeuTraL AD neural network are the same as the transformation module of the RUIDS system. The batch size of these networks is 256 and the number of epoch is 30. Each algorithm is run 10 times, and the mean value is taken as the experimental result.

#### 4.1.3. Implementation details

In order to compare with the previous excellent algorithms, we adopt the data processing method from Deng et al Zong et al. (2018). For the KDDCUP dataset and UNSW-NB15 dataset, we perform one-hot processing on the categorical features, and the processed feature dimensions become 121-dimensional and 179-dimensional, respectively. For the CICIDS17 dataset, remove dirty data containing Nan and Infinity. For the division of the data set, we take half of the normal data as the training set and take the addition normal data and all anomaly data as the test set.

The RUIDS includes a transformer-based self-supervised learning scheme and a masked context reconstruction module. In the self-supervised learning scheme, we designed a learnable transformation set with 11 different transformations. Each transformation is consist of 2 fully connected layers with RELU activations. The parameters of the two linear layers are FC (input_dimension, hidden_dimension, RELU) and FC (hidden_dimension,input_dimension, RELU).

In the masked context reconstruction process, we slice each dataset by taking $C = 100$ data objects as a context according to the original order. Last parts objects less than $C$ are dropped. In each context, we randomly sample $r\%$ of objects, and we set $r = 90$ in the experiment. The sample context changed the dimension through a linear layer before being put into the transformer-based encoder module. The transformer embedding is followed by a normalization layer. In the masked context reconstruction module, the output of the encoder and the positional embedding are spliced as the input of the decoder, which also consisted of a linear layer, a transformer structure, and a normalization layer. The reconstructions of masked samples are get from a linear layer after the output of the transformer decoder. The network parameters of the three different datasets are shown as Table.2.

### 4.2. The effectiveness of RUIDS

The intrusion detection results on four datasets are demonstrated in Table 3, which includes the accuracy, F1-score, and AUC value performance. According to the table, our method achieves the best performance in terms of all metrics compared with all other methods on all datasets.

Among the three traditional anomaly detection algorithms, the overall effect of the ONE-SVM algorithm is better than the LOF and IF algorithms. This is because, in the process of algorithm implementation, the ONE-SVM algorithm learns a hyperplane based on normal data, and the training data in our experiment is not doped with abnormal data, which makes the hyperplane learned by the ONE-SVM algorithm more efficient. The LOF algorithm needs to compare with the surrounding data during the training process and determine the abnormality of the data according to the set threshold. The algorithm requires a low proportion of data anomalies, and the data sets except KDDCUP have a high proportion of anomalies in the experiment, resulting in poor detection results of the algorithm. The IF algorithm is also sensitive to the global sparse points and is not good at dealing with local relatively sparse points, and the result of detection for a dataset with a large number of abnormal data is not good.

Overall, deep learning-based methods outperform traditional machine learning, which indicates that neural networks are more capable of extracting data features. Meanwhile, in these methods, we set precise thresholds for anomaly detection. For example, if the abnormal proportion in the test data is 20%, then we determine that the 20% samples with the largest outliers are abnormal. This approach also improves the effect of detection. These detection methods perform better on the KDD dataset than on the other three datasets. This is because the other three datasets have a higher proportion of anomalies, and the UNSW-NB15 dataset with the highest proportion of anomalies has the worst detection effect.

### 4.3. The robustness of RUIDS

At present, most of the unsupervised anomaly detection methods based on deep learning with good effect need to use clean data for training to learn a better feature representation of normal data. These methods put forward a high requirement for training data that most or all of the training data be clean data. In practical applications, we cannot determine the abnormality of training data, which requires the algorithm of abnormality detection to be highly robust and insensitive to abnormal data.

In experiments, we test the robustness of algorithms. We added 5%, 10%, 15%, 20%, 25%, and 30% of contamination data to the original clean training data respectively, and also reduced the added part in the corresponding test data. The results of deep-learning-based algorithms with different degrees of abnormality are shown from Fig. 3 to Fig. 6. Compared with other algorithms, our algorithm has the highest robustness. As the proportion of contamination data in the training data increases, although the performance of our algorithm declines, the declining trend is slower than other algorithms. The performance of our algorithm is better than other algorithms in the same training set. RUIDS system performs differently on different datasets. The best performance of our algorithm is shown on the KDD dataset. When the contamination data increases by 30%, the accuracy, F1-score, and the AUC of our algorithm only decrease by 1.3%, 4.4%, and 2.6%, respectively. The algorithm has the worst effect on the UNSW-NB15 data set, which may be due to the high abnormality proportion of the UNSW-NB15 dataset itself. When 30% contamination data is added to the training data, the accuracy, F1-score, and the AUC of the algorithm drop by 13.8% and 8.4%, and 22.2% respectively.

Contrary to the RUIDS algorithm, the result of the DAGMM algorithm drops mostly on the KDD dataset. The DAGMM algorithm needs to map high-dimensional data to a low-dimensional space and perform density estimation in the low-dimensional space. The algorithm is sensitive to the distribution of the data set. In the low-dimensional space, some abnormal objects are hidden in the normal objects, and the detection effect is reduced. As shown in Fig. 4, the effect of the DEEP-SVDD algorithm on the UNSW-NB15 dataset

**Table 2**
Implementation framework settings for different datasets .

| | KDD | UNSW-NB15 | CICIDS17 |
|---|---|---|---|
| Transformation | FC(121,60,RELU) | FC(179,90,RELU) | FC(78,40,RELU) |
| | FC(60,121,RELU) | FC(90,179,RELU) | FC(40,78,RELU) |
| Reconstruction | FC(121,60) | FC(179,88) | FC(78,40) |
| | Transformer Block(60) | Transformer Block(88) | Transformer Block(40) |
| | LayerNorm(60) | LayerNorm(88) | LayerNorm(40) |
| | FC(60,100) | FC(88,160) | FC(40,60) |
| | Transformer Block(100) | Transformer Block(160) | Transformer Block(60) |
| | LayerNorm(100) | LayerNorm(160) | LayerNorm(60) |
| | FC(100,121) | FC(160,179) | FC(60,78) |

**Table 3**
The intrusion detection results with the state-of-the-art methods .

| | KDD | | | UNSW-NB15 | | | CICIDS-WED | | | CICIDS-FRI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | F1 | AUC | ACC | F1 | AUC | ACC | F1 | AUC | ACC | F1 | AUC |
| ONE-SVM | 0.8072 | 0.7734 | 0.8564 | 0.8271 | 0.8943 | 0.7031 | 0.7243 | 0.7809 | 0.7098 | 0.8563 | 0.9085 | 0.7493 |
| LOF | 0.7555 | 0.6284 | 0.7231 | 0.2782 | 0.2070 | 0.5421 | 0.4418 | 0.1381 | 0.4884 | 0.2789 | 0.0973 | 0.4461 |
| IF | 0.7801 | 0.7494 | 0.8361 | 0.8456 | 0.9129 | 0.5938 | 0.7094 | 0.7719 | 0.6937 | 0.7961 | 0.8750 | 0.6442 |
| DAGMM | 0.9071 | 0.8589 | 0.9679 | 0.7718 | 0.8592 | 0.7542 | 0.6512 | 0.6738 | 0.6453 | 0.7479 | 0.8224 | 0.7096 |
| DEEP-SVDD | 0.9911 | 0.9866 | 0.9900 | 0.7622 | 0.8532 | 0.6139 | 0.8842 | 0.8916 | 0.8838 | 0.9022 | 0.9314 | 0.8803 |
| GOAD | 0.9900 | 0.9848 | 0.9887 | 0.8551 | 0.9106 | 0.7647 | 0.7376 | 0.7551 | 0.7362 | 0.9586 | 0.9710 | 0.9493 |
| NeuTraL AD | 0.9973 | 0.9959 | 0.9969 | 0.8813 | 0.9267 | 0.8072 | 0.8949 | 0.9017 | 0.8944 | 0.9829 | 0.9880 | 0.9791 |
| RUIDS | **0.9998** | **0.9997** | **0.9997** | **0.9262** | **0.9544** | **0.8802** | **0.9802** | **0.9815** | **0.9801** | **0.9939** | **0.9957** | **0.9925** |

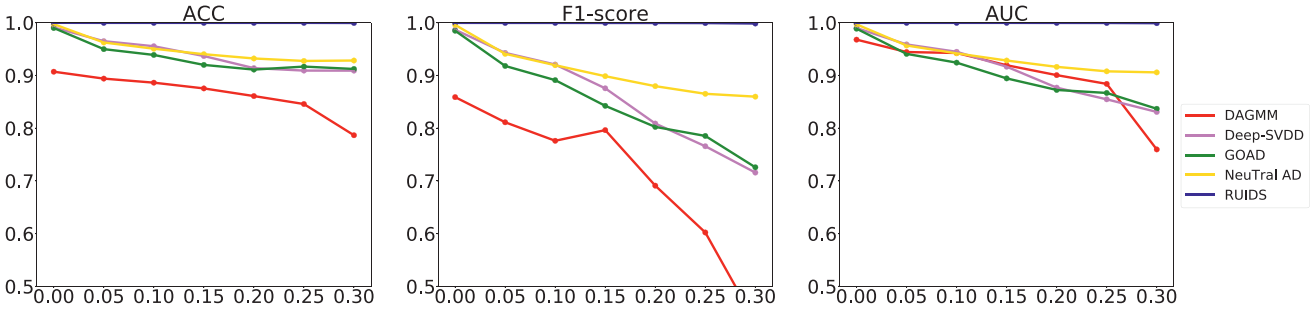



**Fig. 3.** The detection results on KDDCUP dataset with different abnormal proportion.

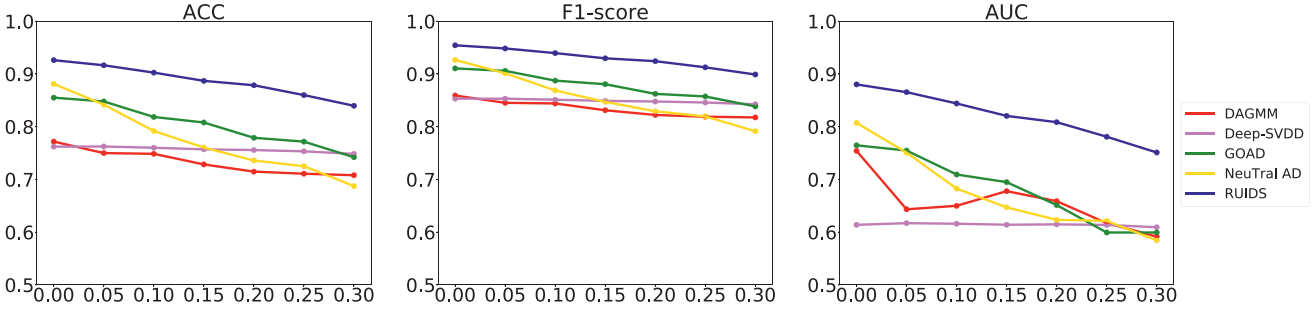



**Fig. 4.** The detection results on UNSW-NB15 dataset with different abnormal proportion.

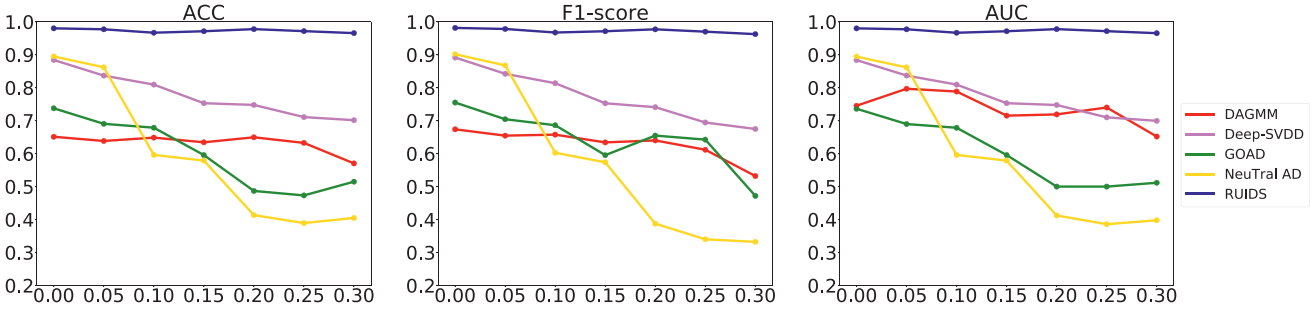



**Fig. 5.** The detection results on CICIDS-WED dataset with different abnormal proportion.

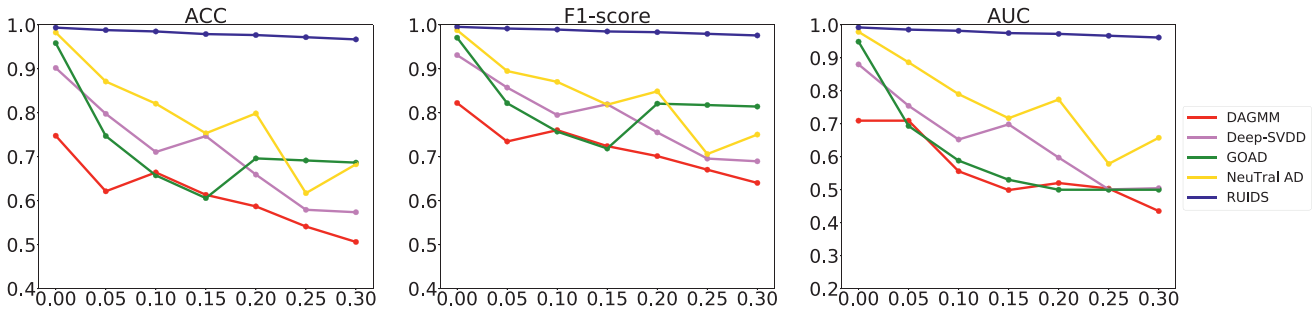



**Fig. 6.** The detection results on CICIDS-FRI dataset with different abnormal proportion.

**Table 4**
The intrusion detection results with different loss function .

| | KDD | | | UNSW-NB15 | | | CICIDS-WED | | | CICIDS-FRI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | F1 | AUC | ACC | F1 | AUC | ACC | F1 | AUC | ACC | F1 | AUC |
| RUIDS | **0.9998** | **0.9997** | **0.9997** | **0.9262** | **0.9544** | **0.8802** | **0.9802** | **0.9815** | **0.9801** | **0.9939** | **0.9957** | **0.9925** |
| Only contrastive loss | 0.9995 | 0.9984 | 0.9990 | 0.8146 | 0.8839 | 0.7121 | 0.8848 | 0.8745 | 0.8840 | 0.9601 | 0.9712 | 0.9594 |
| Only reconstruction loss | 0.8031 | 0.3840 | 0.6334 | 0.8314 | 0.8944 | 0.7382 | 0.7720 | 0.7515 | 0.7704 | 0.7652 | 0.8290 | 0.7272 |

is always poor, and the effect on the other three datasets decreases significantly with the increase of contamination objects. The effect of the GOAD algorithm on the KDDCUP dataset decreased slowest and decreased to varying degrees on other datasets. The NeuTral AD algorithm has the worst effect on the CICIDS-WED dataset. When the proportion of contamination data increases to 20%, it is difficult for the algorithm to effectively detect anomalies.

### 4.4. Ablation study

The loss function of the RUIDS algorithm consists of two parts, which are the transformer-based self-supervised contrastive loss and the masked context reconstruction loss. To verify the effectiveness of these two loss functions, we conducted a comparative experiment. In the experiment, we set the anomaly ratio of the training data set to 30% and tested the performance of the algorithm with only one of the loss functions, and with both loss functions as Table.4 shows.

Through experiments, we found that results on the four datasets proved that the performance with both loss functions considered is better than the performance with only one loss function. We can say that both loss functions have a positive impact on the performance of the algorithm. The performance of the algorithm has been significantly reduced if there is only reconstruction loss on KDD, CICIDS-WED, and CICIDS-FRI datasets. It means that contrastive loss plays a more important impact than reconstruction loss on these three datasets. On the UNSW-NB15 dataset, the reconstruction loss has a greater effect on the performance of the algorithm than the contrastive loss. Different anomaly proportions of datasets result in different sensitivity to the two loss functions. For datasets with a high proportion of abnormality, the improvement effect of reconstruction loss is more obvious. For datasets with a low proportion of abnormality, the improvement effect of contrastive loss is better than reconstruction loss.

### 4.5. Parameter sensitivity test

To enhance the robustness of the RUIDS algorithm, we mask some objects in one context. Masked contexts are reconstructed through a one-layer transformer decoder structure. We test the impact of mask scale size on the experiment results. Fig. 7 shows the effect with different mask ratios on four datasets.

The experimental results show that with the increase of the mask ratio, the performance of the algorithm decreases to different degrees. For the KDD and UNSW-NB15 datasets, the effect on the performance of the algorithm with different mask ratios is almost negligible when the ratio is less than 0.6. The performance decreases linearly with the increase of mask ratio if the mask ratio is higher than 0.6. For the CICIDS-WED and CICIDS-FRI datasets, the performance of the algorithm decreases with the increase of the mask ratio, and the downward trend is increasing. The loss function of the algorithm consists of two parts: masked context reconstruction loss and transformer-based contrastive loss. The masked context reconstruction loss is calculated by reconstructing the masked context, and the contrastive loss is calculated by the unmasked context. As the mask ratio increases, the masked context reconstruction loss has a bigger impact on the al-

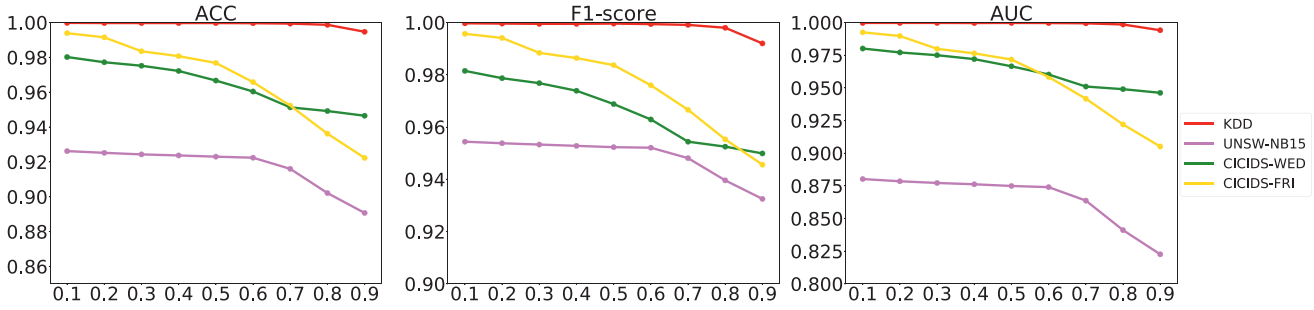
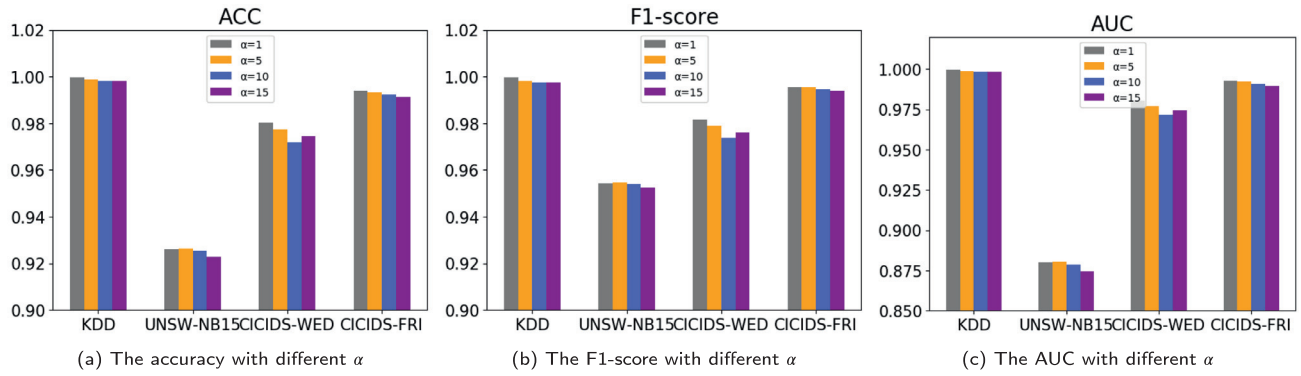
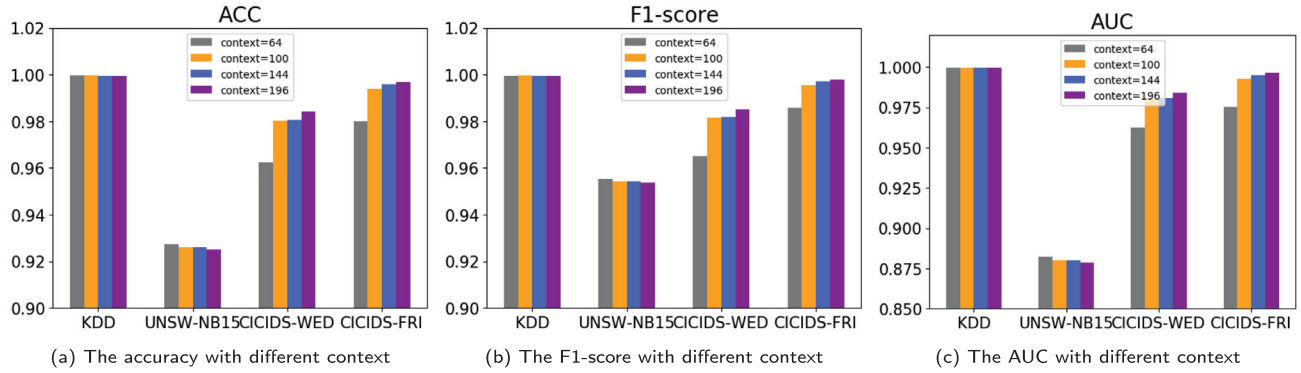



**Fig. 7.** The RUIDS detection results with different mask ratio.




(a) The accuracy with different $\alpha$ (b) The F1-score with different $\alpha$ (c) The AUC with different $\alpha$

**Fig. 8.** The RUIDS detection results with different $\alpha$.




(a) The accuracy with different context (b) The F1-score with different context (c) The AUC with different context

**Fig. 9.** The RUIDS detection results with different context sizes.

gorithm loss. This is corroborated by the contribution of these two loss functions discussed earlier.

Our algorithm consists of a self-supervised scheme and a masked context reconstruction module, each corresponding to a loss function. In the previous discussion, we tested that both loss functions have a positive effect on the final performance improvement of the algorithm. The loss is $L = L_{con} + \alpha L_{rec}$ we test the influence of performances with different $\alpha$ values on four datasets. The results are shown in Fig. 8. We can find that the value of $\alpha$ has little effect on the overall algorithm. Compared with the self-supervised learning loss, the masked reconstruction loss has less influence on the overall loss function. This also confirms the previous results on the contributions of different losses.

In our experiment, we slice the intrusion data into contexts for processing. We set the size of the context $C = 100$ in previous tests. Now we test the impact of context size on performance and the result is shown as Fig. 9. The size of the context hardly has any effect on the KDD dataset. On the CICID-WED and CICIDS-FRI data sets, the detection performance increases with the size of the context increase. The test performance is negatively related to the context size on the UNSW-NB15 data set.

## 5. Conclusion

The network intrusion detection system is very important to network security. Unsupervised intrusion detection methods that do not require labeling data solve the problems of high manual labeling costs and data contamination. In this work, we proposed a robust unsupervised intrusion detection system,i.e, RUIDS, by introducing masked context reconstruction into a transformer-based self-supervised learning scheme. The transformer-based self-supervised scheme is designed to learn the intrinsic relationship within contexts by a set of learnable transformations and a one-layer transformer encoder module. The masked context reconstruction module can learn more discriminative representations which magnify the abnormal intrusion behaviors through a transformer decoder module. By applying the RUIDS scheme to 4 datasets, the superior performance shows its effectiveness and robustness. The thorough ablation study approves that the self-supervised learning scheme and mask context reconstruction module both contribute to intrusion detection.

## Declaration of Competing Interest

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

## CRediT authorship contribution statement

**Wei Wang:** Methodology, Software, Writing – original draft. **Songlei Jian:** Conceptualization, Investigation. **Yusong Tan:** Resources, Project administration. **Qingbo Wu:** Supervision. **Chenlin Huang:** Writing – review & editing.

## Data availability

Data will be made available on request.

## References

Alom, M.Z., Taha, T.M., 2017. Network intrusion detection for cyber security using unsupervised deep learning approaches. In: 2017 IEEE National Aerospace and Electronics Conference (NAECON), pp. 63–69. doi:10.1109/NAECON.2017.8268746.

Bergman, L., Hoshen, Y., 2020. Classification-Based anomaly detection for general data. arXiv e-prints. arXiv:2005.02359

Beula Rani, B.J., Sumathi M. E, L., 2020. Survey on applying gan for anomaly detection. In: 2020 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–5. doi:10.1109/ICCCI48352.2020.9104046.
Breunig, M., Kriegel, H.-P., Ng, R., Sander, J., 2000. Lof: identifying density-based local outliers, Vol. 29, pp. 93–104. doi:10.1145/342009.335388.
Buczak, A.L., Guven, E., 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. 18 (2), 1153–1176.
Chen, Y., Abraham, A., Yang, J., 2005. Feature selection and intrusion detection using hybrid flexible neural tree, pp. 439–444. doi:10.1007/11427469_71.
Falco, F., Zoppi, T., Barbosa Vieira da Silva, C., Santos, A., Fonseca, B., Ceccarelli, A., Bondavalli, A., 2019. Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection, pp. 318–327. doi:10.1145/3297280.3297314.
Feng, J.-C., Hong, F.-T., Zheng, W.-S., 2021. MIST: Multiple instance self-Training framework for video anomaly detection. arXiv e-prints. arXiv:2104.01633
He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2021. Masked autoencoders are scalable vision learners. arXiv e-prints. arXiv:2111.06377
Ho, C., Yow, K.-C., Zhu, Z., Aravamuthan, S., 2022. Network intrusion detection via flow-to-image conversion and vision transformer classification. IEEE Access PP. doi:10.1109/ACCESS.2022.3200034. 1–1
Javaid, A., Niyaz, Q., Sun, W., Alam, M., 2016. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), pp. 21–26.
Ji, P., Zhang, T., Li, H., Salzmann, M., Reid, I., 2017. Deep subspace clustering networks.
Li, C.-L., Sohn, K., Yoon, J., Pfister, T., 2021. Cutpaste: self-Supervised learning for anomaly detection and localization. arXiv e-prints. arXiv:2104.04015
Lin, S., Clark, R., Birke, R., Schönborn, S., Trigoni, N., Roberts, S.J., 2020. Anomaly detection for time series using vae-lstm hybrid model. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 4322–4326.
Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data 6, 3:1–3:39.
Ma, Q., Zheng, J., Li, S., Cottrell, G., 2019. Learning representations for time series clustering.
Malhotra, P., Vig, L., Shroff, G., Agarwal, P., 2015. Long short term memory networks for anomaly detection in time series.
Moustafa, N., Slay, J., 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. doi:10.1109/MilCIS.2015.7348942.
Nisioti, A., Mylonas, A., Yoo, P.D., Katos, V., 2018. From intrusion detection to attacker attribution: acomprehensive survey of unsupervised methods. IEEE Commun. Surv. Tutor. 20 (4), 3369–3388. doi:10.1109/COMST.2018.2854724.
O'Reilly, C., Gluhak, A., Imran, M.A., 2016. Distributed anomaly detection using minimum volume elliptical principal component analysis. IEEE Trans. Knowl. Data Eng. 28 (9), 2320–2333. doi:10.1109/TKDE.2016.2555804.
Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., Rudolph, M., 2021. Neural transformation learning for deep anomaly detection beyond images.
Ruff, L., Vandermeulen, R., Görnitz, N., Deecke, L., Siddiqui, S., Binder, A., Müller, E., Kloft, M., 2018. Deep one-class classification.
Sadaf, K., Sultana, J., 2020. Intrusion detection based on autoencoder and isolation forest in fog computing. IEEE Access 8, 167059–167068. doi:10.1109/ACCESS.2020.3022855.
Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J., 1999. Support vector method for novelty detection, Vol. 12, pp. 582–588.
Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y., 2018. Spectralnet: spectral clustering using deep neural networks. arXiv e-prints. arXiv:1801.01587
Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP.
Sohn, K., Li, C.-L., Yoon, J., Jin, M., Pfister, T., 2020. Learning and evaluating representations for deep one-class classification. arXiv e-prints. arXiv:2011.02578
Tran, C.-P., Tran, D.-K., 2018. Anomaly detection in postfix mail log using principal component analysis. In: 2018 10th International Conference on Knowledge and Systems Engineering (KSE), pp. 107–112. doi:10.1109/KSE.2018.8573410.
Tuli, S., Casale, G., Jennings, N.R., 2022. Tranad: deep transformer networks for anomaly detection in multivariate time series data. arXiv e-prints. arXiv:2201.07284
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. arXiv e-prints. arXiv:1706.03762
Wang, H., Gu, J., Wang, S., 2017. An effective intrusion detection framework based on svm with feature augmentation. Knowl. Based Syst. 136, 130–139.
Wang, W., Jian, S., Tan, Y., Wu, Q., Huang, C., 2021. Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions. Comput. Secur. 112, 102537. doi:10.1016/j.cose.2021.102537.
Wang, Y., Qin, C., Wei, R., Xu, Y., Bai, Y., Fu, Y., 2021. SLA$^2$P: self-supervised anomaly detection with adversarial perturbation. arXiv e-prints. arXiv:2111.12896
Xie, J., Girshick, R., Farhadi, A., 2015. Unsupervised deep embedding for clustering analysis. arXiv e-prints. arXiv:1511.06335
Xu, W., Fan, Y., 2022. Intrusion detection systems based on logarithmic autoencoder and xgboost. Secur. Commun. Netw. 2022, 1–8. doi:10.1155/2022/9068724.
Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M., 2016. Towards K-means-friendly spaces: simultaneous deep learning and clustering. arXiv e-prints. arXiv:1610.04794
Yang, Y., Fu, H., Gao, S., Zhou, Y., Shi, W., 2022. Intrusion detection: a model based on the improved vision transformer. Trans. Emerg. Telecommun. Technol. 33. doi:10.1002/ett.4522.
Zhang, J., Li, C.-G., You, C., Qi, X., Zhang, H., Guo, J., Lin, Z., 2019. Self-supervised convolutional subspace clustering network. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5468–5477. doi:10.1109/CVPR.2019.00562.
Zhao, J., Lu, D., Ma, K., Zhang, Y., Zheng, Y., 2020. Deep Image Clustering with Category-Style Representation, pp. 54–70.
Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., ki Cho, D., Chen, H., 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. ICLR.

**Wei Wang** received the M.S. degree in computer science and technology from the National University of Defense Technology, in 2015, where she is currently pursuing the Ph.D. degree. Her research interests include intrusion detection and artificial intelligence in computer system.

**Songlei Jian** received the B.Sc. degree and Ph.D. degree in computer science from College of Computer, National University of Defense Technology, Changsha, China, in 2013 and 2019, respectively. She is currently an associate professor with the College of Computer, NUDT. Her research interests include representation learning, anomaly detection and artificial intelligence in computer system.

**Yusong Tan** received the Ph.D. degree in computer science and technology from the National University of Defense Technology, in 2004, where he is currently a Professor and Doctor's supervisor with the College of Computer, NUDT. His research interests include cloud computing and intrusion detection.

**Qingbo Wu** received the Ph.D. degree in computer science and technology from the National University of Defense Technology, in 2010, where he is currently a Professor and master's supervisor with the College of Computer, NUDT. His research interests include operating systems and network security

**Chenlin Huang** received his Ph.D. in computer science from the National University of Defense Technology in 2005, where he is currently a Professor and master's supervisor with the College of Computer, NUDT. His major research fields include trust management, security, operating system, and cloud computing.