applied sciences

MDPI

Article

Intrusion Detection Model Based on Improved Transformer

Yi Liu and Lanjian Wu *

School of Computers, Guangdong University of Technology, Guangzhou 510006, China
* Correspondence: 2112005018@mail2.gdut.edu.cn

Abstract: This paper proposes an enhanced Transformer-based intrusion detection model to tackle the challenges of lengthy training time, inaccurate detection of overlapping classes, and poor performance in multi-class classification of current intrusion detection models. Specifically, the proposed model includes the following: (i) A data processing strategy that initially reduces the data dimension using a stacked auto-encoder to speed up training. In addition, a novel under-sampling method based on the KNN principle is introduced, along with the Borderline-SMOTE over-sampling method, for hybrid data sampling that balances the dataset while addressing the issue of low detection accuracy in overlapping data classes. (ii) An improved position encoding method for the Transformer model that effectively learns the dependencies between features by embedding the position information of features, resulting in better classification accuracy. (iii) A two-stage learning strategy in which the model first performs rough binary prediction (determining whether it is an illegal intrusion) and then inputs the prediction value and original features together for further multi-class prediction (predicting the intrusion category), addressing the issue of low accuracy in multi-class classification. Experimental results on the official NSL-KDD test set demonstrate that the proposed model achieves an accuracy of 88.7% and an *F*1-*score* of 88.2% in binary classification and an accuracy of 84.1% and an *F*1-*score* of 83.8% in multi-class classification. Compared to existing intrusion detection models, our model exhibits higher accuracy and *F*1-*score* and trains faster than other models.

Keywords: Transformer; intrusion detection; auto-encoder; NSL-KDD; deep learning

Citation: Liu, Y.; Wu, L. Intrusion Detection Model Based on Improved Transformer. *Appl. Sci.* **2023**, *13*, 6251. https://doi.org/10.3390/app13106251

Academic Editor: Sergio Toscani

Received: 2 April 2023
Revised: 17 May 2023
Accepted: 17 May 2023
Published: 19 May 2023

1. Introduction

The rapid growth of the internet has brought significant convenience, but it has also led to an increasing number of network security problems. In today's world, security is of paramount concern as intruders have become more sophisticated with the advancement of technology [1]. Hackers employ various techniques to bypass firewalls, enabling them to infiltrate network systems and cause damage to the internal infrastructure or collect individuals' private information. Given the rising threats posed by intruders, network intrusion detection has emerged as a critical research direction in network security.

Intrusion detection systems can be divided into two categories: network-based intrusion detection systems (NIDSs) and host-based intrusion detection systems (HIDSs), depending on the type of intrusion behavior being monitored [2]. NIDSs monitor local network traffic by examining data packets to detect intrusion behavior, while HIDSs analyze multiple sources of information collected on the local host, such as system data, log files, and disk resources. Traditional intrusion detection techniques include methods such as entropy-based approaches and redundancy optimization. Entropy-based approaches are used to detect anomalies, such as DDoS attacks in IEEE802.16-based networks, by calculating the entropy of network traffic [3]. This method analyzes statistical and entropy-based features of incoming traffic to determine whether an attack has occurred. However, this approach has some limitations, such as being less effective when dealing with encrypted traffic or low traffic volume. Redundancy optimization is another commonly used technique in intrusion detection, which improves the accuracy and reliability of detection by performing the same detection algorithm multiple times. The most widely used technique is Triple

Modular Redundancy (TMR) [4], which processes input data through three detection modules and compares the output. If there is inconsistency, an attack can be detected. However, the main drawback of redundancy optimization is its high computational cost, requiring a significant amount of hardware resources and time. In recent years, machine learning techniques have gained popularity in the field of intrusion detection. By analyzing large amounts of data, these techniques can discover intrusion features and patterns, leading to improved detection accuracy and speed.

As artificial intelligence continues to advance, an increasing number of algorithms have been proposed [5–7]. Researchers are now exploring the potential of applying machine learning and deep learning to network intrusion detection classification in order to improve the detection rate of intrusion detection systems.

Traditional machine learning algorithms, including decision trees [8], random forests [9], K-nearest neighbors [10], and Bayesian [11] methods, have been widely used in intrusion detection systems with good results. However, these methods may not be suitable for handling large amounts of high-dimensional data. As network technology has developed, network traffic data and features have become increasingly complex. Intrusion detection systems based on traditional machine learning algorithms may no longer be sufficient to meet the demands of network security.

Deep learning has gained widespread use in the field of intrusion detection. Compared to traditional machine learning algorithms, deep learning algorithms are better suited to handle high-dimensional data and complex features. Intrusion detection models based on deep learning commonly use convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and generative adversarial networks (GANs) as underlying models. In reference [12], a CNN-based intrusion detection model was built, and extensive experiments were conducted to optimize its structure, resulting in improved performance. In reference [13], a combination of a CNN and RNN was used to improve the performance of intrusion detection systems. The CNN was used for convolution to capture local features, and the RNN was used to capture temporal features. This model achieved good accuracy in both binary and multi-class classification tasks. In reference [14], a BiLSTM-based intrusion detection model was proposed, which outperformed traditional LSTM and other state-of-the-art models in terms of accuracy, precision, recall, and *F*1-*score*. In reference [15], a technique for oversampling based on GANs and feature selection was proposed. This technique first modeled a complex high-dimensional attack distribution using Gradient Penalty Wasserstein GAN to generate additional attack samples. Then, a subset of features representing the entire dataset was selected using variance analysis. Finally, a rebalanced low-dimensional dataset was generated for machine learning training. In summary, deep learning algorithms have made significant breakthroughs in the field of intrusion detection and have become an important technology. These algorithms can automatically learn features from input data and are used to detect potential network intrusion behaviors.

Although intrusion detection systems have made significant progress with the development of deep learning, they still face three challenges: long training times for models, imbalanced datasets, and poor performance in multi-class classification. Therefore, this paper proposes an enhanced Transformer-based intrusion detection model that offers the following contributions:

1. This paper proposes a data processing strategy to address the challenges of high-dimensional features and class imbalance in intrusion detection datasets. Specifically, two techniques are proposed: (i) A stacked auto-encoder is used to reduce the dimensionality of the dataset by encoding the data features based on their original distribution. This not only accelerates model training but also preserves the information content of the data features. (ii) A new under-sampling method based on the K-nearest neighbors (KNN) algorithm is proposed, which under-samples normal samples and over-samples abnormal samples using Borderline-SMOTE. This hybrid

sampling approach balances the dataset while mitigating class overlap issues, thereby improving the detection performance of the model.
2. To enhance the classification performance of the model, this paper proposes an improved position encoding method for the Transformer. By incorporating positional information from the features in the intrusion detection dataset, the model can capture dependencies among the features, thereby enhancing its detection capability.
3. To enhance the model's capability to handle multiple classes, this paper proposes a two-stage learning strategy. This strategy involves an initial coarse binary prediction, followed by inputting this prediction, along with the original features, into a multi-classification model. This results in more accurate multi-classification predictions.

This chapter provides an overview of intrusion detection, including its background and current research status using machine learning and deep learning techniques. It also highlights three key challenges in intrusion detection and describes how this paper addresses these issues. The subsequent chapters are structured as follows:

1. Section 2, Related Work, presents the current research status of intrusion detection in addressing the aforementioned challenges and discusses the limitations of existing research. It also examines the use of a Transformer in intrusion detection and identifies its shortcomings. Finally, it introduces the proposed improvements in this paper to address these limitations.
2. Section 3, Materials and Methods, presents the proposed model and its various modules and improvements.
3. Section 4, Results, presents the experimental results and discusses their implications in light of the research contributions.
4. Section 5, Discussion, provides an in-depth analysis of the proposed model in this paper, including its strengths, limitations, and practical implications. It also compares the model to existing methods, identifies potential areas for future research, and emphasizes its contributions to intrusion detection.
5. Section 6, Conclusions, presents the overall conclusions of this paper.

2. Related Work

Despite significant progress in intrusion detection using machine learning and deep learning, three key challenges remain: long model training times, imbalanced datasets, and poor performance in multi-class classification. While many researchers have proposed improvements to address these challenges, the impact of these improvements has been relatively modest.

In dealing with class imbalances, intrusion detection datasets often exhibit significant class imbalance, which may cause algorithms to favor predicting the more numerous class and thus lower detection accuracy. To address this issue, researchers often perform sampling on the dataset to balance it. Jiang et al. [16] suggested a detection framework that combines deep hierarchical networks with hybrid sampling techniques. In particular, they employed the one-sided selection algorithm and SMOTE technique to perform under-sampling and over-sampling, respectively, in order to balance the dataset. Zhang et al. [17] proposed another technique for processing unbalanced datasets, which combines SMOTE over-sampling with clustering-based under-sampling using a Gaussian mixture model. Their intrusion detection framework effectively addresses the class imbalance problem and improves detection accuracy. Yan et al. [18] employed an enhanced local adaptive synthetic minority over-sampling technique to address dataset imbalance and utilized an RNN for detecting various types of traffic anomalies, leading to improved accuracy in the detection process. However, these sampling methods are designed to achieve a balanced dataset in terms of quantity, without considering the issue of class overlap in the intrusion detection dataset. The difficulty of detecting data models in the overlapping regions of classes can greatly increase, resulting in a lower detection rate for the models.

In dealing with slow model training speed, the increase in feature dimensionality and quantity in intrusion detection datasets greatly extends the training time. To address this

issue, researchers often utilize dimensionality reduction methods to speed up the model training process. Zhou et al. [19] achieved good accuracy performance by combining an auto-encoder and a residual network. They reconstructed the network using an auto-encoder to perform feature extraction, and then used the extracted features to train a designed residual network. Similarly, Liu et al. [20] employed Principal Component Analysis (PCA) to reduce dimensionality, extracting a subset of principal component features that contain maximum information. The processed data was then fed to the recurrent neural networks for classification, resulting in a high accuracy rate. However, these dimensionality reduction methods do not take into account the loss of information caused by dimensionality reduction, which in turn results in a decrease in the model's classification ability. Moreover, slow model training speed is not necessarily only due to the increase in the number and dimensions of the dataset, as deep learning models with a deeper hierarchy and a larger number of trainable parameters can also lead to slow model training.

In dealing with the low multi-classification ability of intrusion detection models, researchers usually improve the model's multi-class detection ability through optimization. To address this issue, Hassan et al. [21] proposed a hybrid intrusion detection model by combining convolutional neural networks (CNNs) and long short-term memory (LSTM) networks. The CNN is used to extract deep-level features, while the LSTM captures long-term dependencies between these features. To prevent overfitting, the weight matrix in the LSTM network is regularized using drop-connect. This hybrid model achieves high accuracy in intrusion detection classification. Guo et al. [22] proposed a method for detecting attacks without any prior knowledge by combining Sub-Space Clustering (SSC) and One-Class Support Vector Machine (OCSVM). Rehman et al. [23] proposed a combined model that employs convolutional neural networks (CNNs) and attention-based gated cyclic units for detecting both single and hybrid attacks, resulting in improved attack detection performance of the model. Yuqing et al. [24] introduced a novel method for traffic analysis by converting data traffic into pixel points in bytes. The resulting images are then processed using a CNN through operations such as convolution and pooling to obtain classification results. Their approach achieved high accuracy in both binary and multi-classification problems. However, their improvement in multi-class detection capability is not significant enough and requires further enhancement.

In recent years, the Transformer model proposed by Vaswani et al. [25] has shown superior performance in parallel training compared to other deep learning models, significantly improving the training efficiency and reducing the training time. Consequently, researchers have begun to apply Transformers to the field of intrusion detection. In [26], a Robust Transformation-based Intrusion Detection System (RTIDS) was proposed, which used position embedding to associate sequence information between features and stacked the encoder and decoder variants of the Transformer to learn low-dimensional feature representations from high-dimensional raw data. The self-attention mechanism was applied to facilitate the classification of network traffic types. However, the encoding of the input features into low-dimensional representations by the encoder becomes ineffective when the low-dimensional features are fed into the decoder, as the data is transformed back into high dimensions. In [27], an improved Visual Transformer (ViT)-based intrusion detection model was proposed, which was combined with a sliding window mechanism to enhance ViT's local feature modeling ability. A hierarchical focal loss function was adopted to improve the classification performance and mitigate the problem of imbalanced data. However, the use of focal loss alone is insufficient to solve the issue of imbalanced datasets, even with weight modifications, as it does not address the quantitative imbalance of the data. In [28], a combination of convolutional neural networks (CNNs) and a Transformer was proposed for intrusion detection, which captured both the global and local correlations between packets. However, the model did not show significant improvement in multi-class classification. In [29], Transformer-based transfer learning was used to learn network feature representations, and a hybrid CNN-Long Short-Term Memory (CNN-LSTM) model

was applied to detect different types of attacks from deep features, with the synthetic minority over-sampling technique (SMOTE) used to balance the anomalous traffic. However, SMOTE may lead to overlapping of classes, making it difficult for the model to classify the synthesized data, and the use of CNN-LSTM as the base model resulted in longer training time.

This paper proposes an improved Transformer-based intrusion detection model to tackle the challenges of imbalanced data, slow model training, and poor multi-class classification performance. The model incorporates a stacked auto-encoder to reduce dimensionality while preserving the original features. We introduce a hybrid sampling method based on the KNN principle to under-sample normal samples and the Borderline-SMOTE technique to over-sample abnormal samples. This method balances the dataset and addresses the issue of class overlap in intrusion detection datasets. We also propose an enhanced Transformer position encoding method that embeds positional information, improving classification accuracy by allowing the model to learn feature dependencies. As the Transformer has a parallel training approach, it speeds up the training process. Furthermore, we propose a two-stage learning strategy that involves rough binary classification prediction followed by multi-class detection with the original feature and prediction values. Our experimental results show that our proposed model effectively addresses these challenges and achieves faster model training speed and good detection accuracy.

3. Materials and Methods

3.1. Model Construction Materials

Figure 1 illustrates the structure of our enhanced intrusion detection model based on a Transformer, which can be divided into three parts as follows:

The first part is the data processing strategy, which involves the numerical and normalization transformation of input data, dimension reduction of the dataset through the encoding layer of the stacked auto-encoder (SAE), under-sampling of normal samples using KNN, and the use of a hybrid sampling method consisting of Borderline-SMOTE for over-sampling of abnormal samples to obtain a balanced dataset, while mitigating the problem of data class overlap.

The second part is the first stage of learning, where the balanced dataset is embedded with positional information using an improved position encoding method, features are extracted using the Transformer encoder, and the binary classification model is learned and trained using softmax function.

The third part is the second stage of learning, where the binary classification model of the first stage is used to make binary predictions on the balanced dataset, the predicted values are merged with the original features to form a new dataset, which is then embedded with positional information using the improved position encoding method, features are extracted using the Transformer encoder, and the multi-class model is learned and trained using softmax.

Compared to other intrusion detection models, our model has the following advantages:

1. Given the reconfigurable nature of SAE for feature handling, we propose to utilize its encoding layer for reducing the dimensionality of the input features. This ensures that the same amount of information content is preserved in the features despite the reduction in dimensionality, thereby expediting the model training process and reducing computational overheads.
2. This paper proposes a novel under-sampling method based on the characteristics of KNN, combined with the Borderline-SMOTE over-sampling method to achieve hybrid sampling. This method not only balances the dataset but also alleviates the problem of class overlap, thereby improving the classification performance of the model.
3. This paper proposes an improved method for position encoding in the Transformer model, which enhances the model's ability to capture dependencies between fea-

tures by incorporating their positional information, thereby improving the model's classification and detection capabilities.

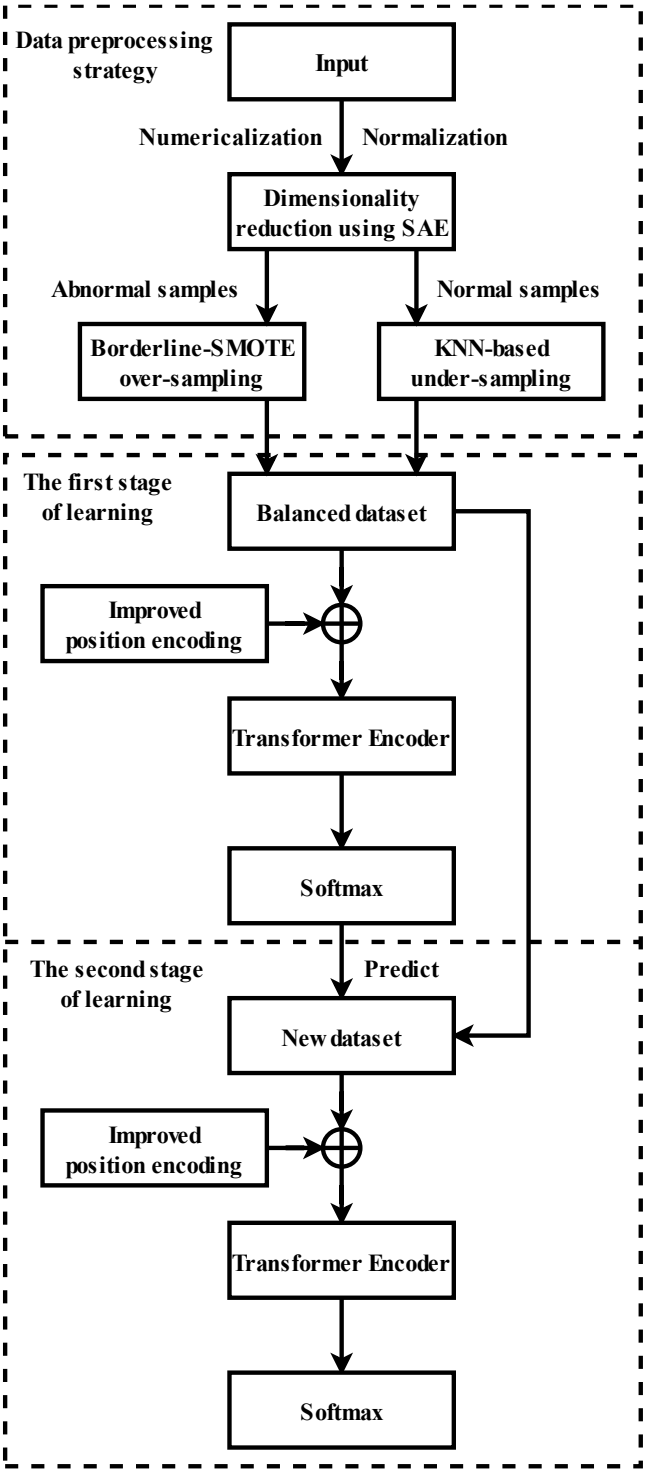

Figure 1. Improved Transformer-based intrusion detection model.

3.2. Data Preprocessing Strategy

This subsection provides a detailed explanation of the data processing approach proposed in this paper. To address the issue of high dimensionality in the dataset, which often results in long sampling and training times, we propose the utilization of stacked autoencoders for data feature dimensionality reduction. We also introduce a hybrid sampling

approach that involves under-sampling the normal samples and over-sampling the abnormal samples in the training set. In addition, we present a novel under-sampling method that leverages the properties of KNN in conjunction with the Borderline-SMOTE over-sampling method for hybrid sampling. This approach effectively resolves the challenges of class overlap and class imbalance in the dataset.

3.2.1. Numericalization and Normalization

Intrusion detection datasets often contain character-based features, which cannot be processed by computers as they only recognize numerical data. Therefore, character-based features are encoded into numeric values by using one-hot encoding. However, the resulting dataset may contain discrete and continuous states, leading to significant differences in individual feature values. This, in turn, can cause the gradient to disperse during backpropagation, slowing down the learning process and reducing the model's ability to extract deeper features. To address this issue, we propose normalizing the dataset after one-hot encoding. In this paper, we utilize minimize–maximize normalization to rescale the data and map it to a range of [0, 1]. The calculation formula is shown as follows:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

In the formula, x represents a specific data value, $x_{\min}$ is the minimum value of the column feature, $x_{\max}$ is the maximum value of the column feature, and x^* is the resulting normalized data value.

3.2.2. Dimensionality Reduction

After numericalization and normalization, the dataset's increased dimensionality results in longer training times and sparse data, leading to a decrease in model accuracy. Therefore, dimensionality reduction becomes necessary. This paper proposes to use the encoding layer of the stacked auto-encoder for dimensionality reduction, based on the reconfigurability of the original distribution of data features by the auto-encoder. This ensures that the amount of information contained in the data features remains unchanged after reduction.

The stacked auto-encoder (SAE) is an unsupervised learning model that utilizes multiple layers of pre-trained auto-encoders. The training process involves a layer-by-layer greedy training strategy, where one layer is trained at a time, and training only starts for the next layer after the current layer is trained. This approach initializes each layer with a reasonable value, leading to faster convergence and better accuracy.

The principle of SAE is to utilize the input data X as a reference to guide the neural network to learn a mapping relationship that can reconstruct the data X^R, where X^R is an approximation of X. Hence, the feature h_2 resulting from the dimensionality reduction of the encoding layer of SAE needs to retain all the information contained in the original feature X, enabling the reconstructed X^R to approximate X. Building upon this, this paper proposes to apply the encoding layer of SAE for dimensionality reduction of the input features, as illustrated in Figure 2.

The encoding layer consists of f_1 and f_2, which map the input X to h_2, while the decoding layer consists of g_1 and g_2, which reconstruct h_2 to X^R. The calculation formulas are shown as follows:

$$h_2 = f_2(f_1(X)) \tag{2}$$

$$X^R = g_1(g_2(h_2)) \tag{3}$$

$$X^R \approx X \tag{4}$$

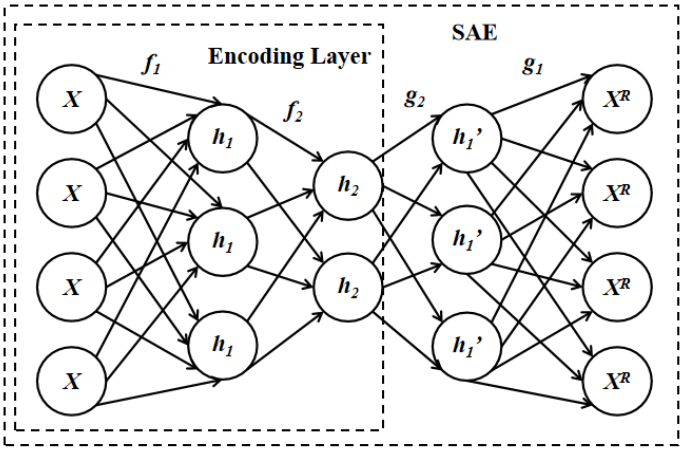

Figure 2. Stacked auto-encoder.

This paper proposes to utilize the encoding layer of the stacked auto-encoder (SAE) for dimensionality reduction, which offers several benefits. Firstly, it provides control over the dimensionality reduction features, as each layer is initialized with a reasonable value after separate training. Secondly, intrusion detection classification tasks typically involve a large number of neurons in the neural network and more trainable parameters, and using the encoding layer of SAE for dimensionality reduction can simplify the complexity of the problem and facilitate the task. Additionally, the reconfigurability of SAE for data features ensures that the amount of information contained in the data features remains unchanged after dimensionality reduction.

3.2.3. Hybrid Sampling

Current class imbalance problems are generally solved through under-sampling or over-sampling methods, but existing sampling methods aim to obtain a balanced dataset in terms of quantity, without considering class overlap. Intrusion detection datasets exhibit a clear class overlap problem, which can be defined as the intersection between normal and abnormal samples. In the overlapping region, even if the samples belong to different categories, their feature attributes are similar due to the similarity between feature attributes. Due to the similarity between feature attributes, the model finds it difficult to classify samples in the overlapping region, leading to a decrease in classification accuracy.

In existing under-sampling methods, random under-sampling deletes random samples, which leads to the loss of useful information even if the dataset is balanced in terms of quantity. In existing over-sampling methods, random over-sampling randomly selects samples for over-sampling, and SMOTE over-sampling randomly selects minority class samples for over-sampling. These over-sampling methods have a common feature, which is to randomly select samples for over-sampling without considering the class information of neighboring samples. This can lead to overlap between over-sampled samples and samples from different classes, making it difficult for the model to classify them. Based on the characteristics of KNN, this paper proposes a new under-sampling method. Then, this paper uses this method to under-sample normal samples and over-sample abnormal samples using Borderline-SMOTE, effectively solving the problems of class imbalance and class overlap.

The under-sampling method proposed in this paper is based on the KNN principle, which can identify and remove normal samples that overlap with abnormal samples. The KNN principle determines the class of a test sample based on the classes of its K-nearest neighbors. Specifically, this method checks if a normal sample is a class-overlapping sample by evaluating the number of normal and abnormal samples among its K-nearest neighbors. If the number of normal samples is less than that of abnormal samples, the sample is considered a class-overlapping sample and is removed. Algorithm 1 presents the under-sampling method based on KNN.

Algorithm 1. KNN-based under-sampling algorithm

Input:
All normal samples in the training set
Output:
Normal samples after under-sampling in the training set
Procedure:
(1) Include all normal samples in the training set as the samples to be tested.
(2) Calculate the distance between each sample to be tested and all other samples.
(3) Rank the distances and select the five nearest samples.
(4) Count the number of samples in the same normal class as the sample to be tested (denoted as 's').
(5) If 's' is less than 3, it suggests that there are more than half of abnormal samples surrounding the sample. In such cases, the sample is classified as class-overlapping and removed.

The under-sampling method used in this paper is Borderline-SMOTE. Borderline-SMOTE is an improvement upon the SMOTE algorithm. The SMOTE over-sampling principle involves obtaining the K-nearest neighbors of each minority class sample x by calculating the distance between it and other minority class samples and then creating a new sample point X_{new} by randomly selecting a sample x_i from its K-nearest neighbors. The calculation formula is shown as follows:

$$X_{new} = x + rand(0,1) \times (x_i - x) \tag{5}$$

SMOTE adopts a global approach by randomly synthesizing new samples for all instances of the minority class, regardless of their distribution or proximity to other samples. However, this method can generate redundant instances and lose useful information, leading to reduced classification accuracy. This is because SMOTE does not consider the class distribution of the nearest neighbors when producing synthetic instances. As a result, the generated synthetic samples may overlap, resulting in suboptimal classification results. To address this issue, we propose using Borderline-SMOTE for oversampling. Borderline-SMOTE is a method for generating synthetic samples in the minority class located near the decision boundary between the minority and majority classes. It utilizes a targeted oversampling approach, where synthetic samples are only created for the "borderline" minority class samples.

Unlike other over-sampling methods that randomly over-sample minority class samples, Borderline-SMOTE first divides the minority class samples into three categories: Safe, Danger, and Noise, as shown in Figure 3.

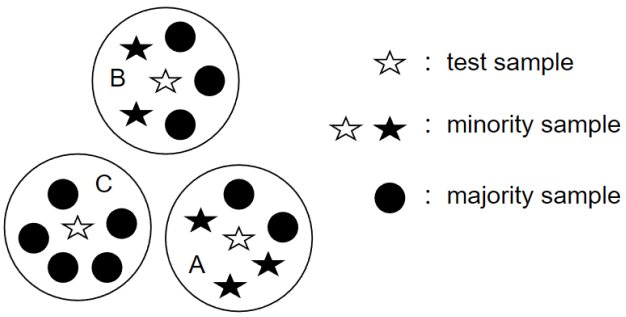

Figure 3. Three types of samples.

Here, A represents the Safe sample, B represents the Danger sample, and C represents the Noise sample. The steps to determine the type are shown in Algorithm 2. Finally, only the samples labeled as Danger were over-sampled.

Algorithm 2. Borderline-SMOTE Categorization

Input:
All abnormal samples in the training set
Output:
Three types of samples
Procedure:
(1) All minority samples are labeled as samples to be tested.
(2) Calculate the distance between each sample to be tested and all other samples.
(3) Rank the distances and select the five nearest samples.
(4) Samples for which more than half of their K-nearest neighbors are also minority samples are labeled as Safe, as shown as A.
(5) Samples for which more than half of their K-nearest neighbors are from the majority class are labeled as Danger, as shown as B.
(6) Samples for which all their K-nearest neighbors are majority samples are labeled as Noise, as shown as C.

After dividing minority class samples into three categories, the Borderline-SMOTE method can easily learn and classify Safe class samples because their neighbors are mostly from the same class. Noise class samples, on the other hand, have neighbors exclusively from the majority class and can be considered as outliers, over-sampling these samples may lead to an increase in noise and negatively affect the model performance. Danger class samples mostly have neighbors from the majority class, causing class overlap and making it difficult for the model to learn. Therefore, Borderline-SMOTE only over-samples Danger class samples to increase the number of minority samples in the overlapping region and improve the model's ability to distinguish minority samples in this area.

In intrusion detection datasets, normal samples are often more abundant than abnormal ones, which belong to the minority class. To address the issues of class imbalance and class overlap, a novel under-sampling method based on the characteristics of KNN is proposed in this paper. The method under-samples normal samples while using Borderline-SMOTE to over-sample abnormal samples. This hybrid sampling approach balances the dataset by reducing the number of normal samples and increasing the number of abnormal samples. Moreover, because the KNN under-sampling method only removes samples that belong to overlapping classes, and Borderline-SMOTE only over-samples Danger class samples, the combination of these two methods can effectively address class overlap issues.

3.3. Improved Transformer

A Transformer was initially applied to natural language processing tasks, abandoning the traditional recurrent neural network (RNN) and convolutional neural network (CNN) structures and solely relying on the attention mechanism to perform machine translation tasks, achieving excellent results. Compared to RNN-based sequential neural networks, the Transformer is superior. RNN training is iterative and sequential, resulting in particularly lengthy training times. In contrast, Transformer training is parallel, allowing all features to be trained simultaneously, dramatically increasing computational efficiency and reducing model training time. Therefore, in this paper, a Transformer was used as the base model to learn and extract features, thereby accelerating the model training speed.

The Transformer is composed of an encoder and a decoder, but for network intrusion detection tasks that do not require decoding, unlike sequence-to-sequence tasks such as machine translation, only the encoder is needed to learn and extract features, which can be combined with softmax for classification. The Transformer classification structure is shown in Figure 4. The Transformer encoder consists of two sub-layers: multi-head attention mechanism and feed-forward neural network, with residual modules and normalization modules in each sub-layer. The multi-head attention mechanism allows the model to focus on different aspects of information, producing multiple subspaces that attend to different aspects of information, thus enhancing the model's performance.

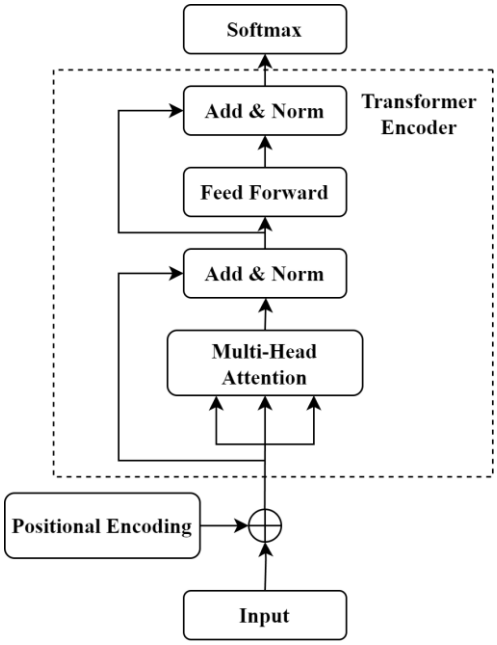

Figure 4. Transformer classification structure.

Improved Location Encoding

The Transformer model first encodes the input with a position encoding. The calculation formulas are shown as follows:

$$PE_{(pos,2i)} = \sin(pos/10,000^{2i/d_x}) \tag{6}$$

$$PE_{(pos,2i+1)} = \cos(pos/10,000^{2i/d_x}) \tag{7}$$

In the equation, the variable *pos* represents the position of a word within a text sequence sample, which is a value ranging from 0 to the maximum sequence length. d_x denotes the dimension of the text encoding, while i is the index of a word within the encoding vector, ranging from 0 to d_x. The location embedding function has a period that varies from 2π to $10{,}000 \times 2\pi$, and each location in the encoding dimension is assigned a different combination of values of the sine and cosine functions with varying periods. This method generates distinct texture location data, which allows the model to capture the relationships between positions and the temporal features of natural language. By incorporating such information into the model, it can better capture the semantic meaning and structure of the text data, ultimately enhancing its performance in various natural language processing tasks.

In the context of natural language processing, machine translation involves encoding text-based input features. However, in intrusion detection tasks, the input features are typically numeric and do not require encoding. Figure 5 illustrates the fundamental difference between these two tasks. Encoding numeric features may alter the inherent information in the data and affect the amount of information that can be extracted by the model. Consequently, the model's ability to learn effectively from the features can be diminished, thereby reducing the classification performance.

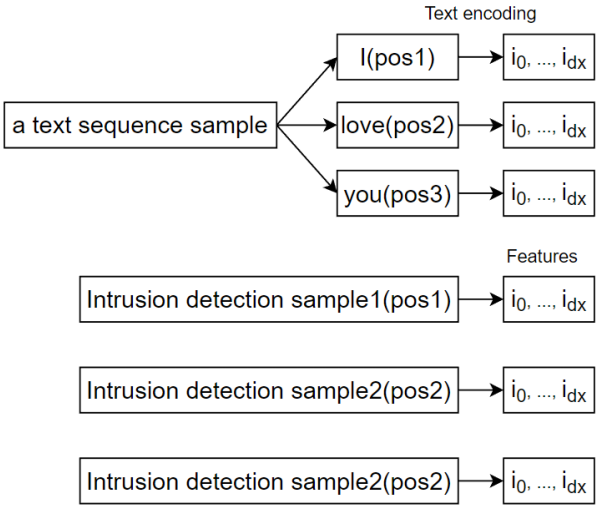

Figure 5. Location encoding for different tasks.

It is crucial to carefully consider whether positional encoding is necessary for a particular type of data before applying it in a machine learning model. When applying positional encoding to text samples, encoding the text is a necessary step. In the positional encoding formula, the variable *pos* represents the index of each word in the text sequence, while *i* represents the index of each element in the encoded vector of the word. The positional encoding method embeds the relationship between the feature vectors of each word in the text sequence, which represents the relationship between each word. However, for intrusion detection samples, text encoding is not required, as intrusion detection samples mostly consist of numerical features. If the position is encoded according to the text samples, the variable *pos* in the positional encoding formula would represent the index of each sample, while *i* would represent the feature index of the sample. In this case, the positional encoding method would embed the relationship between each sample. However, in intrusion detection, the samples are independent of each other, and the embedded information would be irrelevant and invalid.

Although the samples in intrusion detection datasets are independent from each other, there is a correlation among their features. By embedding position codes that represent the position information of the features, the model can learn the dependency between feature positions and improve the learning performance. In other words, the model can capture the relationships between features in different positions, which can enhance its ability to learn and generalize.

After analyzing the above, this paper proposes an improved position encoding method for the Transformer of setting *pos* to 1 while letting *i* represent the positional index of each feature within a segment of samples. In the absence of *pos*, only *i* remains, and the encoding formula generated by the combination of values of sine and cosine functions with different periods will represent the positional information associated with *i*. As *i* represents the positional index of the feature, the proposed positional encoding embedding will capture the positional information of each feature, allowing the model to effectively learn the dependencies between feature positions. Consequently, the improved positional encoding formulas are shown as follows:

$$PE_{(pos,2i)} = \sin(1/10,000^{2i/d_x}) \tag{8}$$

$$PE_{(pos,2i+1)} = \cos(1/10,000^{2i/d_x}) \tag{9}$$

In the equation, d_x represents the feature dimension of the sample and i represents the index of a feature within a segment of the sample, with i ranging from 0 to d_x.

The proposed position encoding method in this paper is more suitable for intrusion detection tasks than the previous approach. While the samples in intrusion detection

datasets are independent, their features are often correlated. By enhancing the position encoding to include feature position information, the Transformer model can capture the positional dependencies between sample features, which increases the information available to the model and enhances its classification performance. The proposed method is therefore more appropriate for intrusion detection tasks, where feature correlations play an important role in identifying anomalies.

3.4. Two-Stage Learning Strategy

This paper proposes a two-stage learning strategy that performs a rough binary classification before multi-class classification. It is particularly suitable for the intrusion detection dataset, which is highly imbalanced and requires the model to effectively learn from the limited number of negative samples. The first stage of binary classification enables the model to better distinguish between normal and abnormal samples, and the predicted results of this stage are then used as additional features for the second stage of multi-classification. This approach provides the multi-classification model with more information and helps it to better classify different types of attacks.

After the data processing strategy is completed, the proposed model can be divided into two parts according to the two-stage learning strategy. The first part involves embedding position information into balanced datasets by using an improved position encoding method. The Transformer encoder is then utilized to extract features and softmax is used to train the first-stage binary classification model. In the second part, the binary classification model trained in the first stage is used to predict the balanced dataset, and the predicted values are combined with the original features to form a new dataset. The improved position encoding method is again employed to embed position information, and the Transformer encoder is utilized to extract features. Finally, softmax is used to train the second-stage multi-class classification model.

3.5. Loss Function

Focal loss [30] is a loss function that was originally proposed for object detection tasks with highly imbalanced datasets. In this paper, it is applied to handle the highly imbalanced intrusion detection dataset. The focal loss function adjusts the weights of positive and negative samples to enable the model to prioritize difficult-to-classify samples, typically those belonging to the minority class in imbalanced datasets. This helps to alleviate the issue of data imbalance and improves the model's ability to accurately classify both positive and negative samples. The application of the focal loss function enables the model to better handle the data imbalance and improve its classification ability.

4. Results

4.1. Experimental Environment and Datasets

The experimental hardware environment utilized in this paper was equipped with an Intel Core i5-10300H 64-bit processor, 16GB of RAM, and a GTX1660ti graphics card. The experimental platform employed TensorFlow 2.2.0 and Keras 2.3.1 frameworks, and Python 3.7 was utilized for coding implementation.

The NSL-KDD dataset [31] is a commonly used dataset in intrusion detection research. It is an improved version of the KDD-CUP-99 dataset, with duplicate and redundant records removed. The dataset contains both normal and anomalous network traffic and is divided into training and testing subsets. The training set consists of 125,973 samples, while the test set consists of 22,543 samples.

The distribution of normal and anomalous samples in the training set of the NSL-KDD dataset is highly imbalanced, with only a small proportion of samples being abnormal. The distribution is presented in Figure 6. To address this issue, the hybrid sampling strategy of KNN under-sampling and Borderline-SMOTE over-sampling is used to balance the dataset. After hybrid sampling, the ratio between normal and abnormal samples in the dataset was

balanced at 1:1. The distribution of normal and abnormal samples after hybrid sampling is shown in Figure 7.

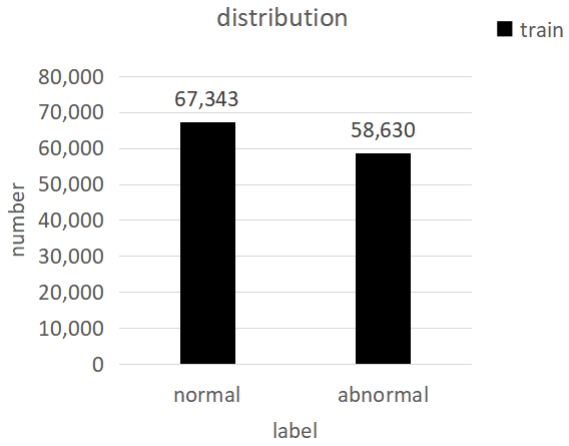

Figure 6. Distribution of data in the training set.

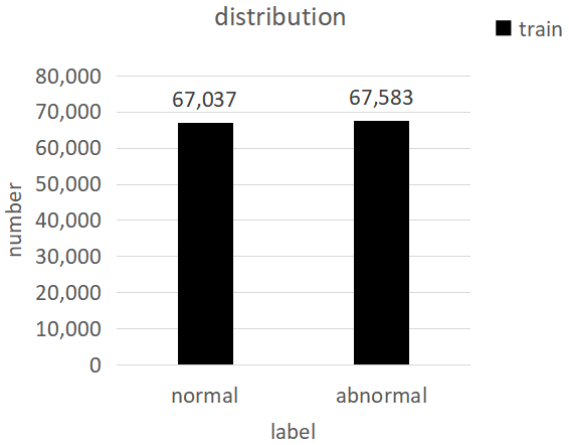

Figure 7. Distribution of data in the training set after hybrid sampling.

4.2. Assessment Indicators

There are generally four evaluation metrics for models, which are *accuracy*, *precision*, *recall*, and *F*1-*score*. Their formulas are shown below:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$precision = \frac{TP}{TP + FP} \tag{11}$$

$$recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \tag{13}$$

TP represents the number of true positives, *FP* represents the number of false positives, *TN* represents the number of true negatives, and *FN* represents the number of false negatives. However, because *precision* and *recall* often conflict with each other, accuracy and *F*1-*score* are used as the main evaluation criteria in this study. The larger the values of accuracy and *F*1-*score*, the better the performance of the model. Additionally, this study added model training time as a metric to evaluate the speed of model training.

4.3. Experimental Results and Discussion

To fully validate the effectiveness of the proposed model, several experiments were designed in this study. In Section 4.3.1, different dimensionality reduction methods were compared to verify the superiority of SAE. Section 4.3.2 conducted experiments with different dimensionality reduction levels to analyze the impact of SAE on model accuracy. Section 4.3.3 tested different sampling methods to compare the hybrid sampling method proposed in this study with other sampling methods. In Section 4.3.4, performance analysis and comparative experiments were conducted to evaluate the intrusion detection capability of the proposed model in binary and multi-class classification, and it was compared with existing models. Additionally, to test the robustness of the model, further performance testing was conducted using the UNSW-NB15 dataset. Section 4.3.5 conducted three ablation experiments to accurately evaluate the effectiveness of each module in the proposed model.

4.3.1. Comparison Experiments of Different Dimensionality Reduction Methods

To validate the effectiveness and applicability of the feature dimensionality reduction method employed in this paper, we conducted comparative experiments under the same experimental conditions with different dimensionality reduction methods: we compared the dimensionality reduction method used in this paper (SAE) with existing dimensionality reduction methods such as PCA [20] and AE [19]. Finally, we used the proposed overall model for classification and selected the highest accuracy and *F*1-*score* for each dimensionality reduction method for comparison. The comparative results are shown in Table 1.

Table 1. Comparison experiments of different dimensionality reduction methods.

Dimensionality Reduction Method	Binary Classification		Multi-Classification	
	Accuracy	***F*1-*Score***	**Accuracy**	***F*1-*Score***
PCA	0.851	0.846	0.820	0.816
AE	0.878	0.871	0.835	0.829
SAE	0.887	0.882	0.841	0.838

According to Table 1, the accuracy of the proposed model using SAE for dimensionality reduction achieved 88.7% in binary classification with an *F*1-*score* of 88.2%, which is a 3.6% and 0.9% improvement over using PCA and AE, respectively. In multi-class classification, the accuracy reached 84.1% with an *F*1-*score* of 83.8%, which is a 2.1% and 0.6% improvement over using PCA and AE, respectively. Analysis of the reasons for this shows that PCA relies more on variance when reducing data, and non-principal components with low variance may contain important information on sample differences. This leads to a reduction in the amount of information contained in the features during the dimensionality reduction process. AE trains a single-layer encoder directly during training, and excessive reduction can result in a loss of information in the reduced data features. However, SAE uses greedy layer-wise training and initializes the parameters for each layer, ensuring control over the reduced features. The layer-wise dimensionality reduction ensures that the amount of information contained in the data features remains unchanged, allowing the model to obtain the most amount of information and achieve the highest accuracy and *F*1-*score* in classification.

4.3.2. Experiments on Different Dimensionality Reduction Levels

After numerical and standardization preprocessing, the dimension of the NSL-KDD dataset was reduced to 122. To prepare the NSL-KDD dataset for training the Transformer encoder classifier, a stacked auto-encoder was employed to further reduce the dataset's dimension after numerical and standardization preprocessing. The encoding layers of the SAE were used for dimensionality reduction, and various feature subsets consisting of different numbers of features were input into the overall model for binary and multi-

class classification. The impact of varying degrees of dimensionality reduction on model accuracy was evaluated, and the results are shown in Figure 8.

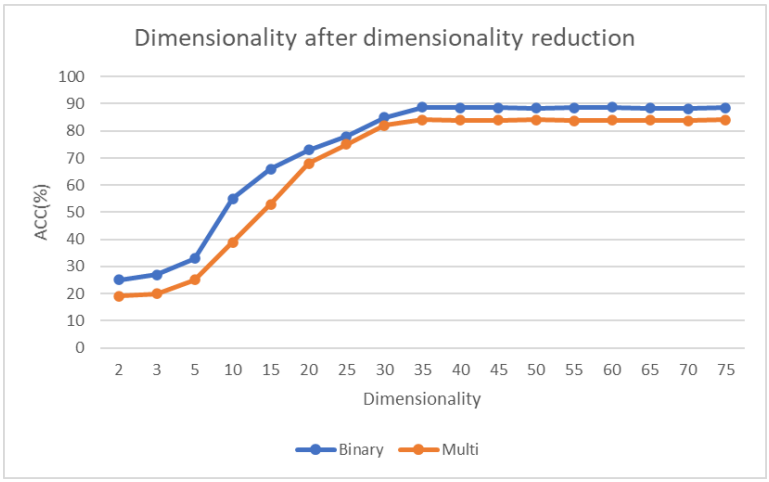

Figure 8. The impact of different degrees of dimensionality reduction on the model accuracy.

From Figure 8, it is observed that the accuracy increased rapidly in the beginning as the number of selected features increased, and it eventually stabilized. The highest accuracy was achieved when the number of features was approximately 35. Analysis of the reasons for this shows that when the feature dimension is reduced to a minimum, the amount of information contained in the features is not sufficient for the model to learn and train effectively, resulting in the lowest accuracy. As the feature dimension increases, the amount of information contained in the features also increases, leading to an improvement in model accuracy. When the feature dimension reaches 35, the accuracy tends to balance.

4.3.3. Comparison Experiments of Different Sampling Methods

To address the issues of imbalanced dataset and class overlapping, a hybrid sampling method (KNN-based under-sampling and Borderline-SMOTE over-sampling techniques) was used in this paper to handle the dataset. To verify the effectiveness of the proposed method, two comparative experiments were set up in this section using different sampling methods:

1. Under the same experimental conditions, five different single sampling methods were used to handle imbalanced datasets: random over-sampling, SMOTE, Borderline-SMOTE, random under-sampling, and KNN-based under-sampling. Finally, binary and multi-class experiments were conducted using the proposed model, and the results are shown in Table 2 to verify the effectiveness of the proposed method.
2. Under the same experimental conditions, the above five sampling methods were randomly combined using a mixed sampling approach to handle imbalanced datasets. Finally, binary and multi-class classification experiments were performed using the proposed model, and the results are shown in Table 3.

Table 2. Comparison experiments of different individual sampling methods.

Sampling Method	Binary Classification		Multi-Classification	
	Accuracy	***F*1-*Score***	**Accuracy**	***F*1-*Score***
Random over-sampling	0.819	0.813	0.809	0.805
SMOTE	0.833	0.827	0.816	0.813
Borderline-SMOTE	0.845	0.841	0.824	0.818
Random under-sampling	0.811	0.807	0.807	0.803
KNN-based under-sampling	0.830	0.825	0.819	0.814

Table 3. Comparison experiments of different hybrid sampling methods.

Hybrid Method					Binary Classification		Multi-Classification	
Under-Sampling		Over-Sampling			Accuracy	*F1-Score*	Accuracy	*F1-Score*
Random	KNN-Based	Random	SMOTE	Borderline SMOTE				
√		√			0.833	0.828	0.809	0.805
√			√		0.847	0.843	0.816	0.813
√				√	0.858	0.852	0.825	0.820
	√	√			0.849	0.844	0.827	0.824
	√		√		0.869	0.865	0.833	0.829
	√			√	0.887	0.882	0.841	0.838

According to Table 2, Borderline-SMOTE achieved an accuracy of 84.5% and an *F*1-*score* of 84.1% in binary classification. In comparison to random over-sampling and SMOTE, Borderline-SMOTE improved the accuracy by 2.6% and 1.2%, respectively. In multi-class classification, Borderline-SMOTE achieved an accuracy of 82.4% and an *F*1-*score* of 81.8%. Borderline-SMOTE improved the accuracy by 1.6% and 0.8% in comparison to random over-sampling and SMOTE, respectively. In the case of under-sampling methods, KNN-based under-sampling achieved an accuracy of 83% and an *F*1-*score* of 82.5% in binary classification. KNN-based under-sampling improved the accuracy by 2.6% compared to random under-sampling. In multi-class classification, KNN-based under-sampling achieved an accuracy of 81.9% and an *F*1-*score* of 81.4%, with an accuracy improvement of 1.2% in comparison to random under-sampling. Analysis of the reasons for this shows that in over-sampling methods, random over-sampling and SMOTE randomly sample the data points, while Borderline-SMOTE over-samples only the Danger class samples. This approach balances the dataset while also alleviating the class overlap problem, thereby enhancing the separability of the data and improving the model's performance. Similarly, in under-sampling methods, KNN-based under-sampling selectively removes overlapping samples that are prone to cause misclassification by the model. This approach balances the dataset while also alleviating the class overlap problem, thereby enhancing the separability of the data and improving the model's performance.

According to Table 3, the proposed hybrid sampling algorithm (KNN-based under-sampling and Borderline-SMOTE over-sampling techniques) outperformed other combinations of hybrid sampling algorithms with the highest accuracy and *F*1-*score* on both binary and multi-class classification tasks. Analysis indicates that the combination of these two techniques not only balances the dataset in terms of quantity but also maximally mitigates class overlap by complementing each other's effects on overlapping samples, improving the separability of the dataset and thus enhancing the model's detection ability.

4.3.4. Performance Analysis and Comparison Experiments

To validate the detection capability of the model, this study first selected recently proposed intrusion detection models, including CNN [32], CNN-LSTM [32], CBA-CLSVE [33], SSC-OCSVM [22], CNN-GRU [34], and FCNN-SE [35], which have shown good performance on the NSL-KDD dataset. Additionally, several intrusion detection models using a Transformer as the base model were selected for comparison, namely RTIDS [26], VIT [27], and CNN-Transformer [28]. To ensure the validity of the experiments, the NSL-KDD dataset was not re-partitioned into training and testing sets under the same experimental conditions, and the official training and testing sets specified by NSL-KDD were used for binary and multiclass comparison experiments. The binary classification results, compared with other models, are presented in Table 4.

Table 4. Comparison experiment of binary classification ability.

Model	Accuracy	*F*1-*Score*	Training Time (s)
CNN	0.793	0.785	22
CNN-LSTM	0.801	0.794	52
CBA-CLSVE	0.861	0.854	25
SSC-OCSVM	0.833	0.827	17
CNN-GRU	0.813	0.805	38
FCNN-SE	0.856	0.851	24
RTIDS	0.870	0.864	30
VIT	0.859	0.855	15
CNN-Transformer	0.866	0.861	25
Ours	0.887	0.882	8

In Table 4, the results of our proposed model demonstrate its superiority over other state-of-the-art models in terms of both accuracy and *F*1-*score*, achieving an impressive accuracy rate of 88.7% and an *F*1-*score* of 88.2%. These results indicate the effectiveness of our proposed model in accurately predicting the target variable and suggest its potential for practical applications. Furthermore, the proposed model also exhibited faster training speed than the other models, indicating its efficiency and scalability. Table 5 presents the results of the multi-classification experiment and compares them with other models.

Table 5. Comparison experiment of multi-classification ability.

Model	Accuracy	*F*1-*Score*	Training Time (s)
CNN	0.788	0.782	23
CNN-LSTM	0.797	0.791	55
CBA-CLSVE	0.832	0.825	28
SSC-OCSVM	0.815	0.811	19
CNN-GRU	0.800	0.795	41
FCNN-SE	0.830	0.824	27
RTIDS	0.835	0.830	35
VIT	0.828	0824	20
CNN-Transformer	0.831	0.827	27
Ours	0.841	0.838	15

In Table 5, our experimental results indicate that our proposed model outperforms other models in the multi-classification task, achieving an accuracy of 84.1% and an *F*1-*score* of 83.8%. This performance is superior to other models by 5.3%, 4.4%, 0.9%, 2.6%, 4.1%, 1.1%, 0.6%, 1.3%, and 1% in accuracy, respectively. Furthermore, our proposed model exhibits faster training speed than other models. These results suggest that our proposed model offers a significant improvement over existing models for the multi-classification task.

To further demonstrate the robustness of our model, we conducted additional tests on the UNSW-NB15 dataset and compared it with the models mentioned earlier under the same experimental conditions (by splitting the UNSW-NB15 dataset into training and testing sets with a ratio of 7:3). The results of our multi-class classification experiment on the testing set are shown in Table 6.

The experimental results show that our model has a better detection capability on the UNSW-NB15 dataset compared to other models, indicating the robustness of our proposed model. In the multi-classification task, our model achieves an accuracy of 87.5% and an *F*1-*score* of 87.3%, outperforming other models by 4.6%, 4.9%, 0.8%, 2.6%, 3.2%, 1.1%, 0.2%, 1.9%, and 0.5% in accuracy, respectively. Additionally, the training speed of our model is faster.

Table 6. Comparison experiment of multi-classification ability on UNSW-15 dataset.

Model	Accuracy	*F*1-*Score*	Training Time (s)
CNN	0.829	0.827	26
CNN-LSTM	0.826	0.806	59
CBA-CLSVE	0.867	0.862	33
SSC-OCSVM	0.849	0.846	23
CNN-GRU	0.843	0.840	45
FCNN-SE	0.864	0.861	29
RTIDS	0.873	0.869	38
VIT	0.856	0.854	24
CNN-Transformer	0.870	0.865	32
Ours	0.875	0.873	19

The three experiments conducted above demonstrate that the proposed model outperforms existing intrusion detection models and has a faster model training speed, as well as a certain level of robustness. Moreover, compared to recently proposed Transformer-based models, the proposed model in this paper has higher accuracy and a faster model training speed. These results confirm the effectiveness and efficiency of the proposed model in detecting network intrusions, indicating its potential application in real-world scenarios.

4.3.5. Ablation Experiment

In order to verify the effect of each module in the proposed model on the overall performance, we conducted three ablation experiments on the model:

1. We conducted ablation experiments to assess the impact of the proposed improved position encoding on the classification ability of the Transformer model. The results are presented in Table 7.
2. We conducted ablation studies on the binary classification model proposed in this paper, including the data processing strategy and the first stage of learning, in order to verify the effectiveness of each module in the model. The results of the experimental analysis are listed in Table 8.
3. We conducted ablation experiments on the multi-classification model. There are three parts of the model that have an impact on the classification effect, namely improved positional encoding, hybrid sampling, and a two-stage learning strategy. The effect of each module was verified by comparing the accuracy and *F*1-*score* before and after the addition of the module. The results of the experimental analysis are shown in Table 9.

Table 7. Ablation experiment of the Transformer model before and after the addition of the improved position encoding.

Model	Binary Classification		Multi-Classification	
	Accuracy	*F*1-*Score*	Accuracy	*F*1-*Score*
Transformer before improvement	0.793	0.788	0.787	0.782
Transformer after improvement	0.804	0.799	0.801	0.794

Table 8. Ablation experiment of the binary classification model.

Module				Accuracy	*F*1-*Score*
Transformer	Improved Positional Encoding	Borderline-SMOTE Over-Sampling	KNN-Based Under-Sampling		
√				0.793	0.788
√	√			0.804	0.799
√	√	√		0.845	0.841
√	√	√	√	0.887	0.882

Table 9. Ablation experiment of the multi-classification model.

Module				Accuracy	*F*1-*Score*
Transformer	**Improved Positional Encoding**	**Hybrid Sampling**	**Two-Stage Learning Strategy**		
√				0.787	0.782
√	√			0.801	0.794
√	√	√		0.830	0.825
√	√	√	√	0.841	0.838

The results presented in Table 7 demonstrate a significant improvement in the accuracy and *F*1-*score* of the Transformer model after the proposed positional encoding method was implemented, as compared to its performance before the improvement. This improvement was observed in both binary and multi-classification tasks, indicating the effectiveness of the proposed method across different types of classification problems. These results highlight the importance of properly encoding positional information in the Transformer model and suggest that the proposed method can be a valuable addition to the existing approaches for enhancing the performance of Transformer-based models.

Table 8 presents the results of our experiments examining the impact of each module on the performance of the binary classification model. As can be seen from the results, the removal of any module leads to a significant decrease in the accuracy and *F*1-*score* of the model, highlighting the critical role that each module plays in achieving high performance. Moreover, the synergy between all modules is observed when all modules are present, resulting in the highest accuracy and *F*1-*score*. These findings underscore the importance of a holistic design approach for the binary classification model, where each component is meticulously designed and optimized to achieve optimal performance. In summary, the results of our experiments indicate that the performance of the binary classification model is highly dependent on the effective integration and cooperation of all its constituent modules.

In Table 9, it is evident that the absence of any module in the total multi-classification model leads to a significant decrease in both accuracy and *F*1-*score*. However, the model exhibits the highest levels of accuracy and *F*1-*score* when all modules are present. These results emphasize the critical role of each module in the overall performance of the multi-classification model and highlight the importance of a comprehensive design approach that optimizes the interaction of all modules to achieve optimal performance.

5. Discussion

To address the three issues of slow model training, imbalanced datasets, and poor multi-class detection performance in intrusion detection models, this paper proposes an improved Transformer model. Through multiple experiments, we have demonstrated the effectiveness of the proposed model in addressing these issues, not only improving the model's training speed but also enhancing its classification detection ability. Despite the notable improvements in detection accuracy attained by the proposed model, there remain certain limitations that warrant attention. Specifically, the multi-class detection capability of the model, although enhanced, still requires further development. Future research endeavors may include the exploration of alternative strategies to enhance the binary classification performance of the proposed model. This may involve investigating diverse model architectures, such as incorporating attention mechanisms or exploring novel loss functions that are better equipped to capture the unique characteristics of the dataset. Additionally, efforts could be made to improve the interpretability of the model by analyzing the attention weights of the Transformer model and identifying significant features for intrusion detection. These approaches may culminate in further improvements in the overall multi-classification detection performance and can have far-reaching implications beyond the scope of intrusion detection. Overall, this study contributes to the knowledge system by proposing a new approach and demonstrating its effectiveness in

addressing the three issues in intrusion detection models, while also emphasizing the need for further research and improvement in this area.

6. Conclusions

This paper proposes an improved Transformer-based intrusion detection model. Specifically, this paper first proposes a data processing strategy that uses SAE to reduce the dimensionality of the dataset while ensuring the same amount of information, speeding up model training. Additionally, a KNN-based under-sampling method is proposed, combined with the Borderline-SMOTE over-sampling method to perform mixed sampling on the dataset, balancing the dataset and greatly alleviating the class overlap problem. Then, an improved position encoding method for Transformer is proposed, learning the position dependencies between features by embedding the positional relationships of features, thereby improving the model's detection ability. Finally, a two-stage learning strategy is proposed to enhance the model's multi-classification ability.

Through comparative experiments on different dimensionality reduction methods, degrees of dimensionality reduction, sampling methods, model abilities, and model ablation, this paper has demonstrated the strong detection ability and faster model training speed of the proposed model in intrusion detection models, providing promising prospects for real-time applications of intrusion detection systems.

However, the proposed model in this paper still has some shortcomings while improving the detection accuracy: Firstly, the model's ability for multi-class detection has a large room for improvement, which requires a focus on developing more powerful and scalable algorithms for processing. Secondly, it is necessary to explore the impact of incorporating more different data sources into our analysis. Thirdly, although the model has improved the detection accuracy of a small number of samples, the improvement effect is not significant. In future research, we will further study the model algorithm to further improve the accuracy of detection of a small number of samples, enhance the overall classification effect of the model, and improve the robustness of the model.

Author Contributions: Methodology, L.W. and Y.L.; validation, L.W. and Y.L.; writing—original draft preparation, L.W.; writing—review and editing, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The NSL-KDD dataset used in this research is an open access dataset and can be found at https://www.unb.ca/cic/datasets/nsl.html (accessed on 27 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, H.W.; Han, B.A.; Su, J.S.; Wang, X.Y. A High-Performance Intrusion Detection Method Based on Combining Supervised and Unsupervised Learning. In Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 7–11 November 2018; pp. 1803–1810.
2. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 686–728. [CrossRef]
3. Shojaei, M.; Movahhedinia, N.; Tork Ladani, B. An entropy based approach for DDoS attack detection in IEEE 802.16 based networks. In Proceedings of the Advances in Information and Computer Security: 6th International Workshop, IWSEC 2011, Tokyo, Japan, 8–10 November 2011; pp. 129–143.
4. Babić, I.; Miljković, A.; Čabarkapa, M.; Nikolić, V.; Đorđević, A.; Ranđelović, M.; Ranđelović, D. Triple modular redundancy optimization for threshold determination in intrusion detection systems. *Symmetry* **2021**, *13*, 557. [CrossRef]

5. Huang, T.; Zhao, R.; Bi, L.; Zhang, D.; Lu, C. Neural embedding singular value decomposition for collaborative filtering. *IEEE Trans. Neural. Netw. Learn. Syst.* **2021**, *33*, 6021–6029. [CrossRef] [PubMed]
6. Zheng, J.; Lu, C.; Hao, C.; Chen, D.; Guo, D. Improving the generalization ability of deep neural networks for cross-domain visual recognition. *IEEE Trans. Cogn. Develop. Syst.* **2020**, *13*, 607–620. [CrossRef]
7. Arora, V.; Verma, K.; Leekha, R.S.; Lee, K.; Choi, C.; Gupta, T.; Bhatia, K. Transfer learning model to indicate heart health status using phonocardiogram. *Comput. Mater. Contin.* **2021**, *69*, 4151–4168. [CrossRef]
8. Ingre, B.; Yadav, A.; Soni, A.K. Decision tree based intrusion detection system for NSL-KDD dataset. In *Information and Communication Technology for Intelligent Systems (ICTIS 2017)*; Springer International Publishing: Cham, Switzerland, 2018; Volume 2, pp. 207–218.
9. Zhang, J.; Zulkernine, M.; Haque, A. Random-forests-based network intrusion detection systems. *IEEE Trans. Syst. Man Cybern. Syst. Part C Appl. Rev.* **2008**, *38*, 649–659. [CrossRef]
10. Liao, Y.; Vemuri, V.R. Use of k-nearest neighbor classifier for intrusion detection. *Comput. Secur.* **2002**, *21*, 439–448. [CrossRef]
11. Mahmood, H.A.; Hashem, S.H. Network intrusion detection system (NIDS) in cloud environment based on hidden Naïve Bayes multiclass classifier. *Al-Mustansiriyah J. Sci.* **2018**, *28*, 134–142. [CrossRef]
12. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. CNN-based network intrusion detection against denial-of-service attacks. *Electronics* **2020**, *9*, 916. [CrossRef]
13. Khan, M.A. HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes* **2021**, *9*, 834. [CrossRef]
14. Imrana, Y.; Xiang, Y.; Ali, L.; Abdul-Rauf, Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst. Appl.* **2021**, *185*, 115524. [CrossRef]
15. Liu, X.; Li, T.; Zhang, R.; Wu, D.; Liu, Y.; Yang, Z. A GAN and feature selection-based oversampling technique for intrusion detection. *Secur. Commun. Netw.* **2021**, *2021*, 9947059. [CrossRef]
16. Jiang, K.Y.; Wang, W.Y.; Wang, A.L.; Wu, H.B. Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network. *IEEE Access* **2020**, *8*, 32464–32476. [CrossRef]
17. Zhang, H.P.; Huang, L.L.; Wu, C.Q.; Li, Z.B. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [CrossRef]
18. Yan, B.H.; Han, G.D. Combinatorial Intrusion Detection Model Based on Deep Recurrent Neural Network and Improved SMOTE Algorithm. *Chin. J. Netw. Inf. Secur.* **2018**, *4*, 48–59.
19. Zhou, P.; Zhou, Z.; Wang, L.; Zhao, W. Network intrusion detection method based on autoencoder and RESNET. *Comput. Appl. Res.* **2020**, *37*, 224–226.
20. Liu, J.; Sun, X.; Jin, J. Intrusion detection model based on principal component analysis and cyclic neural network. *Chin. J. Inf. Technol.* **2020**, *34*, 105–112.
21. Hassan, M.M.; Gumaei, A.; Alsanad, A.; Alrubaian, M.; Fortino, G. A hybrid deep learning model for efficient intrusion detection in big data environment. *Inform. Sci.* **2020**, *513*, 386–396. [CrossRef]
22. Pu, G.; Wang, L.; Shen, J.; Dong, F. A hybrid unsupervised clustering-based anomaly detection method. *Tsinghua Sci. Technol.* **2021**, *26*, 146–153. [CrossRef]
23. Javed, A.R.; Rehman, S.U.; Khan, M.U.; Alazab, M.; Reddy, G.T. CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network Using CNN and Attention-Based GRU. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1456–1466. [CrossRef]
24. Yuqing, Z.; Ying, D.; Caiyun, L.; Kenan, L.; Hongyu, S. Situation, trends and prospects of deep learning applied to cyberspace security. *J. Comput. Res. Dev.* **2018**, *55*, 1117–1142.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
26. Wu, Z.; Zhang, H.; Wang, P.; Sun, Z. RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access* **2022**, *10*, 64375–64387. [CrossRef]
27. Yang, Y.G.; Fu, H.M.; Gao, S.; Zhou, Y.H.; Shi, W.M. Intrusion detection: A model based on the improved vision transformer. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4522. [CrossRef]
28. Zhang, Z.; Wang, L. An Efficient Intrusion Detection Model Based on Convolutional Neural Network and Transformer. In Proceedings of the 2021 Ninth International Conference on Advanced Cloud and Big Data (CBD), Xi'an, China, 26–27 March 2022; pp. 248–254.
29. Ullah, F.; Ullah, S.; Srivastava, G.; Lin, J.C.-W. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. *Digit. Commun. Netw.* 2023, *in press*. [CrossRef]
30. Zhang, F.; Chan, P.P.K.; Biggio, B.; Yeung, D.S.; Roli, F. Adversarial Feature Selection Against Evasion Attacks. *IEEE Trans. Cybern.* **2016**, *46*, 766–777. [CrossRef]
31. Archibe, U.K. NSL Data. 2006. Available online: http://nsl.cs.unb.ca/NSL-KDD (accessed on 27 March 2023).
32. Zainel, H.; Koçak, C. LAN Intrusion Detection Using Convolutional Neural Networks. *Appl. Sci.* **2022**, *12*, 6645. [CrossRef]
33. Shen, Y.; Zheng, K.; Yang, Y.; Liu, S.; Huang, M. CBA-CLSVE: A Class-Level Soft-Voting Ensemble Based on the Chaos Bat Algorithm for Intrusion Detection. *Appl. Sci.* **2022**, *12*, 11298. [CrossRef]

34. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network Intrusion Detection Model Based on CNN and GRU. *Appl. Sci.* **2022**, *12*, 4184. [CrossRef]
35. Chen, C.; Song, Y.; Yue, S.; Xu, X.; Zhou, L.; Lv, Q.; Yang, L. FCNN-SE: An Intrusion Detection Model Based on a Fusion CNN and Stacked Ensemble. *Appl. Sci.* **2022**, *12*, 8601. [CrossRef]