electronics

Article

A Network Intrusion Detection Model Based on BiLSTM with Multi-Head Attention Mechanism

Jingqi Zhang [1], Xin Zhang [1], Zhaojun Liu [1], Fa Fu [1,*], Yihan Jiao [1] and Fei Xu [2]

[1] College of Computer Science and Technology, Hainan University, Haikou 570228, China; 20203102578@hainanu.edu.cn (J.Z.); 20203101194@hainanu.edu.cn (X.Z.); 20203101299@hainanu.edu.cn (Z.L.); 20203101191@hainanu.edu.cn (Y.J.)
[2] College of Civil and Architecture Engineering, Hainan University, Haikou 570228, China; 20203102564@hainanu.edu.cn
* Correspondence: fufa@hainanu.edu.cn

Abstract: A network intrusion detection tool can identify and detect potential malicious activities or attacks by monitoring network traffic and system logs. The data within intrusion detection networks possesses characteristics that include a high degree of feature dimension and an unbalanced distribution across categories. Currently, the actual detection accuracy of some detection models is relatively low. To solve these problems, we propose a network intrusion detection model based on multi-head attention and BiLSTM (Bidirectional Long Short-Term Memory), which can introduce different attention weights for each vector in the feature vector that strengthen the relationship between some vectors and the detection attack type. The model also utilizes the advantage that BiLSTM can capture long-distance dependency relationships to obtain a higher detection accuracy. This model combined the advantages of the two models, adding a dropout layer between the two models to improve the detection accuracy while preventing training overfitting. Through experimental analysis, the network intrusion detection model that utilizes multi-head attention and BilSTM achieved an accuracy of 98.29%, 95.19%, and 99.08% on the KDDCUP99, NSLKDD, and CICIDS2017 datasets, respectively.

Keywords: intrusion detection; deep learning; multi-head attention; BiLSTM

Citation: Zhang, J.; Zhang, X.; Liu, Z.; Fu, F.; Jiao, Y.; Xu, F. A Network Intrusion Detection Model Based on BiLSTM with Multi-Head Attention Mechanism. *Electronics* **2023**, *12*, 4170. https://doi.org/10.3390/electronics12194170

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 31 August 2023
Revised: 29 September 2023
Accepted: 6 October 2023
Published: 8 October 2023

1. Introduction

In recent years, network intrusion has gradually expanded, resulting in the theft of personal privacy and becoming the main attack platform [1]. Intrusion detection, as one of the most important network security protection tools after firewalls, plays a more important role in network security defense systems [2]. It can be defined as "network security devices that monitor network traffic to find unexpected patterns" [3]. Intrusion detection is the process of monitoring network traffic or system activity for unauthorized access, policy violations, and other malicious activities. The intrusion detection aims to identify potential security breaches and alert security personnel so they can take appropriate action to prevent further damage. There are various IDS (intrusion detection systems) to be used, including host-based IDS and network-based IDS. Host-based IDS monitor activity on a single device or server, while network-based IDS monitor all traffic on a network segment. Intrusion detection is a significant component of a comprehensive cybersecurity strategy. It can detect intrusions effectively by monitoring the status and activities of the protection system. Therefore, it has the ability to discover unauthorized or abnormal network behaviors.

As mentioned above, intrusion detection has three types, which are host-based detection, network-based detection, and collaborative detection [4]. HIDS (Host-based intrusion detection) is located in the software component of the monitored system, which mainly monitors activities within the host, such as system or shell program logs. Based on the detection techniques, there are two types of intrusion detection, including misuse detection [5]

and anomaly detection [6]. Misuse detection, also known as signature-based detection, is an application of signature matching to identify intrusions. It can effectively detect known attacks and has a low rate of false alarms.

However, some machine learning techniques have some shortcomings, such as training time being too long in large training sets, too sensitive to irrelevant attributes, and so on [7], researchers try hard to adopt deep learning technology to solve these problems. Currently, artificial intelligence technology is constantly developing, and multitudinous methods of machine learning or deep learning have been applied to intrusion detection systems [8]. The methods that use machine learning have better performance than classical intrusion detection methods. These methods have the ability to learn from a quantity of intrusion data to build an intrusion detection model for distinguishing whether there is an intrusion or not. But it still has some problems, such as the need for plentiful training samples, taking a long time, and relying on feature selection.

Deep learning is usually a modification of artificial neural networks for feature extraction, perception, and learning. Its applications are now used in scores of fields, such as speech recognition, unmanned vehicles, image recognition and classification, natural language processing, bioinformatics, etc. There are various neural network models using deep learning technology. In this paper, we propose an intrusion detection model based on multi-head attention with BiLSTM. The model completes the selection and extraction of features based on the multi-head attention mechanism, and captures the dependencies of vectors over longer distances through the BiLSTM model, thus improving the accuracy and efficiency of identifying network intrusions.

2. Related Research

The first part of the related literature that uses BiLSTM for intrusion detection: S.Siviamohan et al. [9] proposed a local university intrusion detection method based on RNN-BiLSTM (Bidirectional Long Short-Term Memory Recurrent Neural Network), which uses a two-step mechanism to solve the network problem. The experimental results show that BiLSTM is better than all other RNN (Recurrent Neural Network) architectures in terms of classification accuracy, and the prediction accuracy on the CICIDS2017 dataset reaches 98.48%, but it does not specify whether it is dual classification or multi-classification.

Nelly Elsayed et al. [10] produced an intrusion detection model by using BiLSTM and CNN (Convolutional Neural Network). The BiLSTM recursive behavior is used to save the information used for intrusion detection, while the CNN perfectly extracts the data features. It can be implemented and applied to lots of smart home network gateways.

Huang Chi et al. [11] created a network intrusion detection method, which uses CNN and BiLSTM. The former extracts local parallel features, solving the problem of incomplete local feature extraction. The latter is used to extract long-distance related features, taking into account the influence of attributes before and after each data point in the sequence data, which can improve accuracy.

Liangkang Zhang et al. [12] produced a new model based on mean control, CNN and BiLSTM. During data preprocessing, the data standardization of mean control is used to standardize the original data, and then the CNN-BiLSTM algorithm is combined to predict.

The following are the related literature of the works that use an attention mechanism for intrusion detection: Jingyi Wang et al. [13] proposed an intrusion detection model that uses an attention mechanism. A SSAE (Stacked Sparse Autoencoder) was constructed to extract the high-level feature representation of related information. Furthermore, the double-layer BiGRU (Bidirectional Gated Recurrent Unit) network with an attention mechanism was used to classify data.

Haixia Hou et al. [14] proposed a method that uses HLSTM (Hierarchical LSTM) and an attention mechanism. First of all, in order to extract sequence features across multiple hierarchical structures on network record sequences, researchers used HLSTM. Then, the attention layer's function is to capture the correlation between features, redistribute the

weight of features, and adaptively map the importance of each feature to different network attack categories.

Yalong Song et al. [15] proposed a mechanism using BiGRU and a multi-head attention mechanism. It can manage the data and capture the correlation between data and features.

The above articles all use artificial intelligence to predict intrusion detection but do not make a detailed classification prediction of the type of intrusion. Some articles [13] use binary classification, some articles [11,12,14] divide the main categories of network intrusion, and some articles [9,10,15] do not explain the classification clearly. The above-mentioned articles do not make a detailed prediction of intrusion detection. In order to resolve this situation and improve our classification and precision, we propose an intrusion detection model based on BiLSTM and a multi-head attention mechanism.

3. Model Methodology

In order to be suitable for the current NIDS (Network Intrusion Detection Systems) methods and their characteristics, we propose a new detection method to complete intrusion detection, which uses multi-head Attention and BiLSTM. The whole model consists of two phases, including the training phase and the prediction phase. In the first phase, the model's goal is mainly to learn the original vector features of the network intrusion data, training the network to adjust the weight parameters. Then, it can realize the training of the proposed model through true value comparison and loss function calculation. In the prediction phase, the predicted data was put into our model to obtain the final prediction results, this model can also calculate the relevant performance metrics. The overall training and evaluation structure is shown in Figure 1. A more detailed network model structure is shown in Figure 2.

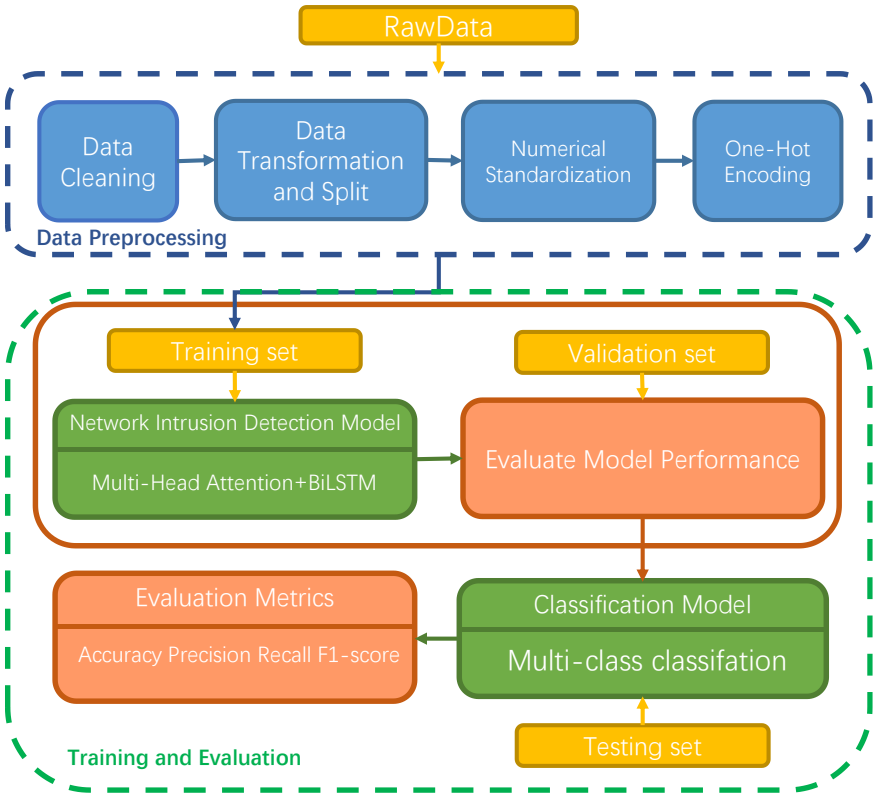

Figure 1. Overall structure of the model based on multi-head attention and BiLSTM.

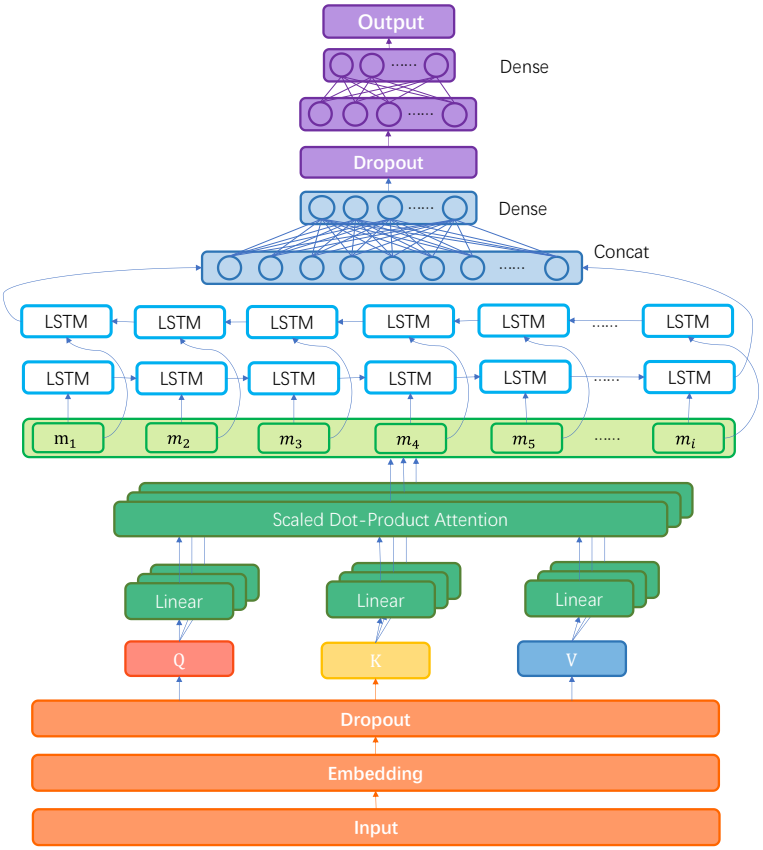

Figure 2. Intrusion detection model network structure.

3.1. Embedding

In Figure 2, we detail the model structure. First of all, we take the processed data and use the embedding layer to transform each feature of intrusion detection into the form of a vector.

The embedding layer is aimed to raise the dimension, and the input data is used to represent each feature value in the original vector (i ranges from 1 to 41 for KDDCUP99 and NSLKDD datasets, i ranges from 1 to 15 for CICIDS2017 datasets). Furthermore, $ai = Embedding(xi)$, where ai is the one-dimensional vector corresponding to each feature with a length of 32. At this time, the original vector is transformed into a two-dimensional vector (taking the NSLKDD dataset as an example, the data is transformed from one-dimensional data with a length of 41 to two-dimensional data with a length of [41, 32], where 41 is the number of features, 32 represents the embedding dimension). We hope that the features can be enlarged so that the model is capable of learning more characteristics of network intrusion activities with embedding. After the embedding layer is a dropout layer, which is used to improve the generalization ability of our proposed model. Without the dropout layer, the model we designed is prone to overfitting which leads to the low prediction accuracy of the test set. Therefore, the addition of the dropout layer can prevent overfitting to a certain extent, and that is, $ai = Dropout(ai)$.

3.2. Multi-Head Attention

Then, the processed data is passed to the multi-head attention [16] mechanism model. The reason why we use the mechanism is that it can pay attention to the important features in the vector, which is designed by imitating human vision [17]. It can give higher weights to important features and lower weights to other features. The data entering the multi-head attention mechanism model part is represented as $S = (a_1, a_2, \cdots, a_i)$, S will be multiplied with the weight data of the attention mechanism to obtain Q, K and V, that is, $q_i = SW_{qj}$,

$k_i = SW_{kj}$, and $v_i = SW_{vj}$, the range of j is from 1 to 3. The weight matrices W_q, W_k, and W_v can be continuously trained through learning, so the model's fitting ability can be further improved. The similarity matrix of different features is obtained by multiplying Q and K^T, and the similarity relationship between different features is obtained. After that, the similarity is normalized by the Softmax function, which reduces the amount of calculation to a certain extent. Finally, the obtained result is multiplied by V to obtain the data with the same dimension as the input according to the Equations (1) and (2). In Equation (2), d_K represents the dimension of the K matrix. Finally, we can obtain the final result according to Equation (3). Its structure is shown in Figure 3 schematically.

$$head_j(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

$$Softmax(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{2}$$

$$T = Concat(head_1, head_2, head_3) \tag{3}$$

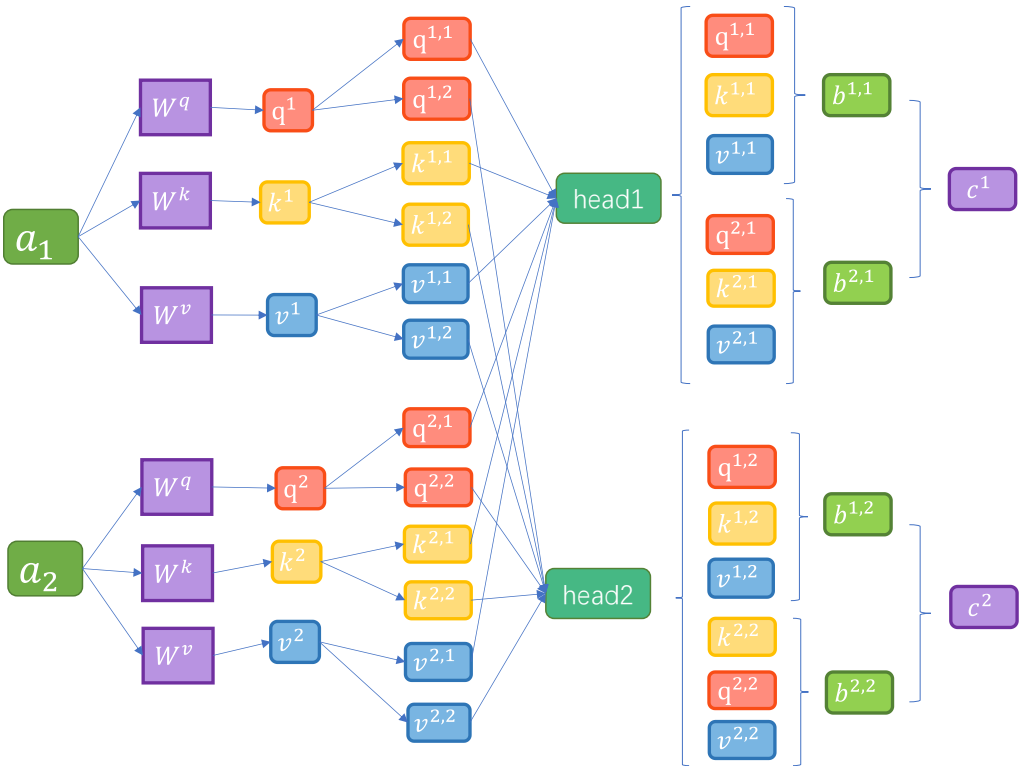

Figure 3. Schematic diagram of the multi-headed attention mechanism.

3.3. BiLSTM

After that, the data weighted by the attention mechanism is fed into the BiLSTM model. LSTM is a kind of RNN, that can learn and remember long-term dependencies, capture the relationship between different features in the feature vector, and will not encounter the problem of gradient disappearance or gradient explosion [18]. Graves et al. [19] reported an important improvement in classification accuracy when using LSTM in a bidirectional architecture. The feature vector of intrusion detection is not a time series data, but it can analyze the relationship between distant features, associate different features, and then make predictions. In this case, the output is a one-dimensional vector of length 128.

In the previous section, we were given the data T generated by the multi-head attention mechanism. Data T is a two-dimensional vector composed of multiple one-dimensional vectors, which we denote as $T = (m_1, m_2, \ldots, m_i)$.

We believe that for the detection of a certain relationship in the data, LSTM has the ability to capture this longer distance dependence while being able to avoid gradient disappearance, gradient explosion, and other problems. However, only using LSTM cannot encode back-to-back information. Therefore, we use BiLSTM to improve the ability to capture bidirectional features.

BiLSTM is composed of several small structures, with one basic unit consisting of four layers, which are the input layer, forward propagation layer, backward propagation layer, and output layer. The forward propagation layer is in charge of extracting the forward features of the vector from front to back, while the backward transmission layer is in charge of extracting the reverse features of the input sequence from back to front. The output layer integrates the data output from the forward propagation layer and the backward propagation layer. We want to extract the forward-backward correlation of the vectors, so the output formula of BiLSTM is shown in Equation (4).

$$h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}] \tag{4}$$

where $\oplus$ denotes the summation calculation of the corresponding elements. $\overrightarrow{h_i}$ denotes the forward output, $\overleftarrow{h_i}$ denotes the backward output. Finally, h_i denotes the result of the summation of the corresponding elements.

Among them, the BiLSTM network structure has lots of single LSTM structures, and the individual LSTM structure is shown in Figure 4.

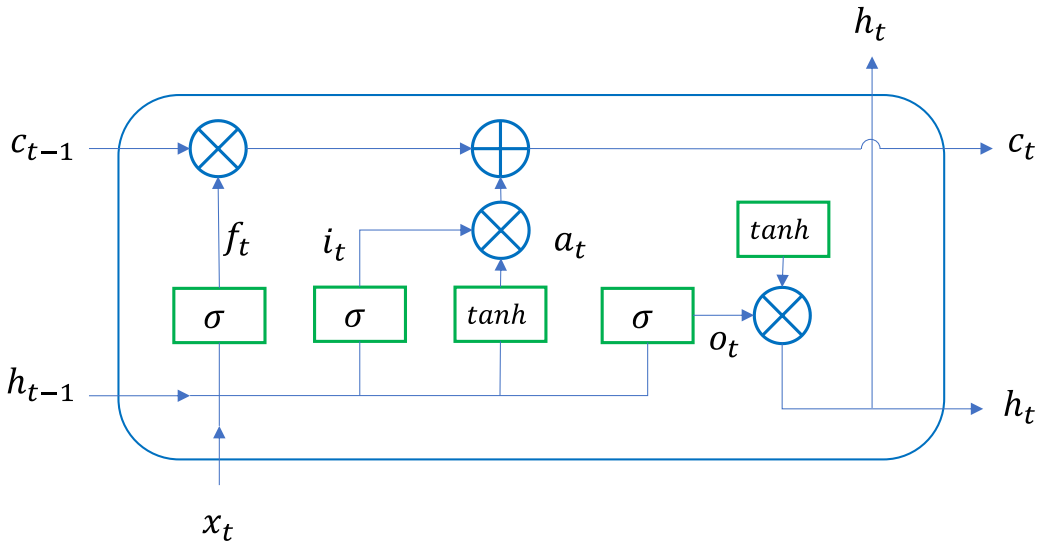

Figure 4. LSTM network structure.

LSTM adds three gating structures in the hidden layer, namely the forget gate, input gate, and output gate, and it also adds a new hidden cell state. In Figure 4, $f(t)$, $i(t)$, and $o(t)$ represent the forget gate, input gate, and output gate at time t, and $a(t)$ represents the initial feature extraction of $h(t-1)$ and $c(t)$ at time t. All formulas are shown in Equations (5)–(8).

$$f(t) = \sigma(W_f h_{t-1} + U_f m_t + b_f) \tag{5}$$

$$i(t) = \sigma(W_i h_{t-1} + U_i m_t + b_i) \tag{6}$$

$$a(t) = \tanh(W_a h_{t-1} + U_a m_t + b_a) \tag{7}$$

$$o(t) = \sigma(W_O h_{t-1} + U_O m_t + b_O) \tag{8}$$

where x_t represents the input at time t. h_{t-1} represents the hidden layer status value at time $t-1$. W_f, W_i, W_o represent the weight parameter of h_{t-1} in the feature extraction process

of the forget gate, input gate, and output gate. U_f, U_i, U_o represent the weight parameter of x_t in the feature extraction process of forget gate, input gate, and output gate. b_f, b_i, and b_o represent the forget gate, input gate, output gate, and offset value in the process of feature extraction, respectively. The related functions are shown in Equation (9) [20] and Equation (10) [21]:

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{9}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

The results of the forget gate and output gate calculations act on $c(t-1)$, which constitutes the cell state $c(t)$ at moment t, denoted as Equation (11). The final hidden state $h(t)$ at moment t is derived from the output gate o(t) as well as the cell state $c(t)$, denoted as Equation (12). where $\odot$ represents Hadamard product.

$$c(t) = c(t-1) \odot f(t) + i(t) \odot a(t) \tag{11}$$

$$h(t) = o(t) \odot \tanh(c(t)) \tag{12}$$

3.4. Dense Layer

Dense layers are a type of layer that helps the neural network to better understand the data and improve the accuracy of the model's predictions. We add two dense layers after BiLSTM, but each uses different activation functions. The first dense layer includes the ReLU activation function [22] (shown in Equation (13)). It aims to learn the features in the data and distinguish different features that improve the accuracy of the model prediction. The ReLU function is very simple and fast to calculate, which is very suitable for large-scale deep neural networks. Meanwhile, the ReLU function can effectively prevent the case of gradient disappearance, and the gradient is constant to 1 when the input is bigger than 0.

$$ReLU(x) = max(0, x) \tag{13}$$

The second dense layer uses the Softmax activation function (shown in Equation (2)), which is often used in multi-class classification problems, so it is very suitable for our network detection model. Then, the output of the dense layer also corresponds to the number of attack types. In the end, our model predicts all the mentioned attack types in the dataset, so it is of great importance to use the Softmax activation function.

Combining multi-head attention with BiLSTM has several advantages, including:

- Improved sequence modeling: BiLSTM is a type of RNN that can effectively model sequential data in both forward and backward directions. However, when combined with multi-head attention, they can further enhance the model's ability to capture long-range dependencies and improve the quality of sequence modeling.
- Increased interpretability: Multi-head attention mechanism allows the model to attend to distinct parts of the input sequence selectively, providing more transparency and interpretability to the model's decision-making process. This is particularly useful in detection tasks such as network intrusion detection.
- Robustness to noise and variations: By attending to multiple parts of the input sequence, the model becomes more robust to variations and noise in the data.
- Scalability: The combination of multi-head attention with BiLSTM allows the model to scale well to larger datasets and more complex tasks without compromising performance or accuracy. This makes it an effective approach for handling large-scale network intrusion detection tasks.

4. Implementation Details

In our experiments, the hardware environment is as follows: the CPU model is Intel Core i7-10750H, the GPU model is NVIDIA GeForce RTX2060 with Max-Q Design, the memory on the GPU card is 6 GB and the RAM on the computer is 32 GB.

The language and platform (software) environment are as follows: the operating system used in the experiment is Windows 11, and the programming environment is Python 3.9. The Keras deep learning framework and Scikit-learn framework are used to help us build the model and process the data.

We divided the data set into three parts, each part has a different function. The training set accounted for 64%, the validation set accounted for 16%, and the test set accounted for 20%. A total of 60 rounds of model training were performed, the batch size was 512, the random number seed was 0, and the number of heads of the multi-head attention mechanism was 3. The Adam algorithm [23] is used as the optimizer of the model, and the value of the learning rate is 0.0003. Adam is able to adaptively adjust the learning rate based on the gradient information and adjust the momentum to avoid falling into a local minimum too early. We add two dense layers after BiLSTM and in the output part, but they have distinct activation functions, one is the ReLU activation function, and the other is the Softmax activation function. Meanwhile, a dropout layer is added after the embedding layer and between the two dense layers with parameters of 0.8 and 0.3, respectively. The whole training time of the model is 55 min on the KDDCUP99 dataset, 4 min 45 s on the NSLKDD dataset, and 33 min on the CICIDS2017 dataset.

The core code of the model we designed is shown in Figure 5. The code is written in Python and built under the Keras deep learning framework.

```
inputs = Input(name='inputs', shape=[max_words, ], dtype='float64')
x = Embedding(top_words, input_length=max_words, output_dim=embed_dim)(inputs)
x = Dropout(0.8)(x)
x = MultiHeadAttention(num_heads=3, key_dim=embed_dim)(x, x, x)
x = Bidirectional(LSTM(hidden_dim[0]))(x)
x = Dense(64, activation='relu')(x)
x = Dropout(0.3)(x)
output = Dense(num_labels, activation='softmax')(x)
model = Model(inputs=inputs, outputs=output)
```

Figure 5. The core code of the model.

5. Experiment and Results

In the model experiment, in order to ensure the accuracy of the experiment, we use three data sets, namely the KDDCUP99 data set, the NSLKDD data set, and the CICIDS2017 data set, which can carry out a more comprehensive evaluation of our model.

5.1. Introduction to the Data Set

The KDDCUP99 dataset [24] is a widely used benchmark dataset in the field of network intrusion detection. It was created for not only the KDD Cup 1999 Data Mining but also the Knowledge Discovery competition. It aims to develop efficient algorithms for detecting network intrusions from TCP/IP network traffic data. This dataset consists of a large collection of network traffic data captured from a simulated environment. It also contains a variety of features extracted from network packets, such as protocol types, service types, and so on.

The NSLKDD dataset [25] is an improved version of the KDD Cup 1999 dataset. This dataset is widely used for network intrusion detection research. NSLKDD stands for "NSL-KDD Intrusion Detection Dataset". It was developed to address some limitations and drawbacks of the original KDD Cup dataset. The types of attacks are shown in Table 1.

Table 1. KDDCUP99 and NSLKDD dataset attack types.

Category	Attack	Interpretation
Normal	Normal	Normal network activity
DOS	back, land, neptune, pod, smurf, teardrop	Denial-of-Service (DoS) attack is a type of cyber attack where a perpetrator attempts to make a website or network resource unavailable to its intended users by overwhelming it with traffic or other types of data.
Probing	ipsweep, nmap, portsweep, satan	Surveillance and other detection activities.
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster	Remote-to-Local (R2L) attack is a type of cyber attack where an attacker tries to gain unauthorized access to a target system by exploiting vulnerabilities in remote services or applications.
U2R	buffer overflow, loadmodule, perl, rootkit	User-to-Root (U2R) attackis a type of cyber attack where an attacker with limited privileges on a system attempts to gain root-level access.

The CICIDS2017 dataset [26], also known as the Canadian Institute for Cybersecurity Intrusion Detection System (CIC-IDS2017), is a comprehensive dataset designed for evaluating NIDS. Its authors are researchers at the University of New Brunswick in Canada. This dataset consists of various network traffic features extracted from different types of network traffic, including normal traffic and several types of attacks. It also simulates a real-world network environment to provide a realistic representation of network traffic. A short description of its files is shown in Table 2. Descriptions of all the datasets are shown in Table 3 (Table 3 is the data set size that has been balanced by using the algorithm SMOTE. The relevant content is presented in Section 5.2.4).

Table 2. Description of files containing CICIDS2017 dataset.

Name of Files	Day Activity	Attacks Found	Advantage	Goal
Monday WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)	This is a dataset that can further meet real-world standards, covering attack standards from 11 countries, making it more reliable and available [26].	Using this dataset can help improve our model's generalization ability and improve its accuracy in modern intrusion detection predictions, rather than just being applicable to the past.
Tuesday WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator		
Wednesday workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed		
Thursday-WorkingHours Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS		
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Thursday	Benign, Infiltration		
Friday-WorkingHours Morning.pcap_ISCX.csv	Friday	Benign, Bot		
Friday-WorkingHours-Afternoon PortScan.pcap_ISCX.csv	Friday	Benign, PortScan		
Friday-WorkingHours-Afternoon DDos.pcap_ISCX.csv	Friday	Benign, DDoS		

Table 3. Description of all datasets.

Dataset	Training Set	Validation Set	Test Set	Total	Input Vector Features	Number of Labels
KDDCUP99	3,108,950	777,237	971,547	4,857,734	41	40
NSLKDD	95,050	23,762	29,704	148,516	41	40
CICIDS2017	498,741	124,685	155,857	779283	78	15

5.2. Data Processing

We used numerical division, data normalization, one-hot encoding, and data balance to process the data set. For KDDCUP99 and NSLKDD data sets, the dimension after data processing is 41×1, and the data set has 40 classification tags, one of which is the normal category, and the other 39 represent various attack types. After one-hot encoding, they are converted into 40-dimensional vectors, so the dimension of the output layer is 40. For the CICIDS2017 dataset, the dimension after data processing is 78×1, and the dataset has 15 classification labels in total, one of which is the normal category, and the other 14 labels represent various attack types. After one-hot encoding, it is converted to a 15-dimensional vector, so the output dimension is 15.

5.2.1. Data Conversion

For the KDDCUP99 dataset, we convert the text type data into numeric types to facilitate our training and testing of the model, such as character type data protocal_type, service, and state, where protocal_type has 3 protocol types, service has 70 network service types, while flag haves 11 network connection types. For instance, the three protocol types characterized by protocol type are TCP, UDP, and ICMP, which we convert to 0, 1, and 2.

5.2.2. Data Normalization

The variation in individual features of the numerically processed data is large, and normalizing this data can avoid causing gradient dispersion when using the backpropagation algorithm. The problem with not normalizing the data is that the magnitude of the gradient decreases with backpropagation, which slows down the growth of the update weights of the intrusion detection model, resulting in a complex dataset of features that are not well extracted by deep learning. We use the normalization method of z-score to convert all the data of KDDCUP99 to $[-1, 1]$, as shown in Equation (14).

$$m'_i = \frac{m_i - \overline{m}}{x} \tag{14}$$

We use m_i and m'_i to represent the value of the data sample before and after normalization. Meanwhile, $\overline{m}$ is used to represent the average data value of the feature before normalization.

5.2.3. One-Hot Encoding

In the feature vectors of the three datasets, we did not use one-hot encoding for processing, because the original data has too many features; thus, using it may lead to the curse of dimensionality and reduce the accuracy. We perform one-hot encoding for the label values (predicted attack types) of the datasets. For the KDDCUP99 dataset and NSLKDD dataset, we convert 40 different categories into one-dimensional vectors with a length of 40 by one-hot encoding. For the CICIDS2017 dataset, we convert 15 different categories into one-dimensional vectors with a length of 15 by one-hot encoding. This encoding is more conducive to model training.

5.2.4. Dataset Balanced

For the KDDCUP99 and CICIDS2017 datasets, there is an imbalance between normal samples and attack samples. To make the dataset more balanced, we use the SMOTE (Synthetic Minority Over-sampling Technique) [27] algorithm to over-sample a small number of samples. To ensure the balance of the dataset and the generalization of our model, we randomly under-sample the samples with a large number of samples.

SMOTE works by creating synthetic examples of the minority class that are strategically placed between existing instances of the minority class. The algorithm randomly selects a minority instance and looks for its k nearest neighbors. It then selects one of these neighbors and calculates the difference between the feature values of the two instances. It multiplies this difference by a random value between 0 and 1 and adds it to the feature values of the selected minority instance. This generates a synthetic instance that is similar to the original minority instance but slightly different.

By repeating this process for multiple instances of the minority samples, SMOTE increases the number of these samples in the dataset. This operation helps to balance the class distribution and provides more training examples for the minority class, and it also improves the performance of machine learning algorithms. The size of the dataset after processing is shown in Table 3.

5.3. Evaluation Criteria

We use Accuracy, Precision, Recall, and $F1$-Score as the metrics for model evaluation, and the formulas for all metrics are shown in Table 4. Where TN indicates the number of correctly predicted negative samples, TP indicates the number of correctly predicted positive samples, FN indicates the number of incorrectly predicted negative samples, and FP indicates the number of incorrectly predicted positive samples.

Table 4. Model evaluation metrics.

Metric	Mathematical Formulae
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	$Precison = \frac{TP}{TP+FP}$
Recall	$Recall = \frac{TP}{TP+FN}$
$F1$-Score	$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision \times Recall}$

5.4. Model Review

We have tested and evaluated our model using metrics in Table 4. The result is shown in Table 5. Through the test data, we obtain the information that the accuracy of our model on the KDDCUP99 dataset is 98.29%, the Precision against normal samples (unattacked samples) is 0.97, Recall is 1, and $F1$ is 0.98. The accuracy of our model on the NSLKDD dataset is 95.19%, the Precision against normal samples (unattacked samples) is 0.95, Recall is 0.98, and $F1$ is 0.97. For the CICIDS2017 dataset, the accuracy is 99.08%, the Precision against normal samples (unattacked samples) is 1, Recall is 0.99, and $F1$ is 0.99. It was found that changing the ratio of the training set and test set size did not have an impact on the detection results of the samples. The data results are shown in Figure 6.

Table 5. Performance of different network structures on the dataset.

Dataset	Structure	Accuracy(%)	Precision	Recall	*F*1-Score
KDDCUP99	Transformer	85.71	0.88	0.82	0.85
	BiLSTM	98.25	0.97	1	0.98
	Attention	71.65	0.63	0.99	0.77
	Multi-Head Attention	71.54	0.63	0.98	0.77
	Attention + BiLSTM	97.96	0.97	1	0.98

Table 5. *Cont.*

Dataset	Structure	Accuracy(%)	Precision	Recall	*F*1-Score
	Multi-Head Attention + BiLSTM	98.29	0.97	1	0.98
NSLKDD	Transformer	73.26	0.75	0.81	0.78
	BiLSTM	95.13	0.96	0.97	0.97
	Attention	65.01	0.76	0.62	0.68
	Multi-Head Attention	65.01	0.76	0.62	0.68
	Attention + BiLSTM	94.7	0.95	0.98	0.96
	Multi-Head Attention + BiLSTM	95.19	0.95	0.98	0.97
CICID17	Transformer	97.94	0.98	0.97	0.97
	BiLSTM	98.51	0.99	0.98	0.99
	Attention	97.75	0.98	0.97	0.98
	Multi-Head Attention	97.88	0.99	0.97	0.98
	Attention + BiLSTM	97.24	0.97	0.97	0.97
	Multi-Head Attention + BiLSTM	99.08	1	0.99	0.99

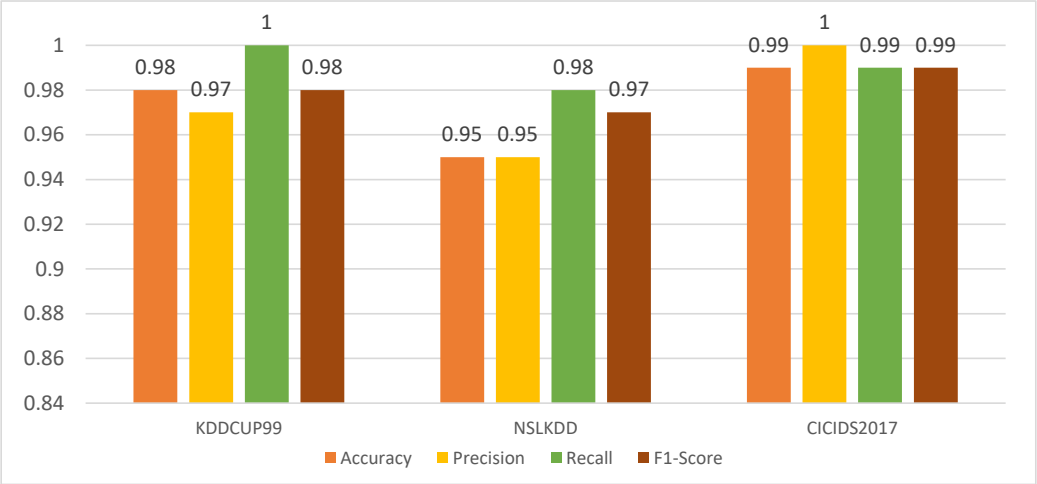

Figure 6. Performance evaluation of three datasets.

5.5. Model Accuracy and Loss Variation

Next, Figures 7–9 show the loss and accuracy during the training of the KDDCUP99 dataset, NSLKDD dataset, and CICIDS2017 dataset, respectively. The experiment carried out 60 rounds of training. For the KDDCUP99 dataset, the curve became smoother after 10 rounds of training, and finally, the value of loss dropped to 0.0049. For the NSLKDD dataset, the curve became smoother after 30 rounds of training, and the value of loss dropped to 0.1481. For the CICIDS2017 dataset, the curve became smoother after 20 rounds of training, and the value of loss dropped to 0.0209. It can be seen from the training process that the model proposed by us has better learning ability and fitting speed. The normalized confusion matrix of the CICIDS2017 dataset is shown in Figure 10.

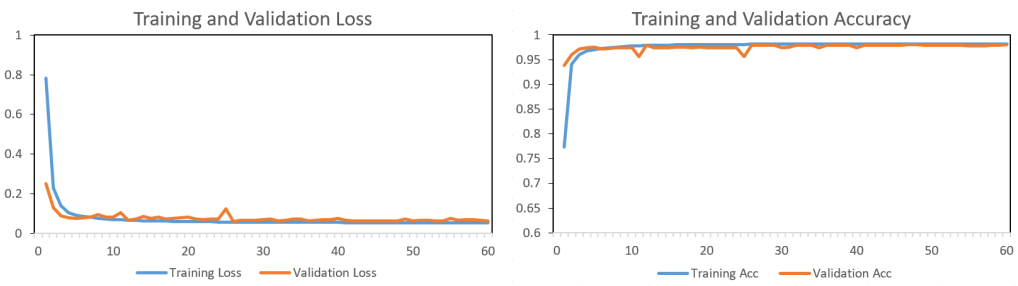

Figure 7. Accuracy and loss variation of model training on the KDDCUP99 dataset.

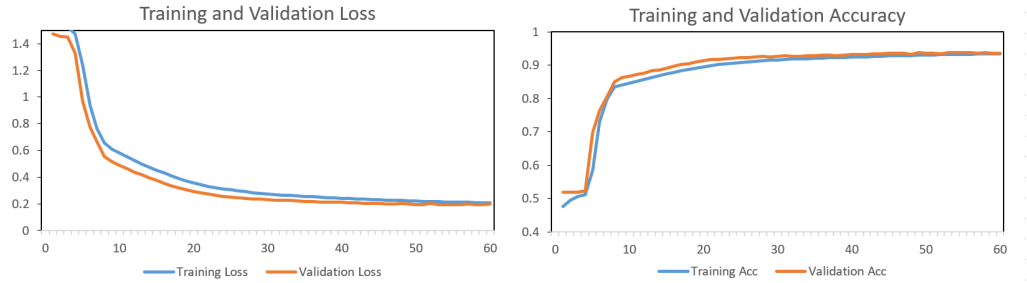

Figure 8. Accuracy and loss variation of model training on the NSLKDD dataset.

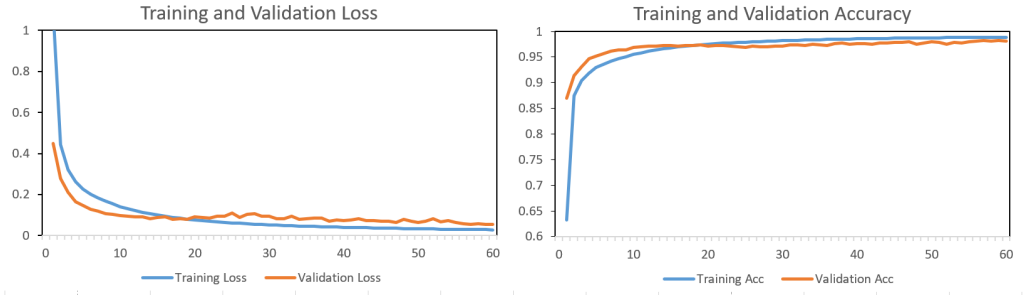

Figure 9. Accuracy and loss variation of model training on the CICIDS2017 dataset.

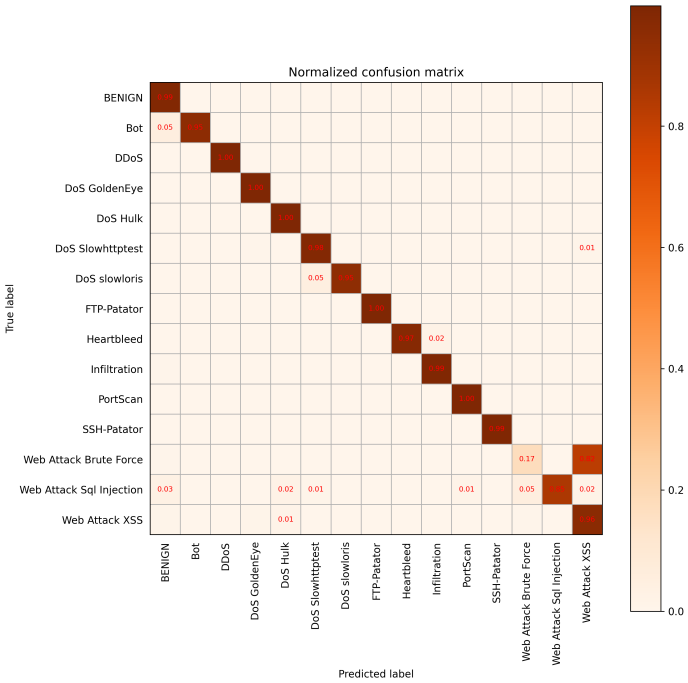

Figure 10. The normalized confusion matrix of CICIDS2017 dataset.

5.6. Ablation Experiments

We believe that the reasons for the high accuracy as well as the better performance of the proposed model are:

- We use BiLSTM to build our model. On the one hand, it can capture the bidirectional features better, on the other hand, it has the ability to avoid situations such as gradient disappearance and gradient explosion, which are very suitable for network intrusion detection.

- The addition of the multi-headed attention mechanism allows different attention weights for each vector in the feature vector to strengthen the relationship between certain vectors and the type of detected attacks, which improves the accuracy of detection. It also avoids the problem of over-focusing attention on its position.

We have verified the performance and accuracy of the Transformer, BiLSTM, Attention, Multi-Head Attention, Attention + BiLSTM, and Multi-Head attention + BiLSTM structures on three data sets. For Multi-Head Attention, we obtain the best head parameter of 3 according to pre-training. In Table 5, we show the performance indicators of different structures on three datasets by using four metrics shown in Table 4 for normal samples (samples that are not under attack). It can be seen from the results that the Multi-Head Attention + BiLSTM structure has higher accuracy and higher F1-Score.

From the data in Table 5, we can obtain the information that the classification effect of the single attention mechanism model or the multi-head attention mechanism model is not good. Furthermore, the performance on the KDDCUP99 dataset and NSLKDD dataset is even worse. The Transformer model alone is not fully competent for the task of intrusion detection, and the accuracy is relatively low. However, when the multi-head attention mechanism is combined with the BiLSTM model, the accuracy of intrusion detection recognition can be increased, and its detection accuracy is also better than that of the BiLSTM model alone, indicating that the combination of the two can show the advantages of both, it also can adapt well to the task of intrusion detection.

5.7. Comparison with Other Models

The experiments of the model were trained and predicted on the KDDCUP99, NSLKDD, and CICIDS2017 datasets, and we compared our model with other models, comparing metrics including accuracy as well as F1-score. the final results are shown in Table 6. The accuracy is shown in Figure 11.

In Table 6 and Figure 11, the information we have obtained indicates that our proposed model has better predictive performance, which means that more refined intrusion detection can be performed. At the same time, we also achieve the most refined classification and inform the relevant people with more specific information.

Table 6. Comparison with other models on the dataset.

Dataset	Algorithm	Accuracy (%)	*F*1-Score (%)
KDDCUP99	CLAIRE [28]	93.58	95.9
	MCLDM [29]	93.94	96.06
	Improved LSTM [30]	97.79	96.95
	Our Method	98.29	98
NSLKDD	BAT [31]	84.25	-
	ICVAE-DNN [32]	85.97	86.27
	Deep AE [33]	87	81.21
	Our Method	95.19	97
CICIDS2017	KELM [34]	97.15	-
	CLAIRE [28]	98	-
	CNN [35]	98	98
	Our Method	99.08	99

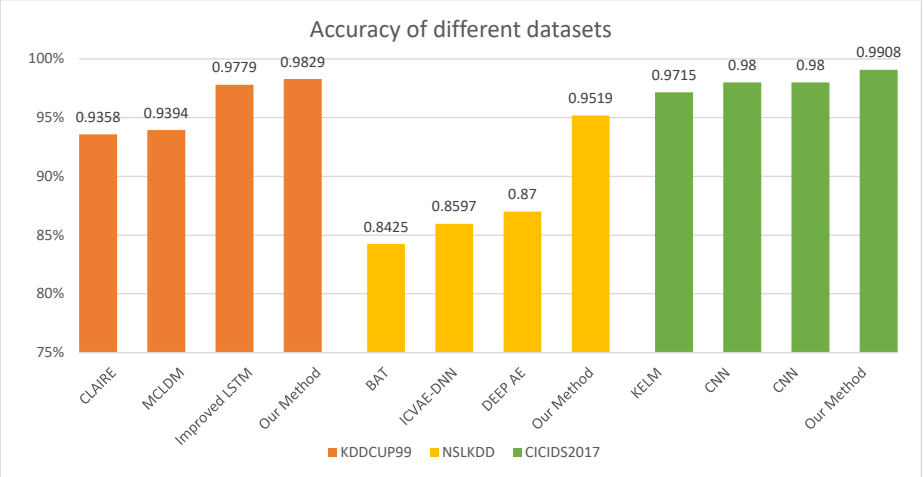

Figure 11. Comparison of different models on three datasets.

6. Conclusions

In this paper, we propose an intrusion detection model based on a multi-head attention mechanism and BiLSTM. The embedding layers can convert sparse high-dimensional feature vectors into low-dimensional feature vectors. This operation can fuse a large amount of valuable information. Then, we try to use the attention mechanism to introduce different attention weights for each vector in the feature vector, not only strengthening the relationship between certain vectors and the type of detected attacks but also improving the accuracy of detection. We also improve the use of a multi-head attention mechanism to avoid focusing too much attention on certain elements in the vector. Finally, we apply the BiLSTM network to detect some kind of relationship that exists in the data, while LSTM aims to capture long-distance dependencies and also can avoid lots of situations like gradient disappearance and gradient explosion. The experimental comparison shows that our proposed model has better accuracy and *F*1-score on the KDDCUP99, NSLKDD, and CICIDS2017 datasets than other models, and it is more accurate for multiple types of intrusion detection than other binary intrusion detection models.

Of course, our model still has some shortcomings. For example, our model is a multi-classification model, hoping to make more detailed and accurate predictions for different network intrusions. If the intrusion is a new type, our model can not give its definition or report what kind of specific intrusion method it is, but it can still be classified as an intrusion type rather than a normal network activity for professionals to study. In addition, the normal samples of the KDDCUP99 dataset and the normal samples of the CICIDS2017 dataset are too large. To ensure the availability of the detection model, we use oversampling and undersampling to ensure the balance of the data, but the sampling process has great randomness, which may delete some important information in the majority sample. In future improvements, we will try to solve these problems.

Author Contributions: Conceptualization, J.Z., X.Z., Z.L. and F.F.; methodology, J.Z.; software, J.Z. and Y.J.; validation, J.Z. and Y.J.; formal analysis, J.Z.; investigation, X.Z.; resources, X.Z.; data curation, J.Z.; writing—Original draft preparation, J.Z. and X.Z.; writing—Review and editing, Z.L., F.F. and F.X.; supervision, F.F.; project administration, J.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Hainan Province Science and Technology Special Fund (Grant No. ZDYF2021GXJS006), Haikou Science and Technology Plan Project (Grant No. 2022-007) and Key Laboratory of PK System Technologies Research of Hainan, China.

Data Availability Statement: The datasets analyzed during this study are available from the corresponding author upon reasonable request.

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

References

1. Manzoor, I.; Kumar, N. A feature reduced intrusion detection system using ANN classifier. *Expert Syst. Appl.* **2017**, *88*, 249–257.
2. Thapa, S.; Mailewa, A. The role of intrusion detection/prevention systems in modern computer networks: A review. In Proceedings of the Conference: Midwest Instruction and Computing Symposium (MICS), Online, 3–4 April 2020 ; Volume 53, pp. 1–14.
3. Patgiri, R.; Varshney, U.; Akutota, T.; Kunde, R. An investigation on intrusion detection system using machine learning. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1684–1691.
4. Liu, M.; Xue, Z.; Xu, X.; Zhong, C.; Chen, J. Host-based intrusion detection system with system calls: Review and future trends. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–36. [CrossRef]
5. Pu, G.; Wang, L.; Shen, J.; Dong, F. A hybrid unsupervised clustering-based anomaly detection method. *Tsinghua Sci. Technol.* **2020**, *26*, 146–153. [CrossRef]
6. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
7. Momand, A.; Jan, S.U.; Ramzan, N. A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning: Deep Learning, Datasets, and Attack Taxonomy. *J. Sens.* **2023**, *2023*, 6048087. [CrossRef]
8. Liu, H.; Lang, B. Machine learning and deep learning methods for intrusion detection systems: A survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
9. Sivamohan, S.; Sridhar, S.; Krishnaveni, S. An effective recurrent neural network (RNN) based intrusion detection via bi-directional long short-term memory. In Proceedings of the 2021 international conference on intelligent technologies (CONIT), Hubli, India, 25–27 June 2021; pp. 1–5.
10. Elsayed, N.; Zaghloul, Z.S.; Azumah, S.W.; Li, C. Intrusion detection system in smart home network using bidirectional LSTM and convolutional neural networks hybrid model. In Proceedings of the 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Lansing, MI, USA, 9–11 August 2021; pp. 55–58.
11. Chi, H.; Lin, C. Industrial Intrusion Detection System Based on CNN-Attention-BILSTM Network. In Proceedings of the 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), Huaihua City, China, 15–17 July 2022; pp. 32–39.
12. Zhang, L.; Huang, J.; Zhang, Y.; Zhang, G. Intrusion detection model of CNN-BiLSTM algorithm based on mean control. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; pp. 22–27.
13. Wang, J.; Chen, N.; Yu, J.; Jin, Y.; Li, Y. An efficient intrusion detection model combined bidirectional gated recurrent units with attention mechanism. In Proceedings of the 2020 7th International Conference on Behavioural and Social Computing (BESC), Bournemouth, UK, 5–7 November 2020; pp. 1–6.
14. Hou, H.; Di, Z.; Zhang, M.; Yuan, D. An Intrusion Detection Method for Cyber Monintoring Using Attention based Hierarchical LSTM. In Proceedings of the 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Jinan, China, 6–8 May 2022; pp. 125–130.
15. Song, Y.; Zhang, D.; Li, Y.; Shi, S.; Duan, P.; Wei, J. Intrusion Detection for Internet of Things Networks using Attention Mechanism and BiGRU. In Proceedings of the 2023 5th International Conference on Electronic Engineering and Informatics (EEI), Wuhan, China, 30 June–2 July 2023; pp. 227–230.
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
17. Liu, C.; Liu, Y.; Yan, Y.; Wang, J. An intrusion detection model with hierarchical attention mechanism. *IEEE Access* **2020**, *8*, 67542–67554. [CrossRef]
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
19. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef]
20. Medsker, L.R.; Jain, L. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 2.
21. Schtickzelle, M. Pierre-François Verhulst (1804–1849). La première découverte de la fonction logistique. *Population* **1981**, *3*, 541–556. [CrossRef]
22. Sudjianto, A.; Knauth, W.; Singh, R.; Yang, Z.; Zhang, A. Unwrapping the black box of deep relu networks: Interpretability, diagnostics, and simplification. *arXiv* **2020**, arXiv:2011.04041.
23. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
24. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE symposium on computational intelligence for security and defense applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
25. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *Int. J. Eng. Res. Technol.* **2013**, *2*, 1848–1853.

26. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108–116.
27. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
28. Andresini, G.; Appice, A.; Malerba, D. Nearest cluster-based intrusion detection through convolutional neural networks. *Knowl.-Based Syst.* **2021**, *216*, 106798. [CrossRef]
29. Luo, J.; Zhang, Y.; Wu, Y.; Xu, Y.; Guo, X.; Shang, B. A Multi-Channel Contrastive Learning Network Based Intrusion Detection Method. *Electronics* **2023**, *12*, 949. [CrossRef]
30. Zhang, L.; Yan, H.; Zhu, Q. An Improved LSTM Network Intrusion Detection Method. In Proceedings of the 2020 IEEE 6th International conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1765–1769.
31. Su, T.; Sun, H.; Zhu, J.; Wang, S.; Li, Y. BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset. *IEEE Access* **2020**, *8*, 29575–29585. [CrossRef]
32. Yang, Y.; Zheng, K.; Wu, C.; Yang, Y. Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network. *Sensors* **2019**, *19*, 2528. [CrossRef]
33. Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing* **2020**, *387*, 51–62. [CrossRef]
34. Wang, Z.; Zeng, Y.; Liu, Y.; Li, D. Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. *IEEE Access* **2021**, *9*, 16062–16091. [CrossRef]
35. Mendonça, R.V.; Teodoro, A.A.; Rosa, R.L.; Saadi, M.; Melgarejo, D.C.; Nardelli, P.H.; Rodríguez, D.Z. Intrusion detection system based on fast hierarchical deep convolutional neural network. *IEEE Access* **2021**, *9*, 61024–61034. [CrossRef]