JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 935-950 (2023)
DOI: 10.6688/JISE.202307_39(4).0014

A Densely Stacked Attention Method for Cyberattack Detection

HAIXIA HOU[1,*], DAOJUN LIANG[2,3,*],
MINGQIANG ZHANG[3,4] AND DONGFENG YUAN[3,†]
[1]*College of Science and Information Science, Qingdao Agricultural University, 266109 China*
[2]*School of Information Science and Engineering, Shandong University, Qingdao, 266299 China*
[3]*Shandong Key Laboratory of Wireless Mobile Communication Technologies, Jinan, 250000 China*
[4]*School of Cyber Science and Engineering, Qufu Normal University, 273165 China*
E-mail: hxhou@qau.edu.cn; {liangdaojun; mqzhang}@mail.sdu.edu.cn;
[†]*dfyuan@sdu.edu.cn*

Cyberattack Detection plays a vital role in network security and is an important means to maintain network security. In order to enhance the security and improve the detection ability of malicious intrusion behavior in the network, this paper proposes a multi-layer Dense Attention (Denseformer) model. The model is composed of multiple transformer-like structures, and each layer is stacked of multiple encoder and decoder sub-layers. The encoder and decoder include self-attention and cross-attention mechanisms, and their features are obtained by cross-fusion of multi-branch structures. By sharing information among multiple encoder-decoder layers, Denseformer can use the attention mechanism to process unserialized input source samples. On the whole, Denseformer is like an attention network embedded on the dense layer, making it easier to handle correlations between features. By stacking the encoding and decoding modules with attention, Denseformer has better generalization performance than other models, thereby improving its cyberattack detection accuracy. The experimental results show that, without other complex training techniques, the proposed method achieves 85.65% on the NSL-KDD dataset.

Keywords: cyberattack detection, intrusion detection system, deep learning, attention mechanism, data scarcity

1. INTRODUCTION

With the rapid updates and iterations of the Internet of Things (IoT), Artificial Intelligence (AI), and Cloud Computing, network information security problems have become increasingly prominent and are facing huge challenges in recent years. Cyberattack detection is a main means to ensure network security. The role of the cyberattack detection model is to monitor and analyze network communications, and identify abnormal behaviors in the network through active response. Intrusion Detection System (IDS) can be defined as a security service. It monitors and investigates system events to identify unauthorized access to system resources [1].

Received June 2, 2022; revised October 3, 2022; accepted November 25, 2022.
Communicated by Meng Chang Chen.
[†] Corresponding author.
[*] The authors contributed equally to this work and should be considered co-first authors.

There are many methods to detect cyberattacks, including traditional machine methods [2–7] and deep learning methods [8–12]. Some classic methods, including KNN, SVM, SOM, random forest, *etc*., are all used for network attack detection. The advantages of these methods are fast training and strong interpretability, but the disadvantages are low detection accuracy and poor generalization performance. For example, these methods may have a larger false position rate [1]. There is also a lot of work in dealing with unbalanced data [5–7, 13] for cyberattack detection problems. Conventional cyberattack detection models usually perform poorly in unbalanced datasets, because they cause the classification results to be biased towards classes with large number of samples. At present, researchers usually use under-sampling [6, 13] or over-sampling [5, 7] methods to deal with the problem of data imbalance in network intrusion detection. When unbalanced data is processed by under-sampling, it is easy to lose most of the sample information, resulting in a decrease in the classification accuracy of categories with many samples. Some oversampling methods, for example, SMOTE [5] oversampling samples based on simple interpolation operations, it is easy to create redundant data samples to increase the difficulty of model training.

In recent years, deep learning has developed rapidly with its powerful representation and generalization capabilities. For example, sequential structures such as Recurrent Neural Network (RNN) [14], Long Short-Term Memory (LSTM) [15], and Transformer [16] are suitable for Natural Language Processing (NLP). And CNN [17] structures such as ResNet [18] and DenseNet [19] are suitable for image processing. On the cyberattack detection task, many works use deep learning to improve the accuracy and generalization of the model, such as Deep Belief Networks [8, 20, 21], deep AutoEncoder-like models [9,22–24], CNN-like models [25–27], RNN-like models [28], LSTM-like models [11], Attention-like models [12, 29]. These models take the detection data as a time series and input them into the encoder-decoder architecture to improve their generalization capability. Inspired by this, we consider the cyberattack detection dataset as real features plus varying noises, and use a stacked multi-layer encoder-decoder structure to capture the relationship between real features. In this paper, we introduce the Dense Attention (Denseformer) layer for cyberattack detection tasks, which has high robustness to noise, thus effectively preserving the real features and improving its generalization performance.

The paper is organized as follows. In Section 2, we introduce some relate work about intrusion detection. Then, the proposed Denseformer is included in Section 3. Section 4 introduces experiments result of the proposed method. Finally, we conclude the paper in Section 6.

2. RELATED WORK

Cyberattack detection problems is widely studied because of its importance to network security. The existing research methods include machine learning methods and deep learning methods. Classical machine learning methods are widely used because of their fast training speed and strong interpretability. In [2], the Tree-Seed Algorithm (TSA) is introduced to extract the effective feature of the input data, and KNN is used for classification. In [3], the authors have reviewed the application of Self-Organizing Mapping (SOM) in intrusion detection. The authors in [4] developed a combining classifier model

based on tree-based algorithms for cyberattack detection. Their algorithm is mainly based on Weka software, using data mining methods.

With the development and popularity of deep learning, many deep learning methods have also been used in cyberattack detection. The Deep Belief Networks (DBN) is introduced to the field of cyberattack detection in [8], and a cyberattack detection model based on DBN is proposed to apply in intrusion recognition domain. The authors in [9] proposed an asymmetric deep AutoEncoder (AE) for unsupervised feature learning to realize cyberattack detection. In [11], the authors proposed an IDS detection method based on hierarchical LSTM (HLSTM) network. With the introduction of hierarchical LSTM, the network can learn multiple time levels on complex network traffic sequences. The authors in [10] use ResNet for cyberattack detection, which converted the intrusion data into image signals and conducted binary classification evaluation in the NSL-KDD dataset.

The author in [25] use the layer of Conv, FC, LSTM and its variants to construct different models such as 3Conv+BiLSTM, Conv+LSTM, 4Conv+2FC and Conv+2FC to select optimal network architecture. They found that CNN and its variant architectures have significantly performed well because of its capability to extract high level features. The Multi-distributed Variational AutoEncoder (MVAE) is proposed in [22], it samples more distinguishable latent feature to improve the accuracy in detecting intrusions. However, this method is a binary classification method that detects both normal and abnormal instructions and cannot be used to detect classes of attacks. In [24], the CNN is introduced for the task of network intrusion detection. They convert the raw data into the image format to fit the CNN inputs. The authors in [20] propose a fuzzy aggregation approach using the modified density peak clustering algorithm (MDPCA) and deep belief networks (DBNs). MDPCA is used to divide the training set into several subsets with similar sets of attributes. Then, each subset is used to train its own sub-DBNs classifier. Finally, the output of all sub-DBNs classifiers is aggregated based on fuzzy membership weights. In [23], the authors propose an effective self-taught learning (STL)-IDS method based on the STL framework. After the training stage, the features extracted by STL-IDS are fed into the SVM to improve its detection capability for intrusion and classification accuracy. In [28], the authors propose a deep learning approach for intrusion detection using recurrent neural networks (RNN-IDS) and investigate the effect of the number of neurons on the performance of the model. In [27], the authors propose a deep learning approach for intrusion detection using a multi-convolutional neural network (multi-CNN) fusion method, which divide the features into four parts and introduced CNN to detect intrusion attack.

In recent years, Transformer [16] and its variants [30, 31] has been widely used in the field of NLP and Computer Vision (CV) due to its attention mechanism and powerful learning capabilities. Its success is mainly attributed to its powerful self-attention mechanism and cross-attention mechanism. Although the Transformer abandons the traditional CNN [18, 19, 32] and RNN [14, 15] design, the whole network structure is completely composed of the attention mechanism, and the whole encoder is required to input the decoder, which increases the distance from the encoder to the decoder in the forward process. In this paper, we investigate the unification of encoder and decoder into separate layers and stacking them consecutively. The proposed method not only shortens the distance from the encoder to the decoder, and enables the network information to be converted between the self-attention and the cross-attention, but also captures the long-term

dependencies between features and the ability to suppress noise. Therefore, our motivation is to stack the encoder-encoder modules densely (Denseformer) and using Attention mechanism in each sublayer for features extracting and instruction detection.

3. METHODS

In the cyberattack detection task, the data features change greatly and are accompanied by a lot of noise. The correlation between the features is difficult to model, which directly affect the detection performance of the model. In order to better capture the correlation between features, Denseformer is proposed to realize the gradual extraction of data features by stacking multiple attention layers. The stacking of multiple layers of attention can capture and model long-range dependencies between features of intrusion data, thereby improving the performance of cyberattack detection. In this section, we first introduce dense attention layers suitable for cyberattack detection tasks, and then introduce the overall architecture of Denseformer.

3.1 Dense Attention Layer

The attention layer in Transformer is designed for language models, making it unable to adapt to the characteristics of cyberattack detection data. In order to stack the attention in multiple layers, it is necessary to slightly modify the Transformer structure so that the information flows to the encoder and decoder respectively. Here, let's review the Transformer structure. It is an encoder and decoder structure. Each encoder and decoder part has multiple attention layers stacked. Each attention module contains two layers of structure, one is the Multi-Head Attention (MHA) layer, and the other is the feedforward layer. Unlike the encoder, the decoder contains an additional layer of cross-attention mechanism to process the output from the encoder layer. Therefore, it has a three-part structure: self-attention sublayer, cross-attention sublayer and feedforward layer. Fig. 1(a) shows the overall structure of dense attention layer, where the attention mechanism can be expressed as

$$Att(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where Q, K, and V are learnable query, key, and value matrices respectively. d is the dimension of the input vector.

Performing atention operations on multiple inputs at the same time will result in an MHA mechanism

$$\begin{aligned} &MHA(Q,K,V) = [H_1,H_2,...,H_h]W_{Att}, \\ &where \quad H_i = Att(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \tag{2}$$

where W_i^Q, W_i^K, W_i^V and W_{Att} are learnable projection matrices, and H_i represents the ith attention head. Note that unlike self-attention, the key and value of cross-attention in the decoder come from the output of the encoder. Then, we can stack the encoder-decoder layer to build the Denseformer.

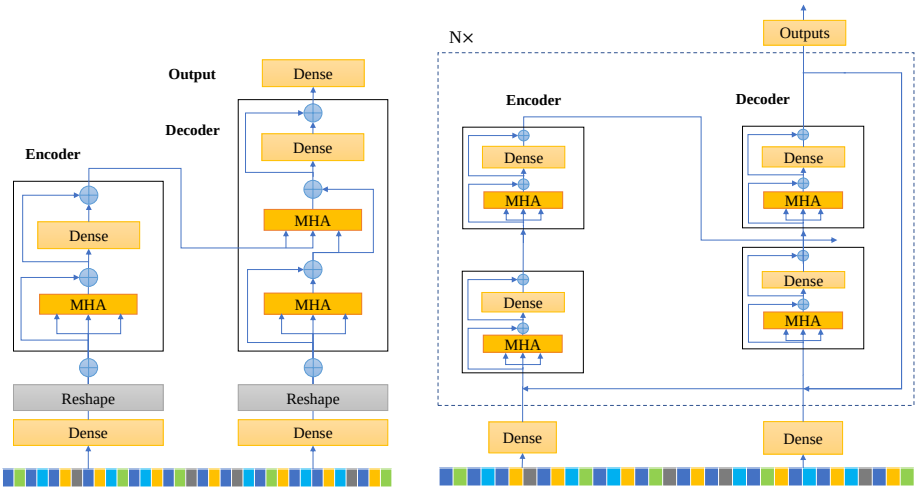


Fig. 1. (a) Dense Attention Layer; (b) Denseformer network architecture with stacked multiple encoder-decoder layers.

3.2 Denseformer

In order to better process the cyberattack detection data, we densely stack the multi-layer encoder and decoder structure, each layer of the encoder and decoder appearing in pairs form a dense attention layers. Therefore, Denseformer is a stacked multiple dense attention layers. The overall structure of Denseformer is shown in Fig. 1 (b), which consists of a stacked multi-layer encoder-decoder structure. Before the data flows to the encoder and decoder, the data needs to be reshaped. Here, we first perform a dense connection operation on it, and then reshape the output tensor to obtain the input of the encoder and decoder respectively. The structure of encoder and decoder is similar to Transformer. The difference is that Transformer is a network structure designed for natural language. For more general data, the data needs to be preprocessed to have sequences and tokens structure. The dense attention layer is divided into two branches from the data input part, and reshape them respectively to adapt to the sequences and tokens structure. The structure can be formalized as

$$X_i = Reshape(\sigma(W_i X_{in})) \quad i = 1,2, \tag{3}$$

where σ is the Relu [33] activation function, X_{in} is input data and X_i is the output feature of the ith branch. After processing the input data, the data will flow to the encoder-decoder structure. The encoder layer E can be formulated as

$$\begin{aligned} Q,K,V &= W_q X_1, W_k X_1, W_v X_1 \\ E_{SA} &= SA(Q,K,V), \\ E_{out1} &= W_e(E_{SA} + X_1) + E_{SA}, \end{aligned} \tag{4}$$

where SA stands for self-attention. Replacing X_1 in Eq. (4) with E_{out1} will result in the final output of encoder E_{out2}. Moreover, the decoder D can be formulated as

$$\begin{aligned} Q,K,V &= W_q X_2, W_k X_2, W_v X_2 \\ D_{SA} &= SA(Q,K,V), \\ D_{out1} &= W_{d1}(D_{SA}+X_1)+D_{SA}. \\ D_{CA} &= CA(W_q E_{out2}, W_k E_{out2}, W_v D_{out1}), \\ D_{out2} &= W_{d2}(D_{CA}+X_1)+D_{CA}, \end{aligned} \tag{5}$$

where CA stands for cross-attention. Then, the encoder and decoder structures appearing in pairs at each layer are defined as $T_i(i=1,2,...)$. The structure of T_i can be formalized as

$$T(X_1,X_2) = D(E(X_1),X_2), \tag{6}$$

Therefore, the whole structure of Denseformer can be expressed as: First, the information of the encoder in T_i will flow to the decoder. The input features X_1 first flow to the encoder part of F_1^e, and the features X_2 flow to the decoder part of F_1^d. Then, the Denseformer of T_1, T_2, and T_3 are connected in sequence. Finally, the final results are output F_{out} through the dense layer. The whole process can be formalized as

$$\begin{aligned} F_1^e, F_1^d &= T_1(X_1,X_2), \\ F_e^2, F_d^2 &= T_2(F_e^1, F_d^1), \\ F_e^3, F_d^3 &= T_3(F_e^2, F_d^2), \\ F_{out} &= W F_d^3, \end{aligned} \tag{7}$$

where W is the weight of the last layer. The encoder-decoder structure in Denseformer makes it more suitable for datasets with large noise levels like NSL-KDD. The encoder of each layer is used to extract coarse-grained features, and the decoder is used for refining these features to obtain fine-grained features. After the above process, Denseformer can effectively process cyberattack detection data through the stacked attention mechanism of multi-layer encoder-decoder.

4. DATASETS AND PROCESSING

4.1 Datasets

KDD-CUP99: This dataset is 9 weeks of network connection data collected from a simulated US Air Force LAN, which divided into marked training data and unmarked test data. The test data and training data have different probability distributions. It contains some types of attacks that do not appear in the training data, which makes cyberattack detection more realistic. The training dataset contains 1 normal identification type and 22 training attack types. In addition, 14 kinds of attacks only appeared in the test dataset.

NSL-KDD: NSL-KDD has solved the data redundancy problems in the KDD99 dataset. It can be used as an effective benchmark dataset to help researchers compare different intrusion detection methods. The NSL-KDD training set and test set are reasonable,

and the evaluation results of different research work will be consistent and comparable. Table 1 describes the sample distribution of this dataset.

Table 1. The detial of the NSL-KDD datasets.

	Total	Normal	Dos	Probe	R2L	U2L
NSL-KDD Train+	125973	67343	45927	11656	995	52
NSL-KDD Test+	22544	9711	7458	2421	2754	200
NSL-KDD Test-21	11850	2152	4342	2402	2754	200

4.2 Data Processing

Because the attributes of each column of NSL-KDD are quite different, three of them are strings. Therefore, we first perform one-hot encoding on these strings, then perform log operations on all attribute values $x = log(x+1)$. Finally, the Gaussian normalization is performed on each attribute. The whole process is as $X_i = \frac{X_i - \mu}{\delta}$, where μ and δ are the mean and variance of each attribute in the training set, respectively.

5. EXPERIMENTS

Hyperparameter Settings: The hyperparameters are setted based on the performance on validation set. All the networks use the Pytorch framework, and trained on one NVIDIA Tesla V100 GPU using Adam optimization method [34]. The batch size on each GPU is set to 128 for 200 epochs. The initial learning rate is set to 0.1.

Network Settings: For Denseformer, the initial dimension of dense layer are setted to 128, and the input size of the attention module is 8.

5.1 Evaluation Metric

In our experiment, multiple commonly metrics are used for evaluation[1]

- Accuracy

$$Acc = \frac{TP+TN}{TP+FP+FN+TN}. \tag{8}$$

- Recall Socre

$$Rec = \sum_i W_i \frac{TP_i}{TP_i+FN_i}. \tag{9}$$

where the index i is the subscript of the classes and W_i is the weight of each class.

- F1 Socre

$$F1 = \frac{2TP}{2TP+FN+FP}. \tag{10}$$

[1]In this list of items, TP, TN, FP, FN are the abbreviations for True Positive, True Negatives, False Positive, False Negative, respectively.

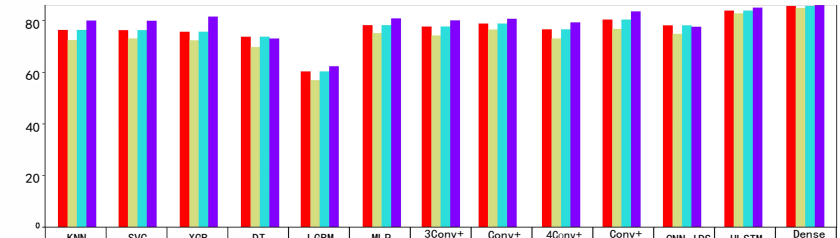
- Precision

$$Precision = \frac{TP}{TP+FP}. \tag{11}$$

Since cyberattack detection is a multi-classification problem, we use weighted Recall and F1 socre as the performance indicators of the model. The weighted average Recall is an improvement of macro average Recall, so we can better measure the coverage of the sample and the proportion of true positive samples in the positive samples.

5.2 Performance on NSL-KDD Test+

In the experiments of this paper, the comparison models used are K-Nearest Neighbor (KNN), Support Vector Classification (SVC), XGBoost (XGB), Decision Tree (DT), Light Gradient Boosting Machine (LGBM), Multi-Layer Perceptron (MLP) are implemented by scikit-learn machine learning library. The 3Conv+BiLSTM, Conv+LSTM, 4Conv+2FC and Conv+2FC are implemented as [22, 25], the CNN-IDS are implemented as [26], and the HLSTM are implemented as [11].



	KNN	SVC	XGB	DT	LGBM	MLP	3Conv+ BiLSTM	Conv+ LSTM	4Conv+ 2FC	Conv+ 2FC	CNN-IDS	HLSTM	Dense former
Accuracy	76. 38	76. 31	75. 64	73. 72	60. 29	78. 22	77. 66	78. 84	76. 55	80. 4	78. 14	83. 88	85. 64
F1 Score	72. 44	73. 04	72. 31	69. 75	56. 86	75. 1	74. 17	76. 53	73. 04	76. 71	74. 82	82. 73	84. 89
Recall	76. 38	76. 31	75. 64	73. 72	60. 29	78. 22	77. 66	78. 84	76. 55	80. 4	78. 14	83. 88	85. 64
Precision	79. 96	79. 91	81. 55	73. 01	62. 28	80. 86	80. 1	80. 67	79. 3	83. 56	77. 61	84. 97	85. 97

Fig. 2. Comparison between different models on NSL-KDD test+ dataset.

We use different networks to conduct experiments on two datasets: NSL-KDD Test+ and NSL-KDD Test-21. The experimental results are shown in Figs. 2 and 3. It can be found that for different indicators, such as Accuracy, Recall and F1 score and Precision, Denseformer has shown good generalization performance on these indicators.

On the NSL-KDD Test+ dataset, Denseformer achieves high performance on the all indicators respectively. In terms of Accuracy, Recall and F1 metrics, the performance of Denseformer has improved significantly. In terms of Accuracy metrics, Denseformer has 2% and 9% performance improvements compared to HLSTM [11] and CNN-IDS [26].

Compared with other methods in [25], Denseformer has a significant improvement in various performance indicators. Among all traditional machine learning methods, the LGBM method achieved the worst results, and the SVC method and MLP method achieved higher accuracy of 76.31% and 78.22%, respectively. The xgboost embedding learning XGB method is only 1.92% higher in accuracy than the DT. In the deep learning method, Conv+2FC and HLSTM have achieved an accuracy of more than 80%, and Denseformer has a large margin gains compare to these methods.

Table 2 compares other advanced deep learning methods. The accuracy of Denseformer has exceeded the current most advanced deep learning methods. For example, in NSL-KDD Test+, Denseformer exceeds BAT-MC [12] by approximately 1.49% points. This model uses multi-layer convolution and BiLSTM. Compared with the DBN+MDPCA [20] model that uses deep belief networks and density clustering methods, Denseformer has a 4.3% performance improvement. In particular, Denseformer outperforms Variational AutoEncoder [29] with a whole encoder-decoder structure by about 5.16%.

Table 2. Performance comparison between Denseformer and other models.

Dataset	Model	Accuracy (%)
NSL-KDD Test+	DNN-5 [21]	78.50
	AE-RL [24]	80.16
	STL+SVM [23]	80.48
	VAE [29]	81.13
	RNN-IDS [28]	81.29
	Multi-CNN [27]	81.33
	DBN+MDPCA [20]	82.08
	Transformer [16]	83.09
	BAT-MC [12]	84.15
	Denseformer	**85.64**
NSL-KDD Test-21	VAE [29]	64.30
	RNN-IDS [28]	64.67
	Multi-CNN [27]	64.81
	DBN+MDPCA [20]	66.18
	Transformer [16]	67.97
	BAT-MC [12]	69.42
	Denseformer	**72.95**

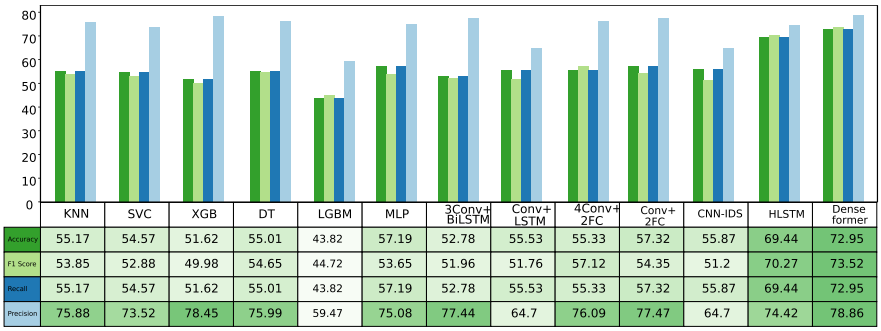
5.3 Performance on NSL-KDD Test-21

On the NSL-KDD Test-21 dataset, Denseformer achieves the best results in all indicators. Denseformer is better than the performance of HLSTM, and greatly exceed other models by a large margin, including shallow learning methods and deep learning methods. For example, the shallow methods such as SVC, XGB and CNN-IDS [26], as well as a hybrid of CNN+LSTM [22, 25].

Table 2 also shows the performance of various methods on the NSL-KDD Test-21 datasets. It can be seen that Denseformer surpasses other state-of-the-art methods. Denseformer can capture the global information between the local and global position, which is beneficial to the improvement of performance. It is worth noting that Denseformer use encoder-decoder with attention mechanism between multiple layers, which means that the network architecture is very effective for network intrusion detection tasks. In particular, Denseformer outperforms Variational AutoEncoder [29] with a whole encoder-decoder structure by about 8.65%.

Compared with the current state-of-the-art models, Denseformer has 2.64%, 2.70%, 2.64% and 2.72% average improvement in four metrics such as Accuracy, Recall, F1 and

precision, which means that it can significantly increase the detection rate of cyberattacks. This method can prevent security problems such as data and privacy leakage, server failure, *etc*., and greatly improve the security guarantee that is urgently needed in network environments such as the Internet, the Internet of Things, and cloud-edge computing.



	KNN	SVC	XGB	DT	LGBM	MLP	3Conv+ BiLSTM	Conv+ LSTM	4Conv+ 2FC	Conv+ 2FC	CNN-IDS	HLSTM	Dense former
Accuracy	55.17	54.57	51.62	55.01	43.82	57.19	52.78	55.53	55.33	57.32	55.87	69.44	72.95
F1 Score	53.85	52.88	49.98	54.65	44.72	53.65	51.96	51.76	57.12	54.35	51.2	70.27	73.52
Recall	55.17	54.57	51.62	55.01	43.82	57.19	52.78	55.53	55.33	57.32	55.87	69.44	72.95
Precision	75.88	73.52	78.45	75.99	59.47	75.08	77.44	64.7	76.09	77.47	64.7	74.42	78.86

Fig. 3. Comparison between different models on NSL-KDD test-21 dataset.

Table 3. Ablation experiments with denseformer layers replaced by other layers.

Model	Conv		Dense		LSTM		Denseformer	
Dataset	Test+	Test-21	Test+	Test-21	Test+	Test-21	Test+	Test-21
Accuracy	78.14	58.69	82.51	67.42	83.88	70.39	**85.64**	**72.95**
F1 Score	74.82	56.50	81.86	69.08	82.73	69.14	**84.89**	**73.52**
Recall	78.14	58.69	82.51	67.42	83.88	70.39	**85.64**	**72.95**
Precision	77.61	66.52	83.98	77.40	84.97	75.15	**85.97**	**78.86**

5.4 Ablation Experiments

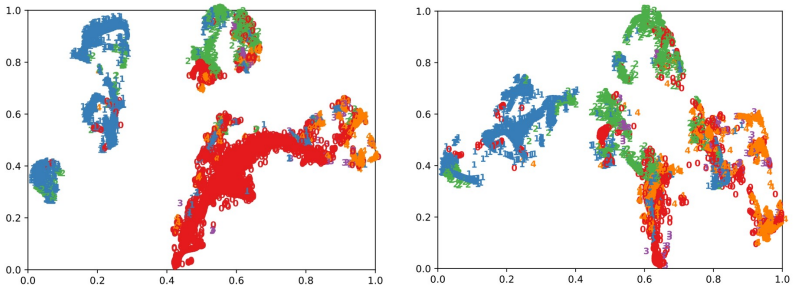
To verify the effectiveness and generalization of Denseformer, we did ablation experiments on replacing the Denseformer layer with other layers and the depth of the Denseformer, respectively. Table 3 shows that Denseformer was still able to outperform Conv (Convolution), Dense (Full Connection) and LSTM layers on the NSL-KDD Test+ and NSL-KDD Test-21 datasets, demonstrating the effectiveness of the Denseformer layer.

Table 3 also shows that if we replace the encoder or decoder in Desneformer with other layers, such as convolution, Dense or LSTM, it will lead to performance degradation. This shows that using an encoder with a self-attention mechanism to extract features and using a decoder with cross-attention to refine features is very important for intrusion detection tasks. The encoder-decoder structure with attention mechanism greatly reduces overfitting to noise, which is also the key to improving performance in intrusion detection tasks.

Table 4 shows that as the number of layers increases, the performance of Denseformer gradually gets better. However, the performance decreases when the number of layers exceeds 3, and the highest accuracy metric can be achieved at a depth of 4. This indicates that both underfitting and overfitting of the model affect its performance, and that the same hyperparameters do not consistently achieve the best performance on all metrics.

Table 4. An ablation study on the depth of denseformer layer.

Depth	1		2		3		4		5	
Dataset	Test+	Test-21	Test+	Test-21	Test+	Test-21	Test+	Test-21	Test+	Test-21
Accuracy	82.54	67.07	84.76	71.73	**85.64**	**72.95**	85.57	72.87	84.36	70.76
F1 Score	80.42	66.91	83.80	72.05	**84.89**	**73.52**	84.66	73.55	82.92	70.74
Recall	82.54	67.07	84.76	71.73	**85.64**	**72.95**	85.57	72.87	84.36	70.76
Precision	84.15	79.64	85.19	78.10	85.97	78.86	**86.49**	**80.77**	84.62	78.09



(a) NSL-KDD Test+ (b) NSL-KDD Test-21

Fig. 4. The T-SNE visualization of the denseformer on the NSL-KDD Test datasets; The numbers 0, 1, 2, 3, and 4 indicate 'Normal', 'DoS', 'Probe', 'U2R', 'R2L'.

5.5 Visualization and Analysis

Figs. 4 (a) and (b) respectively describe the T-SNE visualization of the denseformer model on the two test sets. It can be seen that the number of normal categories is much larger than the number of other categories, and there is a greater degree of confusion between the normal category and the R2L category. The Probe category is relatively independent, but it can be confused with other categories, such as the DoS category. On the whole, the samples of each category are obviously unbalanced, and each category is confused with each other.

Fig. 5 depicts the confusion matrix of Denseformer on the two test sets. This model is relatively poor in U2R and R2L attack categories, but the detection success rate in other categories is relatively high. Our analysis is that the stacked attention mechanism of Denseformer increases this part of the penalty, which increases the overall model accuracy.

6. CONCLUSION

In this paper, we study a stacked dense attention network named Denseformer, which unifying the encoder and decoder into separate layers and stacking them consecutively. In order to further enhance the generalization performance of attention layer in cyberattack detection task, this paper proposes a dense attention with stacked multi-layer attention mechanisms. This method not only shortens the distance from the encoder to the decoder, and enables the network information to be converted between self-attention

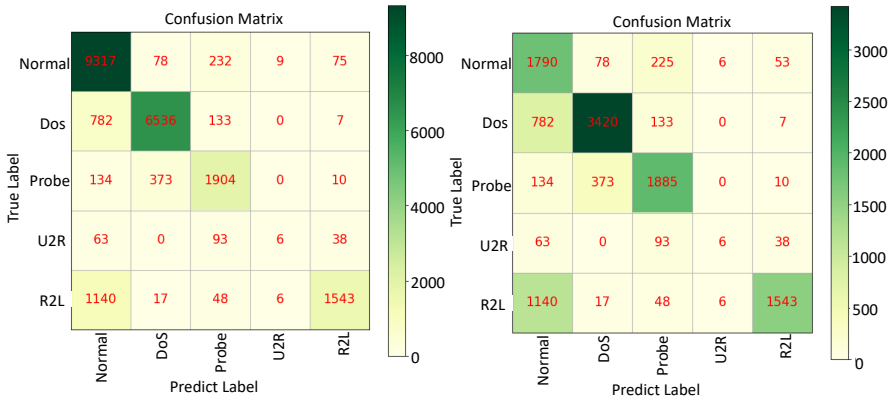


Fig. 5. The confusion matrices of the Denseformer on the NSL-KDD Test+ and NSL-KDD Test-21 datasets.

and cross-attention, but also captures the long-term dependencies between features and the ability to suppress noise. This model also realizes the embedding of the multi-layer encoder-decoder model by stacking the dense attention layers. By using the attention mechanism of multi-layer encoders and decoders, the model can capture the relationship between cyberattack data features, thereby improving model performance. Experimental results show that Denseformer outperforms other models in various performance indicators, which proves that Denseformer model has better generalization performance on cyberattack detection tasks.

ACKNOWLEDGMENT

This work was supported in part by the Project of International Cooperation and Exchanges NSFC under Grant No. 61860206005, the National Natural Science Foundation of China under Grant No. 62271288, the National Natural Science Foundation of China under Grant No. 2021B1515120066 and the Qingdao Agricultural University Doctoral Start-Up Fund under Grant No. 6631122006.

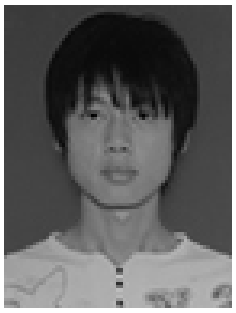
REFERENCES

1. Y. Tang and S. Chen, "An automated signature-based approach against polymorphic internet worms," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, 2007, pp. 879-892.
2. F. Chen, Z. Ye, C. Wang, L. Yan, and R. Wang, "A feature selection approach for network intrusion detection based on tree-seed algorithm and k-nearest neighbor," in *Proceedings of IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems*, 2018, pp. 68-72.

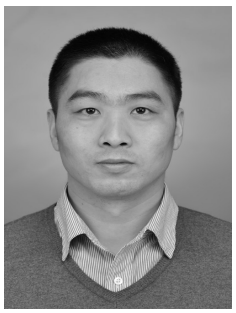
3. X. Qu, L. Yang, K. Guo, L. Ma, M. Sun, M. Ke, and M. Li, "A survey on the development of self-organizing maps for unsupervised intrusion detection," *Mobile Networks and Applications*, 2019, pp. 1-22.
4. J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, Vol. 28, 2017, pp. 1051-1058.
5. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, Vol. 16, 2002, pp. 321-357.
6. I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of Workshop on Learning from Imbalanced Datasets*, Vol. 126, 2003, pp. 1-7.
7. H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of IEEE International Joint Conference on Neural Betworks*, 2008, 1322-1328.
8. N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Proceedings of IEEE 2nd International Conference on Advanced Cloud and Big Data*, 2014, pp. 247-252.
9. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 2, 2018, pp. 41-50.
10. Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Proceedings of International Conference on Neural Information Processing*, 2017, pp. 858-866.
11. H. Hou, Y. Xu, M. Chen, Z. Liu, W. Guo, M. Gao, Y. Xin, and L. Cui, "Hierarchical long short-term memory network for cyberattack detection," *IEEE Access*, Vol. 8, 2020, pp. 90907-90913.
12. T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, Vol. 8, 2020, pp. 29575-29585.
13. P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE Transactions on Information Theory*, Vol. 14, 1968, pp. 515-516.
14. J. L. Elman, "Finding structure in time," *Cognitive Science*, Vol. 14, 1990, pp. 179-211.
15. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, 1997, pp. 1735-1780.
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998-6008.
17. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, 1998, pp. 2278-2324.
18. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

19. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700-4708.
20. Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Applied Sciences*, Vol. 9, 2019, p. 238.
21. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, Vol. 7, 2019, pp. 41525-41550.
22. L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz *et al.*, "Learning latent distribution for distinguishing network traffic in intrusion detection system," in *Proceedings of IEEE International Conference on Communications*, 2019, pp. 1-6.
23. M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, Vol. 6, 2018, pp. 52843-52856.
24. G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, Vol. 159, 2019, pp. 96-109.
25. R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proceedings of International Conference on Advances in Computing, Communications and Informatics*, 2017, pp. 1222-1228.
26. K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, Vol. 6, 2018, pp. 50850-50859.
27. Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial iot based on multi-cnn fusion," *Measurement*, Vol. 154, 2020, p. 107450.
28. C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, Vol. 5, 2017, pp. 21954-21961.
29. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.
30. S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv Preprint*, 2020, arXiv:2006.04768.
31. K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser *et al.*, "Rethinking attention with performers," *arXiv Preprint*, 2020, arXiv:2009.14794.
32. D. Liang, F. Yang, T. Zhang, J. Tian, and P. Yang, "Wpnets and pwnets: from the perspective of channel fusion," *IEEE Access*, Vol. 6, 2018, pp. 34226-34236.
33. X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2012, pp. – –.
34. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Preprint*, 2014, arXiv:1412.6980.

Haixia Hou received the Master degree in Electronics and Communication Engineering at Shangdong University in 2004. She is currently working toward the Ph.D. degree in Information Security at Beijing University of Post and Telecommunications. Her main research interests include information security, user cross-domain behavior analysis, and network security.



Daojun Liang received the BE degree in Computer Science from TaiShan University, China, in 2016, and the MS degree in the School of Information Science and Engineering from the Shandong Normal University. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Shandong University, China. Her research interests include deep learning, computer vision.



Mingqiang Zhang received MS and Ph.D. degrees from Shandong University, China, in 2004 and 2022, respectively. He is currently a Lecturer in School of Cyber Science and Engineering, Qufu Normal University, China. His current research interests include intelligent communication, industrial Internet of Things (IIoT), artificial intelligence and edge computing.

Dongfeng Yuan received the MS degree from the Department of Electrical Engineering, Shandong University, China, in 1988, and the Ph.D. degree from the Department of Electrical Engineering, Tsinghua University, China, in 2000. From 1993 to 1994, he was with the Electrical and Computer Department, University of Calgary, AB, Canada. He was with the Department of Electrical Engineering, University of Erlangen, Germany, from 1998 to 1999, with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA, from 2001 to 2002, with the Department of Electrical Engineering, Munich University of Technology, Germany, in 2005, and with the Department of Electrical Engineering Heriot-Watt University, U.K., in 2006. He is currently a Full Professor with Shandong Key Laboratory of Wireless Mobile Communication Technologies, Shandong University. His current research interests include intellegent communication systems, mobile edge computing and cloud computing, AI and big data processing for communications.