

A malicious traffic detection algorithm based on the combination of traffic statistical feature and BERT text feature

HongPeng Wang¹, YingMing Zeng¹, Jia Hu¹, KaiWei Kong¹ and LinLin Zhang¹.

¹ INSTITUTE 706, THE SECOND ACADEMY OF CHINA AEROSPACE SCIENCE & INDUSTRY CORP.

Abstract. Nowadays, most malicious traffic detection algorithms are based on statistical characteristics for analysis. However, as the behaviors of malicious traffic are constantly evolving, attackers are continually refining their techniques to evade these statistical feature-based detection algorithms. In today's complex network environment, relying solely on statistical features to detect malicious traffic may not be able to identify all malicious traffic. Therefore, this paper proposes a classification detection method that integrates statistical features with BERT text features as the training and testing features of the classification model. The classification model utilizes variational autoencoders to capture malicious traffic's latent patterns and uncommon features. Experimental results show that the proposed method in this paper can classify malicious traffic with an accuracy of 99%, which is significantly better than other malicious traffic detection algorithms. The proposed method combines mixed features with probabilistic modeling, significantly improving the accuracy of detecting malicious traffic and enabling early detection and prevention of potential network attacks and threats.

Keywords: variational autoencoder, network malicious traffic detection, deep learning, BERT.

1 Introduction

As network technology continuously develops, the risk of network security is increasing. According to a security report [1] by Checkpoint Software, global cyber-attacks against enterprises have increased by 29%. Among them, US enterprises suffered an average of 443 attacks per week, and the Asia-Pacific region suffered an average of 1,338 attacks per week. Real-time detection and analysis of network traffic through effective malicious traffic detection algorithms can help identify and prevent potential network attacks and threats, reduce losses, and improve network reliability and stability.

Nowadays, research on malicious traffic detection mainly focuses on several directions, such as statistical analysis, machine learning, deep learning, unsupervised learning, and association analysis [2].

For research on detection methods based on statistical analysis, Rousseeuw et al. [3] used the 3σ model to model statistical feature in network traffic. The model first calculates the mean and variance of historical data. Using the range interval of three standard deviations above and below the mean as the value range for normal traffic indicators, any values outside this range are considered anomalies. The advantage of this method is that it is simple to calculate. However, it is only suitable for scenarios where traffic changes are relatively stable, its accuracy may significantly decrease, and if there are extreme outliers in the historical data used to calculate the mean and variance, the detection results may exhibit significant bias.

For research on machine learning-based detection methods based on the traditional oversampling SMOTE algorithm, Liang et al. [4] proposed the LR-SMOTE algorithm, which combines K-means and SVM methods to make the newly generated samples close to the sample center, to avoid the generation of outlier samples. However, it is generally easy to obtain normal traffic in actual situations. However, it is challenging to obtain malicious traffic, leading to data imbalance between normal traffic and malicious traffic, data distribution, and data noise problems. Data appears in the form of streams, and the data distribution changes over time. The instability of the network may result in indistinct feature distinctions between benign and malicious traffic samples. Moreover, during the oversampling process, the superposition of noise in a small sample data will be aggravated, affecting the detection effect.

For research on Deep learning, DONG et al. [5] increase the depth of the network model by stacking multiple attention modules connected by residual modules and, at the same time, introduce neural networks, pooling layers, batch normalization layers, and activation function layers in the attention module to prevent the model from Overfitting and improving model performance, and finally obtaining the output vector through the DNN model. Although decent results can be achieved, the model complexity requires much training data and computational resources, leading to longer training cycles.

For research on unsupervised clustering method. Pu et al. [6] adopted a hybrid approach based on a clustering algorithm for anomaly detection. The clustering algorithm clusters the data, resulting in two clusters. The cluster with a small amount of data is called an abnormal cluster. When new traffic data is observed, calculate the cluster to which the data belongs and determine whether the traffic data is abnormal. Clustering algorithms rely more on the distance measurement between data. If the distance measurement between data cannot reflect the similarity between data well, the model based on the clustering algorithm will have more false positives and false negatives.

Research on detection methods of association analysis. A method is proposed in the literature [7]. The characteristics of network attack behavior are extracted, the correlation between different features is calculated by association analysis, the entropy is calculated, and the abnormal traffic is identified by comparing the entropy. However, the association analysis algorithm has a disadvantage because it may not accurately capture the association rules between some features.

Although the method introduced above uses machine learning, deep learning, and other algorithms to build classifiers, the traffic features used for model training and prediction still have the problem that the features of the extracted benign traffic samples

are not significantly different from those of malicious traffic samples, it is difficult to capture the complex relationship between the features, and the content covered by the features is single. In order to solve the above problem, this paper proposes a variational autoencoder malicious traffic detection method that integrates statistical features and BERT word vector features. The method combines multi-angle statistical features with strong explanatory features and BERT features containing specific network association information, effectively increasing the diversity of feature information. Moreover, the variational autoencoder with a strong ability to learn the potential features of data is used as the classification model to improve the classification accuracy effectively.

2 The Traffic Detection Approach

This section will provide a detailed overview of the design of a malicious traffic detection algorithm based on the combination of traffic statistical features and BERT text features. Step one involves proposing a feature extraction method that integrates statistical features with BERT text features. The extracted traffic features include not only the multi-angle, comprehensive information, and strong interpretability of statistical features but also contain the information in the URL path of the header request line of the HTTP request message extracted by the BERT model, such as protocol type, host-name, port number, path, query parameters, and anchors. The fusion of traffic statistical features with BERT text features forms hybrid features containing rich information, enabling a more comprehensive understanding and expression of the data. Step 2: Model training. The variational autoencoder (VAE) model is selected to train and classify the characteristics of malicious traffic. Compared to traditional machine learning and deep learning models, the VAE model possesses several advantages. First, the variational autoencoder is an unsupervised learning model that can learn the latent distribution features from data without manually annotating label data. In addition, in the detection of malicious traffic, since the types of malicious traffic are numerous and constantly changing, the use of unsupervised learning models can better adapt to the detection requirements of new types of malicious traffic. In addition, the VAE model can learn latent feature representations of data, map input data to a lower-dimensional latent space, and extract effective feature information for malicious traffic, and this aids in identifying hidden patterns and abnormal features in malicious traffic data, enabling effective recognition and detection of malicious traffic. Finally, the VAE model exhibits a certain degree of robustness to noise variations, capable of handling confounding changes in malicious traffic to some extent.

Fig. 1 illustrates the overall workflow of the proposed method:

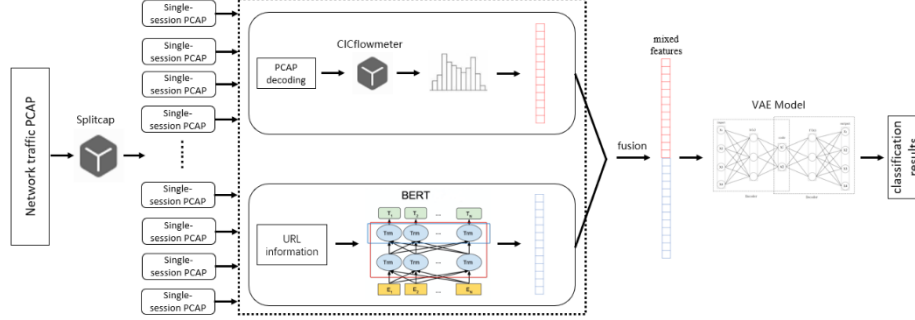


Fig. 1. The flowchart of the methodology in this paper.

2.1 Feature Extraction

Traffic feature extraction is essential for malicious traffic detection. Extracting information-rich, interpretable, and highly distinguishable traffic features can enable the model to identify malicious traffic better [8]. Because statistical features may produce false positives or false negatives due to Fluctuations in the network environment, in order to enable the extracted features to fully express the critical information of network traffic, distinguish different types of network traffic, abnormal behaviors in the traffic, and be specific to different network environments and networks Fluctuations, have a certain degree of robustness. Hence, this paper combines statistical features with BERT text features in the feature extraction.

The process of feature extraction.

Step 1: Parse the PCAP packet byte stream according to the network session and extract and calculate the traffic statistical characteristics based on the byte size. The analyzed statistical characteristics include the minimum forward traffic load, average forward traffic byte, minimum reverse traffic byte value, standard deviation of forward packet header bytes, minimum reverse packet header byte value, minimum flow packet interval, standard deviation of flow packet intervals, SYN, ECE, CWR, FIN, RST, PSN, ACK, URG, PSN, and reverse TCP initial window byte count.

Step 2: Extract the URL information of the request line in the header line of the HTTP request message corresponding to the session in the PCAP package and encode the URL with the BERT model. Because the URL path includes information such as hostname, port number, path, query parameters, etc. When analyzing HTTP URLs, it is essential to consider the context to incorporate semantic information into the generated feature vectors. Based on the Transformer encoder-decoder architecture, the BERT model employs multi-layer self-attention mechanisms to capture dependencies within input text, enabling it to consider both preceding and subsequent contexts of a word and better comprehend the context [9]. This method uses the BERT model trained with more layers, hidden units, and parameters to generate feature vectors from the extracted URL data. Due to the large output vector dimensions of the BERT model, problems

such as large computing resource requirements, overfitting of model training, and high sparsity of training data may occur during model training. Therefore, the high-dimensional vectors extracted by BERT need to be turned into low-dimensional vectors. Discrete Cosine Transform (DCT) is a mathematical transformation method that converts signals or images from the spatial domain to the frequency domain [10]. It has found widespread applications in signal and image processing, particularly in data compression (such as JPEG image compression and MP3 audio compression). One-dimensional Discrete Cosine Transform (DCT) transforms a one-dimensional signal x of length N into a DCT transformation, and its core idea can be represented as Equation (1) (where $x=1, 2, 3, \dots, N-1$).

$$f(x) = \sum_{\eta=0}^{N-1} F_c(\eta)C(x, \eta) \tag{1}$$

transform kernel $C(x, \eta)$ as shown in equation (2):

$$C(x, \mu) = \alpha(\mu)\cos\left[\frac{(2x + 1)\mu\pi}{2N}\right] \tag{2}$$

Normalization Parameter $\alpha(\mu)$ as shown in equation (3):

$$\alpha(\mu) = \begin{cases} \sqrt{\frac{1}{N}}, \mu = 0 \\ \sqrt{\frac{2}{N}}, \mu \neq 0 \end{cases} \tag{3}$$

Using one-dimensional Discrete Cosine Transform reduces the dimensionality of BERT-generated one-dimensional word vectors from 1800 to 100 dimensions.

Step 3: involves concatenating the traffic statistical features from step one with the BERT traffic features obtained in step two, generating new traffic features to be used as the training and testing data for the model.

2.2 VAE Detection Model

Network traffic has the characteristics of large data volume, high latitude, and complex internal relationships. Traditional statistical methods struggle to provide high accuracy and efficiency in analyzing and processing such data. VAE differs from existing statistical analysis methods in that they can efficiently perform inference from training samples to prediction samples, greatly simplifying classification and regression problems while maintaining robustness.

The VAE model framework consists of an encoder (Encoder) and a decoder (Decoder). The encoder maps the input data to a low-dimensional representation in the latent space, and the decoder maps the latent variables back to the original data space [11]. During training, the model adjusts the parameters of the encoder and decoder to

minimize the reconstruction loss, which measures the difference between the input data and the decoded output. By iteratively adjusting these parameters during training, VAE learns to encode the input data into meaningful latent space representations. This optimized latent encoding encapsulates the fundamental features and structure of the data, aiding in accurate reconstruction. The structure of the VAE model framework is illustrated in Fig. 2:

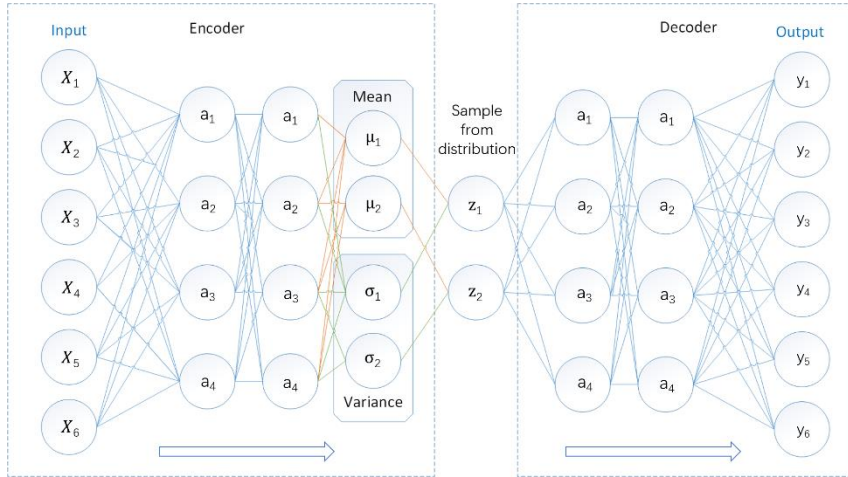


Fig. 2. The flowchart of the methodology in this paper.

The VAE assumes that the benign traffic characteristics obey one probability distribution, and the malicious traffic characteristics obey another probability distribution, with different similarities between the two distributions. By training a VAE probability generative model on a benign traffic training set during the malicious traffic detection process, the VAE model for benign traffic probability distribution can map the probability distribution of the traffic under test to the probability distribution of benign traffic. By comparing the difference in similarity between the generated probability distribution of the traffic under test and the actual probability distribution of benign traffic, it is possible to determine whether the tested traffic data is malicious.

The Model Structure.

The structure of the VAE model is as follows:

(1) Encoder:

Input: Raw 1×176 -dimensional traffic feature data.

Network Structure: Comprised of three fully connected hidden layers. The number of nodes in each layer is 32, 16, and 8, respectively.

Output: Mean and variance features of the latent variables.

(2) The sampling process:

First, calculate the standard deviation through the latent variable's variance, then calculate it based on the mean and standard deviation. Calculate the probability normal

distribution that satisfies the standard deviation and mean from the mean and variance parameters. Ten samples from the normal probability distribution were used to obtain the potential variable. Generate a batch of data samples using the sampled latent variables.

(3) Decoder:

Input: Sampled latent variables.

Network Structure: The reverse of the encoder network structure gradually maps the latent variables back to the original data space.

Output: Reconstructed input data.

(4) Loss Function:

① log probability density: In the given mean and variance, the logarithm of the probability density function can be computed for a certain value under the normal distribution. Generally, a larger logarithm of probability density implies that the original probability value is closer to 1. The logarithm of probability density is shown in equation (4):

$$\log P(x) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(x - \mu)^2}{2\sigma^2} \quad (4)$$

Where μ is the mean of the normal distribution, σ is the standard deviation.

② Kullback-Leibler (KL) divergence loss: Measuring the distance between the latent variables and the prior distribution, prompting the encoder to learn a reasonable latent representation [12]. The formula for KL divergence is shown in equation (5):

$$D_{KL}(p||q) = \sum_{i=1}^N p(x_i) \left(\log \frac{p(x_i)}{q(x_i)} \right) \quad (5)$$

Where x represents the value of the probability distribution, $p(x)$ represents the true probability distribution being approximated or evaluated, $q(x)$ represents the probability distribution used to approximate the true probability distribution p .

The loss function designed in this paper is the KL divergence loss minus the logarithm of the probability density. The better the predictive performance of the model, the smaller the KL divergence and the larger the logarithm of the probability, resulting in a minor difference when these two numbers are subtracted. The formula for Loss function is shown in equation (6):

$$Loss\ function = D_{KL}(p||q) - \log P(x) \quad (6)$$

(5) optimization algorithm: The Adam algorithm combines the advantages of momentum and adaptive learning rates. The Adam algorithm combines the advantages of momentum and adaptive learning rates. During each training iteration, the Adam algorithm computes the gradients for each parameter and updates them using both momentum and adaptive learning rates to achieve efficient model training. This paper selects Adam as the optimization algorithm to achieve efficient model training. The update rules for parameters of the trained model using Adam are shown in equation (7):

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (7)$$

The symbol θ_t represents the current value of the parameter, α denotes the learning rate, \hat{m}_t is the first moment estimate of the gradient (the momentum), \hat{v}_t is the second moment estimate of the gradient (the adaptive learning rate), ϵ is a small number to prevent division by zero. This formula describes the update process of each parameter θ at each time step t .

3 Experiments and Results Analysis

3.1 Experimental Environment

This experimental environment is Windows 11 platform, Intel Corei7, allocated memory 32GB, and graphics card NVIDIA 4070; the experiment uses Python as the programming language and uses the PYTORCH machine learning library to build the VAE model framework.

3.2 Experimental Data

Capture benign traffic packets on the Ethernet interface through the Wireshark tool and extract 39,500 benign traffic packets from the data packets, of which 39,000 pieces of data are used for training and 500 pieces of data are used for testing. Malicious traffic selects the CICIDS2017 network intrusion detection data set created by the Canadian Cyber Security Research Institute. This data set provides a collection of samples of network traffic in the real world. This paper selects benign traffic for training, botnets, brute force cracking, brute-force FTP, brute-force SSH, port scanning, SQL injection, and XSS attacks malicious traffic for testing.

The test experiment compares benign traffic and malicious traffic. The experiment outputs the logarithmic probability density of each traffic sample after sigmoid mapping. By comparing the interval range of the logarithmic probability density of benign traffic samples and malicious traffic samples, the effectiveness of the model classification is judged. If the logarithmic probability density is greater than 0.4, it is judged to be benign traffic, and if the logarithmic probability density is less than 0.4, it is judged to be malicious traffic. Iteratively train for 300 times.

This paper defines the model performance evaluation indicators as TP (Number of malicious traffic correctly detected as malicious by the algorithm), FP (The number of benign traffic detected as malicious by the algorithm), FN (The number of malicious traffic detected as benign traffic by the algorithm), TN (The number of benign traffic detected as benign by the algorithm).

The accuracy rate (the proportion of the total number of correctly predicted benign traffic and malicious traffic). The accuracy rate is shown in equation (8):

$$\text{Accuracy Rate} = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (8)$$

The recall rate (the proportion of malicious traffic detected as malicious traffic by the algorithm to all malicious traffic). The recall rate is shown in equation (9):

$$\text{Recall Rate} = \frac{TP}{TP + FN} * 100\% \quad (9)$$

The false negative rate (the proportion of malicious traffic detected as benign traffic by the algorithm to all malicious traffic). The false negative rate is shown in equation (10):

$$\text{False Negative Rate} = \frac{FN}{TP + FN} * 100\% \quad (10)$$

The false positive rate (the proportion of benign traffic detected as malicious traffic to all benign traffic), The false positive rate is shown in equation (11):

$$\text{False Positive Rate} = \frac{FP}{TN + FP} * 100\% \quad (11)$$

True Positive Rate (the proportion of benign traffic detected as benign traffic) accounts for the amount of all benign traffic), The True Positive Rate is shown in equation (12):

$$\text{True Positive Rate} = \frac{TN}{TN + FP} * 100\% \quad (12)$$

3.3 Experimental Result

First, test the effect of VAE detection using only traffic statistical features for training and testing. Test results are shown in Figures 3 to 8 (the blue dots are benign traffic samples, the red triangles are malicious traffic samples, such as brute force cracking, Brute-Force FTP, Port scanning and so on). The test result statistics are shown in the table 1:

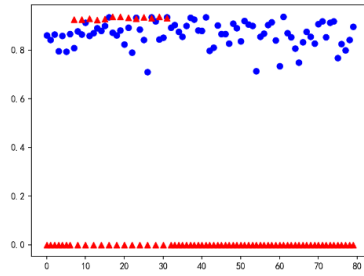


Fig. 3. benign and brute force cracking

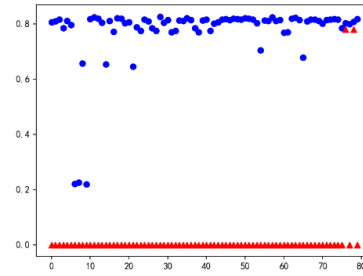


Fig. 4. benign and Brute-Force FTP

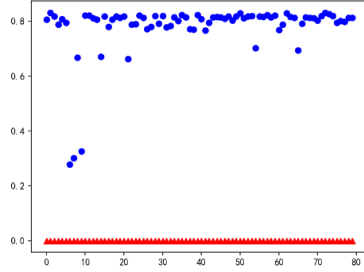


Fig. 4. benign and Port scanning

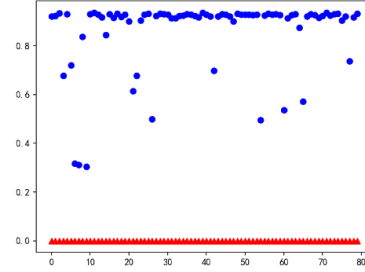


Fig. 6. benign and SQL injection

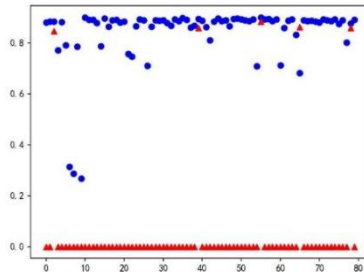


Fig. 7. benign and Brute-Force SSH

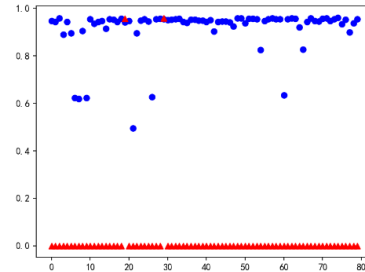


Fig. 8. benign and XSS attacks

Table 1. Classification effect of simple statistical features.

Traffic category	Accuracy	Recall	False Negative Rate	False Positive Rate
Brute Force	91.875%	83.75%	16.25%	0
Brute-Force FTP	96.875%	97.5%	2.5%	3.75%
Port scanning	98.125%	100%	0	3.75%
SQL inject	98.125%	100%	0	3.75%
Brute-Force SSH	95%	93.75%	6.25%	3.75%
XSS attacks	98.75%	97.5%	2.5%	0

According to the above table, it can be seen that the VAE model trained using only statistical features has an average accuracy of 96.458% in detecting seven types of malicious traffic, which can achieve accurate identification of malicious traffic.

The following Figures 9 to 14 shows the effect of inputting the feature vector into the VAE model for testing using statistical features and Bert word vector features. The test effect is as follows (the blue dots are benign traffic samples, and the red triangles are malicious traffic samples). The test results are shown in Table 2:

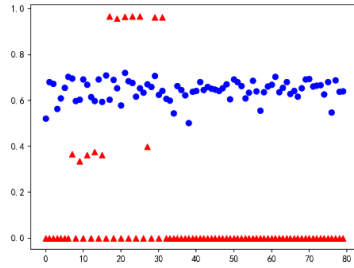


Fig. 9. benign and brute force cracking

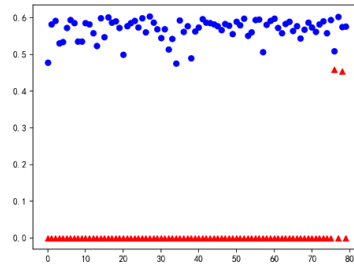


Fig. 10. benign and Brute-Force FTP

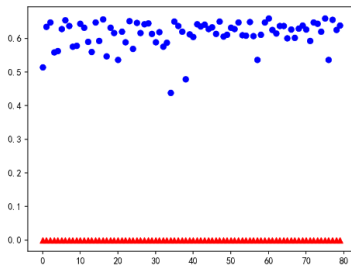


Fig. 11. benign and Port scanning

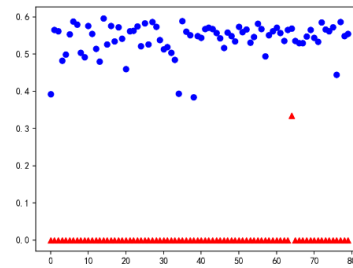


Fig. 12. benign and SQL inject

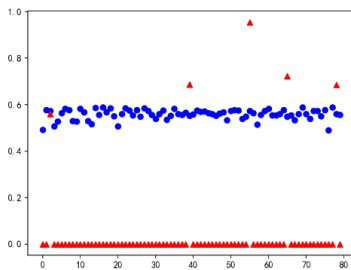


Fig. 13. benign and Brute-Force SSH

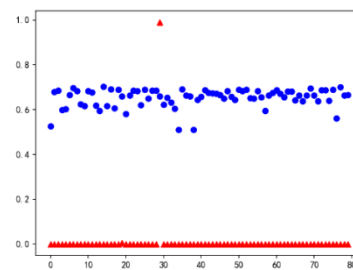


Fig. 14. benign and XSS attacks

Table 2. Classification effect of fusion of statistical features and BERT features.

Traffic category	Accuracy	Recall	False Negative Rate	False Positive Rate
Brute Force	95.625%	91.25%	8.75%	0
Brute-Force FTP	98.89%	97.5%	2.5%	0
Port scanning	100%	100%	0	0
SQL inject	98.125%	100%	0	3.75%

Brute-Force	96.875%	93.75%	6.25%	0
SSH				
XSS attacks	99.375%	98.75%	1.25%	0

Table 2 lists the model's classification results for various types of malicious traffic. It can be seen that the average accuracy and recall rate of this model for six types of malicious traffic reached more than 98.148%, which has excellent classification accuracy. The new features that combine statistical features and BERT word vector features have different improvements in the detection effect of six types of malicious traffic. Detecting malicious traffic for brute-force cracking and XSS attacks is more accurate, and detecting benign traffic for brute-force FTP, port scanning, and brute-force SSH is better improved. It shows that the method proposed in this paper to integrate traffic statistical features and BERT word vector features has a good effect on the model's malicious traffic detection method.

In order to verify the innovation of the method proposed in this paper, the models proposed in other papers using the CICIDS2017 data set for malicious traffic detection were selected as effect comparison models. The paper [13] proposed an improved K-Means clustering model based on genetic algorithm. The paper [14] proposes a way to encode traffic session bytes using word embedding, extracts the temporal characteristics of the session through an extraction algorithm that integrates a multi-head attention mechanism, and uses an LSTM classifier for classification. Paper [15] proposes a traffic anomaly detection method based on the bidirectional LSTM model. This method is based on SSAE for special feature extraction of traffic data and integrates the local timing information extracted by the bidirectional LSTM model and the global information extracted using the multi-head attention mechanism to classify and detect traffic. Paper [16] proposes an interpretable abnormal traffic detection model based on sparse autoencoders. This detection model uses a sparse autoencoder to learn normal traffic characteristics. On this basis, it introduces threshold iteration to select the best threshold to improve the detection rate of the model. The results of the detection effect of this model compared with other models are shown in the table 3:

Table 3. Classification Comparison of classification effects of this model with other models.

algorithm	Accuracy	recall rate	True Positive Rate
The method proposed in this paper	98.148%	96.875%	99.375%
Improved K-Means algorithm based on genetic algorithm	93.522%	97.548%	95.438%
Network malicious traffic detection algorithm incorporating multi-head attention mechanism	98.90%	97.18%	97.44%
Traffic anomaly detection method based on bidirectional LSTM model	98.44	98.46	98.69
Interpretable anomaly traffic detection based on sparse autoencoder	92%	99.2%	93.35%

It can be seen from the comparison data in the above table that the algorithm proposed in this paper has a good improvement in classification effect compared with the

traditional machine learning deep learning method. Compared with the latest traffic detection algorithm, the accuracy and recall rate are the same. However, because the algorithm proposed in this paper can learn the probability distribution characteristics of benign traffic very well, it has higher True Positive Rate for detecting benign traffic.

The above experimental results show that the model proposed in this paper has good comprehensive performance for malicious identification of unknown traffic. VAE only uses benign traffic for training. By learning the probability distribution of benign traffic, the model is trained by minimizing the reconstruction error and maximizing the difference between the prior distribution of the potential representation. Therefore, it is beneficial for the detection of various malicious traffic. The high generalization ability also dramatically improves the accuracy of malicious traffic detection.

4 Conclusion and Future Outlook

In view of the problems of high false alarm rates and insufficient generalization ability of current malicious traffic detection methods, this paper proposes a new set of traffic detection features, which combines the multi-angle of statistical features, comprehensive information, easy-to-understand analysis, and strong interpretability. Features and Bert integrate contextual semantic information to express better the meaning of words, which helps optimize the detection effect of the traffic detection model and improve the quality of model detection.

The detection model uses a variational autoencoder (VAE) to detect malicious traffic. VAE combines variational inference with an autoencoder to model input data from the probability distribution of samples in the latent space. Minimizing redundancy, Construction error, and KL divergence (comparing the distribution of the latent space with the prior distribution) makes the learned latent representation closer to the prior distribution. This allows the model to learn more meaningful and continuous latent representations, improves the accuracy of model detection, and improves the accuracy and generalization capabilities of the model.

The malicious traffic detection method proposed in this paper can effectively detect malicious traffic and play a very important supporting and assisting role in resource optimization and control. However, there is still much work to be done in the future. Currently, some newer malicious traffic types use artificial intelligence and machine learning to simulate benign traffic to carry out malicious activities. Data feature extraction and feature fusion processes will be optimized in the future, and word embedding technology will be used. The request information in the HTTP header will be represented as an embedded vector feature through natural language processing technology. On this basis, a small number of malicious traffic samples are enhanced further to improve the detection performance of malicious traffic samples. For variational autoencoding models, the latent space representation can be improved to capture the latent structure of the data better. In addition, other reconstruction functions can be tried, or complex decoder structures can be adopted, such as deep generative models or convolutional neural networks. The model has been improved for its detection performance.

References

1. Check Point Software.2021 Cyber Attack Trends Mid Year Report[R/OL]. (2021-07-22) [2022-05-06].
2. Camino LVCAMÁHSIMAVC: Malicious traffic detection on sampled network flow data with novelty-detection-based models. *Scientific Reports*, 2023, 13(1):15446-15446.
3. ROUSSEUW P J, HUBERT M: Anomaly Detection by Robust Statistics[R/OL]. (2017-10-14)[2022-05-09].
4. Liang XW, Jiang AP, Li T, et al: LR-SMOTE—An improved unbalanced data set over-sampling based on K-means and SVM. *Knowledge-based Systems*, 2020, 196: 105845.
5. DONG W Y, LI H T, WANG R M, et al.: Network traffic anomaly detection model based on stacked convolutional attention. *Computer Engineering*, 2022, 48 (9) : 12-19.
6. PU G, WANG L J, SHEN J, et al.: A Hybrid Unsupervised Clustering-Based Anomaly Detection Method. *Tsinghua Science and Technology*, 2021, 26(2):146-153.
7. Jinfu C, Yuhao C, Saihua C , et al.: An optimized feature extraction algorithm for abnormal network traffic detection. *Future Generation Computer Systems*,2023,149330-342.
8. LU Gang, GUO Ronghua, ZHOU Ying, et al.: Review of Malicious Traffic Feature Extraction. *Netinfo Security*, 2018, 18 (9): 1-9.
9. Zhao JS, Song MX, Gao X, Zhu QM: Research on Text Representation in Natural Language Processing. *Ruan Jian Xue Bao/Journal of Software*, 2022, 33(1): 102–128
10. Alberto L A , Alfonso MS , Antonio J M .: An Algorithmic-Based Fault Detection Technique for the 1-D Discrete Cosine Transform. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*,2020,28(5):1-5.
11. Dong L, Yong W, Chenhong L, et al.: An improved autoencoder for recommendation to alleviate the vanishing gradient problem. *Knowledge-Based Systems*,2023,263
12. Zhou X, Lin K D, Hu X, et al.: Robust parameter design based on Kullback-Leibler divergence. *Computers Industrial Engineering*,2019,135913-921.
13. ZHAO Jing, LI Jun, LONG Chun, DU Guan-Yao, WAN Wei, WEI Jin-Xia: Unknown Malicious Traffic Detection Based on Integrated SVM and Bagging. *Computer System Applications*, 2022, 31(10):51–59.
14. ZHAO Zhongbin, CAI Manchun, LU Tianliang.: Network Malicious Traffic Detection Incorporating Multi-Head Attention Mechanism. *Frontiers of Data & Computer*,2022, 4(5):60-67.
15. ZHAO Yu, HUO Yonghua, HUANG Wei, et al.: Traffic; Anomaly Detection Method Based on Bidirectional LSTM Models. *Radio Engineering*,2023,53(7):1712-1718.
16. LIU Yuxiao, CHEN Wei, ZHANG Tianyue, et al.: Explainable Anomaly Traffic Detection Based on Sparse Autoencoders. *Netinfo Security*, 2023, 23(7): 74-85.