Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 3, pp. 1083–1091, DOI 10.12694/scpe.v26i3.4375

COMPUTER NETWORK ATTACK DETECTION BASED ON JOINT CNN-LSTM MODEL WITH ATTENTION MECHANISM

MIAO JIANG[*] AND PEI LI[†]

Abstract. This paper addresses the problem that category imbalance in the traffic data set limits the detection performance of classification models for a few classes of attack traffic. The proposed method, which we call Jcla-detect, is based on a joint attention mechanism and a 1-D convolutional neural network (1DCNN)-Bi LSTM model. First, the Borderline SMOTE technique is used to pre-process the imbalanced training samples of traffic data during the data preparation step. This balances the various forms of traffic data and makes it possible for the subsequent model to correctly train the various types of data. After training a 1DCNN-BiLSTM model and a joint attention mechanism using the traffic data, the model extracts and classifies the local and long-range sequence characteristics. Then, by assigning a weight to the features that are helpful for categorization based on their significance, the attention mechanism raises the detection rate of the few assault types. The experimental results show that this method is effective in increasing the minority class attack traffic detection rate, as the method's detection accuracy can reach 93.17 for the URL dataset and it improves the detection rate of U2R attack traffic in the URL dataset by at least 13.70%.

Key words: Traffic anomaly detection; Category imbalance; CNN,Bi-LSTM; Attention mechanism

1. Introduction.

Web-based apps and services are becoming more and more important in people's lives as more and more network node devices are linked to the Internet. A significant number of network access devices—75 billion devices, to be exact—will be online by 2025, predicts Statista, a statistical research resource [1]. The flaws and vulnerabilities present in the protocols, operating systems, and application software that are employed in network attacks are constantly evolving and expanding in tandem with the substantial growth of the Internet. Traffic anomaly detection, an effective technique for network and information system security, is widely used to detect malicious behavior in network traffic [2].

Researchers have developed machine learning techniques to categorise and forecast massive amounts of traffic data for traffic anomaly detection as the amount of traffic data grows [3, 4]. A single classifier,it was discovered that typical machine learning techniques did not produce sufficient traffic anomaly detection results and that their detection performance was more feature-dependent. The majority of them prioritise feature engineering and feature selection, and thus frequently generate false alarms [5].

Numerous deep learning techniques have been used in recent years to study traffic anomaly detection by automatically extracting high-level features from the underlying traffic features through the neural network search space, with some promising research outcomes. To enhance the detection performance for NSL-KDD, [6] suggested a traffic anomaly detection approach combining stacked denoising self-encoder with soft max. [7] used densely linked CNN to detect traffic anomalies and increase detection precision using the KDDcup 99 dataset. On the CICIDS2017 dataset, [8] assessed the comparative detection performance of three neural networks, including LSTM, and discovered that Bi LSTM had the highest detection accuracy. In order to learn enough to improve detection outcomes, the majority of traffic anomaly detection approaches based on classical machine learning models and deep learning models need a lot of sample data. There is a notable class imbalance in the traffic statistics, with a considerable variation in the percentage of each attack class among the anomalous samples. Anomalies are typically greatly outnumbered by normal samples [9]. The majority class samples will outnumber the minority class samples when feeding this unbalanced traffic data training set straight into traditional classification models for learning and training, as is the case with less anomalous data and substantially unbalanced traffic data. Additionally, the few attacks with high threat levels may be

[*] Shangqiu Institute of Technology, School of Information and Electronic Engineering, HeNan Shangqiu 476000, China (jm20515@163.com).

[†] Shangqiu Institute of Technology, School of Information and Electronic Engineering, HeNan Shangqiu 476000, China.

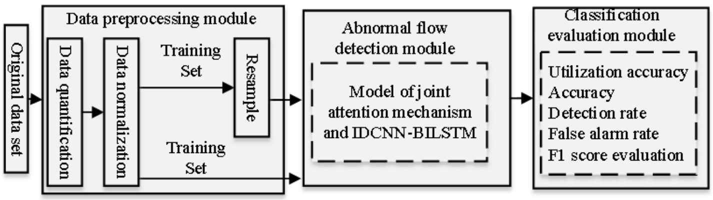

Fig. 2.1: Flow anomaly detection framework.

mistakenly identified as benign traffic or other attack classes, increasing risks to the network, devices, and users [10]. Therefore, the class imbalance issue in network traffic anomaly detection needs to be addressed in order to detect malicious activity in the network successfully.

Data and algorithms have been used by researchers to primarily address the category imbalance problem in traffic anomaly detection [11, 12]. On the data side, resampling techniques like synthetic minority class oversampling, adaptive synthetic sampling, and balanced resampling are mostly used to balance the different types of traffic data. On the algorithmic side, by enhancing algorithms or utilising integrated approaches, the detecting capability is increased. However, there is still a lot of space for improvement in the current research's detection rate of minority class attack traffic [13, 14]. This study suggests the Jcla-detect traffic anomaly detection method, which combines data improvement techniques with deep learning models to increase the detection rate of minority attack classes [15] to address the class imbalance problem in traffic anomaly detection.

The following are this paper's main contributions:

(1) In order to improve the detection performance of highly imbalanced traffic data in terms of both balanced data and improved models, this paper proposes a method for detecting traffic anomalies based on a joint attention mechanism and a 1DCNN-BiLSTM model.

(2) Using 1DCNN and Bi LSTM to extract local and long-range sequence features from network traffic data, respectively, we design a joint attention mechanism and a deep learning hybrid model of 1DCNN-BiLSTM for traffic anomaly detection in this paper. We also add an efficient attention mechanism to each block of 1DCNN and the end of Bi LSTM to focus on features that are crucial for classification and increase the detection rate of a few attack classes.

(3) This study conducts tests using the URL dataset and, using a number of evaluation criteria, compares the proposed method with several existing standard machine learning methods and methods that perform better on the problem of traffic data imbalance. The testing results demonstrate the method's superiority in the detection performance of imbalanced traffic data and its ability to greatly enhance the detection rate of a few classes of attack traffic.

2. The proposed model.

2.1. Traffic Anomaly Detection Framework.

With a minimal number of harmful traffic samples, the traffic anomaly detection approach in this study aims to achieve excellent detection performance on traffic data. In this regard, the suggested data resampling methods and deep learning network models are combined in the traffic anomaly detection method. Figure 2.1 depicts the proposed method's overall detection structure, which is made up of three primary modules: data pre-processing, traffic anomaly detection, and classification and evaluation.

The data pre-processing module quantizes, normalizes, and resamples the original traffic feature data for training. Quantization and normalization allow the data to fit the deep learning model's input format specifications, while data resampling allows traffic data to be balanced and lessens the influence and bias of unbalanced initial data categories on the detection results.

In order to effectively detect a small amount of attack traffic with a high threat level, this paper develops a joint attention mechanism and a 1DCNN-BiLSTM model for deep traffic feature extraction and learning on the pre-processed traffic data.

The model's detection outcomes are reviewed and analysed in the classification evaluation module utilising a range of detection evaluation markers.

2.2. Data pre-processing.

(1) Measurement. Since the traffic feature data in this study includes non-numerical features that must be converted into numerical features, such as protocol type, service, and flag in the NSL-KDD dataset, the Label Encoder() function is used for label encoding. It is also necessary to convert the sample category labels into numerical form. For dichotomous classifications, the normal and assault labels are represented by 0 and1, respectively, and for multiple classifications, by a distinct heat for each sort of attack.

(2) Normalization. In order to reduce the size difference between the feature values in the flow data set, to avoid the impact of numerical magnitude differences and unit differences on the detection results, and to ensure that the detection results are valid, the Min-Max normalization method is used to map each feature data to the [0,1] interval, and its formula is shown in Eq2.1.

$$y_1 = \left\{ \; x_n = \frac{x - X_{\min}}{X_{\max} - X_{\min}} \right. \tag{2.1}$$

where x is each eigenvalue of the feature column X , $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the feature column X, respectively.

(3) Excessive sampling. When deep learning classifiers don't learn enough features during model training, it can make it harder for the model to identify specific kinds of assault samples. During the data preprocessing stage, it is required to oversample the minority class attack traffic so that the deep learning model may fully and efficiently understand the boundaries of each class in the traffic sample space. Although boundary samples are more important for generalization, misclassification is more likely to occur with them. Newly synthesized attack class samples need to be near the class boundaries in order to provide enough information for learning and detection. Using the borderline SMOTE approach, we oversample anomalous traffic samples in this investigation. We first locate the attack samples at the edges, then we regenerate the attack samples, and finally we add the freshly generated samples to the traffic data training set.

For each attack sample x in the training set, calculate its m nearest neighbors, if the number of normal samples in the nearest neighbors of x is more than the attack samples, then x as a boundary sample of the attack class is likely to be misclassified as a normal sample, and such boundary samples need to be oversampled. In the sampling process, the k nearest neighbor attack samples of the attack sample x are calculated, and $n(1 < n < k)$ attack samples are randomly selected from them. The formula for generating new samples of the attack class traffic is shown in equation (2):

$$y_1 = \left\{ \; T_n = T_i + rand(0,1) \times |T_j - T_i| \right. \tag{2.2}$$

where T_n is the newly generated sample, T_i is the boundary sample, T_j is the neighbor of T , and $rand(0,1)$ means generating a random number in the interval [0,1].

2.3. Traffic Detection Model. Since traffic data may be essentially thought of as sequence data with backward and forward correlation, traffic feature data, like the NSL-KDD dataset, exhibits significant correlation and backward and forward sequence dependency between separate features of the same sequence. The sequence learning model can be used to train the dataset in order to detect this type of assault by capturing the deeper features and correlation of the traffic data before and after the detection period. Over time, probe attacks could show up as a persistent change in traffic characteristics.

This study develops a traffic anomaly detection model with a joint attention mechanism and 1DCNN-BiLSTM to fully learn the traffic feature data and successfully extract its deep and complex features in order to increase the detection rate of minority attack traffic. 1DCNN is a good choice for sequence processing in this model since it can perform more non-linear transformation and give traffic sequence features a greater local feature learning power. Although 1DCNN has limited capability for long-distance learning models, the network traffic data has a time series structure and can classify recent long-distance connections based on earlier connections. Learning long distance sequence characteristics is the primary application of bi LSTM

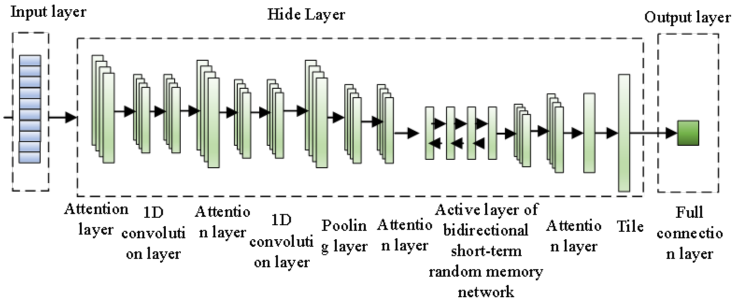

Fig. 2.2: Joint attention mechanism and 1DCNN-BiLSTM model structure.

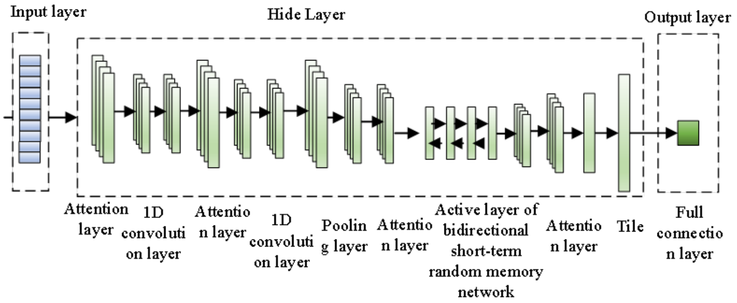

Fig. 2.3: 1 DCNN structure diagram.

networks. The collected deep traffic features from the 1DCNN are fed into the Bi LSTM to learn more sequential association patterns between deep traffic feature vectors across extended distances. In order to further enhance the model's performance in identifying unbalanced traffic data, this paper increases the weights of the traffic category-related features during the feature learning process. This causes the model to gravitate toward features that are more crucial for anomalous traffic detection.

To develop a multi-layer structure that includes a 1-dimensional convolutional layer, a pooling layer, an attention layer, a Bi-LSTM, a tiling layer, a fully connected layer, etc. to learn the correlation and local properties of normal and malicious traffic data sequences. Figure 2.2 depicts the structure of the model. Preprocessed traffic data is introduced into the model via the input layer, and the detection outcomes are then computed via the hidden layer and output via the output layer.

2.3.1. 1DCNN. For feature recognition, 1DCNN is a CNN that collects sequence data as a 1-dimensional grid. Despite having only one dimension, 1DCNN has the translation invariance of 2DCNN, which is advantageous for recognising features. Using this information as a foundation, this study generates the traffic feature data as sequence data with benign and malicious labels, and then applies 1DCNN to the traffic data to achieve local feature extraction. Figure 2.3 illustrates the structure of the 1DCNN model, which uses stacking of 1-dimensional convolutional and pooling layers to address the issue of local feature loss.

The first layer of convolution in one dimension is essential for feature extraction. By training the traffic data to produce an ideal set of convolution kernels with the least amount of loss, complicated traffic features can be automatically extracted using convolution kernels (filters). The i th sample of the flow data can be represented as an m dimensional feature vector $x_i \in R^m$, and multiple consecutive vectors $x_i, x_{i+1}, ..., x_j$ can be represented as $x_{i:j}$, 1-dimensional convolution is performed only in the vertical direction of the flow feature

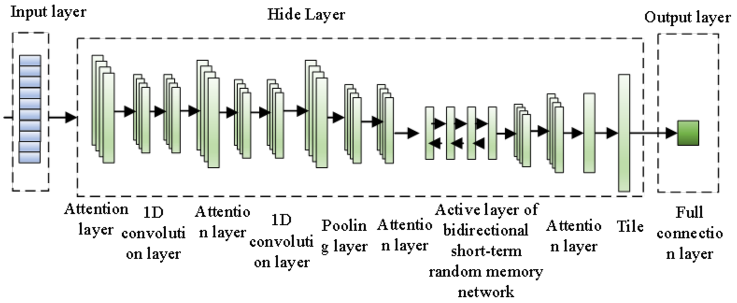

Fig. 2.4: Bi LSTM structure diagram.

data sequence, so the width of its convolution kernel is the dimension of the flow feature, and a feature mapping is constructed by applying a convolution operation to the input flow data using filter w to achieve local space feature extraction, which is calculated as Eq. (3) shows:

$$y_1 = \{ \; h_i = f(\omega \otimes x_{i:j} + b) \tag{2.3}$$

where b is the bias value and $f()$ denotes the nonlinear activation function linear rectification function for the convolution calculation. To retrieve the most crucial information, the pooling layer further aggregates and keeps the short-term features that the convolution layer extracted. The maximum pooling layer is employed in this study to merge the highest feature values from each convolutional layer's feature vectors. A 1 x n-dimensional data feature is created after the operation in the 1-dimensional convolutional and pooling layers, which effectively examines and preserves the local properties of the traffic data sequence.

2.3.2. Bi-LSTM. By learning the connection between the forward and reverse of sequence data, the Bi LSTM variant of the LSTM model improves it and gives it an advantage in classification tasks. The long-range sequence learning capabilities of the LSTM model are also present in Bi LSTM. The input traffic data is used in this study to train the forward and reverse LSTM of the Bi LSTM, whose structure is shown in Figure 2.4 and consists of an input layer, a forward hidden layer, a reverse hidden layer, and an output layer. On the other hand, the reverse LSTM retrieves the deep traffic feature sequence's reverse features from backward to forward. The input deep traffic feature sequence's forward properties are extracted by the forward LSTM. Both are combined in the output layer.

Bi LSTM effectively exploits the temporal features present in data before and after network traffic to improve model training, allowing the model to learn sequence features comprehensively.

2.3.3. The attention mechanism. The fundamental principle of the attention mechanism states that while irrelevant and useless information is ignored in favor of extracting features from more crucial and important information, limited attentional resources are allocated to a small number of crucial pieces of information that require special attention. In order to improve the detection rate of a small sample size of attack samples, it is more beneficial to implement an attention mechanism that assigns matching weights to different traffic features that are used to identify attacks when it comes to traffic anomaly detection. This paper introduces the bi LSTM network and the 1DCNN network, respectively, by means of the attention mechanism. The attention layer is added to the end of the convolutional block for the 1DCNN in order to solve the problem where the convolutional neural network only focuses on local characteristics and leads to erroneous learning of global information. The attention technique uses a weighted summing of its hidden layer vector output expressions to improve detection results with Bi LSTM. The attention mechanism uses probability to assign weights instead of the original random allocations.

3. Experimental results and analysis. On the URL dataset, Jcla-detect surpasses the machine learning techniques DT and LR. The results of Jcla-detect and the original CNN method are nearly identical. The

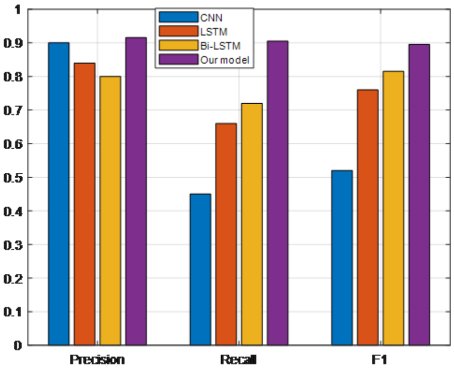

Fig. 3.1: F1 result of malicious URL data.

Table 3.1: Experimental results on TREC and 20NG.

Method	TREC	20NG	Average
SA-CNN	94.20	83.41	88.796
CNN	93.61	82.57	88.081
NB	90.41	82.78	86.586
KNN	86.53	75.29	80.901
LR	95.37	80.93	88.141
RF	90.73	73.45	82.081
DT	93.85	73.36	83.596
GBDT	93.89	67.58	80.726

proposed Jcla-detect is far more effective than the original CNN method in this regard because it can discover and visualise dangerous code sections.

3.1. Token segmentation and char segmentation. The experimental outcomes of the first CNN and Jcla-detect3 (LSTM) on the URL dataset are displayed in Figure 3.1. It would not be appropriate to utilise accuracy as an assessment metric at this time because malicious URLs typically make up a small portion of all URLs. Instead, we use F1 to assess the experimental findings for URLs. As can be seen, token segmentation produces better outcomes than char segmentation. This shows that the token segmentation method is successful in detecting URLs. Regarding the F1 assessment metric, Jcla-detect does not demonstrate a higher advantage over the original CNN approach, most likely because the performance of the original CNN itself is so high that it is very challenging to improve it.

The experimental results of Jcla-detect on two short text datasets are shown in Table 3.1, indicating that Jcla-detect outperforms the original CNN method.

We also use Bayesian, logistical regression, KNN, and GBDT methods to conduct experimental comparisons on the TREC and 20NG datasets. Jcla-detect performs better than these machine learning techniques, in our experience. This shows that the sequence attention technique works on both the short text dataset and the URL dataset.

3.2. LSTM model and Markov model. The Jcla-detect model itself was the focus of this paper's relevant research and experiments. The experimental Jcla-detect findings utilising LSTM and Markov language models on URL data sets in token partition mode are displayed in Figure 3.2. Compared to Jcla-detect-5 (LSTM) and Jcla-detect-3 (Mark-ov), Jcla-detect-3 (LSTM) is more suitable for anomaly detection utilising the token partition method, as shown in Figure 6. Jcla-detect-3 (LSTM) is suggested for URL detection tasks due to the URL length restriction and word correlation.

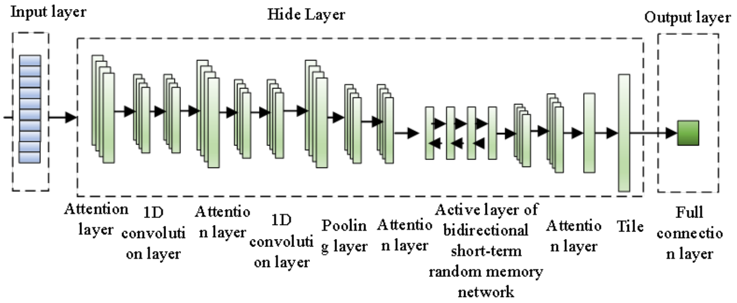

Fig. 3.2: Jcla-detect token partition results of three structures.

Table 3.2: Results of TREC and 20NG.

Methods	TREC	20NG
SA-CNN-3	94.20	83.409
SA-CNN-5	95.50	82.546
SA-CNN-M	93.42	80.457

3.3. Context length impactl. The experimental findings of Jcla-detect on two sets of brief text samples are presented in Table 3.2. TREC data set average length is 10, and 20NG average length is 5. Table Table 3.2 shows that Jcla-detect-5 (LSTM) outperforms competing approaches on TREC data, indicating that the average text length influences model choice.

The length of the URL is another potential factor that could influence the outcome of URL anomaly detection in addition to the token partition method previously mentioned. When the URL is brief, we might think about utilising Jcla-detect-3 (LSTM) to find anomalies. By examining Jcla-detect, we can observe that Jcla-detect-3 (LSTM) and Jcla-detect-5 (LSTM) are more suited for detecting URLs with longer lengths and that both models can produce more accurate results. Short URLs are unable to give additional information for anomaly identification, which further reduces the model's ability to detect anomalies. Therefore, we must take into account the length of the text while classifying texts or detecting anomalies, and then choose the right model to test.

3.4. Analysis of visualization results. For the sake of simplicity, this article provides some concrete examples to illustrate the results more intuitively. As shown in Figure 7, the darker the color is, the higher the value of attention is. By comparing the color depth, we can easily know which parts are malicious code areas. As shown in Figure 3.3, Local include file attacks attempt to access sensitive files on the server through the code "../". In article 3, char (106) is a statement to test whether the server can execute SQL functions. Because passwd, script, (;) and passwd are all offensive parts of malicious URLs, they all have high attention values. However, html and com11 have low attention values, because they are part of the normal code area. An interesting fact is that when the "." token is located in the normal URL, it has a low attention value. At this time, it is surrounded by oo4xccc and html, and the "." in the malicious code area has a high attention value. At this time, Its context is "/" and "./". This means that the context information generated by the external language model (LSTM/Markov) is valid and meaningful.

The overall effectiveness of several models on the CTU-13 data set is displayed in Table 3.3. The suggested model has better precision and recall rates than PCNN and HDM, and its training time is roughly a third of that of PCNN and HDCM. This is because the model suggested in this research can simultaneously learn tasks from three different domains, and training time is reduced through parameter sharing across domains.

The confidence interval shown in Table 3.4 was calculated using a paired sampling t test with a 90%

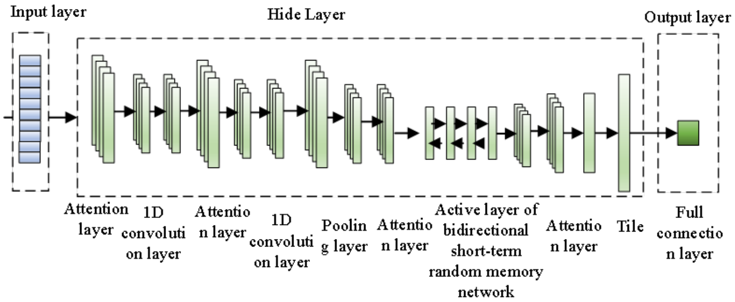

Fig. 3.3: Visualization result of malicious URL attention value of Chinese image title.

Table 3.3: Overall performance of models.

Measures	PCNN	HDCM	Proposed model
Precision (%)	97.5	94.8	99.4
Recall (%)	94.7	94.7	98.0
Sensitivity (%)	93.8	92.1	97.2
Accuracy (%)	95.6	93.4	99.1
Training time (s)	25.42	21.75	9.19

Table 3.4: Confidence interval.

-	$\mu_1 - \mu_3$	$\mu_2 - \mu_3$
Precision (%)	(3.561,1.316)	(6.711,0.985)
Recall (%)	(6.671,0.482)	(6.735,0.281)

confidence level. T stands for the average value of each measurement, and the subscripts 1, 2, and 3 denote the PCNN, HDM, and model that was proposed in this paper, respectively. There are 24 degrees of freedom (the abnormal category is 9 and there are 3 models), and the results were presented as a confidence interval. The findings demonstrate that the model put forward in this study has a higher recall rate than PCNN and HDM models and a higher precision than HDM models. The outcomes also demonstrate that the model put forward in this study is capable of handling three tasks from distinct domains simultaneously without degrading its performance in any one domain.

4. Conclusion.

To enhance the detection rate of minority attack classes, we propose Jcla-detect, a novel method aimed at balancing the various types of traffic data. By addressing the inherent class imbalance, Jcla-detect ensures that the model is not biased towards majority traffic data, allowing it to learn more robust and accurate features from underrepresented attack traffic. The model undergoes comprehensive training across all traffic types, ensuring that both majority and minority classes are equally represented in the learning process.

One of the key innovations in our approach is the incorporation of a joint attention mechanism. This mechanism plays a critical role in refining the model's ability to focus on the most relevant features from different types of traffic data. By attending to both global and local patterns within the dataset, the joint attention mechanism enables the model to more effectively differentiate between benign and attack traffic, even in the presence of subtle variations. This enhanced feature extraction process is particularly beneficial for detecting minority attack traffic, which may exhibit less obvious characteristics compared to the majority class.

The experimental results demonstrate the significant impact of Jcla-detect in improving the detection rate of minority attack traffic. Specifically, our method was tested on the URL dataset, which contains both majority normal traffic and minority attack traffic. The findings indicate that Jcla-detect achieves a detection accuracy of 93.17%, outperforming traditional methods and highlighting the efficiency of our approach. This high accuracy is a direct result of the balanced training process and the joint attention mechanism, which allows the model to capture subtle features in attack traffic that may otherwise be overlooked.

Furthermore, Jcla-detect not only improves detection accuracy but also demonstrates greater robustness in dealing with complex and heterogeneous traffic data. The method effectively handles different traffic patterns, reducing the false positive rate and improving the overall reliability of the detection system. By balancing the data distribution and enhancing the feature learning process through attention mechanisms, Jcla-detect addresses key challenges in minority class detection, offering a promising solution for real-world applications where attack detection is critical.

In conclusion, the combination of data balancing and joint attention mechanisms in Jcla-detect significantly improves the detection of minority attack traffic, achieving high accuracy and robustness. These findings suggest that Jcla-detect can be a valuable tool in cybersecurity systems, particularly in environments with imbalanced traffic data. Future work could explore the adaptability of this approach to other datasets and attack types, further expanding its applicability and effectiveness.

Data Availability. The experimental data used to support the findings of this study are available from the corresponding author upon request.

REFERENCES

[1] Hameed, K., Garg, S., Amin, M. B., Kang, B., Khan, A., *A context-aware information-based clone node attack detection scheme in internet of things.* Journal of network and computer applications (2022). 197.

[2] Weathersby, A., Washington, M., *Extracting network based attack narratives through use of the cyber kill chain: a replication study.* it - Information Technology, (2022). 64(1-2), 29-42.

[3] Tang, D., Wang, X., Yan, Y., Zhang, D., Zhao, H..*Adms: an online attack detection and mitigation system for ldos attacks via sdn.* Computer communications(Jan.), (2022).181.

[4] Ashiku, L., Dagli, C. , *Network intrusion detection system using deep learning.* Procedia Computer Science, (2021).185(1), 239-247.

[5] Venugopal, E., *A comparative analysis on hybrid svm for network intrusion detection system.* Turkish Journal of Computer and Mathematics Education (TURCOMAT), (2021). 12(2), 2674-2679.

[6] Winanta, A., Rochsantiningsih, D., Supriyadi, S. *Exploring efl classroom interaction: an analysis of teacher talk at senior high school level.* ELS Journal on Interdisciplinary Studies in Humanities, 3(3),(2020) 328-343.

[7] Sushma, E., *A review of the cluster based mobile adhoc network intrusion detection system.* Turkish Journal of Computer and Mathematics Education (TURCOMAT), (2021).12(2), 2070-2076.

[8] Han, S., Han, S., Liang, D., Liang, D., Haneda, M., Haneda, M. *A case study of two south korean middle school efl teachers' practices: instructional stances and use of classroom materials.* Classroom Discourse, 12(1-2),(2021) 56-74.

[9] Mao, B., Liu, J., Lai, Y., Sun, M. . *Mif: a multi-step attack scenario reconstruction and attack chains extraction method based on multi-information fusion.* Computer Networks(3), (2021). 108340.

[10] Lawal, M. A., Shaikh, R. A., Hassan, S. R. . *A ddos attack mitigation framework for iot networks using fog computing.* Procedia Computer Science, (2021).182(8), 13-20.

[11] Nandhakumar, E. . *A hybrid adaptive development algorithm and machine learning based method for intrusion detection and prevention system.* Turkish Journal of Computer and Mathematics Education (TURCOMAT),(2021). 12(5), 1226-1236.

[12] Nicolai, K. E.*A green gambit: the development of environmental foreign policy in morocco.* The Journal of North African Studies, 27(4), (2022) 714-740.

[13] Robinson, L., Brown, T., Gledhill, K., Isbel, S., Parsons, D., Etherington, J., et al.*'learning in and out of lockdown': a comparison of two groups of undergraduate occupational therapy students' engagement in online-only and blended education approaches during the covid-19 pandemic.* Australian Occupational Therapy Journal, 69(3), (2022) 301-315.

[14] Xiong, Y., Jin, M., Wang, J., & Wang, X. *Synergistic effect to enhance hydrogen generation of fe2o3/ce0.8sm0.1gd0.1o1.9 in water-gas shift with chemical looping.* International Journal of Energy Research, 46(7), (2022) 9733-9747.

[15] Hofmann, V., & C. M. Müller. *Challenging behaviour in students with intellectual disabilities: the role of individual and classmates' communication skills.* Journal of Intellectual Disability Research, 66(4), (2022) 353-367.

Edited by: Ashish Bagwari
Special issue on: Adaptive AI-ML Technique for 6G/ Emerging Wireless Networks
Received: Jul 25, 2024
Accepted: Oct 11, 2024