Deep Learning Based DDoS Attack Detection

Ziyi Xu

Communication University of China, Hainan International College, No.1 Dingfuzhuang East Street, Chaoyang District, Beijing, China

Abstract. Nowadays, one of the biggest risks to network security is Distributed Denial of Service (DDoS) assaults, which cause disruptions to services by flooding systems with malicious traffic. Traditional approaches to detection, based on statistical thresholds and signature-based mechanisms, respectively, can hardly cope with the increasing complexity of such an attack. In order to improve detection accuracy and generalization, this research suggests a deep learning-based detection model that combines the Long Short-Term Memory (LSTM) network architecture with Convolutional Neural Networks (CNN). On the CICDDoS2019 dataset, which included several DDoS attack versions, the suggested model was trained and evaluated. The hybrid CNN-LSTM has extraction capabilities regarding both the spatial and temporal features of network traffic data, showing highly efficient performance. The classification resulting from this model yielded high accuracy with robust results for different attack scenarios. Results reflect the potential superiority of the given model in detecting DDoS attacks. Even though the performance was sound, the model still showed certain shortfalls, which were revealed when particular types of attacks were tested. Future work may be directed at further refining the model architecture, including optimizing diversity in training to allow for even better detection capabilities.

1 Introduction

DDoS assaults have become one of the biggest cybersecurity concerns in this day and age, as the internet is growing at an accelerated pace. DDoS-based attacks target servers by sending a large number of requests, which, in turn, results in the inability of the server to process valid traffic and thus make the network resources unavailable [1]. Due to the increasing complexity and frequency of DDoS, effective detection and mitigation of this threat have become an important research area in cybersecurity.

Traditional DDoS detection systems are designed based on statistical methods or threshold-based mechanisms; these are inadequate to deal with new, sophisticated types of attacks [2]. As a result, machine learning and deep learning-based detection techniques have received the majority of recent attention. These can learn from large datasets in such a way that automatically extracts the relevant features of DDoS attacks to be used for better efficiency and accuracy in detection.

Corresponding author: xuzy@cuc.edu.cn

This study provides a hybrid CNN-LSTM network model for DDoS attack detection. Experimental validation on the CICDDoS2019 dataset demonstrates that the proposed model detects various kinds of DDoS attacks with excellent performance not only in classification accuracy but also in generalization capability. In this paper, a detailed description is given about the architecture of the proposed model, data preprocessing methods, experimental results, and the corresponding analysis.

2 Related Work

Recently, deep learning has found increased application in the detection of DDoS attacks. Shieh et al. identified a method that combines deep learning with the Gaussian Mixture Model (GMM), where unknown types of DDoS attack were detected properly [3]. Alghazzawi et al. proposed a hybrid deep learning model that improves the efficacy of DDoS detection by optimizing feature selection. The approach increases not only the model's precision but also its computational efficiency, bringing the importance of performing feature selection in optimizing DDoS detection [4].

Akgün et al. made the assumption that a deep learning model and data preprocessing will be used in a DDoS assault intrusion detection system. They evaluated the real-time performance and detection accuracy of models built on CNN, LSTM, and Deep Neural Network (DNN). Experimental results show that CNN-based models have reached high accuracy and powerful performance in DDoS attack detection using deep learning [5].

Particularly, Abdallah et al. submitted a design to fit in an anomaly detection system through a hybrid CNN-LSTM architecture specifically for the detection of DDoS attacks happening in Software Defined Networks. In their model, they managed to embed spatiotemporal features, making it a worthy one in complex network environments for attack detection [6]. Moreover, Mousa and Abdullah combined the stacked autoencoder with a checkpoint network to a large extent improve the performance in anomaly-based DDoS detection, especially the detection performance in large-scale complex attacks [7]. Further, Alghazzawi et al. have suggested a detection technique in DDoS using an architecture of hybrid CNN-Bidirectional Long Short-Term Memory (BiLSTM). The maximum reached accuracy on the CICDDoS2019 dataset was 94.52% [8]. Such experiments show that by assembling various algorithms, the performance and accuracy of the model greatly increase.

Numerous studies demonstrate effective deep learning-based techniques for DDoS attack detection. The recent advances in feature selection optimization and various model architectures have greatly increased the efficiency and accuracy of DDoS detection. Based on such work, this paper adopts hybrid modeling by combining the CNN-LSTM model, which is further validated on CICDDoS2019 datasets, showing superior synthesis performance in all attack scenarios.

3 Methodology

3.1 Preprocessing

Realistic data are mostly not idealized, so it is necessary to carry out data preprocessing before feeding it into the training phase. Missing value and outlier handling are the first issues to be tackled in the model. The method of deleting samples with more missing values is used to ensure completeness and consistency in data. For outliers, this study uses the z-score method, which uses the standard deviation to determine each data point's distance from the mean in order to identify outliers, and usually sets a threshold beyond which data points are considered outliers and removed.

The linear correlation between features and the target label is then measured using the Pearson correlation coefficient throughout the feature selection phase. The calculation of Pearson's correlation coefficient is given by Formula (1).

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where r is the correlation coefficient. Correlation coefficients are useful for determining which features are less powerful in predicting the target outcome and which are highly correlated with other features, thus removing redundancy and simplifying the model, since including correlated variables increases the risk of overfitting.

Finally, this research uses a standardization strategy for all data and describes the features in the form of a standard normal distribution with a mean of 0 and a standard deviation of 1, imposing a common scale on all attributes.

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

where x is the initial feature value, z is the normalized value, μ is the feature mean, and σ is the feature standard deviation. The process of normalization not only helps the gradient descent algorithm to converge faster but also guarantees that each individual feature contributes equally to training.

3.2 Convolutional Neural Network

The fundamental premise of CNN, a deep learning model that is frequently used to process structured data, is to accomplish dimensionality reduction through pooling layers and extract local features from the data using convolutional operations. In particular, in order for the data to be identified effectively, CNN makes much use of the extraction of the object features through many stacked convolution and pooling layers [9]. The first layer is a one-dimensional convolutional layer (1D-CNN) that works through convolutional kernels with size 3. The number of said convolutional kernels in both is 64. Through the convolutional operation, local time-dependent patterns are extracted using the input data, manifested in the form of linear combinations of features from adjacent time steps, from the following formula (3).

$$y_i = \sum_{j=1}^{k} x_{i+j-1} \cdot w_j + b \tag{3}$$

where y_i is the output value at position i, $x_{i+j-1}$is the input value at the corresponding position, $w_j$is the weight of the convolutional kernel, and b is the bias term. This formula shows that a feature map is generated as the convolutional layer aggregates useful local information through a process of weighted summation.

The layer that follows the convolution layer is the Max Pooling layer, which has the effect of reducing the spatial scale every time. It drifts a fixed-size window over a feature map and selects the maximum input within each window to avoid overfitting as it retains the most salient information and reduces, to an extent, the model's complexity. This layer can significantly reduce the number of parameters while retaining the information most important to the model's classification task.

However, the CNN perceptron is relatively flawed and inadequate due to the long training period and limited network generalization, so the research in this paper combines CNN with another common deep learning algorithm to get a better algorithm.

3.3 Long Short-Term Memory

LSTM is a specifically designed form of recurrent neural networks in dealing with and predicting long-term dependencies inherent in time series data. In contrast to the traditional Recurrent Neural Networks (RNNs), due to its self-innovated gating mechanism, LSTM will control the information flow and thus keep the important information retention and transmission in the long time series, avoiding the gradient vanishing problem, which makes itself perform well on the tasks dealing with long time dependency [10].

Therefore, the proposed approach can significantly improve the model's long and short-term dependence by feeding the feature vectors output from the CNN into an LSTM layer. The designed LSTM layer itself contains 128 hidden units controlling exactly how information flows through several input gates, forget gates, and output gates capturing and memorizing important patterns in the data sequence. The basic formula of LSTM is as formula (4).

$$h_t = o_t \odot \tanh(c_t) \tag{4}$$

where h_t is the hidden state at time step t, o_t is the activation value of the output gate, and c_t represents the cell state, modulated by the nonlinear activation function $tanh$. These commented equations reflect how this LSTM takes advantage of intrinsic mechanisms of inner gates and memory cells to maintain sequence information memorized and updated.

The LSTM layer is followed by a Dropout layer to increase the model's ability to generalize. This will randomly eliminate part of the neurons at a ratio of 0.5 so that it can prevent the model from fitting too well. In addition to this, Dropout can make the model be more resistant to noise in the training process and can enhance its performance on unknown data.

4 Experiments

4.1 Dataset Description

The public dataset CICDDoS2019 was chosen to be used in this study. This Dataset was published in 2019 by the Canadian Institute for Cybersecurity (CIC). It has been specifically designed to support research and evaluation techniques of detection for DDoS attacks. This dataset is aimed at providing a modern, comprehensive collection of DDoS attack data, with coverage over various types of DDoS attacks and normal network traffic. The CICDDoS2019 dataset was well labeled. Each of the traffic records were classified as normal traffic and attack traffic, with the attack traffic divided into some specific types of attacks. The labels are directly used as training data for the algorithms applied in this experiment.

4.2 Model Training

Ten training epochs were used to gradually optimize the model's weights and biases during the trial. From the first to the tenth epoch, the training accuracy increased linearly, from 98.19% to 98.75%. At the same time, the training loss dropped from 0.0620 down to 0.0511, thus proving that the model effectively and efficiently captured essential features of data in the course of learning. Performance on the validation set was similarly very impressive, with validation accuracy starting at 98.97% in the first epoch and remaining high at 98.70% by the tenth epoch. Validation loss fluctuated somewhat, but the strong generalization on test data was maintained.

4.3 Confusion Matrix Analysis

There is a more natural method to comprehend how well the model predicts across many categories: through the confusion matrix. Table 1 is confusion matrix of the model. As the confusion matrix shows, the model rightly classifies 966,491 samples as benign, only misclassifying 162 as attacks. This indicates that this model did an excellent job in correctly classifying samples, particularly in terms of telling apart normal traffic and common varieties of DDoS attacks such as Network Time Protocol (NTP) Amplification Attack and Synchronize (SYN) Flood Attack.

Table 1. Confusion matrix of the model.

3812	40	198	0	104
162	966491	4862	10405	53
936	1926	235665	121	98
75	4840	3618	610439	89
11	105	7	0	275638

On the other hand, the confusion matrix also denotes where the model had problems. In Domain Name System (DNS) Amplification Attack category, the model correctly classified 3,812, while it misclassified 104 samples as benign and 198 as other Attack Types. The classification of SYN attacks also showed some challenges, as 4,862 samples were misclassified as NTP attacks, while 10,405 were classified as SYN attacks—this possibly showing overlap in feature space between these two attack types. Results might suggest that while there are many things at which the model is good, it improves upon reducing most of the misclassification rates for some categories.

4.4 Model Evaluation

One key measure of a model's utility is how well it performs on a test set. By evaluating 2,119,695 test samples, the model obtain the following key metrics. These model evaluation indicators can be seen in Table 2.

Table 2. Model evaluation indicators.

Precision	**Recall**	**F1-score**
0	0.76	0.92
1	0.99	0.98
2	0.96	0.99
3	0.98	0.99
4	1.00	1.00

For the test dataset, the overall accuracy of the model came out to be 98.70%, which proved to be pretty effective. There are some differences in the model's performance between categories, according to the categorization report. For the category of DNS assaults, the precision is 0.76, the recall is 0.92, and the F1-Score is 0.83. These results indicate a major misclassification problem with this class, probably as a result of data complexity or relatively small sample size. On the other hand, this model did very well in classifying instances of normal traffic, with precision and recall both at 0.99 for an F1-Score of 0.99 also. This means this model has high accuracy and reliability in detecting normal traffic. It also did very well on the detection of other common types of DDoS attacks, including NTP Amplification Attack, SYN Flood Attack and User Datagram Protocol (UDP) Amplification Attack. In all these cases, precision and recall are also very high, while the F1-Score was close to 1.0. These

results once more upheld that the model is capable of defining boundaries clearly between the latter-mentioned attack types.

5 Conclusion

In order to efficiently detect various DDoS attack types, this study suggests a deep learning-based model for DDoS attack detection that makes use of CNN and LSTM. Supremacy will be shown on the CICDDoS2019 dataset through various experiments on the most critical items: namely, the distribution with the same classification accuracy, and generalization capability. Although it performs very well in most situations, there is still a need for improving the performance in certain attack type classifications. Based on this, future work should focus on the optimization of dataset diversity, improvement of model structure, and combination with other deep learning methods to enhance model performance in detection.

References

1. Mittal, M., Kumar, K. & Behal, S. Deep learning approaches for detecting DDoS attacks: a systematic review. *Soft Comput.* **27**, 1303 9–13075 (2023). https://doi.org/10.1007/s00500-021-06608-1
2. Cui, Y., Qian, Q., Guo, C., Shen, G., Tian, Y., Xing, H., & Yan, L. Towards DDoS detection mechanisms in Software-Defined Networking. *J. Netw. Comput. Appl.* **190**, 103156 (2021). https://doi.org/10.1016/j.jnca.2021.103156
3. Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., & Miu, D. Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model. *Appl. Sci.* **11**, 5213 (2021). https://doi.org/10.3390/app11115213
4. Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M.Z. Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection. *Appl. Sci.* **11**, 11634 (2021). https://doi.org/10.3390/app112411634
5. Akgun, D., Hizal, S., & Cavusoglu, U. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Comput. Secur.* **118**: 102748 (2022). https://doi.org/10.1016/j.cose.2022.102748
6. Abdallah, M., An Le Khac, N., Jahromi, H., & Delia Jurcut, A. A hybrid CNN-LSTM based approach for anomaly detection systems in SDNs. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, Vienna, Austria, August 2021 (pp. 1-7). https://doi.org/10.1145/3465481.3469190
7. Mousa, A.K., & Abdullah, M.N. An Improved Deep Learning Model for DDoS Detection Based on Hybrid Stacked Autoencoder and Checkpoint Network. *Future Internet.* **15**: 278 (2023). https://doi.org/10.3390/fi15080278
8. Alghazzawi, D., Bamasag, O., Ullah, H., & Asghar, M.Z. Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection. *Appl. Sci.* **11**: 11634 (2021). https://doi.org/10.3390/app112411634
9. Sumathi, S., Rajesh, R., & Lim, S. Recurrent and deep learning neural network models for DDoS attack detection. *J. Sens.* **2022**(1): 8530312 (2022). https://doi.org/10.1155/2022/8530312
10. Staudemeyer, R.C., & Morris, E.R. Understanding LSTM—a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586* (2019). https://doi.org/10.48550/arXiv.1909.09586