Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 5, No. 3, 2023

SCD: A Detection System for DDoS Attacks based on SAE-CNN Networks

Hao Xu, Hequn Xian *

College of Computer Science and Technology, Qingdao University, Qingdao, China
* **Corresponding author**: Hequn Xian (Email: xianhq@126.com)

Abstract: The pervasive application of network technology has given rise to a numerous of network attacks, including Distributed Denial of Service (DDoS) attacks. DDoS attacks can lead to the collapse of network resources, making the target server unable to support legitimate users, which is a critical issue in cyberspace security. In complex real-world network environments, differentiating DDoS attack traffic from normal traffic is a challenging task, making it significant to effectively distinguish between attack types in order to resist DDoS attacks. However, traditional DDoS attack detection methods have certain limitations in terms of data preprocessing and detection efficiency. In this paper, we propose a lightweight framework based on deep learning called SAE-CNN-Detection (SCD), which combines stacked autoencoder network (SAE) and convolutional neural network (CNN) for DDoS attacks detection. The CIC-DDoS2019 dataset is used to simulate network traffic that has suffered from DDoS attacks, and this system employs adaptive preprocessing techniques for the dataset. The results demonstrate that multi-classification experiment achieves an accuracy of 97.2% for DDoS attack types, while the binary classification experiment achieves an accuracy of 99.1%.

Keywords: Distributed Denial of Service; Deep Learning; Stacked Autoencoder Network; Convolutional Neural Network.

1. Introduction

The DDoS attack is a highly damaging network attack that employs multiple types of attacks and coordinated efforts on a large scale. Attackers leverage a large number of compromised hosts, also known as “zombies”, to launch simultaneous DDoS attacks against the target system. DDoS attacks can cause network congestion, system crashes, and exhaustion of input/output bandwidth for databases and servers. This security hazard has become increasingly prevalent in recent years, particularly as a result of the widespread adoption of cloud technology by businesses, coupled with the proliferation of insecure Internet of Things (IoT) devices and zombie networks. DDoS attacks are becoming more and more severe in terms of their scale, frequency, and intricacy, thereby posing a global challenge for enterprises from a security standpoint.

In March 2022, Russian companies and corporate websites suffered a 7-fold surge in the incidence of DDoS attacks, with the most prolonged attack lasting for 145 hours, resulting in considerable damage. Enterprises with data centers, public cloud service providers, internet infrastructure service providers such as DNS providers, emerging game service providers, and basic government network industries are all prime targets for DDoS attacks. The variety of DDoS attack modes makes them difficult to detect, leading to further proliferation.

Several researchers have endeavored to utilize machine learning techniques for the identification of DDoS attacks [1-3]. While these methods have shown improved performance compared to statistical techniques, they still have some limitations. These limitations include the need for extensive prior knowledge and experience in network operations, intricate human extraction of appropriate statistical features during DDoS detection, and limitations in detecting a variety of DDoS attack types. In addition, machine learning methods require model updating to accommodate changes in the system and attack vectors. Furthermore, classical machine learning classification techniques have limited feature extraction capabilities, making them complicated to handle large volumes of traffic data.

Recently, deep learning (DL) has achieved significant success in various application fields, such as facial recognition, image processing, and natural language translation. DL can extract raw features from data without any human intervention [4]. DL models can automatically search for correlations within the raw data, to meet high performance requirements. Feature extraction and selection using DL methods are automatically accomplished during training process. This feature makes DL-based methods an ideal approach for detecting DDoS attacks. Another characteristic of DL-based methods is that, compared to traditional machine learning methods such as random forests, support vector machines, and KNN, deep learning has stronger learning ability. Therefore, they need to learn highly complex patterns to achieve higher accuracy and more functionality (such as key link analysis and attack types classification) within a single framework. Considering these two characteristics, DL model in this paper consists of two stages: feature processing and model detection. The feature processing stage groups and formats the input data, simplifying the complex and immense feature selection process in machine learning methods. Additionally, this stage obtains multiple DDoS attack vectors, solving the bottleneck problem of processing attack vectors. The model detection stage inputs the processed features into a deep learning model to detect whether the input data grouping is a DDoS attack grouping.

The contributions of this paper are as follows:

1) This paper proposes a light-weight classification framework based on SAE-CNN for detection of DDoS attacks (SCD), which can acquire a deep and full-ranged understanding of the network traffic data. The proposed framework’s classification efficiency is evaluated on a publicly available dataset,

demonstrating significant improvements compared to other classic methods.

2) We extract effective features from network traffic data through our preprocessing technology, and then convert them into grayscale images. As an ideal input mode for deep learning, its application can improve the accuracy of classification and thus improve the efficiency of DDoS attacks detection.

2. Related Work

This section highlights the latest state-of-the-art works in DDoS attacks detection that have been proposed in recent years. Most previous research on DDoS attack defense techniques is implemented using network intrusion detection methods [5]. However, with the development of DDoS attack techniques, the situation of DDoS attack defense research is severe. There are many challenges in the detection of new DDoS attacks, especially regarding network traffic processing. Awan et al. [6] employs two machine learning methods, Random Forest (RF) and Multi-Layer Perceptron (MLP), to detect denial of service attacks through the Scikit ML library and the big data framework Spark ML library. Tuan et al. [7] evaluates the performance of machine learning methods such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), Naive Bayes (NB), Decision Trees (DT), and Unsupervised Learning (USML) in terms of accuracy, false alarm rate (FAR), sensitivity, specificity, false positive rate (FPR), AUC, and Matthews Correlation Coefficient (MCC) on the well-known public datasets UNBS-NB15 and KDD99 for detecting zombie network DDoS attacks. Prasad et al. [8] demonstrates DDoS anomaly detection on the open CIC-DDoS2019 dataset using the Stochastic Gradient Boosting (SGB) machine learning model. Alduailij et al. [9] proposed a method for detecting DDoS attacks in cloud computing, applying two feature selection techniques, the Mutual Information (MI) and Random Forest Feature Importance (RFFI).

While machine learning offers an effective approach for DDoS attacks detection, some researchers argue that the accuracy of detection, as demonstrated in previous studies [10-11], tends to decrease as the size of datasets, often characterized by high-dimensional features, increases. Furthermore, these methods pose challenges when manual feature extraction is needed for raw or unlabeled datasets.

Doriguzzi et al. [12] introduced a novel approach to DDoS attacks detection that leverages the features of CNN to categorize traffic as either malicious or benign. Nandi et al. [13] utilized five prevalent feature selection techniques, namely Information Gain, Gain Ratio, Chi-Squared, Relief, and Symmetric Uncertainty, to select the most relevant attributes from the NSLKDD dataset. Subsequently, DDoS packets were separated from the dataset, and Weka was used to discretize the dataset. Employing the same set of classifiers on the NSLDDoS dataset yielded an average DDoS detection rate of 99%. K. Sadaf et al. [14] achieved an accuracy of 88.98% by automating threshold learning for anomaly detection in an autoencoder-based model, through the combination with the unsupervised learning technique of Isolation Forest. Additionally, Can et al. [15] enhanced the unbalanced dataset of CIC-DDoS2019 by using automatic feature selection to solve the imbalance problem in the dataset. The voluminous amount of network traffic data poses a significant challenge to traditional DDoS detection techniques. Traditional feature selection and classification algorithms fail to adapt to the massive data environment, as the detection performance is closely linked to the chosen features and classifiers. There is a relative paucity of research on distinguishing different types of DDoS attacks, and the imbalanced raw traffic data significantly impairs the classification results.

Inspired by the aforementioned research, we propose SCD, a lightweight DDoS attack detection model. This model not only efficiently extracts features from input traffic data, but also classifies multiple types of DDoS attacks.

3. METHODOLOGY

The network fusion structure of the proposed model is shown in Figure 1. SCD integrates advanced deep learning theories, especially SAE and one-dimensional convolution. The optimal structure for detecting DDoS attacks is achieved by merging model operations. The issue of imbalanced datasets is addressed through preprocessing, and the dimension reduction capability of SAE is efficiently combined with the classification ability of the CNN network. The integration of all three aspects enables a thorough and comprehensive understanding of attack traffic.

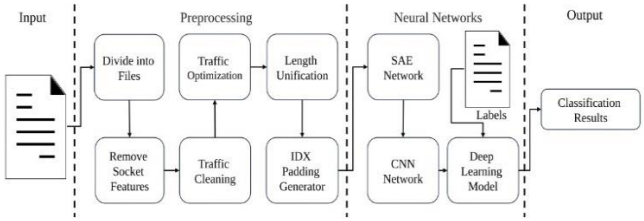

Figure 1. Architecture of vehicular networks.

3.1. Data Preprocessing

The preprocessing procedure is of critical significance. There are three rationales for preprocessing the raw data: (1) the diverse lengths of the raw data collected from the network pose challenges for deep learning models, which require standardized input formats; (2) certain information contained in the raw data, such as port numbers, may adversely affect the accuracy of the results; (3) a standardized format enables streamlined maintenance and facilitates subsequent tasks. The raw data comes from PACP files in the CIC-DDoS2019 dataset, and through our preprocessing technology, we ultimately generate 13 types of grayscale images.

All data packets are defined as a collection and divided into multiple files based on IP addresses and quintuples (source host, destination host, source port, destination port, and transport protocol). Each file corresponds to a session. The original data stream is partitioned into smaller files, and features are extracted from the data of each session stream.

Remove socket features. We remove all of the socket features like destination IP, destination port, timestamp, and flow ID. These features vary from network to network, and we need to train the model with packet characteristics itself. Furthermore, both intruders and normal users can share the same IP address. Therefore, training the DL model with socket information can cause an overfitting problem, as the model can be biased to the socket information. So we remove all unnecessary features.

Traffic cleaning. This is a crucial step in eliminating interference data from traffic packets, removing irrelevant features, and retaining useful ones. The original data often contains missing (NaN) and infinity values that could affect the learning process of DL models. Therefore, these values are removed from the data during the preprocessing phase.

Traffic optimization. This includes removing duplicate and empty files that may impair the learning ability of the framework.

Length unification. We truncate PACP files larger than 784 bytes to 784 bytes and padding files smaller than 784 bytes with 0x00s at the end to supplement them to 784 bytes.

The IDX padding generator. We convert these uniformly sized PCAP files into IDX file of size 28 bytes x 28 bytes, which is a common file format used in the field of deep learning. If mapped to [0,1], these files are treated as grayscale images.

After preprocessing these operations and processing benign traffic data streams, a total of 77,460 grayscale images are generated and a total of 13 types sample images of DDoS attacks are shown in Figure 2.

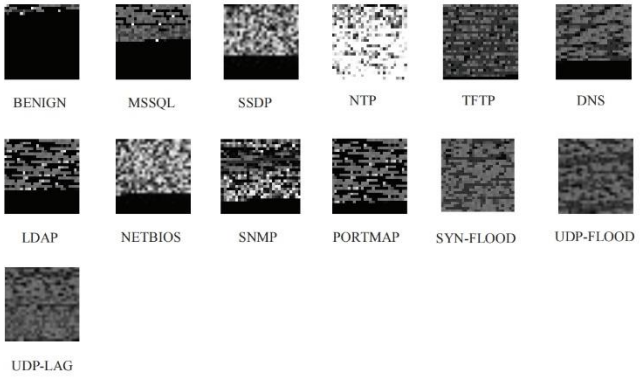

Figure 2. 13 types of grayscale images.

By analyzing these grayscale images, we arrive at the conclusion that various categories of DDoS attacks exhibit a notable level of differentiation, and each category displays a considerable degree of consistency. Therefore, it provides an effective prerequisite for high accuracy DDoS attack multi-classification task.

3.2. SCD Architecture

In this section, we will introduce the architecture of SCD.

The autoencoder is a type of artificial neural network that has found widespread use in a variety of fields, including data noise filtering and image processing. The rationale for employing autoencoders in this work is that, when compared to linear and kernel principal component analysis (PCA) [16], autoencoder can efficiently reduce the dimensionality of input traffic data and significantly enhance the accuracy of detecting subtle anomalies in normal traffic. Additionally, autoencoders are simpler to train and do not require the computationally demanding calculations that kernel PCA does. The SAE is a multi-layer network structure that is formed by linking multiple simple autoencoders together. Figure 3 shows the structure of a single-layer autoencoder, which includes an input layer, a hidden layer, and an output layer.

In the SAE framework, data dimensionality reduction is achieved through the encoding and decoding process. Specifically, the input traffic data is represented by $e=\{e_1,e_2,...,e_n\}$, the hidden layer output is represented by $G_{encoder}=\{g_1,g_2,...,g_m\}$, and the autoencoder output is represented by $p_{decoder}=\{p_1,p_2,...,p_n\}$. During the encoding and decoding process of SAE, the input data undergoes transformations according to formulas (1) and (2):

$$G_{encoder}=f\left(W_a e_n+b_a\right) \tag{1}$$

$$P_{decoder}=h\left(W_\beta G_{encoder}+b_\beta\right) \tag{2}$$

where $f()$ and $h()$ represent activation functions, with sigmoid function being commonly selected. The weight matrix W_α represents the connections between input layer neurons and hidden layer neurons, while b_α represents the offset between input layer neurons and hidden layer neurons. Similarly, weight matrix W_β denotes the connections between output layer neurons and hidden layer neurons, while bias b_β is the offset between output layer neurons and hidden layer neurons. The autoencoder network's training process involves continuous training of parameters $\{W_a,W_\beta\}$ and $\{\mathrm{b}_a,\mathrm{b}_\beta\}$ through forward and backpropagation algorithms to minimize the loss function.

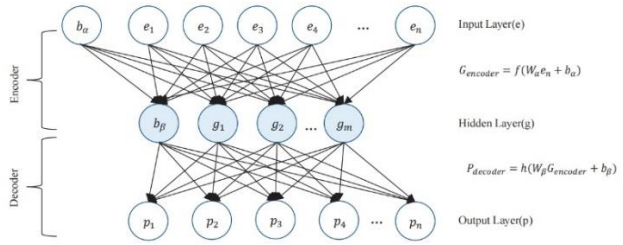

Figure 3. A single layer autoencoder structure in our SAE network.

SAE extracts effective features of data, achieving effective dimensionality reduction and input signal reconstructions in the hidden layer. Upon analyzing SAE and CNN, it can be seen that they both utilize similar principles of forward propagation and error backpropagation based on gradient descent, providing a theoretical basis for network fusion during training. Combining SAE and CNN enables the full utilization of SAE's data dimensionality reduction and CNN's feature extraction and classification abilities, allowing for the detection of DDoS attacks in a single network model. One-dimensional convolution is believed to be an ideal choice for DDoS attacks detection as it captures the spatial dependence between adjacent bytes in network packets and finds discriminative patterns for each attack type, accurately classifying DDoS attack types. The detailed workflow of CNN is shown in Figure 4. Our classification results confirm the effectiveness of one-dimensional convolution in feature extraction for network traffic data. Formula (3) represents the one-dimensional convolution operation:

$$y^j=\gamma\left(\sum_i k^{ij}*x^j\right) \tag{3}$$

the third equation of the model is denoted by y^j, where x^j represents the j_{th} of the input feature map, and i denotes the one-dimensional convolution layer. The convolution kernel k^{ij} is utilized, and the convolution operation is performed with "same" padding, denoted by $*$. The activation function $\gamma()$ used for non-linear processing is rectified linear units (ReLU), and no bias is incorporated into the convolution layer.

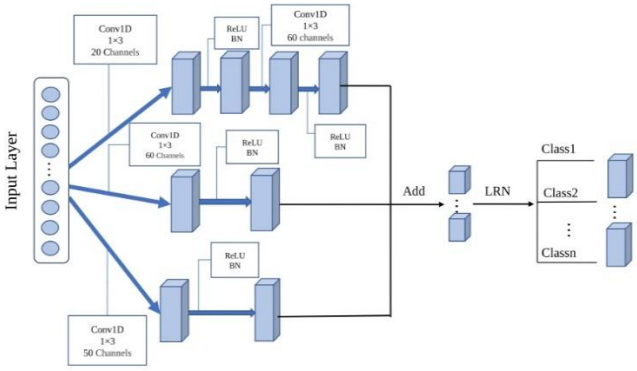

Figure 4. The detailed workflow of CNN

By integrating the robust data feature extraction capability of SAE and the classification ability of CNN. The network architecture is composed of four fundamental parts: data input, data dimension reduction, CNN feature extraction, and data classification. Notably, the high-dimensional vector increases the number of parameters for the network model storage and operation, leading to elevated training time cost. To address issues arising from vectors with high dimensions, a SAE is constructed to reduce the input data dimensionality from 784 to 256-128-64-24. For the CNN feature extraction, a three-branch parallel one-dimensional convolutional neural network structure is constructed based on the Inception module structure. Multiple parallel CNN branches are utilized to extract features with different receptive fields. Specifically, Branch 1 employs 20 1×3 convolution kernels followed by 60 1×3 convolution kernels, Branch 2 uses 60 1×3 convolution kernels, and Branch 3 uses 50 1×3 convolution kernels. After each convolution operation, batch normalization (BN) layer and ReLU layer are added. In the data classification part, the Add operation is utilized to accumulate the extracted feature activation values from the three branches. Local Response Normalization (LRN) is incorporated to penalize abnormal responses for better generalization. Finally, the Softmax classifier is used to obtain the probability results of n classes for classification. Then, in the last stage of the one-dimensional CNN-based SCD classifier, the Softmax classifier obtains the output label. Softmax classifier is defined as follows:

$$L = \frac{\exp(Output^m)}{\sum \exp(Output^i)} \tag{4}$$

where $Output^m$ is the output data of m_{th} neuron in the connected layer. $L = \{l_1, l_2, l_3, ..., l_N\}$ is the complete set of classes and N denotes the total number of classes. The output with the highest probability indicates the class of the input value. To be notified that we used Adam optimizer in SCD.

4. Experimental Analysis

4.1. Environment and Metrics

PyTorch serves as the experimental software framework and operates on a Windows 64-bit operating system. The server consists of an R7 5800H CPU and 16GB of memory, with an Encoder Decoder NVIDIA GeForce RTX 2060 GPU employed as an accelerator. A random selection of 10% of the data is utilized as the testing dataset, while the remaining data is utilized for training purposes. The conventional performance metrics, including accuracy (Acc), precision (Prec), recall (Recall), and F1 score (F1), are employed and defined as follows: $\text{Acc}=\frac{TP+TN}{TP+FP+FN+TN}$, $\text{Prec}=\frac{TP}{TP+FP}$, $\text{Recall}=\frac{TP}{TP+FN}$, $F1=\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$. Where TP is True Positive, namely the number of correctly classified cases as a specific class; FP is False Positive, namely the number of misclassified cases that classified as that class; FN, False Negative, which is the number of cases that are supposed to be classified as that class, yet misclassified as other classes; TN, True Negative, which is the number of cases that correctly classified as not that specific class.

4.2. Datasets

This paper utilizes a current and all-inclusive DDoS attack dataset known as CIC-DDoS2019 [17], which was developed by the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick. The dataset was generated daily and consists of network traffic flow that has been analyzed and labeled as either benign or as a DDoS attack. The CIC-DDoS2019 dataset serves as a representation of real-world data. In this paper, we have classified DDoS attacks into two major categories: reflection-based DDoS attacks and exploitation-based DDoS attacks. Furthermore, we have further divided these attacks into 12 subcategories (MSSQL, SSDP, NTP, TFTP, DNS, LDAP, NETBIOS, SNMP, PORT MAP, SYN-FLOOD, UDP-FLOOD and UDP-LAG), in addition to benign traffic, resulting in a total of 13 classes. Therefore, the experimental process can be simplified into a multi-classification task for network traffic under a DDoS attack, which serves as experiment scenario one. This paper also merges all attack labels into a “DDoS” label combined with the benign label to perform a binary classification task for network traffic under a DDoS attack, which serves as experiment scenario two.

After analysis, there is a problem of data imbalance in the CIC-DDoS2019 dataset, while the PORTMAP, Benign Traffic, and UDP-LAG categories have small percentages of occurrence. This imbalance can lead to a long-tail effect during model training, biasing the classification results towards the majority classes and resulting in reduced sensitivity towards the minority classes, ultimately leading to subpar classification performance. To address this issue, we performed undersampling on the densely populated TFTP class to mitigate its impact on the overall model. Additionally, we applied the SMOTE oversampling technique to generate synthetic data for the sparse classes, thereby augmenting their influence on the overall model. These series of operations collectively contribute to achieving class balance within the dataset.

The content of the SMOTE algorithm can be described as follows: For each sample S in the minority class, the algorithm calculates the Euclidean distance to its nearest neighbors within the same class, obtaining the K-nearest neighbors for that minority class. The synthetic data point S_S is generated by taking the difference between sample S and its closest neighbor S_n multiplying it by a random number r within the range of 0 to 1, and finally adding this difference to the vector of sample S. The formula for the SMOTE algorithm is represented by Equation (5):

$$S_S = S + r\times(\text{S-S}_n) \tag{5}$$

After employing this method, the data quantities of all 13

classes, including various attack types as well as benign traffic, are balanced. A balanced dataset effectively mitigates the occurrence of the long-tail effect during the training process, thereby enhancing the classification accuracy of the SCD model for classes with fewer data instances. This ensures that the SCD model maintains high precision in real-world scenarios where the majority of the traffic is benign.

4.3. Settings

Three sets of experiments were conducted in this paper. In the first set, DDoS attacks were classified into multiple categories, and the results were compared with three other graph-based detection methods known for their excellent performance and compared against other deep learning models studied on the same dataset. In the second set of experiments, the effectiveness of the SCD model on the ISCXIDS dataset was validated to ensure the model's robustness. In the third set of experiments, binary classification was performed on network traffic data labeled as either DDoS attack or benign traffic to detect whether DDoS attacks are occurring, and the results were compared with classical detection methods. The fine-tuning perparameters of the neural networks established in this paper are listed in Table **1**.

Table 1. Parameters For Our Neural Networks.

Parameter	Description
CNN parameter	3 one-dimensional layers+ReLU+BN.
SAE parameter	5 layers.
Training epoch	100.
Optimizer	Adam.
Learning rate	$1 * 10^{-3}$.
Loss function	Cross entropy loss.
Total parameter quantity	$1 * 10^{-6}$.

4.4. Results

In experiment scenario one, the detection performance for each attack label was statistically analyzed and compared with other methods. These methods comprise three popular image testing methods on the same dataset in this paper, AlexNet [18] and ResNet-34 [19]. Furthermore, we compare our method with the DDoS multi-classification network based on DNN proposed by Chartuni A et al. [2] and the DDoS multi-classification network based on CNN proposed by de Assis et al. [20], the experimental results of the experiment scenario one is presented in Table 2.

Table 2. Performance Evaluation of SCD Compared to Other Classic Systems in Multi-Classification Task.

System	Acc	Recall	F1(%)
AlexNet	83.26	81.79	82.24
ResNet-34	93.52	92.78	93.23
Chartuni A et al.	94.21	94.03	94.12
de Assis et al.	95.40	95.70	92.80
SCD	**97.22**	**96.79**	**96.84**

In comparison, the SCD system exhibits superior performance in detecting DDoS attacks and in multiclass classification when compared to other deep learning strategies, with an exceptional accuracy rate of 97.22%.

The results of the multi-classification experiment on DDoS attacks are shown in Figure 5. The figure clearly indicates the SCD model's outstanding detection ability in experimental scenario one.

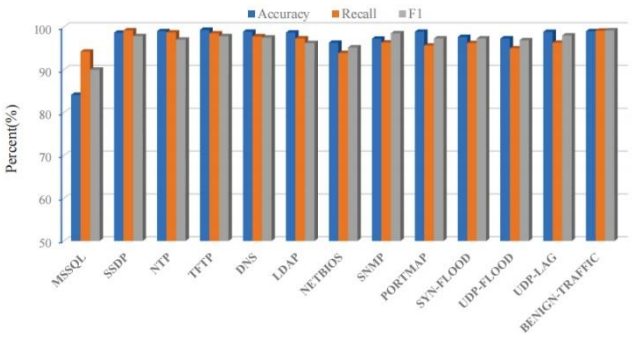

Figure 5. The performance of SCD in classifying each attack types of DDoS in multi-classification task.

The model's detection ability for MSSQL is relatively weak, possibly due to the limited feature attributes used for distinguishing the MSSQL attack category. According to [24], the strong correlation attributes affecting MSSQL detection are port numbers, but the port number attribute cannot be used in model classification tasks and is therefore discarded, which affects the detection performance. The model performed well on other categories due to its relatively small number of parameters and computational complexity, making it suitable for deploying in deep learning environments to detect DDoS attacks. In experiment scenario two, we conducted a comprehensive evaluation of the SCD model's effectiveness using the ISCXIDS dataset, a widely recognized dataset in the field of DDoS attacks. The ISCXIDS dataset encompasses 29 crucial features, including frame. len, frame. protocols, ip.hdr.len, ip.len, and ip.flags.rb. To ensure the model's optimal training, we eliminated irrelevant features from the dataset and applied consistent preprocessing techniques. The resultant data underwent processing and was then utilized as input for the SCD model. Regarding the partitioning of training and testing sets, we allocated 20% of the dataset for testing purposes and utilized the remaining 80% for training. To evaluate the SCD model's efficacy, we performed 100 epochs and documented the Acc and Loss for both the training and testing sets, as illustrated in Figure 6, we smoothed the curve. The figure clearly demonstrates that the training and testing sets reached a stable state after 100 epochs, exhibiting exceptional accuracy rates and minimal loss. Notably, the training set achieved an accuracy of 99.3% with a loss rate of 0.35%, while the testing set achieved an accuracy of 99% with a loss rate of 0.53%.

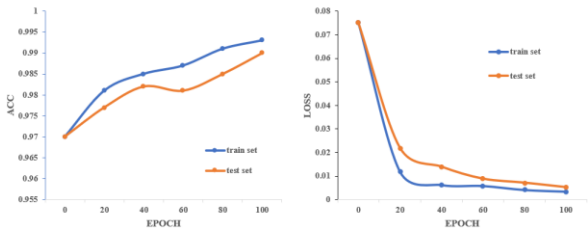

Figure 6. The Acc and Loss rates on the training and testing sets.

The experimental data confirmed the effectiveness of the SCD model on other datasets, demonstrating its robustness. In experiment scenario three, a model was trained to effectively detect DDoS attacks, and the network traffic data with DDoS attack labels and benign labels can be classified into two categories to detect possible DDoS attacks. We compared our model with various classical techniques, evaluating six different algorithms, including Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and

Logistic Regression. The detection results for experiment scenario two are shown in Figure 7. The SCD model outperforms other algorithms in the binary classification task, with an exceptional accuracy rate of 99.1%.

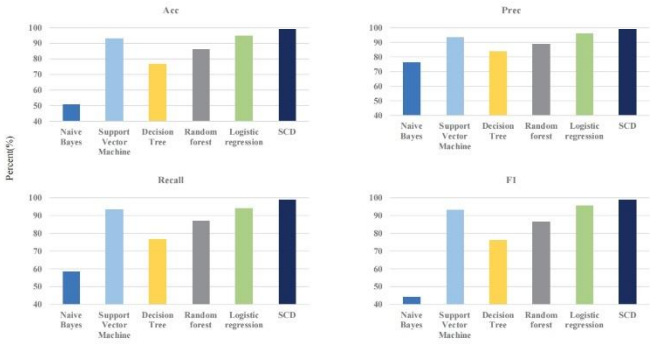

Figure 7. Performance evaluation of SCD compared to other classic algorithms in binary classification task in terms of Acc, Prec, Recall, and F1

Overall, all three experiments demonstrate more substantial improvements compared to the current state-of-the-art method. Therefore, the SCD model proposed in this paper can effectively distinguish DDoS attack types.

5. Conclusion

DDoS attacks are considered one of the most aggressive types of attacks in recent years, causing serious impacts on the entire network system. This paper proposes a new DL-based model, namely SCD, for detecting DDoS attacks. To simulate the network traffic data under the conditions of significant network fluctuations caused by DDoS attacks, the CIC-DDoS2019 dataset is employed in this paper. To address the issues of dirty data, large attribute value intervals, and imbalanced categories, a targeted data preprocessing method is proposed through experiments. The SCD model utilizes both the dimensionality reduction advantage of SAE and the CNN network, achieving lightweight and precise modeling. The results show that the SCD model achieves an accuracy of 97.2% for the multi-classification task and an accuracy of 99.1% for the binary classification task. However, further research is needed to investigate the performance of the proposed model on other datasets.

References

[1] R Amrish, K Bavapriyan, V Gopinaath, A Jawahar, and C Vinoth Kumar. Ddos detection using machine learning techniques. Journal of IoT in Social, Mobile, Analytics, and Cloud, 4(1):24–32, 2022.

[2] Andr´es Chartuni and Jos´ e M´ arquez. Multi-classifier of ddos attacks in computer networks built on neural networks. Applied Sciences, 11(22):10609, 2021.

[3] Firooz B Saghezchi, Georgios Mantas, Manuel A Violas, A Manuel de Oliveira Duarte, and Jonathan Rodriguez. Machine learning for ddos attack detection in industry 4.0 cppss. Electronics, 11(4):602, 2022.

[4] Meenakshi Mittal, Krishan Kumar, and Sunny Behal. Deep learning approaches for detecting ddos attacks: A systematic review. Soft Computing, pages 1–37, 2022.

[5] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In 2017 IEEE 60th international Midwest symposium on circuits and systems (MWSCAS), pages 1597–1600. IEEE, 2017.

[6] Mazhar Javed Awan, Umar Farooq, Hafiz Muhammad Aqeel Babar, Awais Yasin, Haitham Nobanee, Muzammil Hussain, Owais Hakeem, and Azlan Mohd Zain. Real-time ddos attack detection system using big data approach. Sustainability, 13(19):10743, 2021.

[7] Tong Anh Tuan, Hoang Viet Long, Le Hoang Son, Raghvendra Kumar, Ishaani Priyadarshini, and Nguyen Thi Kim Son. Performance evaluation of botnet ddos attack detection using machine learning. Evolutionary Intelligence, 13:283–294, 2020.

[8] M Devendra Prasad, V Prasanta Babu, and C Amarnath. Machine learning ddos detection using stochastic gradient boosting. Int. J.Comput. Sci. Eng, 7(4):157–16, 2019.

[9] Mona Alduailij, Qazi Waqas Khan, Muhammad Tahir, Muhammad Sardaraz, Mai Alduailij, and Fazila Malik. Machine-learning-based ddos attack detection using mutual information and random forest feature importance method. Symmetry, 14(6):1095, 2022.

[10] Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber security. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[11] Mohammed K Alzaylaee, Suleiman Y Y erima, and Sakir Sezer. Dl-droid: Deep learning based android malware detection using real devices. Computers & Security, 89:101663, 2020.

[12] Roberto Doriguzzi-Corin, Stuart Millar, Sandra Scott-Hayward, Jesus Martinez-del Rincon, and Domenico Siracusa. Lucid: A practical,lightweight deep learning solution for ddos attack detection. IEEE Transactions on Network and Service Management, 17(2):876–889, 2020.

[13] Suman Nandi, Santanu Phadikar, and Koushik Majumder. Detection of ddos attack and classification using a hybrid approach. In 2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP), pages 41–47. IEEE, 2020.

[14] Kishwar Sadaf and Jabeen Sultana. Intrusion detection based on autoencoder and isolation forest in fog computing. IEEE Access, 8:167059–167068, 2020.

[15] Duy-Cat Can, Hoang-Quynh Le, and Quang-Thuy Ha. Detection of distributed denial of service attacks using automatic feature selectionwith enhancement for imbalance dataset. In Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, April 7–10, 2021, Proceedings 13, pages 386–398. Springer, 2021.

[16] Sonam Salaria, Sakshi Arora, Nishita Goyal, Pooja Goyal, and Shifaly Sharma. Implementation and analysis of an improved pca technique for ddos detection. In 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), pages 280–285, 2020.

[17] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A Ghorbani. Developing realistic distributed denial of service (ddos) attackdataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST), pages 1–8. IEEE, 2019.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[20] Marcos V . O. de Assis, L. F. Carvalho, J. Rodrigues, Jaime Lloret, and M. L. Proenc ¸a. Near real-time security system applied to sdn environments in iot networks using convolutional neural network. Comput. Electr. Eng., 86:106738, 2020.