Performance Comparison of Machine Learning Models for DDoS Attacks Detection

Panida Khuphiran, Pattara Leelaprute,
Putchong Uthayopas
Department of Computer Engineering, Kasetsart University
Bangkok, Thailand
panida.khu, putchong@ku.th,
pattara.l@ku.ac.th

Kohei Ichikawa, Wassapon Watanakeesuntorn
Graduate School of Science and Technology, NAIST
Nara, Japan
ichikawa@is.naist.jp,
wassapon.watanakeesuntorn.wq0@is.naist.jp

***Abstract*—Distributed denial of service (DDoS) attack is one of the most costly attacks for IT system in terms of time and money. In this paper, the use of machine learning algorithms for DDoS detection has been addressed. The traditional SVM and new emerging deep learning algorithm, namely Deep Feed Forward (DFF), are evaluated. The DARPA Scalable Network Monitoring and DARPA 2009 DDoS attacks dataset is used to test the effectiveness of these two algorithms. The dataset is preprocessed to find the potential speedup of the classification process. From the experiments, DFF deep learning algorithm has achieved a high accuracy of 99.63% with the training time of 289.614 secs. For SVM, the highest accuracy achieved is 93.01%, with the training time of 371.118 secs. Anyway, SVM is able to deliver a faster classification time. Therefore, DFF is suitable for the situation when accuracy is the main concern while SVM can be used when speed of classification is a critical factor.**

***Keywords*—Support vector machine, Deep learning, DDoS attacks**

I. INTRODUCTION

CYber attacks are attempts to gain an illegal access to the computer systems for the purpose of causing damage. Statistics of cyber attacks during 2015-2017 show that the cybercrime is still increasing [1]. Distributed denial-of-service attack or DDoS attack, is one of the most used techniques employed [1]. DDoS attack uses multiple machines to construct the flooding of packets directed to a target machine to make the service disruption on the target machine. This can cause a serious and costly service problem to its users and administrators. Thus, many researchers conduct extensive studies trying to develop techniques and tools to detect the potential DDoS attack as early as possible. One of the main approaches in detecting the DDoS attack is to use various kinds of machine learning techniques. With the recent development of deep learning technique and powerful GPU hardware, there is a great potential to develop tools and techniques to detect the DDoS attack much faster. Moreover, the use of SDN (Software Defined Network) enables many new approaches in handling and isolating the DDoS attack. In this paper, the effectiveness of two techniques for DDoS attack analysis has been evaluated by using DARPA2009 dataset for training set and testing set. One is based on traditional SVM, another one is based on the deep feed forward (DFF) technique. The evaluation results enable developers to understand how to select an appropriate algorithm to use for DDoS detection.

The rest of the paper is organized as follows. First, the related works are described in Section 2. Then, Section 3 presents the approach used in this work. Section 4 discussed the experiments conducted and the results obtained. Finally, Section 5 presents the conclusion and possible future works.

II. RELATED WORKS

SVM is one of the most widely used machine learning algorithms for classification problems with a broad application in health [2], science [3] - [4], and network [5]. Recently, deep learning algorithms that utilize deep hierarchical layers of neural network are gaining much more attention among researchers. Many works [2] - [5] have been conducted to compare the performance and accuracy of these two algorithms for various applications. For instance, the development of Myocardial Infarction [2] detection also compares the performance of SVM with ANN algorithm. In their study, LIBSVM is used for the SVM classification. Another work on an intrusion detection system based on anomaly detection is also experimented with these algorithms and comparing their performance and accuracy. The work on a development of a detection tool for Software-Defined Networking (SDN) uses some features from NSL-KDD Dataset [6] and a DDoS detectable tool [7] - [8]. One of the key performance areas is the performance of the library used for machine learning. Some comparison studies on SVM [9] - [10] found that LIBSVM may not be the best option for SVM algorithm implementation and Thundersvm is utilized for SVM based classification.

Many works [11] - [12] are also conducted to compare the deep learning frameworks. In the study, Theano, Torch, Caffe, Tensorflow, and Deeplearning4J were evaluated. The results showed that Keras used less time for training models when tuning with large epochs and got higher accuracy. Hence, this work decided to employ Keras with Tensorflow as the backend. For the model evaluation, precision, recall, accuracy, and F1 score are selected as the metrics based on the studies done in [2] - [5]. Some features were determined based on the previous researches about the development of detection tools using information from packets [6] - [8], [13]. Through

978-1-5386-8164-0/18/31.002018$IEEE$

the evaluations of this study, we aim to provide appropriate information to choose an optimal DDoS detection algorithms in terms of accuracy and performance.

III. APPROACH

In this work, the objective is to investigate the effectiveness of two algorithms, namely, SVM and deep feed forward (DFF). The standard test data, the 2009 DARPA Intrusion Detection dataset [14] - [15] is used for the evaluation. This dataset contains 10 days network traffic including HTTP, SMTP, and DNS background traffic. The dataset consists of 7000 pcap files. In this work, DARPA-2009 DDoS Attack-20091105 [16] was assigned as a sample DDoS attack dataset. This dataset contains about 6 minutes of SYN flood DDoS attack in 3 pcap files. For the normal packet dataset, we selected some pcap files from the original dataset, that were collected from the same day, based on information from the ground truth table. The ground truth table shows dates, start-end times and worth traffic descriptions for each attack traffic. In order to conduct the analysis, some preprocessing is needed as shown in Fig 1.

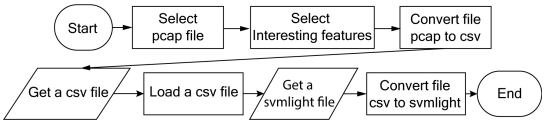

Fig. 1. Overview of pre process data procedure

In our work, we create a machine learning model that detects when a DDoS attack is underway. We aggregate packet information for a certain time window, in our case, 1 second. The machine learning model analyzes the patterns on the number of transferred packets, the number of observed IP addresses and so on over the time windows, and classifies each time window into the time period where a DDoS attack is underway or the time period where no attack is detected. In this work, we have two types of features we are interested in. Table I and Table II present the information for these two types of features respectively.

In the Table I, the first type of network features are aggregated for each time window. In Table II, the second type of features on each packet are added with the previous window-based information.

The total numbers of features of each type are 16 features and 26 features respectively. After pre-processing the datasets, those files are converted into a file in Comma-separated values (CSV) format. Finally, the CSV file is converted into Svmlight format for using with SVM libraries.

SVM is a traditional classification algorithm that can be used to solve both supervised and unsupervised machine learning problems. The tuning of hyper parameters of SVM depends on selected kernel functions. In this research, we selected linear, radial basis function (rbf), polynomial and sigmoid as the kernel functions.

From those kernel equations, the tuning parameters are determined as degree, gamma and coef0. The parameter of

TABLE I
INTERESTING FEATURES DESCRIPTION SECOND TYPE

Feature	**Descriptions**
Status	*The network situation. Attacked by DDoS or not.*
All_Packets	*Number of arrived packets per a time window*
Num_IPpair	*Number of different IP source and IP destination address pairs per a time window*
Num_IPsrc	*Number of different IP source addresses per a time window*
Num_IPdst	*Number of different IP destination addresses per a time window*
Num_Portpair	*Number of different source and destination port pairs per a time window*
Num_Portsrc	*Number of different source ports per a time window*
Num_Portdst	*Number of different destination ports per a time window*
Num_Ether	*Number of packets, where the packet type is ethernet per a time window*
Num_Dot3	*Number of packets, where the packet type is IEEE 802.3 type per a time window*
Num_TCP	*Number of packets, where the packet type is Transmission Control Protocol per a time window*
Num_UDP	*Number of packets, where the packet type is User Datagram Protocol per a time window*
Num_ARP	*Number of packets, where the packet type is Address Resolution Protocol per a time window*
Num_ICMP	*Number of packets, where the packet type is Internet Control Message Protocol type per a time window*
Num_LLC	*Number of packets, where the packet type is logical link control type per a time window*
Num_Len	*Number of different packet lengths per a time window*

TABLE II
INTERESTING FEATURES DESCRIPTION SECOND TYPE

Feature	**Descriptions**
Ether_or_Dot3	*The packet type is ethernet or IEEE 802.3 type*
MAC_src	*MAC address of the source device*
MAC_dst	*MAC address of the destination device*
Ether_type	*The type of ethernet packet,(e.g. 0x0800 is Internet Protocol version 4 (IPv4))*
LLC	*The packet is logical link control (LLC) type or not*
LLC_ssap	*The source SAP field that represents the logical address of LLC packet*
LLC_dsap	*The destination SAP field that represents the logical address of LLC packet*
IP_ttl	*Time to live of the packet*
IP_version	*The version of IP, such as IPv4*
TCP	*The packet is TCP type or not*
UDP	*The packet is UDP type or not*
ARP	*The packet is ARP type or not*
ICMP	*The packet is ICMP type or not*
pLen	*The packet size*
Status	*The network situation is attacked by DDoS or not*
num_ip_pair	*The number of the pair of source ip and destination ip*
all_packets	*Total of packets arrival per a window*
ratio_ip	*Ratio of the number of IP sources divided by the number of IP destinations*
num_ip_src	*Number of ip source*
num_ip_dst	*Number of ip destination*
num_port_pair	*The number of the pair of source port and destination port*
ratio_port	*Ratio of the number of source ports divided by the number of destination ports*
num_port_src	*Number of port source*
num_port_dst	*Number of port destination*
weight_ip	*Weight of the number of top three IP pairs*
weight_port	*Weight of the number of top three port pairs*

degree indicates the dimension. In this work, we used one, two, and three accordingly as the parameter of degree. The C-Support Vector Machine Classification (SVC) is used for the SVM model. SVM uses C parameter to avoid some misclassification in the model. In this experiment, we used 1.0 for the parameter by default.

Deep learning is a class of machine learning model that uses multiple layers of neural networks to solve the problems. In this work, the deep feed forward (DFF) algorithm [17] is used. Fig 2 shows the configuration of the used DFF.

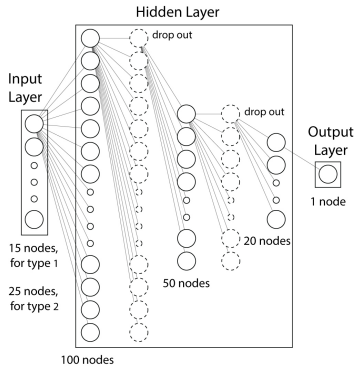

Fig. 2. Simple deep feed forward (DFF) neuron network

In our experiment, we generated the training set and testing set by combining the data of the time windows under the DDoS attack situation and normal situation. The number of samples are shown belows in the Table III.

TABLE III
THE NUMBER OF SAMPLES FOR EACH TYPE

Dataset	DDoS attack	Normal	Total
Window-basis	335	365	700
Packet-basis(S)	331	369	700
Packet-basis(L)	481,903	518,097	1,000,000

Window-basis dataset denotes the dataset in the type of Table I. Packet basis dataset denotes the dataset in the type of Table II. We have two different sample size datasets for packet-basis data, Packet-basis(S) and (L). Packet-basis(S) is prepared for the purpose of comparing the performance with window-basis dataset so that it has the same sample size.

IV. Experimental Results and Discussion

In this work, the experiments were conducted on a server with Ubuntu version 18.04 and GTX 1080 GPU card. The experiments were performed using Keras with Tensorflow backend for deep learning and thundersvm for SVM. The testing data set is 10% of dataset randomly selected based on k-fold validation [18]. The remaining data is used as the training set. The accuracy, recall, precision and F1 score is used as the performance metrics. These values are calculated with k-fold validation step to reduce the possible overfitting in the model. The calculate formula is below in (1), (2), (3), and (4) [19]

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{1}$$

$$Recall(R) = \frac{tp}{tp + fn} \tag{2}$$

$$Precision(P) = \frac{tp}{tp + fp} \tag{3}$$

$$F1score = 2\frac{PxR}{P + R} \tag{4}$$

The results of the experiment of SVM are shown in Table IV and Table V. In this Tables, w denotes the window-basis dataset and p denotes the packet-basis dataset.

TABLE IV
THE BEST ACCURACY FROM EACH KERNELS

Kernel	Accuracy	Recall	Precision	F1-score
Linear (w)	76.00%	0.765	0.744	0.754
Linear (p)	90.28%	0.925	0.883	0.902
Polynomial(w)	93.01%	0.922	0.933	0.927
Polynomial(p)	92.58%	0.894	0.933	0.906
Sigmoid (w)	47.86%	1.000	0.480	0.650
Sigmoid (p)	47.29%	1.000	0.472	0.642
RBF (w)	60.05%	0.912	0.599	0.661
RBF (p)	48.43%	1.000	0.479	0.648

TABLE V
TIME USED FOR EACH KERNELS

Kernel	Training time(s)	Testing time(s)
Linear (w)	331.6156	0.0033
Linear (p)	30.6892	0.0030
Polynomial(w)	371.1180	0.0029
Polynomial(p)	379.4168	0.0031
Sigmoid (w)	0.0507	0.0023
Sigmoid (p)	0.0500	0.0025
RBF (w)	5.9043	0.0038
RBF (p)	172.5352	0.0034

From Table IV, polynomial kernel has achieved the highest accuracy for overall. The different types of data affect with some kernels especially with RBF and Linear kernel. In the experiments of polynomial kernel, we also plotted graphs to find relationships between the parameter of degrees and accuracy and calculation time. The graphs are shown below in Fig 3 (a), (b), (c).

The results from Table V shows that sigmoid takes the least time to establishing the model. However, the model performance is not good enough. From Fig 3, we found that when degree increased, training time and accuracy tend to increase while predict time tends to decrease. Accordingly, the best results in terms of accuracy from each algorithm are shown below in Table VI. A indicates accuracy, R indicates recall and P indicates precision.

From our experiment, Deep Learning got the highest accuracy of 99.63%, recall of 0.994, precision with 0.998 and F1-score is 0.996 while SVM got an accuracy rate of 81.23%, recall of 0.927, precision of 0.756 and F1-score is 0.826 when considered with the same data size and data type.

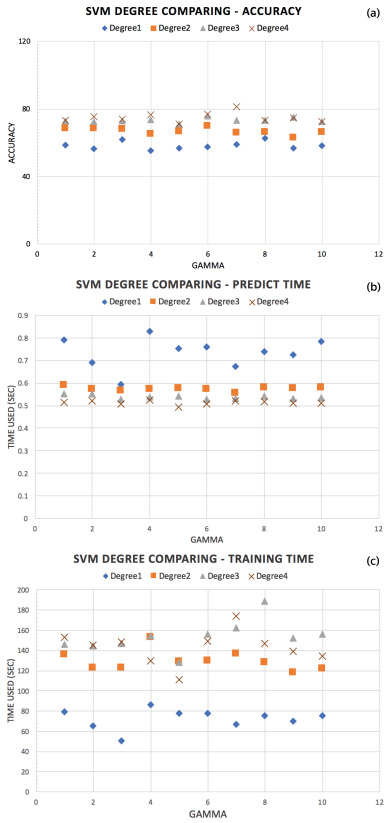

Fig. 3. (a) The relationship between increasing degree and accuracy (b) The relationship between increasing degree and calculation time for testing the model (c) Show the relationship between increasing degree and calculation time for training the model

TABLE VI
THE BEST RESULTS IN TERMS OF ACCURACY FOR EACH ALGORITHM

SVM	Model Performance				Time used (s)	
Data	A(%)	R	P	F1-score	Train	Test
Window-basis	93.01	0.922	0.933	0.927	371.118	0.003
Packet-basis(S)	92.58	0.894	0.933	0.906	379.417	0.003
Packet-basis(L)	81.23	0.927	0.756	0.826	138.260	0.500
Deep learning	Model Performance				Time used (s)	
Data	A(%)	R	P	F1-score	Train	Test
Window-basis	61.30	0.240	0.452	0.295	3.314	0.117
Packet-basis(S)	68.30	0.366	0.922	0.504	3.438	0.115
Packet-basis(L)	99.63	0.994	0.998	0.996	239.614	14.651

V. Conclusion

In this paper, the application of machine learning algorithm for the problem of DDoS attack detection has been addressed. Two algorithms, Support Vector Machine (SVM) and Deep Feed Forward (DFF) have been evaluated to demonstrate the feasibility of applying these algorithms. The experiments have been conducted to compare the performance of these two algorithms. It has been found that DFF can classify the data with a higher accuracy. Therefore, deep learning is a useful choice for the classification of DDoS attack packets in terms of accuracy. However, SVM is an appropriate choice for for faster classification method. In our future work, we plan to examine both two algorithms with the real time data to develop a useful method that is available with the real networks.

Acknowledgment

We use datasets of DARPA Scalable Network Monitoring (SNM) Program Traffic, IMPACT ID USC-LANDER/DARPA_Scalable_Network_Monitoring-20091103/rev8431 . Traces taken 2009-11-03 to 2009-11-12. Provided by the USC/LANDER project (http //www.isi.edu/ant/lander).

References

[1] "2017 cyber attacks statistics," Available at https://www.hackmageddon.com/2018/01/17/2017-cyber-attacks-statistics/ (2018/01/17).
[2] N. AjiBhaskar, "Performance analysis of support vector machine and neural networks in detection of myocardial infarction," *Procedia Computer Science*, vol. 46, pp. 20–30, 2015.
[3] U. F. . J. S. . a. Evgeny Byvatov, and . Gisbert Schneider*, "Comparison of support vector machine and artificial neural network systems for drug/nondrug classification," *American Chemical Society*, vol. 43, pp. 1882–1889, Sep. 2003.
[4] B. M. H. Alhafidh and W. H. Allen, "Comparison and performance analysis of machine learning algorithms for the prediction of human actions in a smart home environment," *ICCDA '17*, 2017.
[5] R. H. Aditya Nur Cahyo and D. Adhipta, "Performance comparison of intrusion detection system based anomaly detection using artificial neural network and support vector machine," *American Institute of Physics*, 2016.
[6] D. M. S. A. R. Z. Tuan A Tang, Lotfi Mhamd and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," *IEEE*, Oct. 2016.
[7] S. T. S. Kokila RT and K. Govindarajan, "Ddos detection and analysis in sdn-based environment using support vector machine classifier," *IEEE*, Dec. 2014.
[8] W. S. Quamar Niyaz and A. Y. Javaid, "A deep learning based ddos detection system in software-defined networking (sdn)," Nov. 2016.
[9] S. P. Behrend, "Design, implementation, and optimization of an advanced i/o framework for parallel support vector machines," Ph.D. dissertation, Reykjavik, 2018.
[10] B. H. Q. L. Zeyi Wen, Jiashuai Shi and J. Chen, "Thundersvm: A fast svm library on gpus and cpus," 2017.
[11] S. K. Vassili Kovalev, Alexander Kalinovsky, "Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?" pp. 99–103, 2016.
[12] "Darpa 2009 ddos attack-20091105," Available at https://github.com/szilard/benchm-dl/blob/master/keras_backend.md (2017/07/14).
[13] N. Moustafa and J. Slay, "Creating novel features to anomaly network detection using darpa-2009 data set," pp. 204–212, Jul. 2015.
[14] "Darpa scalable network monitoring-20091103," Available at https://ant.isi.edu/datasets/readmes/DARPA_Scalable_Network_Monitoring-20091103.README.txt (2018/02/13).
[15] M. Gharaibeh and C. Papadopoulos, "Darpa-2009 intrusion detection dataset report," Tech. Rep.
[16] "Darpa 2009 ddos attack-20091105," Available at https://ant.isi.edu/datasets/readmes/DARPA_2009_DDoS_attack-20091105.README.txt (2014/12/9).
[17] "Introduction-neuron networks," Available at https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/neural_networks.html.
[18] Tzu-TsungWong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," *Elsevier*, vol. 48, no. 9, pp. 2839–2846, Sep. 2015.
[19] D. M. W. Powers, "Evaluation: From precision, recall and f-factor to roc informedness markedness & correlation," South Australia, Tech. Rep., 2007.