Detail Analysis on Machine Learning based Malicious Network Traffic Classification

Sungwoong Yeom, Kyungbaek Kim

Department of Electronics & Computer Engineering,
Chonnam National University
77 Yongbong-ro, Buk-gu, Gwangju, Korea
yeomsw0421@gmail.com, kyungbaekkim@jnu.ac.kr

Abstract

Research of using variety of machine learning techniques to detect malicious traffic is drawing attention recently. In particular, the acceleration of CNN development used in image processing techniques has provided new possibility of network traffic classification. In this paper, we evaluate the performance of machine learning based network traffic classification through CICIDS 2017 dataset in detail. For this detail analysis, we conducted 3-fold cross-validation for Naïve bayes, SVM and CNN based classifier with CICIDS 2017 dataset with accuracy, precision, recall and F-measure. Especially, we analyzed the result of validation in terms of the imbalance and the diversity of training dataset. Through the evaluation results, we show that CNN outperforms traditional machine learning methods in binary classification with sufficient data.

Keywords-component; CNN; Traffic Classification; CICIDS 2017

I. Introduction

As the complexity of network traffic growths with highly developed network technologies, there are many threats, which cause malicious traffic the network, reduce the quality of network services and damage the behavior of specific servers and hosts. Thus, detection and prevention of malicious network traffic of those threats is becoming more important. Analyzing and evaluating malicious traffic helps prevent the cause of attacks.

Recently, as machine learning techniques have matured, research on machine learning-based network traffic classification techniques has drawn attention. These machine learning techniques can generalize data or examples to find out how to perform tasks and learn how to improve themselves from the past data. Packet duration, packet length, time and protocol variables can be used as data to train to learn the network traffic classifier such as Naive Bayes, Decision Tree, SVM, KNN [2] and Random Forest [8]. However, it is difficult to achieve accurate performance when testing with unverified datasets. Therefore, a comprehensive framework is needed for the intrusion detection system benchmarking dataset. In 2016, Gharib et al. [5] identified 11 criteria that are important for building reliable datasets. The 11 criteria of this framework are "Attack Diversity, Anonymity, Available Protocols, Complete Capture, Complete Interaction, Complete Network Configuration, Complete Traffic, Feature Set, Heterogeneity, Labeled Dataset, and Metadata". Researchers at the Canadian Institute of Cyber Security have shown that most datasets are outdated and unreliable for evaluation purposes [11]. CICIDS2017 is a new set of intrusion data, which is not well researched and is likely to contain errors and shortcomings.

This paper show to evaluate the detection performance of malicious network traffic through various machine learning techniques by using CICIDS2017 (an actual benchmark dataset similar to actual network traffic). For the convolution neural network based network traffic classifier, malicious performance is classified by converting a set of data into a standardized image. The classification result of each classifier was verified by 3-fold cross-validation, and the result of each classifier was analyzed accordingly, such as Precision, Recall, and F-measure. In addition, improvement measures were proposed for problems caused by the analysis results.

II. Related Work

A. Naïve Bayes

The Naive Bayes classification technique is a probabilistic classification technique that applies Bayes theory, and the more independence between feature vectors, the better its performance. The Naive Bayes classification algorithm is not as complex as most algorithms, so it can be applied quickly and easily. Although the classification is not more accurate than the complex algorithms, similar results can be obtained. These probabilities can be inferred directly from the data and the attributes of the class can be calculated on the assumption that they are conditionally independent. In fact, this assumption may not be true. Failure to comply with this condition may result in incorrect probability calculations, but these violations may not affect forecast accuracy. Prediction can be accurate even if incorrect probabilities are used in the calculations. Depending on the nature of this probability model, this algorithm can be used in a map learning environment to learn very efficiently.[9]

B. SVM (Support Vector Machine)

SVM is techniques used to categorize datasets with different characteristics using maximum margins among given feature vectors, which have the advantage of stably operating classification techniques by obtaining support Vector between

groups, even if multiple feature vectors are given. SVM is optimizing generalizations that correctly categorize invisible data. This optimization solves the problems that appear in other learning algorithms, such as overfitting. You should learn the SVM just as you have to learn the artificial neural network. Map the educational data of the input space to a higher level of functional space. Determine the linear decision boundary in the feature space by constructing the optimal super plane that distinguishes the class. This allows SVM to obtain nonlinear boundaries in the input space. A support vector is a point in the input space that best defines the boundaries between classes. A kernel function that allows calculations to be performed in the input space avoids potentially difficult calculations in the feature space. The concept of statistical learning theory is used to describe which factors should be controlled for good generalization.[10]

C. CNN (Convolutional Neural Network)

CNN is the latest classification model for image classification, where multiple filters are applied to pixel data in images to extract high-dimensional features and learn the classifiers. At this time, the high-dimensional features extracted exist inside the hidden layer consisting of the Convolution layer, the Pooling layer, and the Fully connected layer, making it difficult to identify detailed information about each feature and to attach any special meaning.

1) Convolution Layer

The Convolution Layer is the core of CNN's design of the block. The parameters of the layer consist of a series of kernels whose incoming fields are small but extend to the depth of the input volume. During forward passage, each filter is converted according to the width and height of the input volume to calculate the dot product between the filter item and the entry and to create a two-dimensional activation map of the filter. As a result, the network learns which filters are enabled when the data entry detects a particular type of function on a given set of data. Stacking activation maps for all filters along depth dimensions forms the entire output volume of the Convolution Layer.

2) Max-Pooling Layer

Pulling is a form of nonlinear down sampling. Max Pooling is the most commonly used, and divides the input into a set of non-overlapping rectangles and outputs a maximum value for each sub-area. Intuitively, the exact location of features is less important than the rough location compared to other features, thus gradually reducing the size of the space in the kernel, reducing the number of parameters and computations in the network and controlling overfitting.

3) Flattening and Fully Connected layer

At the end of the various Convolution, Max-Pooling layers, high levels of inference in the neural network are performed through the Fully-Connected layer. The neurons in the Fully-Connected layer are linked to all activation values in the previous layer, as can be seen in a general artificial neural network. However, in order to pass two-dimensional resources onto these Fully-Connected layers from various previous layers, one-dimensional data must be changed. This is the Flatten layer and produces a vector.

4) Softmax

Inside the vector of the Fully Connected layer is a score for classes. Softmax converts the network's non-normalized output to probabilities and makes the sum of these probabilities equal to 1. In other words, Softmax is used in neural networks to map the probability distribution to the predicted output class.

Table I. Components of Friday Afternoon Dataset

Label	Record Count
BENIGN	97718
DDoS	128027

Table II. Components of Friday Afternoon Dataset

Label	Record Count
BENIGN	127537
PortScan	158930

Table III. Components of Friday Morning Dataset

Label	Record Count
BENIGN	189067
Bot	1966

Table Ⅳ. Components of Thursday Afternoon Dataset

Label	Record Count
BENIGN	288566
Infilteration	36

Table Ⅴ. Components of Thursday Morning Dataset

Label	Record Count
BENIGN	168186
Brute Force	1507
XSS	652
Sql Injection	21

Table Ⅵ. Components of Wednesday Dataset

Label	Record Count
BENIGN	10169
Slowloris	5796
Slowhttptest	5499
Hulk	7187
GoldenEye	6355
Heartbleed	11

Table Ⅶ. Components of Tuesday Dataset

Label	Record Count
BENIGN	432074
FTP-Patator	7938
SSH-Patator	5897

III. Machine Learning Based Malicius Traffic Detection on CICIDS 2017

A. Dataset : CICIDS2017

In this paper, CICIDS 2017 dataset was used to compare each machine learning technique with each attribute group. This dataset was created by the Canadian Institute for Cyber-security (CIC). CICIDS 2017 dataset includes common attacks similar to actual data. It also includes analysis of network traffic using CICFlowMeter, source and destination IP ports, protocols, and attacks. The CIC identified 11 criteria needed to build a reliable set of benchmarks. These criteria are complete network configuration, complete traffic, labeled datasets, full interaction, full capture, available protocols, attack diversity, heterogeneous, functional sets, and metadata. CICIDS2017 dataset consists of labeled network flows (including full packet payload in pcap format), their profiles and machines, and CSV files for deep running purposes. The dataset consists of 8 files, with records for the remaining attacks except normal traffic divided by attacks as shown in Tables 1, 2, 3, 4, 5, 6, and 7. These dataset have different characteristics in terms of data imbalance and variety. The datasets in Table 1 and 2 are balanced with two labels. The datasets in Table 3 and 4 are made up of two labels, of which the amount of BENIGN data is too large for other labels which causes a data imbalance. The datasets in Table 5 made up of four labels, of which BENIGN data that leads to imbalance is too large compared with other labels. The dataset corresponding to Table 6 consists of six labels and much the amount of Heartbleed records in this dataset is small than other labels. The dataset in Table 7 consists of three labels in total and records are imbalanced.

B. Configuration of classifiers

Weka is a collection of machine learning techniques for data mining operations. Weka includes classification, regression, clustering, connection rules, and visualization functions. Weka is an open source software issued under the GNU General Public License and includes data manipulation, result visualization, database connectivity and K-fold Cross-Validation features to complement basic machine running tools. The version of Weka used in this paper is 3.9.3. This tool allows data files in csv or arff format with Weka. K-fold Cross-Validation of Naïve Bayes and SVM is applied to CICIDS 2017 dataset.

When conducting experiments with Naive Bayes and SVM, we used Naive Bayes and SMO classifiers, which Weka provides natively. When conducting experiments of CNN, we changed a record of CICIDS 2017 dataset, consisting of one LABEL and 78 data entries, into an 8-bit 9x9 format image as input of CNN. According to the results obtained through CNN-based traffic classification experiments [1] based on various convolutional neural network configurations, number of Convolution Layer is set to 1, Polling Filter Type is set to 2x2, and number of Hidden Units is set to 1024. During evaluation, Precision, Recall and F1-Score are measured and detail results are shown in Figures 1, 2, 3, 4, 5 and 6.

IV. Discussion about the detection result using different technique by daily activity

Throuth the entire evaluation, CNN and SVM generally have high detection rates. The results can be achieved as shown in Figures 1, 2, 3, 4, 5 and 6 in terms of Precision, Recall and F1-Scores obtained with confusion matrix. We pay attention to the results of CNN mostly. When malicious traffic classification is executed by using each classifier based on the dataset recorded to Table 1 and 2, the results can be obtained as shown in Figure 1 and 2. Table 1 and 2 dataset are composed of two labels, and the amount of data for each label is well balanced. This well-balanced binary classification shows that CNN and SVM have better results than Naive Bayes. Especially, CNN is better than SVM in the manner of processing time, because CNN based classifier is much faster than SVM based classifier.

When traffic classification is run on the basis of Table 3 and 4 dataset, the results are shown in Figure 3 and 4. Table 3 and 4 dataset consists of two labels. However, there is a huge data imbalance between the data on each label. If malicious traffic classification is carried out based on this data, CNN can't detect them. Through this analysis, we recognize that if there is an imbalance in the data within the dataset, the performance of CNN decreases dramatically. With the same reason, CNN and SVM did not work with dataset of Table 5.

When malicious traffic classification is executed based on dataset of Table 6, the results are obtained as shown in Figure 5. Table 6 dataset consists of six labels. The amount of records with BENIGN and DoS Hulk labels are well balanced, while the rest of the labels are imbalanced. Because of this reason, CNN can classify those two labels only.

Table 7 dataset consists of three labels. When malicious traffic classification is carried out based on this data, the performance of CNN is not good. Through this analysis, we recognize that if there are many labels which need to be classified, the performance of CNN decreases.

V. Conclusion and Future works

In this research paper, the machine learning techniques are analyzed in detail for malicious traffic classification using CICIDS 2017 dataset. In particular, the performance of the image processing technique, CNN, is evaluated in detail. As a result of the assessment, CNN has a good performance for binary classifications with sufficient data. However, performance of CNN is not good if there are many labels which need to be classified and there is an imbalance of the data. As a future work, we plan to conduct re-sampling and PCA (Principle Component Analysis) to address data imbalance and improve performance of the convolution neural network with many labels.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2017R1A2B4012559). This research was supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF)& funded by the Korean government (MSIT) (NRF-2019M3E5D1A020 67961).

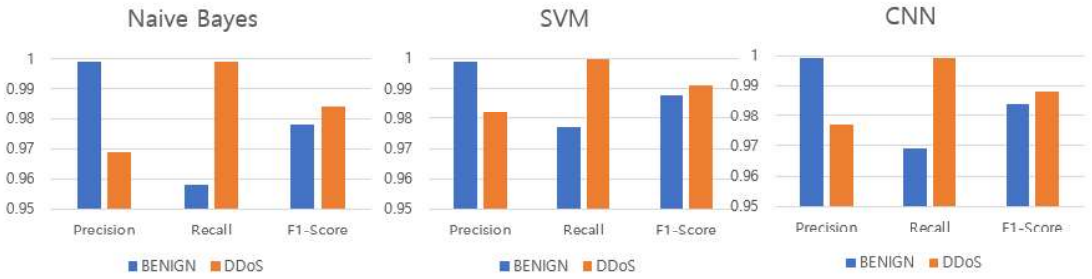

Figure 1 DDoS Classification Result for Friday Afternoon

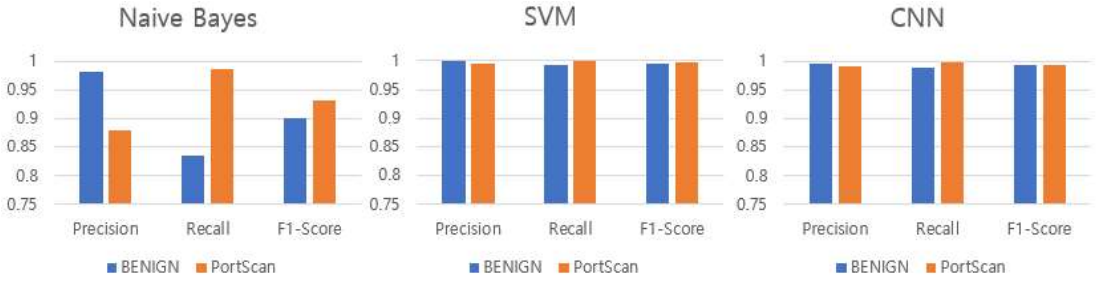

Figure 2 Portscan Classification Result for Friday Afternoon

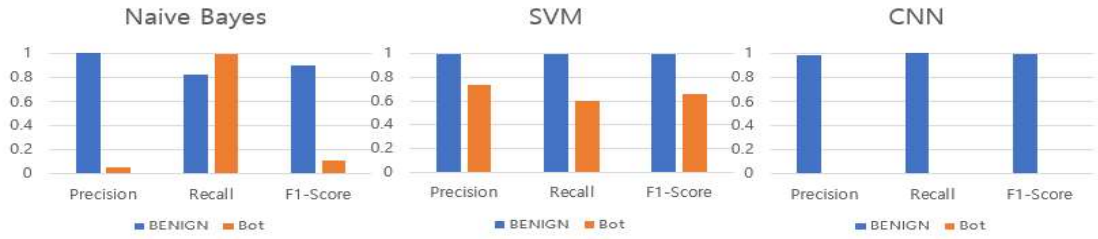

Figure 3 Bot Classification Result for Friday Morning

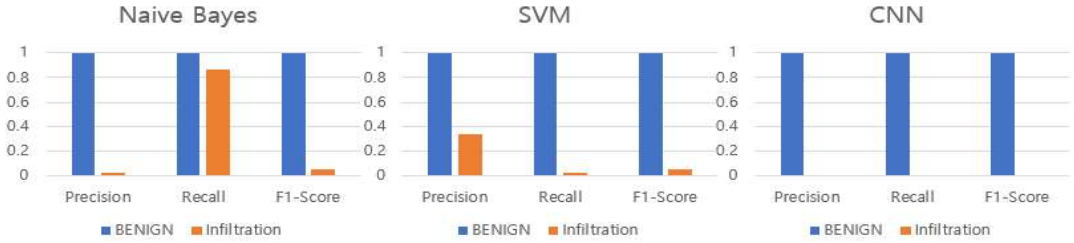

Figure 4 Thursday Afterrnoon Infilteration Classification Result

Figure 5 DoS Classification Result Wednesday

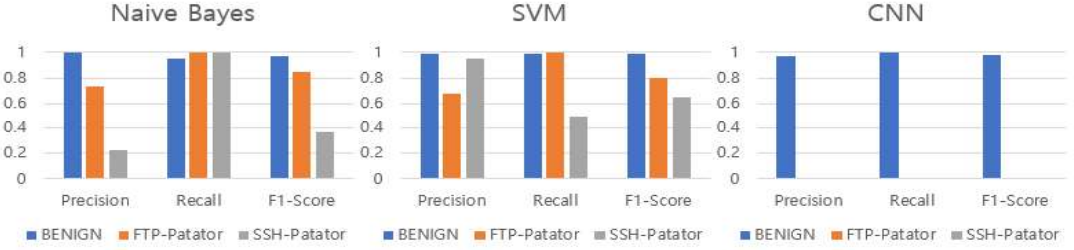

Figure 6 Tuesday Patator Classification Result

References

[1] Sung-woong Yeom, Van-Quyet Nguyen, and Kyungbaek Kim. "Assessing Convolutional Neural Network based Malicious Network Traffic Detection Methods." KNOM REVIEW, Vol. 22, No. 1, pp. 20-29, August, 2019.

[2] Jintae Choi, Sinh-Ngoc Nguyen, Jeongnyeo Kim, Guee-Sang Lee, Kyungbaek Kim, "Performance Comparison of Traffic Classification Techniques for Detecting Malicious Network Traffic." In Proceedings of the International Conference on Smart Media & Applications (SMA 2017) , December 17-19, 2017, Boracay, Philippines

[3] Nguyen, Sinh-Ngoc, et al. "Design and implementation of intrusion detection system using convolutional neural network for dos detection." Proceedings of the 2nd International Conference on Machine Learning and Soft Computing. ACM, 2018.

[4] Panigrahi, Ranjit, and Samarjeet Borah. "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems." International Journal of Engineering & Technology 7.3.24 (2018): 479-482.

[5] Gharib, Amirhossein, et al. "An evaluation framework for intrusion detection dataset." 2016 International Conference on Information Science and Security (ICISS). IEEE, 2016.

[6] Mera, Carlos, and John William Branch. "A survey on class imbalance learning on automatic visual inspection." IEEE Latin America Transactions 12.4 (2014): 657-667.

[7] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization." ICISSP. 2018.

[8] Jun, Li, et al. "Internet traffic classification using machine learning." 2007 Second International Conference on Communications and Networking in China. IEEE, 2007.

[9] https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.analytics.doc/doc/r_naive_bayes_background.html

[10] Sharma, Anand & Sharma, Tanvi & Mansotra, Vibhakar. (2016). Performance Analysis of Data Mining Classification Techniques on Public Health Care Data. International Journal of Innovative Research in Computer and Communication Engineering. 4.

[11] Search UNB [Internet]. University of New Brunswick est.1785. [cited 2019May26]. Available from: https://www.unb.ca/cic/datasets/ids-2017.html