Analysis on Network Traffic Features for Designing Machine Learning based IDS

N Meemongkolkiat and V Suttichaya*

Department of Computer Engineering, Faculty of Engineering, Mahidol University, 25/25 Salaya, Phuttamonthon, Nakhon Pathom, 73170, Thailand.
nannapat.mee@student.mahidol.edu, vasin.sut@mahidol.edu

Abstract. An intrusion detection system (IDS) is the most important technology for securing network systems. It can dynamically monitor network traffic for malicious activities that are aimed to violate confidentiality, integrity, authenticity, and availability of the network. Currently, several Machine Learning (ML) techniques are used to design and implement IDS since ML techniques can capture the complex nature of cyberattacks. However, network traffic information usually contains unimportant features that can deteriorate the efficacy of ML-based IDS. This research analyses the critical features in network traffic to be used for design/implementing the effective ML-based IDS. The selected features are applied to different ML methods to test the effectiveness. This research is conducted on the CICIDS2017 dataset generated by the Canadian Institute of Cybersecurity, using 30 percent of the full datasets and 100 percent of the Wednesday set. The best result achieved for 30 percent of the full set is by using 30 chosen features with the Bagging ensemble classifier giving the accuracy of 99.9 percent with the false-positive rate as low as 0.03 percent. The best result achieved for Wednesday set is by using the Random Forest Classifier which achieves an accuracy of 99.9 percent and a false-positive rate (FPR) of 0.02 percent.

1. Introduction

Nowadays, IDS is used for recognizing network malicious traffics. Various techniques such as pattern matching, statistics-based, and machine learning are used to design IDS. Many studies on machine learning were conducted and showed a high success rate. Many techniques, such as fast kNN (FKNN) [1], were used to reduce the computational time and some use more extensive techniques like Artificial Neural Network (ANN) to achieve an exceptional detection rate [2]. Some other research constructed IDS using ensemble learning and the use of oversampling techniques [3]. However, ML-based IDS does not only limit supervised learning, but some research also investigates unsupervised learning using a protocol-based IDS [4]. Furthermore, a hybrid approach has been done as semi-supervised learning using a hybrid of k-means clustering and ensemble learning [5]. Some challenges of ML-based IDS are because the data usually contain high dimensionality and noise. Feature selection techniques is a technique that reduces the dimensionality of the dataset. The benefit of feature selection is that it reduced computational time and improved the performance of ML-based IDS. There are many techniques of feature selection including Information Gain, Recursive Feature Elimination, and Correlations. Ustebay et al [6] used Recursive Feature Elimination with Random Forest estimators to eliminate unnecessary features from the dataset. It achieved 95 percent accuracy on the...
CICIDS2017 dataset. Furthermore, Aksu et al [7] applied Fisher Score to select the best 30 features to earn an accuracy of 99 percent.

The purpose of this research is to analyze the significant features of network traffic to be used for design and implement machine learning based IDS. Furthermore, the main contribution of this paper is to evaluate the important features in network traffic that are useful in classifying network attacks. Moreover, it will test the effectiveness of different ML methods on selected features by using accuracy, recall, false-positive rate, and F1-Score. Furthermore, the research will be conducted using the CICIDS2017 dataset. This paper is organized as follows; Related works are elaborated in Section 2. The methodology is presented in Section 3. Section 4 presents the results. Discussion and Conclusion are represented in Section 5 and Section 6, respectively.

2. Literature Review

2.1. CICIDS2017 Dataset

The CICIDS2017 dataset is an intrusion detection dataset provided by the Canadian Institute of Cybersecurity and it is a newer dataset, and it contains more up-to-date network attacks. It is a combination of eight CSV files which consisted of a total of approximately three million instances. The eight CSV files are the network traffic for each day, Monday is the only day that has no malicious data. The CICIDS2017 network traffic dataset consists of 15 labels, 14 of which are attacks and one benign. Totally, 288,602 records were missing values. The dataset is considered highly imbalanced since the label with the highest number is benign with 2,359,087 records where the least is heart bleed with 11 records. The in-depth information of the dataset can be found in [8]

2.2. Related Works

Zhou et al [9] used three datasets namely, NSL-KDD, AWID, and CICIDS2017 Wednesday set to classify the dataset into binary and multiclass classification. The Correlation-based feature selection combined with Bat algorithm (CFS-BA) was used as the feature selection technique where the number of features used was 10, 8, and 13 for NSL-KDD, AWID, and CICIDS2017, respectively. The reduced datasets then undergo Voting with C4.5, Random Forest (RF), and Forest with Penalizing Attributes (ForestPA) as estimators. The results achieved for the NSL-KDD dataset were the accuracy of 99.8 percent with the false acceptance rate (FAR) of 0.08 percent. Additionally, the results achieved for the AWID dataset were 99.5 percent with the FAR of 0.15 percent. Lastly, the results for the CICIDS2017 dataset were 99.9 with the FAR of 0.12 percent.

Kurniabudi et al [10] proposed the CICIDS2017 Dataset Feature Analysis with Information Gain for Anomaly Detection. The research used 20 percent of the CICIDS2017 dataset, and the aim was to classify the malicious data into seven classes namely, Normal, Bot, Brute Force, Dos/DDos, Infiltration, Port Scan, and Web Attack. The number of features used was 4, 15, 22, 35, 52, 57, and 77 features in which ranked by using the weight obtained from the Information Gain technique. The classifiers used to classify the CICIDS2017 dataset were Random Forest (RF), Bayes Network (BN), Random Tree (RT), Naïve Bayes (NB), and J48 or C4.5. The best result was achieved by using RF with 22 features which achieved an accuracy of 99.86 percent and the FAR of 0.08 percent. Additionally, the results achieved for the AWID dataset were 99.5 percent with the FAR of 0.15 percent. Lastly, the results for the CICIDS2017 dataset were 99.9 with the FAR of 0.12 percent.

Abdulhammed et al [11] proposed a feature selection technique with Principal Component Analysis (PCA) and Auto Encoder (AE) to classify the CICIDS2017 dataset into binary and multiclass classification. CICIDS2017 dataset was preprocessed and sampled using the Uniform Distribution Based Balancing (UDBB). The UDBB approach used uniform distribution balancing to learn and resample new instances. The dataset was pre-processed with PCA and AE. The experimental results showed that the optimal features for PCA are 10 features, whereas the optimal features for AE are 59. The classifiers used were RF, Bayesian Network (BN), Linear Discriminant Analysis (LDA), and
Quadratic Discriminant Analysis (QDA). The result showed that the accuracy achieved was 99.6% for both binary and multiclass classification when using PCA with RF.

3. Methodology
The process for evaluating the importance of features in the CICIDS2017 dataset is explained in this section. The overall process will be presented in Section 3.1. The process of re-labelling and sampling are elaborated in Section 3.2. The methods and parameters used in feature selection and classification will be explained in Section 3.3 and Section 3.4, respectively.

3.1 The Architecture

Figure 1. System Architecture

Figure 1 illustrates the overall process of the system. The system consisted of three parts, namely Pre-processing, Feature Selection, and Classification. Pre-processing is the procedure for preparing the CICIDS2017 dataset. Then, the feature selection methods were applied to extract the dominant features that are used to classify malicious packages from benign packages. To evaluate the effectiveness of the selected features, various ML algorithms were applied to classify attack classes.

3.2 Preprocessing
The research analyzes both the Wednesday set and the Full set since many related works have been done on both sets. CICIDS2017 contains network traffic records from Monday to Sunday. Totally, 288,602 records were dropped because they contain missing values. Additionally, nominal features, namely Flow ID, Source IP, Destination IP, Protocol, and Timestamp were excluded, according to the suggestion of Ring et al [12].

The class labels in CICIDS2017 are strings that represent the name of network attack methods. The relabeled process changed all labels into integer categories where 0 will present the benign label and label 1 to 14 will represent the attack labels.

The preprocess data was split into four datasets, feature selection dataset, cross-validation dataset, training dataset, and testing dataset. Additionally, standardization was applied to both CICIDS2017 Wednesday and the Full set before feature selection. The purpose of standardization is to scale the mean to zero and the standard deviation to one.

The CICIDS2017 Full sets have approximately three million transactions after the cleansing process. Therefore, the random sampling method is required to reduce the size of the CICIDS2017 Full set dataset. Cochran’s sample size formula [13] is applied to calculate the optimal number of samples. Totally, 30 percent of records in the Full set are randomly chosen to obtain a confidence level of 99 percent and a margin of error of one percent.

3.3 Feature Selection Process
The feature selection process is illustrated in Figure 2 below.
Firstly, Recursive Feature Elimination (RFE) with Random Forest estimator was applied for ranking the features. Secondly, the first 30 features were selected. The reason is many studies used 10 to 35 features in classifying malicious network traffics. Thirdly, the three-fold cross-validation process was applied to find the optimum number of features.

3.4 Classification
The CICIDS2017 Full set and Wednesday set were split as illustrated in Figure 3.

Ten classification algorithms were applied to train and test set to classify malicious traffic data. Most methods are used in many related works who uses this dataset. These methods are as follows:

- **Basic Classifiers**: Decision Tree (DT), Gaussian NB (GNB), KNeighbors (KN), Linear Discriminant Analysis (LDA), Logistic Regression (LR)
- **Ensemble Classifiers**: Adaboost, bagging method, Random Forest (RF) method, Stacking method, and Voting method.

All basic classifiers were applied to both CICIDS2017 Wednesday and Full set. The top three best candidate from basic classifiers were selected to be used in Stacking and Voting.

4. Results
The results from the features selection method and the classification methods will be reported in this section. Section 4.1 will present the result of feature selection. Section 4.2 will present the CICIDS2017 Full set result. Section 4.3 will present the result for the CICIDS2017 Wednesday set.

4.1. Feature Selection
Table 1 shows all 30 selected features from RFE. The features in bold fonts represent the top 18 features selected after the cross-validation process.

|   | Source_Port |   | Flow_Bytess |   | Fwd_IAT_Max |
|---|-------------|---|-------------|---|-------------|
| 2 | Init_Win_bytes_backward | 12 | Bwd_Packet_Length_Mean | 23 | Total_Length_of_Fwd_Packets |
| 3 | Init_Win_bytes_forward | 13 | Bwd_Packet_Length_Max | 25 | Bwd_Header_Length |
| 4 | Avg_Bwd_Segment_Size | 14 | Bwd_Packetss | 24 | Fwd_Header_Length |
| 5 | Average_Packet_Size | 15 | Bwd_Packet_Length | 21 | Fwd_Header_Length |
4.2. CICIDS2017 Full Set Dataset
Table 2 and 3 shows the classification result of the CICIDS2017 Full set. The value in bold fonts indicates the best result for basic and ensemble classifiers. The performance of machine learning algorithms is evaluated using accuracy, precision, recall, false-positive rate, and F1-Score.

**Table 2. Classification results of 18 Features CICIDS2017 Full Set and 30 Percent Sample Size**

| MLA Name                      | Train time (s) | Test time (s) | Accuracy (%) | Recall (%) | FPR (%) | F1-Score (%) |
|-------------------------------|----------------|---------------|--------------|------------|---------|--------------|
| AdaBoostClassifier            | 104.473        | 2.888         | 98.963       | 98.963     | 0.676   | 98.964       |
| BaggingClassifier             | 142.167        | 1.453         | **99.932**   | **99.932** | 0.032   | **99.932**   |
| DecisionTreeClassifier        | 48.053         | 0.115         | **99.924**   | **99.924** | 0.038   | **99.924**   |
| GaussianNB                    | 1.862          | 1.793         | 72.449       | 72.449     | 31.175  | 82.260       |
| KNeighborsClassifier          | 194.835        | 1333.310      | 99.043       | 99.043     | 0.627   | 99.027       |
| LinearDiscriminantAnalysis    | 14.627         | 0.163         | 87.387       | 87.387     | 8.860   | 88.635       |
| LogisticRegression            | 89.920         | 0.133         | 96.790       | 96.790     | 2.621   | 96.669       |
| RandomForestClassifier        | 206.594        | 2.727         | 99.901       | 99.901     | 0.015   | 99.901       |
| StackingClassifier            | 808.655        | 188.438       | 99.916       | 99.916     | 0.064   | 99.916       |
| VotingClassifier              | 77.882         | 196.544       | 99.438       | 99.438     | 0.135   | 99.436       |

**Table 3. Classification results of 30 Features CICIDS2017 Full Set and 30 Percent Sample Size**

| MLA Name                      | Train time (s) | Test time (s) | Accuracy (%) | Recall (%) | FPR (%) | F1-Score (%) |
|-------------------------------|----------------|---------------|--------------|------------|---------|--------------|
| AdaBoostClassifier            | 303.898        | 11.216        | 85.717       | 85.717     | 0.121   | 80.434       |
| BaggingClassifier             | **419.990**    | **3.660**     | **99.939**   | **99.939** | **0.022** | **99.937**   |
| DecisionTreeClassifier        | 53.825         | 0.109         | **99.928**   | **99.928** | 0.036   | **99.928**   |
| GaussianNB                    | 2.011          | 1.913         | 72.449       | 72.449     | 31.175  | 82.260       |
| KNeighborsClassifier          | 196.016        | 1328.410      | 99.043       | 99.043     | 0.627   | 99.027       |
| LinearDiscriminantAnalysis    | 14.914         | 0.116         | 87.387       | 87.387     | 8.860   | 88.635       |
| LogisticRegression            | 92.188         | 0.119         | 96.790       | 96.790     | 2.621   | 96.669       |
| RandomForestClassifier        | 342.112        | 5.383         | 99.899       | 99.899     | 0.048   | 99.895       |
| StackingClassifier            | 4555.030       | 1313.160      | 99.929       | 99.929     | 0.022   | 99.927       |
| VotingClassifier              | 336.292        | 1314.500      | 99.447       | 99.447     | 0.539   | 99.428       |

4.3 CICIDS2017 Wednesday Set Dataset
Table 4 and 5 shows the classification result of the CICIDS2017 Wednesday set. The value in bold fonts indicates the best result for basic and ensemble classifiers.
Table 4. Classification results of 18 Features on CICIDS2017 Wednesday Set

| MLA Name             | Train time (s) | Test time (s) | Accuracy (%) | Recall (%) | FPR (%) | F1-Score (%) |
|----------------------|----------------|---------------|--------------|------------|---------|--------------|
| AdaBoostClassifier   | 84.204         | 3.153         | 94.268       | 94.268     | 2.011   | 94.026       |
| BaggingClassifier    | 40.957         | 0.417         | 99.928       | 99.928     | 0.044   | 99.928       |
| DecisionTreeClassifier | **6.463**     | **0.038**    | **99.912**   | **99.912** | **0.052** | **99.912**   |
| GaussianNB           | 1.017          | 0.151         | 74.424       | 25.576     | 66.224  | 35.555       |
| KNeighborsClassifier | 20.657         | 35.129        | 99.098       | 99.098     | 0.924   | 99.096       |
| LinearDiscriminantAnalysis | 2.906    | 0.039         | 83.638       | 83.638     | 4.176   | 83.038       |
| LogisticRegression   | 51.438         | 0.036         | 84.820       | 84.820     | 1.660   | 83.089       |
| RandomForestClassifier | **96.022**   | **2.406**    | **99.944**   | **99.944** | **0.032** | **99.944**   |
| StackingClassifier   | 442.504        | 25.961        | 99.927       | 99.927     | 0.050   | 99.927       |
| VotingClassifier     | 75.217         | 38.240        | 99.616       | 99.616     | 0.111   | 99.611       |

Table 5. Classification results of 30 Features on CICIDS2017 Wednesday Set

| MLA Name             | Train time (s) | Test time (s) | Accuracy (%) | Recall (%) | FPR (%) | F1-Score (%) |
|----------------------|----------------|---------------|--------------|------------|---------|--------------|
| AdaBoostClassifier   | 103.219        | 3.434         | 91.036       | 91.036     | 2.128   | 91.473       |
| BaggingClassifier    | 57.887         | 0.514         | 99.962       | 99.962     | 0.024   | 99.962       |
| DecisionTreeClassifier | **8.633**     | **0.049**    | **99.958**   | **99.958** | **0.030** | **99.958**   |
| GaussianNB           | 1.069          | 0.232         | 54.218       | 45.782     | 76.96   | 53.687       |
| KNeighborsClassifier | 10.161         | 36.896        | 98.935       | 98.935     | 1.140   | 98.935       |
| LinearDiscriminantAnalysis | 3.954    | 0.053         | 85.990       | 85.990     | 5.47    | 86.220       |
| LogisticRegression   | 43.196         | 0.050         | 81.530       | 81.530     | 15.46   | 82.027       |
| RandomForestClassifier | **112.483**  | **2.355**    | **99.967**   | **99.967** | **0.019** | **99.967**   |
| StackingClassifier   | 174.985        | 32.612        | 99.954       | 99.954     | 0.034   | 99.954       |
| VotingClassifier     | 21.006         | 40.174        | 99.674       | 99.674     | 0.056   | 99.672       |

5. Discussion

The result from Section 4 shows that the Decision tree outperforms other basic classification methods in both Full Set and Wednesday set. Moreover, ensemble classifiers can significantly reduce the FPR. The result also shows that using 18 and 30 features shown no significant difference in terms of accuracy. It should also be pointed out that the features selected from RFE are consistent with many features proposed in Kurniabudi et al. Decision tree, Bagging, and Random Forest classifiers can be used to classify attacks in multi-class classification accurately with an accuracy of more than 99 percent. The three classifiers are shown to be superior to the other classifiers because the base of all three classifiers is tree-type classifier. The advantages of using tree-type classifiers are their ability to handle outliers and unbalanced datasets. Most misclassification in multi-class classification occurs from web attacks. Most web attacks such as SQL injection cannot be classified with network traffic alone, the content of the package may also be required to classify them. It is also worth noting that according to [8] the features that are important in classifying Heartbleed attack are features 7, 28, and 29 from Table 1. All three features are included in the results from RFE where feature 7 also appears in the 18 features set. Therefore, all classification models can correctly classify the Heartbleed attack. As mentioned in Section 2 that Abdulhammed et al uses UDBB to synthesize data where the new instances are randomly chosen from the uniform distribution. However, malicious traffic instances may not be uniformly distributed. Thus, using uniform distribution as an assumption could be inaccurate.
6. Conclusions
This research is conducted on the CICIDS2017 dataset and aims to analyze the significant features of network traffic to be used for design and implement ML-based IDS. RFE was applied to the dataset to eliminate unnecessary features from the dataset. The cross-validation process was applied to select the top 18 features. ML algorithms were applied to classify malicious traffics. The effectiveness of the selected features and ML methods were evaluated using accuracy, recall, false-positive rate, and F1-Score. The result showed that the reduction in the number of features does not affect the accuracy but slightly increases the FPR and reduces the time taken to train and test greatly. The top classifiers are Decision Tree, Random Forest, and Bagging Classifiers.

7. References
[1] Kanneganti, V. K., Swathi, K., & Brao, B. (2020). A Novel Framework for NIDS Throuh Fast Knn Classifier on CICIDS 2017 Dataset. International Journal of Recent Technology and Engineering, 8(5), 3669-3675. doi:10.35940/ijrte.e6580.018520
[2] Andreatos, A., & Moussas, V. (2019). A novel intrusion detection system based on neural networks. MATEC Web of Conferences, 292, 03017. doi:10.1051/matecconf/201929203017
[3] Yulianto, A., Sukarno, P., & Suwastika, N. A. (2019). Improving adaboost-based intrusion detection System (IDS) performance on cic ids 2017 Dataset. Journal of Physics: Conference Series, 1192, 012018. doi:10.1088/1742-6596/1192/1/012018
[4] Labonne, M., Olivereau, A., Polve, B., & Zeghlache, D. (2020). Unsupervised protocol-based intrusion detection for real-world networks. 2020 International Conference on Computing, Networking and Communications (ICNC). doi:10.1109/icnc47757.2020.9049796
[5] Aamir, M., & Zaidi, S. M. (2019). Clustering based semi-supervised machine learning for ddos attack classification. Journal of King Saud University - Computer and Information Sciences. doi:10.1016/j.jksuci.2019.02.003
[6] Ustebay, S., Turgut, Z., & Aydin, M. A. (2018). Intrusion detection system with Recursive Feature elimination by using random forest and deep learning classifier. 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELF). doi:10.1109/ibigdelft.2018.8625318
[7] Aksu, D., Üstebay, S., Aydin, M. A., & Atmaca, T. (2018). Intrusion detection with comparative analysis of supervised learning techniques and fisher score feature selection algorithm. Communications in Computer and Information Science, 141-149. doi:10.1007/978-3-030-00840-6_16
[8] Sharaafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy. doi:10.55220/0006639801080116
[9] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 107247. doi:10.1016/j.comnet.2020.107247
[10] Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M. Y., Bamhdi, A. M., & Budiarto, R. (2020). Cicids-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access, 8, 132911-132921. doi:10.1109/access.2020.3009843
[11] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics, 8(3), 322. doi:10.3390/electronics8030322
[12] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147-167. doi:10.1016/j.cose.2019.06.005
[13] Snedecor, G. W., & Cochran, W. G. (1992). Statistical methods. Ames, IA: Iowa State Univ. Press.