International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 1 (2019), pp. 1-7

A Comparison of Various Machine Learning Algorithms in a Distributed Denial of Service Intrusion

[1]SH Kok, [2]Azween Abdullah, [3]Mahadevan Supramaniam,
[4]Thulasyammal Ramiah Pillai, [5]Ibrahim Abaker Targio Hashem

[1,2,4,5] School of Computer and IT (SoCIT), Taylor's University, Malaysia.
[3]Research & Innovation Management Centre, SEGi University, Malaysia.

ORCIDs: [1](0000-0001-9477-8988) [2](0000-0003-4425-8604) [3](0000-0002-3294-3324)
[4](0000-0001-7611-9540) [5](0000-0002-3734-0899)

Abstract

Machine learning (ML) is used for network intrusion detection because of its prediction ability after training with relevant data. ML provides a good method to detect new and unknown attacks. There are many types of attacks in network intrusion: however, this paper concentrates on distributed denial of service (DDoS). DDoS is similar to denial of service, the different being that the attack comes from multiple sources instead of only one source. In this paper, various classification algorithms are used and their performances are compared. It was found that random forest (RT) gives the best accuracy at 99.97%, while the least accurate algorithm was support vector machine (SVM) at 63.25%.

Keywords: Classification, Distributed Denial of Service, Machine Learning, Random Forest.

I. INTRODUCTION

In network intrusion detection system (NIDS) research, there are three types of detection approaches misused or signature-based, anomaly-based and hybrid-based. Misused or signature-based approach primarily detects known intrusion attacks, whereas anomaly-based approach detects new or unknown intrusion attacks. Hybrid-based detects both known and unknown intrusion attacks.

Machine learning (ML) techniques learn the pattern from past data and make predictions for current data. Since ML recognises patterns, instead of specific signatures, it can be used for hybrid-based approach that can detect small variations from known attacks. New attacks are continuously being created, therefore it is important that NIDS be able to adapt to changes to detect both known and unknown attacks.

There are many types of attack in NIDS. However, this paper focuses on distributed denial of service (DDoS) attacks. DDoS is very similar to denial of service attacks. The difference is that the second has a single source of attack, whereas the first has multiple sources of attack. Both types of attack result in inaccessible network resources, due to the complete consumption of the network resources by these attacks.

The challenge for an effective NIDS is to have a high accuracy rate a with low false positive rate and a low false negative rate. These are some of the main metrics currently being emphasised for NIDS research.

II. RELATED WORK

Extensive research has been done on intrusion detection system (IDS), however this topic will never be obsolete due to the continuous evolving nature of development and advancement, which continually changes the landscape of the network infrastructure. The challenge is to keep up with advances and evolving threats with equally advanced and effective solution. Datasets and classifications of types of attacks are obsolete in comparison to current threats [1]. Therefore, there is a compelling need to update IDS solutions to address the current situation and active research is needed to ensure that these tools do not become obsolete.

DDoS is a type of NIDS that launches its attacks using multiple sources or hosts. This is achieved by controlling multiple compromised hosts that act like a zombie host for the attacker. These zombie hosts continuously send streaming packets to the targeted victim rendering it non-serviceable to legitimate users [2]. Prime targets for this type of attack are websites that offer services via the Internet, such as Twitter, Spotify or Amazon. Loss of service for such websites leads to losses of financial gain [3].

Usually prominent websites are the primary victims of such attacks. Recently Twitter, Spotify, and Amazon suffered interruptions in their services for nearly two hours on 21 Oct, 2016 due to DDoS attacks. Such interruptions in their services lead to huge financial losses.

Feature selection (FS) is an important part of pre-processing. Using a trial and error approach for very low FS (three features as a set), it was found that the three features that have the highest impact on NIDS are, 'from the source to destination time to live value while the packets are alive', 'source TCP window advertisement value', and 'number of connections that contain the same service and source address in the previous 100 connection' [4]. However, the study could not conclude that these are the optimum features, because it did not increase or decrease the feature dimension.

ML can be divided into three main types: supervised, unsupervised and semi-supervised. Supervised algorithms require data to be labelled, then based on the label, they can classify the data according to a distinct pattern for each class or label. Unsupervised algorithms can use data without any labelling. This type of algorithm clusters the data into group(s)

with similar characteristics. Semi-supervised algorithms use data that are partially labelled. The ML types and algorithm are shown in Fig. 1. It has been found that supervised algorithms work well in IDS with previously known attacks, while unsupervised algorithm are more robust with both known and unknown attacks [5].

Table 1 contains a summary of previous studies that have used ML techniques for intrusion detection. The best result obtained was using k-nearest neighbours (kNN) as a classifier with information gain ratio (IGR) for FS. This method resulted in an accuracy of 99.07%. The lowest accuracy was 55.05%, which was produced using naïve Bayes (NB) as a classifier with LDA for FS. However, using this algorithm with other FS, except canonical correlation analysis (CCA), was found to be comparatively better than using NB alone. Therefore, there could be a compatibility issue between the NB algorithm with LDA and CCA.

Extreme learning machine is another name for feedforward neural network that can be used to solve classification, clustering, regression and feature engineering problems. This algorithm has been found to be a very accurate algorithm when the data size is huge. If the data size is smaller, SVM can give better results [6].

Averaged one dependence estimators (AODE) can provide good accuracy when performing binary classification. In addition, their training and testing times has been found to be relatively fast compared to other popular algorithms [7].

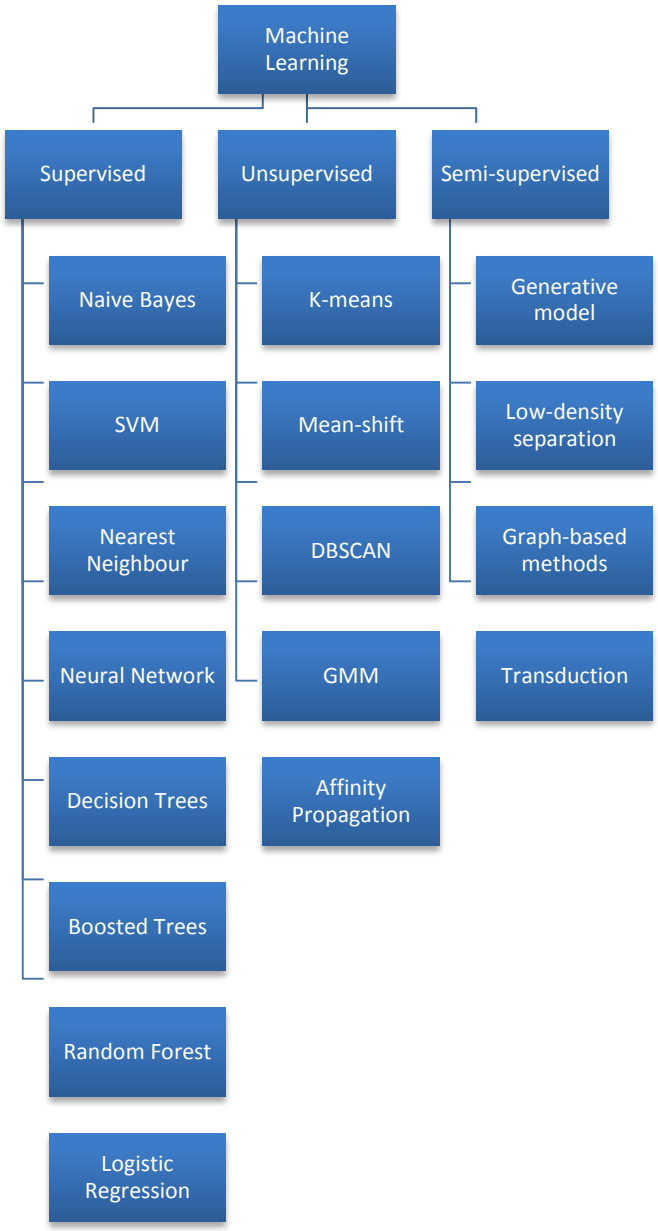

Fig. 1. Classification of ML algorithms

Density peak clustering (DPC) is a method that does not require many parameters. Hybrid algorithm developed based on DPC with kNN has been found to simplify the implementation of a solution. In addition, this hybrid method can effectively detect intrusion attacks and has good performance with respect to accuracy [8].

Lazy learning methodologies have been studied to overcome the limitation of eager learning methods. It has been argued that eager learning methods contribute to losess in performance efficiency when trying to generalise training data prior to receiving queries. This causes an unnecessary use of computational overhead. To overcome this, the use a heuristic weight-based indexing was proposed with a lazy method to overcome the high search complexity that is normally associated with lazy methods [9]. This study used kNN with cross validation as its lazy method.

Table 1. Summary of the performance of ML algorithms

Reference		**[9]**		**[8]**		**[10]**		**[4]**		**[7]**		**[11]**	
Algorithm	FS			CCA (small)	CCA (big)	LDA (small)	LDA (big)			CFS	PCA	IGR	MRMR
kNN		92.30	91.07							97.65	98.87	99.07	98.05
SVM			91.19					98.76		76.61	96.78	94.39	88.93
NB				58.44	59.55	57.93	55.50		75.73	82.66	89.91	90.26	87.56
BN									92.70				
DT										98.99	98.95	97.83	98.78
NN										83.80	97.50	97.70	94.60
RT				94.61	89.75	92.50	93.56						
RF				92.81	90.10	87.75	86.46						
RC				89.08	87.84	92.16	84.30						
REP Tree				94.61	89.46	88.12	93.26						
Bagging				95.53	88.45	89.73	86.00						
Randomised Filtered				87.76	87.10	83.02	75.93						
AODE									94.37				
LWL		90.70											
hwIBK		97.60											
PSO			92.59										
LUS			92.75										
WMA			88.66										
TANN			84.67										
CANN			69.04										
DPNN			87.26										

Note:

kNN	k-Nearest Neighbour	BN	Bayes Network
RC	Random Committee	LWL	Local Weighted Learning
hwIBK	heuristic weighted kNN	PSO	Particle Swarm Optimisation
LUS	Local Unimodal Sampling	WMA	Weighted Majority Algorithm
TANN	Triangle Area based Nearest Neighbours	CANN	Cluster Centre and Nearest Neighbour
DPNN	Hybrid Density Peaks Nearest Neighbours	CCA	Canonical Correlation Analysis
CFS	Correlation based Feature Selection	PCA	Principal Component Analysis
IGR	Information Gain Ratio	MRMR	Minimum Redundancy Maximum Relevance

In the quest to improve IDS, several important characteristics have been identified for future IDS, such as online learning capability, drift concept handling, and the capacity to adjust to any environment. These characteristics help avoid high false alarm ratios [12].

III. METHOD

This paper compared the performances of eight ML algorithms using the CICIDS2017 dataset. Steps taken are as shown in Fig. 2. ML algorithms were used to build Model, as in the diagram. Three important performance metrices [20] were used for the comparison; namely accuracy, true positive rate and false alarm rate.

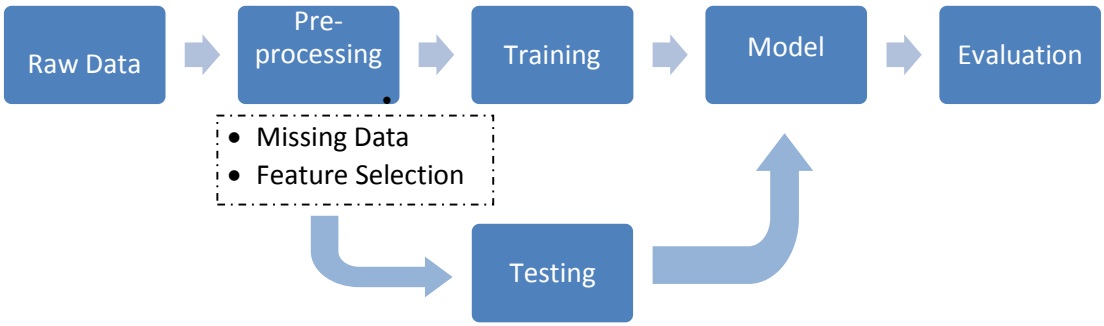

Fig. 2. Diagram of process flow performed

III.I Dataset

We used the CICIDS2017 dataset which was created by the Canadian Institute for Cybersecurity for network security and intrusion detection. This dataset covers six types of attack, and has 79 features. However, for this paper we used only one day (a Friday afternoon) out of a total of five days of data. This constitutes one type of attack, namely DDoS. This attack results in a system overload, and therefore prohibits legitimate usage of the affected network [13].

This study was performed using RStudio version 1.1.453 with caret package version 6.0-80 installed on a Win7 OS. The hardware specifications of the laptop were Intel i5 CPU, 8Gb RAM, and Intel HD Graphics 3000.

III.II Pre-processing

Raw data normally include many imperfections such as missing value, redundancies and inconsistencies. Therefore pre-processing is required to produce a clean dataset that will ensure that an ML technique can build and train a model smoothly without any errors [14]

1) Missing Data

Missing data include empty values or values not compatible with the data format. For example, features with numerical formats must consist of numbers only, and cannot include any symbols or alphabetic characters. The simplest approach is to discard or remove such data all together. However these data points could be important; therefore, we use the maximum likelihood approach [15]. All missing data were replaced with linearly interpolated values.

During this step, it was found that four observations have missing data in two of the features. These features are Flow Bytes/s and Flow Packets/s. All missing data were replaced via linear interpolation using the 'na.approx' command.

2) Feature Selection

All the features that have min, max, mean and std (standarad deviation) values were removed except the mean value of the feature because these values refer to the same features but use different calculated values. We removed these features, namely Fwd Packet Length, Bwd Packet Length, Flow IAT, Fwd IAT, Bwd IAT, Packet Length, Active, and Idle. In addition, Fwd IAT Total, Bwd IAT Total and Packet Length Variance were removed for the same reason.

Features with non-numerical data were also removed to improve the algorithm performance namely Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, and Timestamp. This reduced the total number of features from 84 to a final dataset of 50 features. The feature 'Label' was the dependent variable or the labelled data.

However after the pre-processing step, three algorithms were found to have errors with the prepared dataset. These algorithms were SVM, neural network and nearest neighbour. Further investigation found that two features, Flow Bytes/s and Flow Packets/s included 'Inf' values. Therefore, these two features were removed when running these three algorithms.

III.III Machine Learning

We used a total eight ML algorithms to train and test the prepared dataset. These eight algorithms are supervised algorithms that can produce binary classification.

1) Logistic Regression

This algorithm uses a regression model to find the best-fitting model that describes a dependent variable based on a set of independent variables. The outcomes of the dependent variable consist of only two possible values: true or false. Therefore it is well suited for binary classifications.

2) Naïve Bayes (NB)

This algorithm uses a probability calculation of Bayes' theorem. Each independent variable contributes to the probability of the outcome. It is a powerful knowledge representation and reasoning algorithm under conditions of uncertainty [16].

3) Support Vector Machine (SVM)

This algorithm finds the optimum hyperplane that separates two classes with the maximum distance between the border points of each class. These border points form the support vector. Therefore it is effective for high-dimensional space problems, and is memory efficient. However if the feature count is larger than the number of samples, this technique will have only a mediocre performance [17].

4) Decision Tree (DT)

This algorithm uses a tree structure analogy to represent a series of rules that lead to a class or value [16]. It starts with a root node, which is the best predictor. Then, it progresses through branch nodes to other predictors. Ultimately it reaches the leaf nodes, which represent a decision or classification.

5) Boosted Trees (BT)

This algorithm is based on decision tree with the addition of a boosting method. Instead, of building one large tree, multiple small trees are built. Then the result of each small tree is added, with a weighted value, to obtain a final predictive outcome.

6) Random Forest

This algorithm is similar to BT, where multiple small trees are built. However, it differs in the way it calculates the final predictive outcome. Instead of using a boosting method, it uses a bagging method. This method uses the mean of the individual small trees to obtain the final predictive outcome. This classifier is found to be fast and efficient with large datasets [18].

7) Neural Network

This algorithm uses the brain cell analogy of a neuron. Multiple neurons are arranged in multiple layers. Each neuron takes an input, processes it, and produces an output. This output moves to the next layer of neurons with an applied weight. This process iterate until an outcome is produced. Neural network learns from scenarios and detects zero-day patterns that are similar to those on which it has been trained. Therefore, it can detect known attacks and variants on these attacks [19].

8) Nearest Neighbour

This algorithm uses a voting system to determine its outcome. To determine the classification of a new point, it finds the class that has the most neighbouring points (votes).

III.IV Performance Metrices

Metrices are used to quantify the ML performance. Such metrices can be calculated based on a confusion matrix as shown in Table 2 [20].

1) Accuracy

This metric determines the accuracy, all correct prediction, of the model. It is the model abilities to predict both positive and negative results correctly.

Accuracy: $$\frac{TN + TP}{TN + TP + FP + FN} \quad (1)$$

2) True Positive Rate (TPR)

This metric calculates how often the model is able to predict a positive result correctly. Similar to Accuracy, but difference is it only takes positive observation.

TPR:: $$\frac{TP}{TP + FN} \quad (2)$$

3) False Alarm Rate (FAR)

This metric calculates how often the model is predicting a positive result wrongly. It provides indication of possible error of the model, thus lower value is better.

FAR: $$\frac{FP}{FP + TN} \quad (3)$$

Table 2. Confusion matrix table

		Predicted Class	
		Negative (Normal)	Positive (Attack)
Actual Class	Negative (Normal)	True Negative (TN)	False Positive (FP)
	Positive (Attack)	False Negative (FN)	True Positive (TP)

IV. RESULT

We used various ML algorithms, and their results are shown in Table 3. Random forest produced the highest accuracy of 99.97% with the lowest FAR of 0.02%. SVM had the lowest accuracy at only 63.25% with a high FAR of 36.92%. This

indicates that SVM could not find a good hyperplane of separation, because the data could not be separated via linear regression. A possible improvement would be to use SVM with kernels (5).

Three algorithms took comparatively longer to train, these were SVM, random forest and nearest neighbour. Therefore, for these algorithms, fine-tuning or hardware that has better processing power may be required.

Table 3. Result of the eight classification algorithms using the CICIDS2017 dataset

Algorithm	Accuracy (%)	TPR (%)	FAR (%)
Random Forest	99.97	99.97	0.02
Boosted Trees	99.88	99.94	0.19
Decision Tree	99.78	99.82	0.26
Naïve Bayes	99.77	99.80	0.27
Nearest Neighbour*	99.74	99.75	0.27
Neural Networks*	86.15	80.50	0.51
Logistic Regression	78.94	99.47	32.63
SVM Linear*	63.25	63.31	36.92

* Flow Bytes/s and Flow Packets/s columns were removed

V. DISCUSSION

It appears that the dataset used, CICIDS2017, is well suited for DT algorithm and its derivatives, such as BT and random forest. Random forest produced the best result with the highest accuracy of 99.97%, followed by BT and DT.

The result obtained was based on one day out of a total of five days data and consisted of only one type of attack (DDoS). Therefore, this result is not conclusive and further testing is required to confirm this result.

This study primarily focused on common classifiers; future studies should use more advanced hybrid algorithms to test the same and/or the full CICIDS2017 dataset. In addition to much larger sample data, the full dataset includes six types of attacks.

VI. CONCLUSION

This paper aims to compare nine supervised algorithms' performance towards DDoS intrusion. DDoS attack will result in inaccessible to network resources, due to complete consumption of the network resources. The random forest algorithm produced the best result with an accuracy of 99.97%. This ensemble classifier, which uses the bagging method, can handle outliers and noise in the dataset, which makes it less susceptible to over-fitting. However, random forest took a relatively longer time to compute compared to the other algorithms. Therefore, there is room for improvement and fine-tuning the model could allow it to work in more efficient manner. In addition, this study was done using only one day of data, from a total of five days. To have clear conclusion, it is important to use the full dataset, which includes a total of six types of attacks.

REFERENCES

[1] Nisioti A, Mylonas A, Yoo PD, Member S, Katos V. From Intrusion Detection to Attacker Attribution : A Comprehensive Survey of Unsupervised Methods. IEEE Commun Surv Tutorials. 2018;PP(c):1.

[2] Hafizah S, Ariffin S, Muazzah N, Latiff A, Khairi MHH, Ariffin SHS, et al. A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN). Technol Appl Sci Res [Internet]. 2018;8(2):2724–30. Available from: https://www.researchgate.net/publication/324830666=2

[3] Behal S, Kumar K. Detection of DDoS attacks and flash events using information theory metrics–An empirical investigation. Comput Commun [Internet]. 2017;103:18–28. Available from: http://dx.doi.org/10.1016/j.comcom.2017.02.003

[4] Chowdhury N, Ferens K, Ferens M. Network Intrusion Detection Using Machine Learning. 2010;30–5.

[5] Zamani M, Movahedi M. Machine Learning Techniques for Intrusion Detection.:1–11.

[6] Ahmad I, Basheri M, Iqbal MJ, Rahim A. Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. 2018;33789–95.

[7] Zou X, Feng Y, Li H, Algorithm MC, Hamid IRA, Syafiqah N. Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System. 2018;

[8] Science N, Phenomena C, Li L, Zhang H, Peng H, Yang Y. Chaos, Solitons and Fractals Nearest neighbors based density peaks approach to intrusion detection. Chaos, Solitons Fractals Interdiscip J Nonlinear Sci Nonequilibrium Complex Phenom [Internet]. 2018;110:33–40. Available from: https://doi.org/10.1016/j.chaos.2018.03.010

[9] Chellam A, Ramanathan L, Ramani S. Intrusion Detection in Computer Networks using Lazy Learning Algorithm. Procedia Comput Sci [Internet].

2018;132:928–36. Available from: https://doi.org/10.1016/j.procs.2018.05.108

[10] Dahiya P, Kumar D. Network Intrusion Detection in Big Dataset Using Spark. Procedia Comput Sci [Internet]. 2018;132:253–62. Available from: https://doi.org/10.1016/j.procs.2018.05.169

[11] Biswas SK. Intrusion Detection Using Machine Learning : A Comparison Study. 2018;118(19):101–14.

[12] Ahmad B, Jian W, Ali ZA. Role of Machine Learning and Data Mining in Internet Security : Standing State with Future Directions. 2018;2018.

[13] Sharafaldin I, Habibi Lashkari A, Ghorbani AA. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. Proc 4th Int Conf Inf Syst Secur Priv [Internet]. 2018;(Cic):108–16. Available from: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006639801080116

[14] Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F. A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing. 2017;239:39–57.

[15] García S, Luengo J, Herrera F. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowledge-Based Syst. 2016;98:1–29.

[16] Informatic S, Science C. AN ENSEMBLE APPROACH BASED ON DECISION TREE AND BAYESIAN NETWORK FOR INTRUSION DETECTION. Comput Sci Ser. 2017;15:82–91.

[17] Informatic S, Science C, Science C, Intelligence M, Labs MIR, Roy SS. RANDOM FOREST, SUPPORT VECTOR MACHINE AND NEAREST CENTROID METHODS FOR CLASSIFYING NETWORK INTRUSION. Comput Sci Ser. 2016;14:9–17.

[18] Shams R, Mercer RE. Supervised classification of spam emails with natural language stylometry. Neural Comput Appl. 2016;27(8):2315–31.

[19] Saied A, Overill RE, Radzik T. Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing [Internet]. 2016;172:385–93. Available from: http://dx.doi.org/10.1016/j. neucom.2015.04.101

[20] Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: A review. Appl Soft Comput J. 2010;10(1):1–35.