Global Journal of Engineering and Technology Advances, 2020, 02(02), 001–010

Global Journal of Engineering and Technology Advances

Cross Ref DOI: 10.30574/gjeta

Journal homepage: http://www.gjeta.com

(RESEARCH ARTICLE)

Ensemble classifiers for detection of advanced persistent threats

Okwara Jerry Chizoba [1,*] and Buba Abba Kyari [2]

[1] *Information and Communication Technology Unit, Centre for Entrepreneurship Development and Vocational Studies, The Federal Polytechnic Ado-Ekiti.*
[2] *Faculty of Social and Management Science, Dept. of Business Administration, Yobe State University.*

Publication history: Received on 01 February 2020; revised on 05 February 2020; accepted on 07 February 2020

Article DOI: https://doi.org/10.30574/gjeta.2020.2.2.0007

Abstract

The demand for application of technology in almost all walks of life is in the increase and can be seen to be geared by the paradigm changes in industrial revolutions (current 4.0), IoT/IoE (Internet of Things/Internet of Everything) concept, Internet 2.0, Artificial Intelligence (AI), BYOD (Bring Your Own Device) to mention a few but not without their increased inherent vulnerabilities and exposure to sophisticated and dynamic awaiting threats. Advanced Persistent Threats (APTs) among other malwares are some of the malicious attacks given serious attention as they have shown some level of complexities thereby causing defender solutions to poorly detect them. Poor APT attack tactics understanding, insufficient network traffic log analysis and poor classification are some of the problems identified for poor detection of these attacks. Network traffic logs are used by researchers to analyze the network and track attacks as packets move across network nodes. This research studies attack modelling in order to understand APT attack tactics and generate their dataset through simulation as well as a real dataset for normal operation. The experiment will be simulated on a virtual environment using dimensionality reduction technique on the network traffic log for improved log processing. To improve the APT detection accuracy flawed by their stealthiness, the ensemble of classifiers (Support Vector Machine, Random Forest, Decision Tree) with majority voting is used for better attack classification which resultantly gives a better detection accuracy of 90.47%.

Keywords: Artificial Intelligence; Ensemble; Dimensionality reduction; Network traffic

1. Introduction

There is a growing demand for technology application and development in almost all walks of life leading to flexibility of platforms (hardware, software) that run their day to day operations. This has witnessed an increase in the use of mobile devices, cloud computing and company policies like BYOD (Bring Your Own Device) as a form of support or use for getting works done either onsite or from some remote locations (Rashid et al, 2014). These migrations and developments seems amazing but not without their increased vulnerabilities and exposure to attacks. Again, devices (Routers, Firewall etc.) that enable establishment of communication/access checks for these infrastructure are most times not properly configured, prone to vulnerabilities and or allows access due to trust thereby exposing its asset to possible threats (Randy, 2017; Rashid, et al, 2014). A typical example of how attackers exfiltrated data from their target stealthily would be to use among other means Internet Control Message Protocol (ICMP) echo request, Alshamrani et al (2019); Daniel (2018); Randy (2017); Singh et al (2003) which will evade scanning completely because it is considered benign (Shick and Horneman, 2014; Rashid et al, 2014). As described by Nicho and Khan (2014) an "Advanced Persistent Threat (APT) is a term accorded to a new breed of insidious threats that use multiple attack techniques and vectors conducted by stealth to avoid detection so that hackers can retain control over target systems unnoticed, for long periods of time." Figure 1 shows the characteristic features of an APT that are hindering security solutions from detecting them. The identified features depict the stealthiness Binde et al (2011) of the attacks leading to their poor detection. Code obfuscation Binde et al (2011); Rashid et al (2014) is a stealthy and complex way of presenting malicious

[*] Corresponding author
E-mail address: okwarajerry@gmail.com

codes such that systems find them unclear, unreadable and therefore are not able to determine what they are meant for in some cases while in other cases they present as genuine code but are actually concealing malicious codes (Cert-UK, 2014; Binde et al, 2011). This increases the chance of the malware to propagate the system and span longer periods of time without being detected.

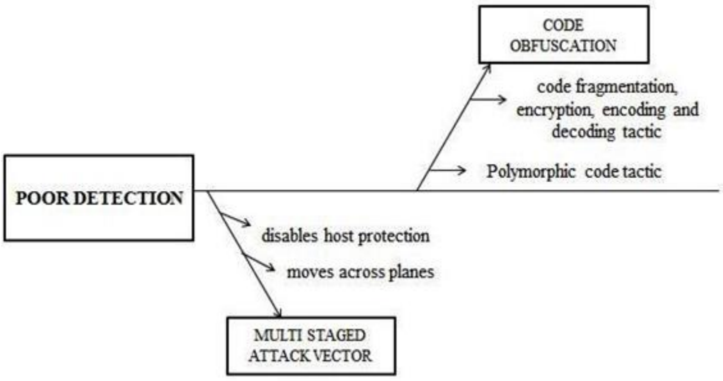

Figure 1 Poor detection of APTs aided by their Stealthiness

2. Internet Control Message Protocol (ICMP) Traffic

ICMP is part of the Internet Protocol suite along with Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) etc. Unlike other protocols, ICMP is not for data exchange rather it is used for establishing the status of the process of end to end communication in a network (Daniel, 2018). It has no dealings with any form of movement of data end to end between devices or any means that needs such services to achieve their objectives or meet their goal. A typical example of the usage of ICMP would be with the common achievement of the use of the PING tool in network communication (Shick and Horneman, 2014). The PING tool is actually used in network communication to confirm and establish fact for the connectivity of devices in a network using ICMP protocol to confirm that a particular computer whose IP address is being PING can respond to a message or determine if they are online.

3. Related work

The use of an anomaly based machine learning system, Machine Learning Advanced Persistent Threat (MLAPT) was proposed by (Ghafir et al, 2018). It is a phased system that uses Threat detection (8 different methods for various APT attack stages), Alert correlation (correlates alerts from the threat detection phase to determine the particular APT attack stage they belong thereby reducing false alert) and Attack prediction phase (correlates results from the alert correlation phase) to predict an early APT attack before its cycle is completed. The Alert correlation phase uses three steps Alert filter to filter alerts for redundancy and reduce noise, Alert clustering to collate related alerts and Correlation indexing which helps to determine how close alerts are in their individual clusters. The attack prediction phase uses Decision tree learning, Support Vector Machine, K-nearest neighbours and Ensemble Classification algorithms to train the prediction model so that the best one with prediction accuracy is considered for use. The solution is able to predict attacks based on the machine learning dependent already known record of monitored network. As against their comparators Brogi and Tong (2016); Giura and Wang, (2012) this work is an autonomous system because it has a phase that can generates its own detection events. Their result from the research saw a reduced case of false positive rates. Its shortcoming is poor coverage of the APT attack lifecycle therefore a need to include more detection modules to cater for that.

In order to use the ensemble of classifiers to improve detection accuracy of novel attacks, Prusti and Jena (2015) in their work supported the fact that single classifiers are not sufficient for improved detection accuracy. The objective was to obtain an improved detection rate with lowered false positive rate and at minimal cost. They used a predictive model based on ensemble to classify normal and attack classes. For ensemble, they used Support Vector Machine, Decision Tree and Neural Network as their combination of base learners. They presented a dataset with 38465 instances using AdaBoost, Logitboost and Bagging ensemble methods with the majority voting combination rule and their result showed AdaBoost to emerge as the best with 97.44% accuracy. Another work on ensemble is that by Mkuzangwe and Nelwamondo, (2017) where they proposed the use of Adaboost (using weighted majority voting combination rule), decision tree (decision stump) and the information gain concept. The idea was performance bound such that the average information gain associated with the features used in building the ensemble is obtained and used a measure for the classification accuracy of their work. The Network Intrusion Detection system was launched using NSL KDD dataset filtered for Neptune and normal connections with classification of both types of connections in perspective. Ensemble method was also used by Sornsuwit and Jaiyen, (2015) in their work on detecting User to Root (U2R) and Remote to Local or User (R2L) attacks. They aimed at removing redundant features to improve their dataset, decrease false alarm rate as well as increase detection accuracy for the attacks in question. Naïve Bayes, Decision Tree, K-Nearest Neighbour, Support Vector Machine and Multilayer Perceptron were used as weak learners with Adaboost ensemble method. Their result showed Naïve Bayes and Multilayer Perceptron with the best result for sensitivity and specificity respectively.

4. Machine learning techniques

Machine learning techniques Ghafir et al (2018); Aburomman and Reaz (2017); Chandran et al (2015); Shah et al (2015) have been applied in researches for APT detection but not without their setbacks resulting from the changing attackers approach and sophistication especially (Nguyen, 2017). The application of Machine learning techniques to solve some of the most difficult issues in computing has witnessed improvements upon existing solutions in terms of providing some form of reinforcements and support for them (Sultana et al, 2018; Nguyen, 2017; Shah et al, 2015; Ford and Siraj, 2014). Within this machine learning techniques lies potentials that may be applicable for providing remedial actions to some challenging and complex situations accorded to their ability to adapt swiftly to new and unknown circumstances.

Figure 2 shows the classifiers which are grouped into single, hybrid and ensemble classifiers as listed by (Shah et al, 2015; Aburomman and Reaz, 2017). These classifiers are models which the process of classification techniques are used to learn from, following a set of training data instances. A test instance is then used on one of the trained model at the testing stage so as to ascertain which class they belong to Tan, (2018); Friedberg et al, (2015). An anomaly detection technique that is classification based operates in two (2) stages with the first one identified as the training stage (learning a classifier with labeled training data) and the second one as testing stage (classifying a test instance into normal or anomalous) (Aljawarneh et al, 2018; Chandola et al, 2009).

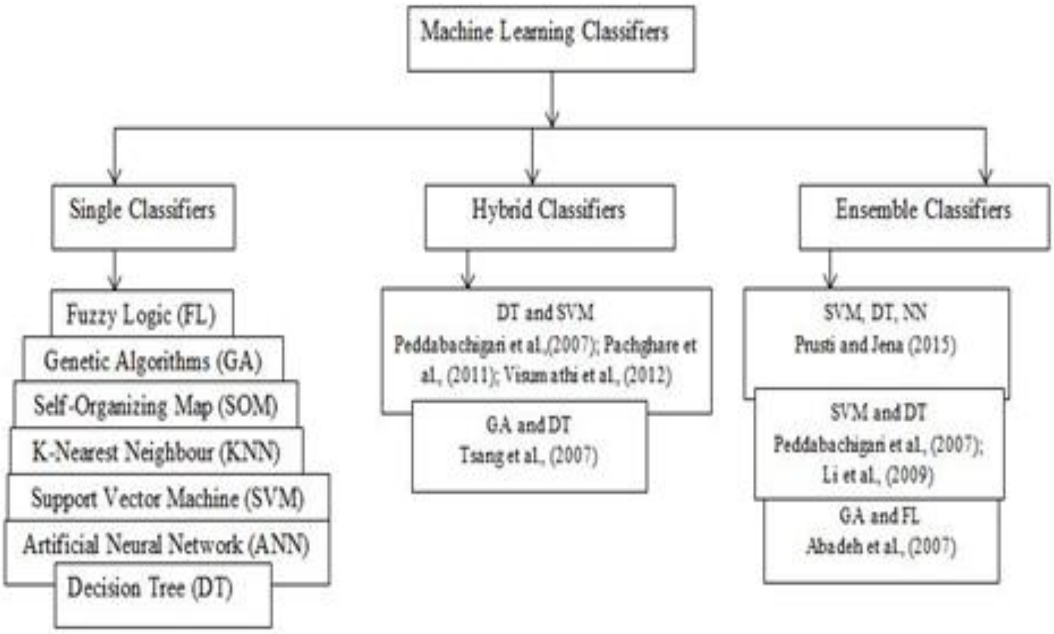
Machine Learning Classifiers
Single Classifiers
Hybrid Classifiers
Ensemble Classifiers
Fuzzy Logic (FL)
Genetic Algorithms (GA)
Self-Organizing Map (SOM)
K-Nearest Neighbour (KNN)
Support Vector Machine (SVM)
Artificial Neural Network (ANN)
Decision Tree (DT)
DT and SVM
Peddabachigari et al.,(2007); Pachghare et al., (2011); Visumathi et al., (2012)
GA and DT
Tsang et al., (2007)
SVM, DT, NN
Prusti and Jena (2015)
SVM and DT
Peddabachigari et al., (2007); Li et al., (2009)
GA and FL
Abadeh et al., (2007)

Figure 2 Machine Learning Classifiers

4.1. Support Vector Machine

Support Vector Machine (SVM) is one of the many machine learning techniques deployed for providing classification and regression solutions to mention a few. The principle of this technique relies on optimal hyper-plane in a high-dimensional space. The goal of support vector machine is to design a hyper plane that classifies all training vectors into two classes such that the best one is the hyper plane that leaves the maximum margin from both classes. The following constraints shows how training data are classified for a binary classification.

For all x elements that are members of a class +1, the following constraints are satisfied:

$w^T x + b \geq +1$

For all x elements that are members of a class -1, the following constraints are satisfied:

$w^T x + b \leq -1$

Following the above, the aim of SVM is to determine the optimal hyper plane define by $w^T x + b = 0$ which maximizes the margin of the two conditions presented in the preceding statements. Having found the optimal plane, the decision function is defined as $f(x) = \text{sign}(w^T x+b)$.

4.2. Random Forest

Random Forest is a classifier which comprises of a collection of decision trees such that for the resulting class to emerge, prediction will depend on the votes received from constituent trees that make up the forest. The emerging model is formed as a result of algorithm obtained from a collection of trees or better still forest of trees where the root node and constituent internal nodes represent the input variables. The available data is represented in a tree form or order thereby making it a lot easier to interpret. The aim of this technique is to have a model that can make prediction based on the provided class attribute or label (normal or attack for this work).

4.3. Decision Tree

In other to make decisions, a Decision tree will leverage the formation of tree structure with the leaves as nodes such that possible solutions are spread across and tends towards the root so as to follow the most efficient possibility. Its learning algorithm is described as below:

- Choose the attribute that has the highest information gain
- Set P_i as probability of an arbitrary tuple in D which belongs to class C_i estimated by $|C_i, D| / |D|$
- Entropy to classify a tuple is computed as:
  Info (D) = $\sum_{i=1}^{m} Pi \log 2\, Pi$ (4.1)
- Information needed to classify D is computed as:
- Info_A (D) = $\sum_{i=1}^{v} \frac{Dj}{D} * Info(Dj)$ (4.2)
- Information gained by branching on attribute A is calculated as:
  Gain_A = Info (D) – Info_A (D) (4.3)

4.4. Dimensionality Reduction

This technique has been widely used for network traffic data preprocessing in order to come up with an ideal output for use with various machine learning processing applications. Following the redundancy found in input data, smaller set of new variables can be found in them such that each is a combination of the input variables that have the similar information as the input; this technique forms the dimensionality reduction process. Principal Component Analysis (PCA) is a statistical dimensionality reduction technique. The main purpose of this is to find a new coordinate system in which the input data can be expressed with many less variables without a significant error (Sorzano et al, 2014). The following are the steps in principal component analysis:

- Mean center the data
- Compute the covariance matrix $\sum$
- Calculate the eigenvalues and eigenvectors of $\sum$
- Eigenvector with largest eigenvalue λ_1 is 1st principal component
- Eigenvector with Kth largest eigenvalue λ_k is Kth principal component
- $\lambda_k / \sum_i \lambda_i$ = proportion of variance captured by Kth principal component

Raw captured network traffic packets data are usually large in size or volume as a result of the fact that the capturing device consider quite a number of attributes which may not be necessary for use when analyzing them or feeding them to support systems (Nguyen, 2017). This case if not properly handled in terms of pre-processing which might result to building up of noise within the resultant dataset (Nguyen, 2017; Ahmed et al, 2016) and consequently constitutes reasons for false alarms in detection systems (Ahmed et al, 2016). Some of the best practices for reducing these large volume of data as identified by Nguyen, (2017) are through the use of dimensional reduction where principal components are selected Ahmed et al, (2016); Shyu et al, (2003) such that more dimensions of variables are mapped and streamlined to fewer ones, clustering (items that have close similarities are grouped), Spearman's rank correlation and statistical sampling. For this work, dimensional reduction is implemented with some principal components selected from the entire dataset (Ahmed et al, 2016).

4.5. Ensemble Classifiers

When a collection of selected Classifiers are trained at the same time to provide solution to an identified common problem and their outputs are combined or aggregated to improve accuracy, the process is referred to as an ensemble method (Aburomman and Reaz 2017; Sornsuwit and Jaiyen, 2015). Under certain conditions where the Classifier output are independent on each other and make errors in an independent manner it is possible that combining the output of several classifiers, we can get a resultant classifier which is better that the constituent Classifiers. The multiple learners will have different decisions and therefore they can be combined by several available ways to determine a particular decision. There are two (2) processes involved in achieving this task where the first one is to make appropriate decision on the selection of ensemble of classifiers that are relevant and sufficient for the task at hand as well as their ability to be diversely used. This entails generating different base learners with different algorithms that will be used for the ensemble. The learners may make there different errors in the instance space but by combining them together, a stronger learner can emerge. The next step in the process would be to come up with a strategy to put the results or decisions of particular Classifiers together such that reinforces accurate decisions and subsequently incapacitates erroneous prone Classifications (Aburomman and Reaz, 2017; Prusti and Jena, 2015). This technique of bringing a selection of classifiers together has recorded successes having been implemented in the area of intrusion detection systems to enhance their performance (Mkuzangwe and Nelwamondo, 2017; Sornsuwit and Jaiyen 2015; Prusti and Jena 2015). By using ensemble, low bias and variance for individual learners is achieved and where both varies for low and high, a balance can be created between them. By combination, statistical, computational and representational issues arising from training data and the hypothesis space can be reduced thereby flawing the potential of choosing the wrong hypothesis as with single classifiers. Ensemble method has been applied for prediction on other domains likes credit card fraud detection, weather forecast aviation and medicine to mention a few.

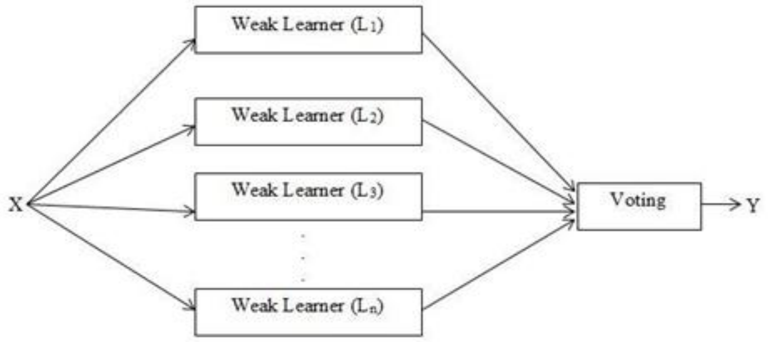

Figure 3 Ensemble of Classifiers Architecture (Sornsuwit and Jaiyen, 2015)

Figure 3 show the architecture of ensemble of classifiers such that the collection of weak learners X which constitute inputs are combined to form a stronger one. In the diagram, there are weak learner L_1, L_2, L_3...L_n forms the set of inputs X and the output of the process is Y which represents the stronger classifier a process which is carried out by way of voting. Voting methods can be through majority, plurality, weighted or soft voting. Majority voting is the commonly used method as classifiers will vote for a particular class label such that the resultant one emerges as having received over half of the entire votes. In any case that no class label gets more than half of the votes, they are rejected and no prediction would be made. On the other hand, Plurality voting adopts or considers the class label with the highest votes in count. There is no rejection here as there would always be a class label with the highest vote count. Weighted voting allows single classifiers that have showed some level of variance in their performance to be combined for reinforcement purpose thereby emerging a stronger learner. In a case where single classifiers generate class probability outputs, Soft voting is adopted. Given that they are all presented with equal opportunities, soft voting would get their average and resultantly obtain a better one (Prusti and Jena, 2015).

5. Results and discussion

This section presents the results obtained from the individual classifiers as well as the ensemble. The tool used for performing the experiment is Weka version 3.8. The Voting algorithm was used for the ensemble and Support Vector Machine, Decision Tree and Random Forest where the individual classifiers applied. The voting algorithm works by using a set of classifiers or models whose predictions are combined in such a way their mean or mode is chosen or they are allowed to vote on the result will be. Majority voting was used as the combination rule for the listed classifiers to determine how the decisions of the models are combined to produce a result.

5.1. Support Vector Machine (SVM)

The following shows results obtained from using SVM classifier on the dataset. Table 1 shows the SVM result for 3148 instance where 60% for training and 40% for testing. It shows that 2843 instance where correctly classified at 90.31% accuracy while 305 instances were incorrectly classified at 9.68% accuracy. A breakdown of the analysis shows that 1871 attack instances were correctly classified as attack, 304 instances of attack were incorrectly classified as normal. In furtherance to the analysis, 1 normal instance was incorrectly classified as an attack while 972 normal instances were correctly classified as normal. The Confusion matrix for the result is shown here:

$$\begin{array}{c} \begin{matrix} a & \quad b \end{matrix} \\ \begin{pmatrix} 1871 & 304 \\ 1 & 972 \end{pmatrix} \end{array} \quad \begin{array}{l} a = \text{Attack} \\ b = \text{Normal} \end{array}$$

Table 1 Result for SVM

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.860	0.001	0.999	0.0860	0.925	0.809	0.930	0.956	Attack
	0.999	0.140	0.762	0.999	0.864	0.809	0.930	0.761	Normal
Weighted Average	0.903	0.044	0.926	0.903	0.906	0.809	0.930	0.896	

5.2. Decision Tree (DT)

The following shows results obtained from using SVM classifier on the dataset. Table 2 shows the DT result for a total of 3148 instances where 60% for training and 40% for testing. It shows that 2847 instances where correctly classified at 90.43% accuracy while 301 instances were incorrectly classified at 9.56% accuracy. A breakdown of the analysis shows that 1874 attack instances were correctly classified as attack, 301 instances of attack were incorrectly classified as normal. In furtherance to the analysis, 0 normal instance was incorrectly classified as an attack while 973 normal instances were correctly classified as normal. The Confusion matrix for the result is shown here:

$$\begin{array}{c} \begin{matrix} a & \quad b \end{matrix} \\ \begin{pmatrix} 1874 & 301 \\ 0 & 973 \end{pmatrix} \end{array} \quad \begin{array}{l} a = \text{Attack} \\ b = \text{Normal} \end{array}$$

Table 2 Result for DT

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.862	0.000	1.000	0.862	0.926	0.811	0.950	0.966	Attack
	1.000	0.138	0.764	1.000	0.866	0.811	0.950	0.828	Normal
Weighted Average	0.904	0.043	0.927	0.904	0.907	0.811	0.950	0.923	

5.3. Random Forest

The following shows results obtained from using RF classifier on the dataset. Table 3 shows the RF result for a total of 3148 instances where 60% for training and 40% for testing. It shows that 2848 instances where correctly classified at 90.47% accuracy while 300 instances were incorrectly classified at 9.52% accuracy. A breakdown of the analysis shows that 1875 attack instances were correctly classified as attack, 300 instances of attack were incorrectly classified as normal. In furtherance to the analysis, 0 normal instance was incorrectly classified as an attack while 973 normal instances were correctly classified as normal. The Confusion matrix for the result is shown here:

$$\begin{array}{c} \begin{array}{cc} a & \quad b \end{array} \\ \begin{pmatrix} 1875 & 300 \\ 0 & 973 \end{pmatrix} \end{array} \qquad \begin{array}{l} a = \text{Attack} \\ b = \text{Normal} \end{array}$$

Table 3 Result for RF

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.862	0.000	1.000	0.862	0.926	0.812	0.954	0.968	Attack
	1.000	0.138	0.764	1.000	0.866	0.812	0.954	0.842	Normal
Weighted Average	0.905	0.043	0.927	0.905	0.907	0.812	0.954	0.929	

5.4. Ensemble of classifiers

The following shows results obtained from using ensemble classifier on the dataset. Table 4 shows the ensemble result for a total of 3148 instances where 60% for training and 40% for testing. It shows that 2848 instances where correctly classified at 90.47% accuracy while 300 instances were incorrectly classified at 9.53% accuracy. A breakdown of the analysis shows that 1875 attack instances were correctly classified as attack, 300 instances of attack were incorrectly classified as normal. In furtherance to the analysis, 0 normal instance was incorrectly classified as an attack while 973 normal instances were correctly classified as normal. The Confusion matrix for the result is shown here:

$$\begin{array}{c} \begin{array}{cc} a & \quad b \end{array} \\ \begin{pmatrix} 1875 & 300 \\ 0 & 973 \end{pmatrix} \end{array} \qquad \begin{array}{l} a = \text{Attack} \\ b = \text{Normal} \end{array}$$

Table 4 Result for ensemble

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.862	0.000	1.000	0.862	0.926	0.812	0.931	0.957	Attack
	1.000	0.138	0.764	1.000	0.866	0.812	0.931	0.764	Normal
Weighted Average	0.905	0.043	0.927	0.905	0.908	0.812	0.931	0.898	

Table 5 show a comparison of the results obtained with applying PCA before running ensemble and PCA after running ensemble. It shows that there is a slight change in the accuracy obtained by using PCA on the dataset before using them for SVM, RF, DT and ensemble classifiers.

Table 5 Results comparison

Classifiers	Without PCA (%)	With PCA (%)
SVM	88.53	90.31
RF	90.37	90.47
DT	90.18	90.43
Ensemble	90.18	90.47

From the results presented, the model recorded 90.47% accuracy on its ability to classify an attack and normal instance of a given network traffic. This result shows an improvement upon the use of the ensemble classifier with Random Forest taking the highest vote from the ensemble using majority voting algorithm for the combination rule. In addition, having applied the dimensionality reduction using the principal component analysis technique which for this work has presented in section 3.4, an improved result can be seen as presented in table 5 The main purpose of this is to find a new coordinate system in which the input data can be expressed with many less variables with less of a significant error. From the confusion matrix provided in the results, class imbalance constituted to the model's capturing of a fraction of the attack class instances as normal.

6. Conclusion

ICMP protocol is a benign connection for testing connectivity to nodes on the network therefore is allowed to pass by Firewalls. This protocol has been exploited for exfiltrating data as shown in this work. Network traffic logs are large in volume and carry information that can be used for detecting attacks if properly analyzed and processed. Using PCA, attributes of the log can be meaningfully reduced to produce a better data which can be fed for further machine learning processes. If attacks and normal network traffic can be properly classified, then the accuracy for detecting an attack would have been improved as the detection system can tell the difference between an attack and a normal traffic. Single classifier can perform low in terms of classification accuracy therefore the need to use ensemble of classifiers for making a decision on the best predicted class to choose from all individual classifiers.

APTs can adopt so many techniques to achieve carrying out their attacks without being noticed by available detection systems. This work used ICMP echo request as a case. In this regards, this work suggest the following for further research:

- HTTP (Hyper Text Transmission Protocol)
- DNS (Domain Names Service)

TCP (Transmission Control Protocol)

Compliance with ethical standards

Acknowledgments

This research would not have been complete without support from Dr. Anazida Binti Zainal, Dr. Nurudeen Mahmud Ibrahim and Juan Ramon Moya.

Disclosure of conflict of interest

The Authors of this Paper whose names appear above, hereby declare that there is no conflict of interest as the case may be.

References

[1] Aburomman AA and Reaz MBI. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & Security*, *65*, 135-152.

[2] Ahmed M, Mahmood ANand Hu J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31.

[3] Aljawarneh S, Aldwairi M and Yassein MB. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152-160.

[4] Alshamrani A, Myneni S, Chowdhary A and Huang D. (2019). A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. IEEE Communications Surveys & Tutorials.

[5] Binde B, McRee R and O'Connor TJ. (2011). Assessing outbound traffic to uncover advanced persistent threat. SANS Institute. Whitepaper, 16.

[6] Brogi G and Tong VVT. (2016). November. Terminaptor: Highlighting advanced persistent threats through information flow tracking. In 8th IFIP International Conference on New Technologies, Mobility and Security.

[7] Cert-UK Code-obfuscation:https://www.ncsc.gov.uk/content/files/protected_files/Guidance_files/Code-obfuscation.pdf..

[8] Chandran S, Hrudya P and Poornachandran P. (2015). August. An efficient classification model for detecting advanced persistent threat. In 2015 international conference on advances in computing, communications and informatics (ICACCI*)* 2001-2009.

[9] Daniel M. (2018). December. An ICMP Reference: Online source.

[10] Ford V and Siraj A. (2014). October. Applications of Machine Learning in Cyber Security. In Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering.

[11] Friedberg I, Skopik F, Settanni G and Fiedler R. (2015). Combating advanced persistent threats: From network event correlation to incident detection. Computers & Security, 48, 35-57.

[12] Ghafir I, Hammoudeh M, Prenosil V, Han L, Hegarty R, Rabie K and Aparicio-Navarro FJ. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. Future Generation Computer Systems, *89*, 349-359.

[13] Giura P and Wang W. (2012). December. A context-based detection framework for advanced persistent threats. In Cyber Security (CyberSecurity), 2012 International Conference on 69-74.

[14] Mkuzangwe NN and Nelwamondo F. (2017). November. Ensemble of classifiers based network intrusion detection system performance bound. In 2017 4th International Conference on Systems and Informatics (ICSAI) 970-974.

[15] Nguyen TN. (2017). Attacking Machine Learning models as part of a Cyber Kill Chain. arXiv preprint arXiv:1705.00564.

[16] Nicho M and Khan S. (2014). Identifying Vulnerabilities of Advanced Persistent Threats: An Organizational Perspective. International Journal of Information Security and Privacy (IJISP), *8*(1), 1-18.

[17] Prusti D and Jena SK. (2015). *An Efficient Intrusion Detection Model Using Ensemble Methods*, Master of Technology Dissertation, Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India.

[18] Randy FS. (2017). Detecting Compromised Systems Analyzing the Top Eight Indicators of Threat Traffic*:* White Paper Commissioned by LogRhythm Oct. 2017.

[19] Rashid A, Ramdhany R, Edwards M, Kibirige Mukisa S, Ali Babar M, Hutchison D and Chitchyan R. (2014). Detecting and preventing data exfiltration.

[20] Shah AA, Hayat MS and Awan MD. (2015). Analysis of Machine Learning Techniques for Intrusion Detection System: A Review. Infinite Study.

[21] Shick D and Horneman A. (2014). Investigating advanced persistent threat 1 (apt1). Software Engineering Institute (SEI) CERT Division Technical Report (CMU/SEI-2014-TR-001). TechnicalReport/2014_005_001_90523.pdf (Accessed 18th March, 2019)

[22] Shyu ML, Chen SC, Sarinnapakorn K and Chang L. (2003). A novel anomaly detection scheme based on principal component classifier. Miami univ coral gables fl dept of electrical and computer engineering.

[23] Singh A, Nordström O, Lu C and Dos Santos AL. (2003). July. Malicious ICMP tunneling: Defense against the vulnerability. In Australasian Conference on Information Security and Privacy 226-236. Springer, Berlin, Heidelberg.

[24] Sornsuwit P and Jaiyen S. (2015). October. Intrusion detection model based on ensemble learning for U2R and R2L attacks. In 2015 7th international conference on information technology and electrical engineering (ICITEE) 354-359.

[25] Sorzano COS, Vargas J and Montano AP. (2014). A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877.

[26] Sultana N, Chilamkurti N, Peng W and Alhadad R. (2018). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, 1-9.

[27] Tan PN. (2018). Introduction to data mining. Pearson Education India.

How to cite this article

Okwara JC and Buba AK. (2020). Ensemble classifiers for detection of advanced persistent threats. Global Journal of Engineering and Technology Advances, 2(2), 01-10.