INTRUSION DETECTION SYSTEM ON A COMPUTER NETWORK USING AN ENSEMBLE OF RANDOMIZABLE FILTERED CLASSIFIER, K-NEAREST NEIGHBOR ALGORITHM

Supported by tetfund TERTIARY EDUCATION TRUST FUND

Asaju, La'aro Bolaji[1], Peter Bamidele Shola[2], Nwadike Franklin[2] and Hambali Moshood Abiola[1]
[1]Department of Computer Science, Federal University Wukari, PMB 1020, Taraba State, Nigeria
[2]Department of Computer Science, University of Ilorin, Kwara State, Nigeria
*Corresponding author: lbasaju@fuwukari.edu.ng

Received: January 29, 2017 **Accepted:** March 30, 2017

Abstract: Intrusion detection is the process of monitor the event occurring in a computer network and analyzing them for signs of intrusions. In recent years, the needs of internet are felt in lives of many people. Accordingly, many studies have been done on security in virtual environments. The earliest techniques such as authentication, firewalls and encryption could not be utilized toprovide the complete internet security. Similarly, the motivations to create a new solution approach and a defense system in cyber environment led to introduction of numerous intrusion detection systems (IDS); i.e. different algorithms. However, the results have shown that using a machine learning and knowledge discovery techniques are very effective and increase the detection accuracy of anomalies on a real time computer networks. Therefore, this study presents an ensemble of randomizable filtered and K-Nearest Neighbor classifier for selecting features in order to enhance network intrusion detection and increase the accuracy of anomaly detection in a real time computer network. Furthermore, data preprocessing and analysis are undertaken using KDDcup99 dataset and a filter, such that best features are selected and irrelevant, redundant, noisy data are removed. The selected features are passed as input to the based classifier for classification and optimization. The based classifier KNN is employed to increase the amount of learning, the efficiency of classification and thereby increasing the authenticity of intrusion detection. Experimental results obtained reveals that the proposed algorithm is very promising accurately detecting anomalies on a computer network.

Keywords: Attack, intrusion detection, KDDCUP99, K-nearest neighbor, randomizable filter

Introduction

Intrusion detection could be thought as the process where events occurring in a computer system or network are monitored and analyzed for possible signs of intrusions. It could be also described as an attempt where the confidentiality, integrity, availability, control of system or network resource are being compromised or bypassed the security mechanisms of a computer or network. The usefulness of intrusion detection is not only to detect successful intrusions, but to also monitor attempts to break security, by providing important information for timely counter measure (Heady *et al.*, 1990; Sundaram, 1996).

However, the problem encountered by many organizations in keeping or protecting the information and infrastructures from unauthorized users (intruders) cannot be over emphasized. Apart from obtaining confidential and privileged information, the intruders are more exciting and fascinating any time they succeed in break the security and have access to unauthorized computer networks. Dueto the ease and availability of the internet, it is very easy tohave access to information on how to attack network and methods used by intruders. In the recent time, studies have shown there is an increasing number of cybercrimes and attacks on various organization networks. The motivation to minimize the attacks by protecting the information and network infrastructures of these organizations led to introduction of numerous solution techniques by the researchers in the domain. The traditional-based protection methods that have been employed for data protections are user authentication and authorization, data encryption, avoiding programming error by interlocution of the structured programming methodology and the firewall approach. However, this methodology is not efficient in terms of anomaly detection and none of these approaches have been able to protect the integrity of network and data completely (Balogun and Jimoh, 2015). Cybercrimes and network attacks have spark the need to build a robust defense system in order to protect networks and computer systems. Cyber-attacks include destabilizing network, gaining unauthorized access to files with privileges, mishandling and misuse of software. An intrusion detection system is to automatically scan network activities and detect attacks on networks.

Intrusion detection system

IDS is a system software that detects attack on a network or computer system. Basically, IDS is classified into: misuse detection and anomaly detection (Durst, 1999). In misuse system, the signature of known attacks is stored in the database. Any data similar to that signature is classified as attacks. Whereas anomaly detection is referred to as statistical knowledge about normal activity. This type of detection approach could be categorized into semi-supervised and unsupervised anomaly detection (Erbacher, 2002). Semi-supervised anomaly detection approach needs a set of purely normal training data from which the profile of normal behavior can be found. However, if such data contains some hidden attacks, then, the approach may fails to detect the future instances of these attacks. On the other hand, in unsupervised anomaly detection approach, the profile of normal behavior is setup with unlabeled training data that consists of both normal as well as anomalous samples. Intrusions correspond to deviations from the normal activity of system. The anomaly detection system has high false positive/ negative alarm rate compared to misuse detection systems. Many drawbacks of conventional approaches are:

- Signature-based IDSs must be automatic to detect each attack and thus must be continually updated with signatures of new attacks.
- Many signature-based IDSs have hardly defined signatures that prevent them from detecting variant of common attacks.
- Anomaly detection approaches usually create a large number of false alarms due to the random nature of users and networks.
- Anomaly detection approaches often need wide "training sets" of system occurrence records in order to characterize normal behavior patterns.
- Application-based IDSs may be weaker than host based IDSs to being attacked and disabled since they run as an application on the host they are monitoring.

K-nearest neighbor

K-Nearest Neighbor (k-NN) is a type of instance based learning method for classifying objects based on the closest training examples in the feature space (Lee *et al.*, 1999). It is a lazy type of learning where the function is approximated locally and all computations are delayed until classification is done. The k-nearest neighbor algorithm is one of the simplest machine learning algorithms. An object is classified based on the majority vote of its neighbors, object is assigned to the most common class amongst its k nearest neighbors. If *k=1*, then the object is merely assigned to the class of its nearest neighbor (Lane, 2000). The k-NN algorithm employs all labelled training instances to serveas a model of the target function. In classification phase, k-NN employs a similarity-based search approach to determine a locally optimal hypothesis function. Generally, k-NN is used for intrusion detection in combination with statistical schemes (anomaly detection) (Lee *et al.*, 1999). The benefit of the k-NN Algorithm as a classifier for an IDS is that, it is analytically tractable. k-NN is simple to implement and uses local information, which can produce highly adaptive behavior. Lastly, another important strength of the KNN algorithm is that it lends itself very easily for parallel implementations (Lee *et al.*, 1999). One of the weak point of K-NN Algorithm as a classifier for an IDS is that itrequired large storage. K-NNs are also recognized to be highly prone to the curse of dimensionality and low rate in classifying test tuples.

Randomizable filter classifier

This method employed an arbitrary classifier on data that has been passed through an arbitrary filter. Similar to the classifier, the structure of the filter worked exclusively on the training data and test instances will be processed by the filter without altering their structure (Hall *et al.*, 2009). In using randomizable filter (RF) as an ensemble base classifier, each base classifier is built using a different random number seed (but based on the same data). The final prediction is a straight average of the predictions generated by the individual base classifiers. Class for creating a committee of random classifiers. The base classifier (that forms the committee members) needs to implement the Randomizable interface.

Network intrusion detection

Intrusion Detection is a problem of identifying unauthorized (Park, 2007; Proenca Jr., 2006) users in a computer system. It is also defined as the problem of protecting computer network systems from being compromised. The fundamental principle in IDS comprising Network Based Intrusion Detection System (NIDS) originated from anomaly Host Intruder Detection System (HIDS) research based on Denning's pioneering work (Denning, 1986). A host-based IDS offers much more germane information than Network-based IDS. HIDS are used efficiently for analyzing the network attacks, for instance, it can provides information on exactly what the attacker did and the commands he used, the files he tamper with, rather than just a vague accusation.

NIDS systems gather information from the network itself rather than from each separate host (Global Info. Assurance Cert.). The NIDS reviews the network attacks while packets are moving through the network. The network sensors are equipped with attack signatures that are rules on system via pattern matching of known signatures, and most network-based systems allow advanced users to define their own signatures (Global Info. Assurance Cert.). Attack on the sensor is based on signature of previous known attacks and the operation of the monitors will be transparent to the users (SANS Penetration Testing).The transparency of the monitors decreases the likelihood that an intruder will be able to discover it and nullify its capabilities without much efforts (Base, 2000). Network Node IDS (NNIDS) agents are deployed on every host within the network being protected (Karthikeyan and Indra, 2010).

Network attack

Attacks were classified, according to the goal of attacker. Each attack type falls into one of the following four categories (Mukkamala, 2003).

Denials-of Service (DoS) denial of service attack is a class of attacks in which an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

Probe or Probing is a class of attacks in which an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits.

User-to-Root (U2R) User to root exploits are a class of attacks in which an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain Root access to the system.

Remote-to-Local(R2L) A remote to user attack is a class of attacks in which an attacker sends packets to a machine over a network-but who does not have an account on that machine; exploits some vulnerability to gain local access as a user of that machine.

Table 1: Classification of attacks on KDD data set

Identify the type	meaning	Specific Classification Identification
Normal	Normal Record	Normal
Dos	Denial of Service Attack	Neptune,pod,land, back,smurf, teardrop
Probing	Monitoring and other exploration activities	Ipsweep,nmap,portsweep,satan etc.
R2L	Unauthorized access from remote machine	Imap,ftp_write,Warezclient ,multihop, phf,spy,guess_passwd, warezmaster
U2R	Unauthorized access to local super user privileges by ordinary users	Loadmodule, buffer_overflow,rootkit, per

Weka 3.7.13 tool was use to run the experiment of the proposed methods. The usage of RF is to remove the noisy and inconsistent data from the kddcup99 connection dataset. 60% of the kddcup99 dataset is utilized as training set while the remaining 40% is employed to test the method. The KNN is used as the classification algorithm in which it is trained to understand the type of data to classify. The framework of the proposed network anomaly detection system is provided in Fig 1.

Proposed IDS system

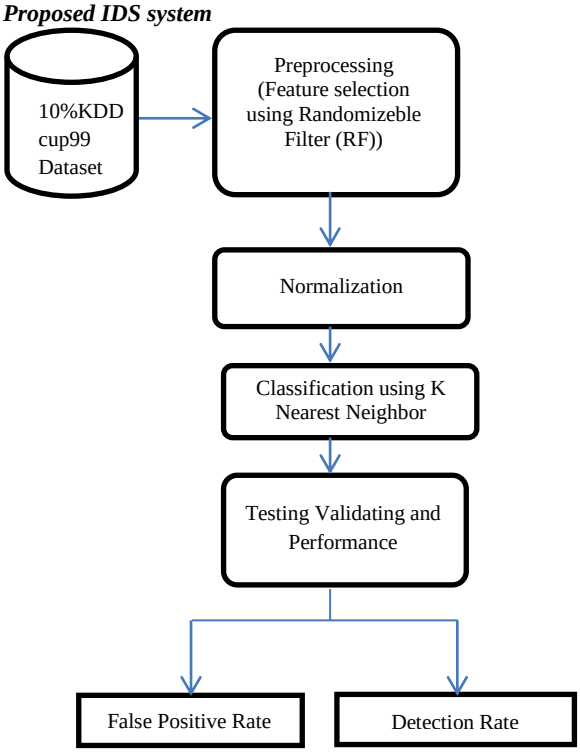

Fig. 1: Block diagram of the proposed IDS

KDDCUPP99 data set description

The data set provided for the 1999 KDD Cup that is originally prepared by MIT Lincoln labs for the 1998 Defense Advanced Research Projects Agency (DARPA'98) Intrusion Detection Evaluation Program, with the objective of evaluating research in intrusion detection, and it has become a benchmark dataset for the evaluation of IDSs.DARPA'98 is about 4 gigabytes of compressed raw (binary) topdump data of 7 weeks of network traffic. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labelled as either normal or an attack, with exactly one specific attack type.The attack type is divided into five major categories: DOS-Denial of Service (e.g. a mail bomb), R2L- Unauthorized access from a remote machine (e.g. sendmail), U2RUnauthorized access to super user or root functions (e.g. a buffer overflow attack), Probing-surveillance and other probing for vulnerabilities (e.g. port scanning) (Chandrasekhar, 2012). For each record, KDDCup99 training data set contains 41 fixed feature attributes and a class identifier. In the 41 fixed feature attributes, nine characteristic properties is the discrete type, and others are continuous. In this paper we will use the subset of the original dataset which consist the distinct records. In order to make the data suitable for intrusion detection, we need to preprocess the data. First in order to reduce the number of attribute we apply the information gain algorithm. Second in order to format the dataset we use the Normalization process to normalize the dataset.

Feature selection

Data Preprocessing is the important task for reducing the attribute of KDD cup 1999 dataset. This process is carried out in two steps (Harshit Saxena, 2014). The first step involves mapping symbolic-valued attributes to numeric valued attributes. In second step attributes are reduced by using Information gain. In this first we calculate the entropy of each attribute and subtract the entropy of each attribute by entropy of class label attribute. This calculates the information gain of each attribute. Then we select only those attribute which have positive information gain and other attributes are discarded. The KDD dataset has 42 attributes and after applying information gain 18 attributes remain.

Proposed algorithm

1: Input: (*S*, *K*, F)
Target matrix: S = $\{x_i, y_i\}_1^N$
Number of neighbors: $K \in \{1, ..., N-1\}$
Filter: F $\{1 \ldots N\} \in X$
Initialize $err \leftarrow 0$
For each query point x_i(i= 1 to N) do
find Knearest neighbors X_K using Euclidean distance
majority vote between the k points to determine the class label of x_i
Filter x_i
If $x_i$is misclassified
$err \leftarrow err + 1/N$
end if
end for
Output: *err*

Adaptation of randomfilter in training k-NN

1:**Input: (T, K, M, F)**
Training set: T ={X, Y} $\in R^{N\ X\ P}$
Number of neighbors: $K \in \{1,...,N-1\}$
Number of iterations: $M \in$R
Filter: F$\{1 \ldots N\} \in X$
2: Initialization: A = $\{1... 1\} \in R^{N\ X\ 1}$
[*the weight vector*]
3: **for***m* = 1 to *M* do
4: **for** each query point *xi* (i= 1 to N) **do**
$D_A(x_i, X) = \frac{(x_i, X)}{A}$
[*calculate the weighted distance*]
5: F (x_i) [*Filter to remove noisy and redundant attribute*]
6: **if** point x_i is classified incorrectly
[*update the neighbors' weights*]

7:$errm \leftarrow errm + \frac{(1)}{N}$
8: renormalizes *A* so that $\sum_{=1}^{n} A(i) = N$
9: **end for**
10: ξm ← errm − errm−1
11: **end for**
12: **Output:** the weight vector *A*

At completion, the learned vector *A* can be used along for k-NN classification at each testing point $\mathbf{t_0}$. Specifically, given the training set $T = \{x_i, y_i\}_1^N$, the distance between $\mathbf{t_0}$and each training point $x_i$is defined as

$D(\mathbf{t0}, xi) = \| \mathbf{t0} - \boldsymbol{x_i} \|_{A_I} = \frac{\sqrt{(\mathbf{t0} - xi)^T - (\mathbf{t0} - xi)}}{A_i}$

Classification

To classify a class-unknown from dataset ***X***, the *k*-NN classifier algorithm ranks the dataset neighbors among the training dataset vectors, and uses the class labels of the *k* most similar neighbors to predict the class of the new dataset. The classes of these neighbors are weighted using the similarity of each neighbor to ***X***, where similarity is measured by Euclidean distance or the cosine value between two dataset vectors. The cosine similarity is defined as follows:

$$\boldsymbol{Sim}(X, D_J) = \frac{\sum_{t_i \in (X \cap D_j)} x_i\ X\ d_{ij}}{||X||_2\ X\ ||D_j||_2}$$

Where
X is the dataset represented as a vector;
D_J is the *j*th training dataset;

t_i is a is the attribute shared by X and D_J;
x_i is the weight of attribute t_i in $\boldsymbol{X}$;
d_{ij} is the weight of attribute t_i in dataset D_J;
$||X||_2$ is the normalization of X
$||D_j||_2$ is the normalization of D_j

Performance measure

Classification Accuracy: It is the ability to predict categorical class labels. This is the simplest scoring measure. It calculates the proportion of correctly classified instances.

- True Positive (TP): If the instance is positive and it is classified as positive
- False Negative (FN): If the instance is positive but it is classified as negative
- True Negative (TN): If the instance is negative and it is Classified as negative
- False Positive (FP): If the instance is negative but it is classified as positive

Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

False Alarm $= \frac{FP}{FP+TN}$

Table 2: Results of classification

Attack Type	Accuracy of Classification		
	KNN	RF + KNN	False Alarm Rate
Normal	99.97%	100%	0%
Probe	99.44%	99.97%	0.03
DoS	98.32%	100%	0%
U2R	86.54%	100%	0%
R2L	98.76%	100%	0%
Summary	96.60%	99.99%	0.1%

Conclusions

Intrusion Detection is a process of detection intrusion in a computer system in order to increase the security. Intrusion detection is an area in which more and more sensitive data are stored and processed in networked system (Harshit Saxena, 2014). We proposed an ensemble of RF Classifier and KNN approach for building IDS. The RF classifier remove irrelevant and noisy data with a reduce dimension. The output is then use as an input to the KNN which then use the dataset for classification. KNN parameter uses the k most similar neighbors to predict the class of the new dataset. The classes of these neighbors are weighted using the similarity of each neighbor to X, where similarity is measure by Euclidean distance or the cosine value between two dataset vectors. We analyze that there are several technique which provide good detection rate in case of Denial of Service (DoS) attack. But fail to achieve good detection rate in case of U2R and R2L attack. Many of the algorithm does not perform well in detecting the attacks like U2R and R2L. We perform series of experiment on KDD Cup 99 for acquiring more accuracy. We have used Confusion matrices for evaluation of our proposed technique and the result are obtained on the basis of evaluation metrics namely, Sensitivity, Specificity and Accuracy. As we saw we got the best result as compared to the previous algorithm and it is clear our technique perform well.

References

Bace RG 2000. *Intrusion detection*. Sams Publishing.

Balogun AO & Jimoh RG 2015. Anomaly Intrusion Detection Using an Hybrid Of Decision Tree And K-Nearest Neighbor. *J. Adva. Scientific Res. & Applic. (JASRA)*, 2(1): 67-74.

Chandrashekhar AM & Raghuveer K 2012. Performance evaluation of data clustering techniques using KDD Cup-99 Intrusion detection data set. *Inter. J. Infor & Network Secu.*, 1(4): 294 - 305.

Denning DE 1987. An intrusion-detection model. *IEEE Transactions on Software Engineering*, 2: 222-232.

Durst R, Champion T, Miller E, Spagnuolo L & Witten B 1999. Testing and evaluating computer intrusion detection systems. *Communications of the ACM*, 42(9): 15-15.

Erbacher RF, Walker KL & Frincke DA 2002. Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics & Applications*, 22(1): 38-47.

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P & Witten IH 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1): 10-18.

Heady R, Luger G, Maccabe A & Servilla M 1990. The architecture of a network level intrusion detection system. Technical Report, Department of Computer Science, University of New Mexico.

Karthikeyan KR & Indra A 2010. Intrusion detection tools and techniques-a survey. *Int. J. Computer Theory & Eng.*, 2(6): 901

Lane TD 2000. Machine learning techniques for the computer security domain of anomaly detection. Ph.D. Thesis, Purdue Univ., West Lafayette, IN, USA

Lee W, Stolfo SJ & Mok KW 1999. A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE Symposium onSecurity and Privacy*, pp. 120-132.

Patcha A & Park JM 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12): 3448-3470.

Proença Jr. ML, Coppelmans C, Botolli M & Mendes LS 2006. Security and reliability in information systems and networks: Baseline to help with network management, pp.149 -157.

Saxena H & Richariya V 2014. Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain. *International Journal of Computer Applications*, 98(6): 25 – 29.

Sundaram A 1996. An introduction to intrusion detection. *ACM Crossroads*, 2(4): 3-7.

Wong WT & Lai CY 2006. Identifying important features for intrusion detection using discriminant analysis and support vector machine. In: *IEEE 2006 Inter. Conf. onMachine Learning and Cybernetics*, pp. 3563-3567.