International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-3 Issue-3, July 2013

Comparative Study of Selected Data Mining Algorithms used for Intrusion Detection

Ajayi Adebowale, Idowu S.A, Anyaehie Amarachi A.

Abstract. -- In the relatively new field of data mining and intrusion detection a lot of techniques have been proposed by various research groups. Researchers continue to find ways of optimizing and enhancing the efficiency of data mining techniques for intrusion attack classification. This paper evaluates the performance of well known classification algorithms for attack classification. The focus is on five of the most popular data mining algorithms that have been applied to intrusion detection research; Decision trees, Naïve bayes, Artificial neural network, K-nearest neighbor algorithm and Support vector machines. We discuss their advantages and disadvantages and finally we induce the NSL-KDD dataset with the respective algorithms to see how they perform.

Key Words— Data mining, Intrusion detection, decision trees, Naive bayes, Artificial neural network, k-nearest neighbor, Support vector Machines, NSL-KDD

I. INTRODUCTION

Intrusion detection starts with instrumentation of a computer network for data collection. Pattern-based software ‘sensors’ monitor the network traffic and raise ‘alarms’ when the traffic matches a saved pattern [3]. Security analysts decide whether these alarms indicate an event serious enough to warrant a response. A response might be to shut down a part of the network, to phone the internet service provider associated with suspicious traffic, or to simply make note of unusual traffic for future reference [14]. If the network is small and signatures are kept up to date, the human analyst solution to intrusion detection works well. But when organizations have a large, complex network , the human analysts quickly become overwhelmed by the number of alarms they need to review. Some large networks generate over one million alarms per day [10]. And that number is increasing. This situation arises from ever increasing attacks on the network, as well as a tendency for sensor patterns to be insufficiently selective (i.e., raise too many false alarms). Commercial tools typically do not provide an enterprise level view of alarms generated by multiple sensor vendors. Commercial intrusion detection software packages tend to be signature-oriented with little or no state information maintained [3]. These limitations have led to the investigation of the application of data mining to intrusion detection.

Manuscript Received July, 2013.
Ajayi Adebowale, Computer Science department, Babcock University, Ilishan, Nigeria.
Idowu S.A, Computer Science department, Babcock University, Ilishan, Nigeria.
Anyaehie Amarachi, Computer Science department, Babcock University, Ilishan, Nigeria.

II. DATA MINING AND INTRUSION DETECTION SYSTEMS

Data Mining is the automated process of going through large amounts of data with the intention to discover useful information about the data that is not obvious. Useful information may include special relations between the data, specific models that of the data that repeats itself, specific patterns, and ways of classifying it or discovering specific values that fall out of the “normal” pattern or model [2]. In other to understand how data mining can help advance intrusion detection, it is important to know how current IDS work to identify an intrusion.
Intrusion detection systems are a combination of hardware and software resources aimed at protecting the confidentiality, availability and integrity of a computer system or network. To an analyst sitting in front of an IDS, an ideal system would alert on all malicious connections, whether it is a known or novel attack [6]. However the search for the ideal IDS continues and the amount of network data is increasing.
Besides the issue of data overload facing network analysts due to increasing complexity and large size of networks, traditional methods for intrusion detection are based on extensive knowledge of signatures of known attacks that are provided by human experts. The signature database has to be manually revised for each new type of intrusion that is discovered. A significant limitation of signature-based methods is that they cannot detect emerging cyber threats. In addition, once a new attack is discovered and its signature developed, often there is a substantial latency in its deployment across networks [8].
Data mining can help improve intrusion detection by addressing the above mentioned problems in the following ways

(i). Remove normal activity from alarm data to allow analysts to focus on real attacks
(ii). Identify false alarm generators and”bad” sensor signatures
(iii). Find anomalous activity that uncovers a real attack
(iv). Identify long, ongoing patterns (different IP address, same activity.

III. DATA MINING ALGORITHMS FOR IMPLEMENTING IDS

Data mining techniques can be differentiated by their different model functions and representation, preference criterion, and algorithms [4]. The main function of the model that we are interested in is classification, as normal, or malicious, or as a particular type of attack. The resultant model is presented in a form depending on the data mining technique used.

Common representations for data mining techniques include rules, decision trees, linear and non-linear functions (including neural nets), instance-based examples, and probability models [10].
A brief overview of five popular algorithms that have been applied to the study of intrusion detection is given below.

A. Decision Trees

Decision tree is a predictive modeling technique most often used for classification in data mining [4]. The Classification algorithm is inductively learned to construct a model from the preclassified data set. Each data item is defined by values of the attributes and classification may be viewed as mapping from a set of attributes to a particular class. Each non-terminal node in the decision tree represents a test or decision on the considered data item. Choice of a certain branch depends upon the outcome of the test. To classify a particular data item, we start at the root node and follow the assertions down until we reach a terminal node (or leaf). A decision is made when a terminal node is approached [8].

An advantage of using decision tree algorithms for IDS is that its construction does not require any domain knowledge. Hence a data mining expert with little knowledge of networking can help build accurate decision tree models. Another significant advantage is that decision trees can handle high dimensional data [11]. This increases the suitability of decision tree algorithms for IDS especially while considering the heterogeneity of network connection data and its ever increasing size. Decision trees are able to process both numerical and categorical data (this suits the alphanumeric nature of network connection data)[12]. Finally, decision tree representations are easy to understand hereby making it easier for the network analyst to identify network trends and deviations from normal traffic [11].
Disadvantages of decision tree algorithms in IDS are that the output attribute must be categorical (normal or anomaly) and limited to one output attribute. Decision tree algorithms are also known to be unstable and trees created from numeric datasets can be complex [8].

B. Naïve Bayes

A Bayesian network is a model that encodes probabilistic relationships among variables of interest [2]. Naive Bayesian classifiers use the Bayes theorem to classify the new instances of data. Each instance is a set of attribute values described by a vector, X = (x1, x2 . . . , xn). Considering m classes, the sample X is assigned to the class Ci if and only if P(X|Ci)P(Ci) > P(X|Cj)P(Cj) for all j in (1, m) such that j =6 I. Naïve Bayesian technique is generally used for intrusion detection in combination with statistical schemes, a procedure that yields several advantages, including the capability of encoding interdependencies between variables and of predicting events, as well as the ability to incorporate both prior knowledge and data [10].
Naïve Bayesian classifiers simplify the computations and exhibit high accuracy and speed when applied to large databases. A disadvantage of using Bayesian networks is that their results are similar to those derived from threshold-based systems, while considerably higher computational effort is required [8]. Lack of available probability data is a significant disadvantage of the naïve Bayesian approach to IDS. Another disadvantage is that in naïve bayes approach it is assumed that the data attributes are conditionally independent [2] which is not always so (it should be noted however that despite this, Bayesian classifiers give satisfactory results because focus is on identifying the classes for the instances, not the exact probabilities).

C. Artificial Neural Networks

Neural networks (NN) are systems modeled based on the working of the human brain [4]. As the human brain consists of millions of neurons that are interconnected by synapses, a neural network is a set of connected input/output units in which each connection has a weight associated with it. The network learns in the learning phase by adjusting the weights so as to be able to predict the correct class label of the input [10].
Neural networks have been used both in anomaly intrusion detection as well as in misuse intrusion detection [13]. For anomaly intrusion detection, neural networks were modeled to learn the typical characteristics of system users and identify statistically significant variations from the user's established behavior. In misuse intrusion detection the neural network would receive data from the network stream and analyze the information for instances of misuse.
An Advantage of neural network algorithms as a classifier in IDS is that it requires less formal statistical training [8]. Neural networks are able to implicitly detect complex nonlinear relationships between dependent and independent variables. Neural networks are also known to exhibit a high tolerance to noisy data (this would come in handy while dealing with noisy connection data). Neural networks also boast of an availability of multiple training algorithms [7].
It can be argued that the "Black box" nature of neural networks as limited its potential as a classifier for IDS [4]. Another disadvantage of the neural network algorithm is its relatively greater computational burden. Artificial neural networks are known for their proneness to over fitting and to require long training time [8].

D. K-Nearest Neighbor

K-Nearest Neighbor (k-NN) is an instance based learning method for classifying objects based on the closest training examples in the feature space [12]. It is a type of lazy learning where the function is only approximated locally and all computations are deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. If k=1, then the object is simply assigned to the class of its nearest neighbor [9].
The k-NN algorithm uses all labeled training instances as a model of the target function. During the classification phase, k-NN uses a similarity-based search strategy to determine a locally optimal hypothesis function. Test instances are compared to the stored instances and are assigned the same class label as the k most similar stored instances. Generally it is used for intrusion detection in combination with statistical schemes (anomaly detection) [12].
An advantage of the K-Nearest Neighbor Algorithm as a classifier for an IDS is that it is analytically tractable. KNN is simple in implementation and it uses local information, which can yield highly adaptive behavior.

Finally, a major strength of the KNN algorithm is that it lends itself very easily to parallel implementations [11].
One of the weaknesses of the K-Nearest Neighbor Algorithm as a classifier for an IDS is its large storage requirements. KNNs are also known to be highly susceptible to the curse of dimensionality and slow in classifying test tuples.

E. Support Vector Machine

Support Vector Machines have been proposed as a novel technique for intrusion detection. An SVM maps input (real-valued) feature vectors into a higher-dimensional feature space through some nonlinear mapping. SVMs are developed on the principle of structural risk minimization [11]. Structural risk minimization seeks to find a hypothesis (h) for which one can find lowest probability of error whereas the traditional learning techniques for pattern recognition are based on the minimization of the empirical risk, which attempt to optimize the performance of the learning set. Computing the hyper plane to separate the data points i.e. training an SVM leads to a quadratic optimization problem [6]. SVMs can learn a larger set of patterns and be able to scale better, because the classification complexity does not depend on the dimensionality of the feature space [12]. SVMs also have the ability to update the training patterns dynamically whenever there is a new pattern during classification.
An advantage of Support Vector Machine as a classifier for an IDS is that they are highly accurate. In addition, Support Vector Machines are able to model complex nonlinear decision boundaries and are less prone to over fitting than other methods.
A disadvantage of SVMs as a classifier for an IDS is its high algorithmic complexity and extensive memory requirements [9]. This consequently makes the speed both in training and testing slow.

IV. EXPERIMENTAL METHODOLOGY

A. The Dataset

The dataset used in this research is the NSL-KDD dataset. NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD cup'99 data set. It is basically a processed version of the KDD cup’99 dataset. This dataset enables researchers to train their algorithms on the full dataset (because of its smaller amount of records) instead of using a portion of the full dataset as in the case of the KDD cup’99 data set. More information about this dataset can be found in [16].

B. Data Mining Tools

Our experiments were done using Weka 3.6.7. Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection.
The Experiments were carried out on a 32-bit Windows 7 Ultimate operating system, with 2 GB of RAM and a Pentium (R) Dual-core CPU at 2.20Hz per core. Due to the iterative nature of the experiments and resultant processing power required, the java heap size for weka-3-6.*7* was set to 1024 MB.

To assess the effectiveness of the algorithms, each one of them was trained on the full NSL-KDD data set using a ten-fold validation test mode in a Weka (Waikato Environment for Knowledge Analysis) environment. To test and evaluate the algorithms we use 10-fold cross validation. In this process the data set is divided into 10 subsets. Each time, one of the 10 subsets is used as the test set and the other k-1 subsets form the training set. Performance statistics are calculated across all 10 trials. This provides a good indication of how well the classifier will perform on unseen data.

C. Performance Measurement Terms

(1). Correctly Classified Instance
The correctly and incorrectly classified instances show the percentage of test instances that were correctly and incorrectly classified. The percentage of correctly classified instances is often called accuracy or sample accuracy.
(2). Kappa Statistics
Kappa is a chance-corrected measure of agreement between the classifications and the true classes. It's calculated by taking the agreement expected by chance away from the observed agreement and dividing by the maximum possible agreement. A value greater than 0 means that the classifier is doing better than chance.
(3) Mean Absolute Error, Root Mean Squared Error, Relative_Absolute_Error
The error rates are used for numeric prediction rather than classification. In numeric prediction, predictions aren't just right or wrong, the error has a magnitude, and these measures reflect that.
Detection of attack is measured by following metrics:
(i). True positive (TP): Corresponds to the number of detected attacks and it is in fact an attack.
(ii). False positive (FP): Or false alarm, corresponds to the number of detected attacks that is in fact normal.
The accuracy of an intrusion detection system is measured regarding to detection rate and false alarm rate.

V. RESULTS AND DISCUSSIONS

A summary of our experiments on the five selected algorithms are given below.

Table 1: Comparisons of different Classification Algorithms on Weka

Algorithm	Method Name	Correctly Classified Instances in Full Dataset (%)	Incorrectly Classified Instances in Full Dataset (%)	Classification Time (seconds)
SVM	functions.SMO	97.3285	2.6715	735.74
ANN	functions.MultilayerPerceptron	95.7594	4.2406	7543
KNN	lazy.IBk	99.4403	0.5597	0.03
NB	bayes.NaiveBayes	89.5919	10.4081	1.6
DT	trees.J48	99.5594	0.4406	11.06

Table 2: Performance of Support Vector Machines, Artificial Neural Network, K-Nearest Neighbour, Nave-Bayes and Decision Tree Algorithms on the full NSL-KDD dataset.

Parameter	SVM	ANN	KNN	NB	DT
Correctly classified instances	24519	24123	25051	22570	25081
Incorrectly classified instances	673	1069	141	2622	111
Kappa Statistic	0.9462	0.9136	0.9888	0.7906	0.9911
Mean Absolute Error	0.0267	0.0545	0.0056	0.1034	0.0064
Root Mean Squared Error	0.1634	0.197	0.0748	0.3152	0.0651
Relative Absolute Error	5.3676%	11.107%	1.1333%	20.7817%	1.2854%

Table 3: True Positive and False Positive Rates of chosen classification Algorithms on Weka

Algorithm	Method Name	TP Class Normal	TP Class Anomaly	FP Class Normal	FP Class Anomaly	Weighted Average TP	Weighted Average FP
SVM	functions.SMO	0.986	0.959	0.041	0.014	0.973	0.029
ANN	functions.MultilayerPerceptron	0.956	0.959	0.041	0.044	0.958	0.043
KNN	lazy.IBk	0.996	0.993	0.007	0.004	0.994	0.006
NB	bayes.NaiveBayes	0.912	0.877	0.123	0.088	0.896	0.106
DT	trees.J48	0.996	0.996	0.004	0.004	0.996	0.004

From the results of our experiments, the decision tree algorithm (J.48; java implementation of Ross Quinlan's C4.5) gave the best detection rate. It correctly classified 25081 instances of the full NSL-KDD dataset, achieving a detection rate of 99.5594%. J.48 also had the best kappa statistic at 0.9911.

Considering computational performance however, K-Nearest Neighbour algorithm (lazy.IBK in Weka) proved to have a faster build time (Time it takes to build model on network training data) at 0.03 seconds while having a detection rate of 94.5919% as shown in table 1.Naïve Bayes (bayes.Naivebayes) had the second best build time at 1.6 seconds but a detection rate of 89.5914%.

Computational performance is particularly important when considering real-time classification of potentially thousands of simultaneous networks traffic. From our experiments, J4.8 (Java implementation of Quinlan's C4.5 decision tree classifier) appears to be the best suited for real-time classification tasks due to its relatively fast classification speed and high detection rate.

VI. CONCLUSION AND FUTURE WORK

While various algorithms have been applied to intrusion detection research it is important to note that each of the approaches has its own advantages and disadvantages. Table 2 shows the performance of these algorithms in classifying connection records (NSL-KDD dataset). Despite the fact that some algorithms gave a better detection rate than the others it is pertinent to note that different classifiers have different knowledge regarding the problem and they approach the problems differently. In literature, it has been suggested that combining more than one data mining algorithm may be used to remove disadvantages of the other [1][5][12]. Thus a combined approach (Ensemble) is advised while selecting a mode to implement intrusion detection system. Future work in this regard involves exploring methods of enhancing the performance of data mining algorithms. We will be applying an ensemble of smart learners and a meta-classification module to the NSL-KDD dataset, in view to enhancing the performance of J4.8 classifier.

REFERENCES

1. Axelsson, S. (2000). "The base-rate fallacy and the difficulty of intrusion detection", ACM Trans. Information and System Security 3 (3), pp. (186-205).
2. Barbara, D., Wu, N. and Jajodia, S. [2001]. "Detecting Novel Network Intrusions Using Bayes Estimators", Proceedings Of the First SIAM Int. Conference on Data Mining, (SDM 2001), Chicago, IL.
3. Bloedorn et al. (2003). Data Mining for Network Intrusion Detection: How to Get Started. The MITRE Corporation McLean, VA.
4. Brugger, S. (2004). Data Mining Methods for Network Intrusion Detection, University of California, Davis.
5. Carbone, P. L. (1997). "Data mining or knowledge discovery in databases: An overview", In Data Management Handbook, New York: Auerbach Publications.
6. Didaci, L., Giacinto, A. & Roli, F. (2002). "Ensemble learning for intrusion detection in computer networks", Proceedings of AI*IA, Workshop on "Apprendimento automatico: metodi e applicazioni", Siena, Italy.
7. Eskin, E., Arnold, A., Preraua, M., Portnoy, L. and Stolfo, S. J. (2002). "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data", (Barbar & Jajodia Eds.), Data Mining for Security Applications. Boston: Kluwer Academic Publishers.
8. Frank, J. (1994). "Artificial intelligence and intrusion detection: Current and future directions", In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology (NIST).
9. Kesavulu, E., Reddy, V. N. and Rajulu, P. G. (2011). "A Study of Intrusion Detection in Data Mining". Proceedings of the World Congress on Engineering 2011 Vol IIIWCE 2011, July 6 - 8, 2011, London, U.K.
10. Lane, T. D. (2000). "Machine Learning Techniques for the computer security domain of anomaly detection", Ph.D. Thesis, Purdue Univ., West Lafayette, IN.
11. Lappas, T. and Pelechrinis, K. (2006). Data Mining Techniques for (Network) Intrusion Detection Systems, Department of Computer Science and Engineering Riverside, Riverside CA. [9]. Lee, W. & Stolfo, S.J. (1998). Data mining approaches for intrusion detection, In Proc. of the Seventh USENIX Security Symp., San Antonio, TX.
12. Lee, W., S. J. Stolfo, & Mok, K. W. (1999). "A data mining framework for building intrusion detection models," In Proc. of the 1999 IEEE Symp. On Security and Privacy (pp. 120-132), Oakland, CA: IEEE Computer Society Press.
13. Lee, W., Stolfo, S.J. & Mok, K.W. (1999). "Mining in a data-flow environment: Experience in network intrusion detection," (Chaudhuri, S. & Madigan, D. Eds.). Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99) (pp. 114-124), San Diego, CA: ACM,
14. Lee, W. & Stolfo, S.J et al. (2000). "A data mining and CIDF based approach for detecting novel and Distributed intrusions", In Proc. of Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France.
15. Mukkamala, S. & Sung, A. H. (2008). "Identifying key variables for intrusion detection using neural networks", Proceedings of 15th International Conference on Computer Communications (pp. 1132-1138).
16. (SANS: FAQ: Data Mining in Intrusion Detection) http://www.sans.org/securityresources/idfaq/data_mining.php
17. Tavallaee,M. , Bagheri, E. , Lu,W. & Ghorbani, A. 2009. "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

AUTHORS PROFILE

Ajayi Adebowale, holds a B.Sc. degree in Mathematics (Computer Science) and an M.Sc degree in Computer science from University of Agriculture Abeokuta, Nigeria and Babcock University, Nigeria respectively. He is a Cisco Certified Network Associate and his research interests include Knowledge discovery in databases, Machine learning and Information security. He can be contacted at deboxyl@gmail.com

Sunday Idowu PhD is an Associate Prof. in the Department of Computer Science, Babcock University, Ilishan-Remo, Ogun State, Nigeria. He holds a Masters degree in Software Engineering, and PhD in computer science from Andrews University, MI, USA and University of Ibadan, Oyo State, Nigeria, respectively. His research areas are Software Engineering, Web Application Development and Security. He has published works in several journals of international repute. He can be contacted at saidowu07@gmail.com

Anyaehie Amarachi, received her B.Sc. in Computer Systems and Information Technology from Eastern Mediterranean University, North Cyprus, in 2011. She is currently studying for an M.Sc. degree in Computer Science at Babcock University, Nigeria. Her current research interests include Project Management, Web application development, and Information Systems Security and Assurance. She can be contacted at amarachi.anyaehie@yahoo.co.uk