Indian Journal of Science and Technology, *Vol 12(37), DOI: 10.17485/ijst/2019/v12i37/147551, October 2019*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Select the Best Machine Learning Algorithms for Prediction and Classification of Intrusions using KDD99 Intrusion Detection Dataset

Sara Tamy[1]*, Hicham Belhadaoui[1], Mahmoud Almostafa Rabbah[1], Nabila Rabbah[2] and Mounir Rifi[1]

[1]RITM Laboratory, ESTC, Hassan II University, BP. 8012, Casablanca, Morocco; saratamy@yahoo.fr, belhadaoui_hicham@yahoo.fr, mrabah@gmail.com, rifi.mounir@gmail.com
[2]Laboratory of Structural Engineering, Intelligent Systems and Electrical Energy, ENSAM, Hassan II University, BP. 20000, Casablanca, Morocco; nabila_rabbah@yahoo.fr

Abstract

Objectives/Methods: The growing prevalence of network attacks is an issue that can affect the availability, confidentiality and integrity of critical information for companies. Thus, Intrusion detection systems are increasingly being used to identify unusual access or attacks to secure internal networks. In this study, we will outline the evolution of large data in the intrusion detection system, and apply three supervised learning methods namely: Naïve Bayes, Random tree, and Support Vector Machines SVM, using the kdd99 data set. The purpose of this research is to detect and predict attacks in order to take preventive action against intrusion risks. **Findings:** Investigational results have demonstrated that the random tree gives the highest accuracy at 100%. The results will be useful in choosing the best classification machine learning algorithm for intrusion prediction. **Application/Improvements:** for simulation and testing the performance of algorithms, we have used WEKA (Waikato environment for knowledge analysis), which includes tools for data preparation, classification, regression, clustering, association rule extraction and visualization.

Keywords: Intrusion Detection System (IDS), Machine Learning, KDD99, Naïve Bayes, Random Tree, Support Vector Machines (SVM)

1. Introduction

Today, Intrusion Detection System (IDS) has a very important role in network security. Especially as the number of attacks targeting confidential information is increasing, ranging from 9 million attacks in June 2004 to over 33 million attacks in less than a year[1]. One of the solutions proposed to solve this problem is the use of Network Intrusion Detection Systems (NIDS), which is used to detect attacks by monitoring network activities[2]. Thus, it is required that these systems be accurate and fast to report attacks to network administrators, quickly, in order to take appropriate countermeasures.

Traditionally, Intrusion Detection Systems are based on human technology to distinguish between intrusive and normal traffic. However the massive and increasing volume of data requires the use of machine learning techniques that provide decision tools for analysts, and automatically generate rules to be applied in order to prevent unauthorized access to the computer network[3].

In this study we will use the Waikato Environment for Knowledge Analysis (WEKA), data mining tool for classification. It firstly classifies the data set and then defines the best algorithm to diagnose and predict the intrusion.

The main contributions of this work are: Select the best classifier for the intrusion detection system, comparison of different data mining algorithms for the kdd99 intrusion detection data set and identification of the best solution based on the performance algorithm for intrusion prediction. The rest of the paper is organized as follows: IDS is presented in Section 2, related work is discussed in Section 3, Section 4 describes the Experiment,

*Author for correspondence

Section 5 explains in detail the experiences of using the proposed machine learning models and Section 6 presents the conclusions and future prospects.

2. Intrusion Detection System

An IDS is a mechanism to identify abnormal or suspicious activities on a given target in order to remedy problems as soon as possible. The IDSs are based on several approaches: Scenarios approach This type of IDS uses a database of signatures, and tries to match a data obtained by the information sources of the system, with that already known and Behavioral Approach detect violations of the security policy of the system by observing the behavior of the users and comparing it with a model of behavior considered normal called profile[4].

In this paper we evaluate the performances of classifiers, while trained to identify signatures of attacks.

3. Related Work

In paper 5, the authors have presented a framework of machine learning for intrusion detection system in order to protect wireless sensor networks. Their system is not limited on particular attacks, while machine learning methods allow creating detection model from training data automatically thus reduce human labor to write signature of attacks or indicate the normal behavior of a sensor node.

In paper 6, the authors have presented two orthogonal and complementary approaches to reduce the number of false positives in intrusion detection by using alert post-processing via data mining and machine learning. Furthermore, these two methods can be used jointly in an alert-management system due to their complementary nature. These concepts have been verified on a variety of data sets, and achieved a significant reduction in the number of false positives in both simulated and real environments.

In paper 7 ther authors have used a hybrid intelligent approach by using a combination of classifiers to make the best decision, thus the performance of the resulting model is ameliorated. The procedure consists of filtering the data under supervision or unsupervised using a classifier or clustered on all training data then the output is applied to another classifier to classify the data. They use a two-class classification strategy and a 10-fold cross validation method to obtain the final results that classify intrusion and normal traffic. The simulation shows that their proposed approach is effective with a high detection rate and a low false alarm rate.

In paper 8, the authors examine four learning algorithms for a breast cancer data set. in their efforts to predict breast cancer and reduce the risk of death. they have used several machine learning algorithms which are: Random forest, Naive Bayes, Supportg Vector Machines SVM, and K-Nearest Neighbors K-NN, to choose the more effective one.

4. Experiment

4.1 Weka

Waikato Environment for Knowledge Analysis (WEKA) is a collection of machine learning algorithms designed to facilitate the application of machine learning techniques to a variety of real-world problems, including tools for data preparation, classification, regression, clustering, association rule extraction and visualization[9].

4.2 KDD-99 Data Set

This database, a standard set to be audited, includes a wide variety of intrusions simulated in a military network environment. Many published studies have showed that KDD99 is the most widely used dataset for IDS and machine learning domains, and it is effectively the dataset for these research areas[10]. This data set contains 22 intrusion types (Table 1), 42 attributes, and 494020 instances (Web-1).

Table 1. Training attack types

Training attack types		
back dos	perl u2r	rootkit u2r
Buffer overflow u2r	phf r2l	satan probe
ftp_write r2l	pod dos	smurf dos
guess_passwd r2l	portsweep probe	spy r2l
imap r2l	nmap probe	teardrop dos
ipsweep probe	neptune dos	warezclient r2l
land dos	multihop r2l	warezmaster r2l
loadmodule u2r		

4.3 Classifiers Used

For our work we will use the following classifiers:

1. Naïve Bayes algorithm simplifies learning by assuming that the functions are independent given class. Despite the fact that independence is generally low in practice; Bayes naive is often in competition with a more sophisticatedclassifiers[11].
2. Random Tree is the supervised Classifier that uses a bagging idea to create a random set of data in order to construct a decision tree. This algorithm can be uses for both classification and regression problem[12].
3. SVMs are a learning technique that can be considered as a new method for training classifiers of polynomial functions, neural networks or basic radial functions. Despite the fact that SVM is considered easier to use than neural networks, users are not familiar with[13].

5. Results and Discussion

5.1 Metrics

In this section we will describe the metrics, evaluate the machine learning methods used, and discuss the results.

Accuracy: The accuracy of detection is given by the percentage of correctly classified instances. It is the number of correct predictions divided by the total number of instances in the data set.

The accuracy can be measured by the following equation:

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

Table 2. Naïve bayes performance

TP Rate	FP Rate	Recall	MCC	Precision	F-Measure	PRC Area	ROC Area	Class
0,975	0,005	0,975	0,686	0,485	0,648	0,957	0,999	back
0,995	0,001	0,995	0,796	0,638	0,777	0,636	0,999	teardrop
0,556	0	0,556	0,109	0,022	0,041	0,026	0,999	loadmodule
0,996	0	0,996	0,997	1	0,998	1	1	neptune
0,5	0,003	0,5	0,038	0,003	0,006	0,002	0,977	rootkit
0,75	0	0,75	0,231	0,071	0,13	0,75	0,995	phf
0,954	0,002	0,954	0,743	0,58	0,722	0,604	0,996	satan
0,133	0	0,133	0,049	0,018	0,032	0,08	0,999	buffer_overflow
0,75	0,003	0,75	0,057	0,004	0,009	0,004	0,997	ftp_write
0,952	0	0,952	0,554	0,323	0,482	0,357	1	land
1	0	1	1	1	1	1	1	spy
0,966	0,009	0,966	0,443	0,205	0,339	0,181	0,994	ipsweep
0,429	0	0,429	0,171	0,068	0,118	0,072	1	multihop
0,999	0	0,999	0,998	1	0,999	1	1	smurf
0,985	0,037	0,985	0,115	0,014	0,028	0,372	0,998	pod
0,333	0	0,333	0,333	0,333	0,333	0,386	1	perl
0,478	0,006	0,478	0,259	0,143	0,22	0,3	0,988	warezclient
0,446	0,001	0,446	0,272	0,167	0,243	0,112	0,995	nmap
0,917	0	0,917	0,36	0,141	0,244	0,317	0,942	imap
0,9	0,001	0,9	0,222	0,055	0,103	0,276	0,994	warezmaster
0,907	0,001	0,907	0,767	0,65	0,757	0,506	0,998	portsweep
0,652	0,001	0,652	0,774	0,997	0,788	0,994	0,999	normal
0,943	0,001	0,943	0,256	0,07	0,13	0,477	0,989	guess_passwd

Table 3. Random tree performanc

TP Rate	FP Rate	Recall	MCC		**Precision**	**F-Measure**	**PRC Area**	**ROC Area**	**Class**
0,997	0	0,997	0,997		0,996	0,997	0,995	0,998	back
1	0	1	0,997		0,994	0,997	0,994	1	teardrop
0,444	0	0,444	0,471		0,5	0,471	0,222	0,722	loadmodule
1	0	1	1		1	1	1	1	neptune
0,1	0	0,1	0,183		0,333	0,154	0,033	0,55	rootkit
0,5	0	0,5	0,577		0,667	0,571	0,333	0,75	phf
0,988	0	0,988	0,99		0,992	0,99	0,982	0,995	satan
0,8	0	0,8	0,775		0,75	0,774	0,6	0,9	buffer_overflow
0,5	0	0,5	0,535		0,571	0,533	0,286	0,75	ftp_write
0,857	0	0,857	0,819		0,783	0,818	0,701	0,929	land
0	0	0	0		0	0	0	0,5	spy
0,983	0	0,983	0,985		0,986	0,985	0,973	0,993	ipsweep
0,429	0	0,429	0,378		0,333	0,375	0,161	0,714	multihop
1	0	1	1		1	1	1	1	smurf
0,981	0	0,981	0,983		0,985	0,983	0,966	0,991	pod
0,667	0	0,667	0,667		0,667	0,667	0,444	0,833	perl
0,981	0	0,981	0,98		0,979	0,98	0,963	0,991	warezclient
0,948	0	0,948	0,948		0,948	0,948	0,899	0,974	nmap
0,75	0	0,75	0,866		1	0,857	0,75	0,875	imap
0,75	0	0,75	0,75		0,75	0,75	0,563	0,875	warezmaster
0,98	0	0,98	0,985		0,99	0,985	0,974	0,992	portsweep
0,999	0	0,999	0,999		0,999	0,999	0,999	1	normal
0,925	0	0,925	0,942		0,961	0,942	0,924	0,972	guess_passwd

Table 4. SVM performance

TP Rate	FP Rate	**Recall**	**MCC**	**Precision**	**F-Measure**	**PRC Area**	**ROC Area**	**Class**
0,997	0	0,997	0,994	0,991	0,994	0,99	1	back
0,998	0	0,998	0,999	1	0,999	1	1	teardrop
0	0	0	0	0	0	0,012	0,994	loadmodule
1	0	1	1	1	1	1	1	neptune
0	0	0	0	0	0	0,001	0,882	rootkit
0	0	0	?	?	?	0,222	1	phf
0,972	0	0,972	0,985	0,998	0,985	0,972	0,998	satan
0,6	0	0,6	0,671	0,75	0,667	0,552	0,999	buffer_overflow
0,5	0	0,5	0,577	0,667	0,571	0,298	0,999	ftp_write
1	0	1	0,977	0,955	0,977	0,955	1	land
0	0	0	?	?	?	0,007	1	spy
0,982	0	0,982	0,984	0,987	0,984	0,975	1	ipsweep
0	0	0	0	0	0	0,003	0,998	multihop

1	0	1	1	1	1	1	1	smurf
0,992	0	0,992	0,992	0,992	0,992	0,992	1	pod
0	0	0	0	0	0	0,35	1	perl
0,915	0	0,915	0,92	0,925	0,92	0,863	1	warezclient
0,965	0	0,965	0,949	0,933	0,949	0,901	0,997	nmap
0,833	0	0,833	0,913	1	0,909	0,851	1	imap
0,75	0	0,75	0,769	0,789	0,769	0,706	0,999	warezmaster
0,994	0	0,994	0,996	0,998	0,996	0,993	0,998	portsweep
0,999	0,001	0,999	0,998	0,998	0,998	0,997	0,999	normal
0,943	0	0,943	0,962	0,98	0,962	0,938	1	guess_passwd

Table 5. Weighted average of classifiers

	TP Rate	FP Rate	Re-call	MCC	Preci-sion	F-Measure	PRC Area	ROC Area	Correctly classified instances	In-correctly classified instances	Time to build model(s)
NB	0,928	0	0,928	0,947	0,989	0,95	0,991	1	0,927	0.072	3.8
Random Tree	1	0	1	0,999	1	1	0,999	1	0,999	0.0004	6.38
SVM	0,999	0	0,999	?	?	?	0,999	1	0,999	0.0007	289.99

Recall: also known as sensitivity is the rate of the positive observations that are correctly predicted as positive. The sensitivity or the true positive rate (TPR) is defined by:

$$Sensitivity = \frac{TP}{TP + FN} \quad (2)$$

while the specificity or the True Negative Rate (TNR) is given by:

$$Specificity = \frac{TN}{FP + TN} \quad (3)$$

Precision: Percentage of correctly classified elements for a given class:

$$\Pr ecision = \frac{TP}{TP + FP} \quad (4)$$

F-measure: Combination of precision and recall.

$$F - measure = \frac{2*Precision*Sensitivity}{Precision+Sensitivity} \quad (5)$$

5.2 Result and Discussion

To implement and evaluate the classifiers, we apply the 10-fold cross-validation test which is a method used to evaluate predictive models. It divides the original set in a training sample to build the model, and a set of tests to evaluate it. After applying the pre-treatment and preparation methods, we try to analyze the data and determine the distribution of values in terms of effectiveness and efficiency Table 2-4 present the result of simulation.

In order to compare the performance of the classifiers we have used the weighted average of classifiers, and were based on the number of correctly classified instances, the number of incorrectly classified instances, precision and the model build time (Table 5).**Table 5.** Weighted average of classifiers

After obtaining these results we can visualize it as shown in Figure 1 (graph that illustrates the performance of the classifiers). Random Tree is the best classifier for kdd99 data set with 100% of precision, 100% true positive rate and 0% false positive rate. The time to build the model is longer than naive bayes which guarantees only 98.9% of precision. SVM cannot classify the data set correctly, and it takes a long time to build a model (289.99s).

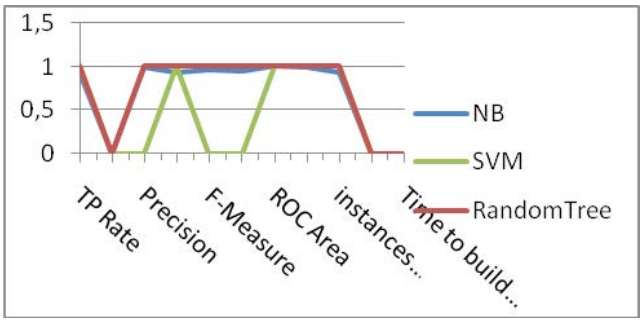

Figure 1. **Performance of the classifiers.**

6. Conclusion

In this article, we have analyzed the kdd99 intrusion detection dataset using tree machine learning algorithms, namely: Naïve Bayes, Random Tree and SVM. The results show that the Random Tree algorithm is the best way to classify all the data. The global performance of naive bayes and the SVM algorithm is unacceptable. Therefore, our future work is to optimise intrusion detection system employing decision tree algorithm and using python programming language.

7. References

1. Panda M, Patra MR. Network intrusion detection using naive bayes, International Journal of Computer Science and Network Security. 2007; 7(12):258–63.
2. Gudadhe M, Prasad P, Wankhade K. A new data mining based network intrusion detection model. In: 2010 International Conference on Computer and Communication Technology; 2010. p. 731–35. https://doi.org/10.1109/ICCCT.2010.5640375.
3. Sinclair C, Pierce L, Matzner S. An application of machine learning to network intrusion detection. In: Proceedings 15th Annual Computer Security Applications Conference; 1999. p. 371–77.
4. Sara T, Rabbah N, Rabbah MA. Study of Strategies for Real-Time Supervision of Industrial Network Security; 2018. p. 1–5.
5. Yu Z, Tsai JJP. A framework of machine learning based intrusion detection for wireless sensor networks. In: 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing; 2008. p. 272–79. https://doi.org/10.1109/SUTC.2008.39. PMCid: PMC2604134.
6. Pietraszek T, Tanner A. Data mining and machine learning-towards reducing false positives in intrusion detection, Information Security Technical Report. 2005; 10(3):169–83. https://doi.org/10.1016/j.istr.2005.07.001.
7. Panda M, Abraham A, Patra MR. A hybrid intelligent approach for network intrusion detection, Procedia Engineering. 2012; 30:1–9. https://doi.org/10.1016/j.proeng.2012.01.827.
8. Khourdifi Y, Bahaj M. Selecting Best Machine Learning Techniques for Breast Cancer Prediction and Diagnosis. In: International Conference Europe Middle East and North Africa Information Systems and Technologies to Support Learning. Springer, Cham.; 2018. p. 565–71. https://doi.org/10.1007/978-3-030-03577-8_61.
9. Witten IH, Frank E, Trigg LE. WEKA: Practical machine learning tools and techniques with Java implementations; 1999. p. 1–5.
10. Özgür A, Erdem H. A review of KDD99 datasetusage in intrusion detection and machine learning between 2010 and 2015, Peer. J. Preprints. 2016; 4:1–22. https://doi.org/10.7287/peerj.preprints.1954.
11. Rish I. An empiricalstudy of the naive Bayes classifier. In: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence; 2001. p. 41–46.
12. Kalmegh S. Analysis of WEKA data miningalgorithm-reptree, simple cart and random tree for classification of Indian news, International Journal of Innovative Science, Engineering and Technology. 2015; 2(2):438–46.
13. Osuna E, Freund R, Girosit F. Training support vector machines: An application to face detection. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 1997. p. 130–36.