International Journal of Security and Its Applications
Vol.7, No.5 (2013), pp.67-80
http://dx.doi.org/10.14257/ijsia.2013.7.5.06

A New Fast and High Performance Intrusion Detection System

Ahmed Ahmim[1] and Nacira Ghoualmi-Zine[2]

Laboratory of Networks and Systems, Department of Computer Science
Badji Mokhtar-Annaba University, Algeria
[1]ahmed.ahmim@lrs-annaba.net, [2]ghoualmi@lrs-annaba.net

Abstract

The cyber-attacks represent one of the most dangerous secret weapons. Intrusion detection system is an important tool to protect our systems and networks against the various forms of attacks. The purpose of this paper is to build a fast and high performance hybrid hierarchical intrusion detection system called NFPHIDS that possesses the following characteristics: have a short training time, detect the low frequent attacks, give a high detection rate for frequent attacks, and give a low false alarm rate. NFPHIDS contains two levels. The first one includes four fast classifiers Random Forest, Simple Cart, Best first decision tree, Naive Bayes used for their excellent performance on the detection of respectively Normal behavior and DOS, Probe, R2L, and U2R. Only five outputs of the first level are selected, and used as inputs of the second level that contains Naïve Bayes as final classifier. The experimentation on KDD99 shows the high performance of our model compared to the results obtained by some well-known classifiers.

Keywords: *Computer Security, Intrusion Detection System, Hierarchical IDS, Hybrid IDS*

1. Introduction

The Internet is an essential part of modern life. We need the Internet for job, shopping, education, and communication, etc. With the increased use of the Internet, the security of computer networks has become a crucial problem due to the importance and sensitivity of the information communicated. The cyber attack represents one of the most dangerous secret weapons. Computer security rallies methods, techniques and tools used to protect systems, data, and services against the accidental or intentional threats in order to ensure confidentiality, availability, and integrity [1]. Nowadays, different techniques and methods are developed to implement the security policy such as authentication, cryptography, firewalls, proxies, antivirus, Virtual Private Network (VPN), and Intrusion Detection System (IDS).

An IDS is either software or hardware system that automates the monitoring process of events occurring in a computer system or network, by analyzing them, to notify the probable security problems [2]. We can classify the intrusion detection system into two categories: anomaly detection and misuse detection [3]. If an IDS uses information about the normal behavior of the system which it monitors, we qualify it as anomaly detection. If an IDS uses information about attacks, we qualify it as misuse detection [4].

Different methods and techniques are used for intrusion detection. In the early stage, the artificial intelligence and learning machine are used. However, face problems like huge network traffic volumes, highly imbalanced data distribution, difficulty to take decision boundaries between normal behavior and attack, and a requirement for continuous adaptation to a constantly changing environment these techniques have shown limitations [5]. And the data mining techniques are used to deal with these limitations.

ISSN: 1738-9976 IJSIA

To build a high performance IDS, various data mining techniques are tested such as Fuzzy Logic [6], Naïve Bayes [7], RIPPER [8], Decision Trees [9], Support Vector Machines [10], and Artificial Neural Networks [11].

This paper is organized as follows. Section 2 presents the related works. We outline in Section 3 the structure and operation mode of the NFPHIDS. The experiments are discussed in section four. Finally, the Section 5 draws the conclusion.

2. Related Works

The main drawback of the simple one level IDS is its inability to combine a high true positive rate of frequent attacks and normal behavior on one hand, and a good positive rate of the low frequent attacks on the other hand. To deal with the above inconveniences, and enhance the performance of IDS, some hierarchical and hybrid models based on a different type of classifiers are developed such as: SHIDS and PHIDS [12], HPCANN [13], IDS based on evolutionary soft computing model using Neuro-fuzzy classifiers [14], IDS based hybrid RBF/Elman neural networks [15], FC-ANN [16], IDS based on hierarchical clustering and support vector machines [17].

SHIDS [12] is a serial hierarchical IDS incrementally build. At first, they have only the normal classifier layer. The normal connection is allowed, but the mischievous connection is detected and stored within a database. When the number of attack reaches the threshold, the clustering algorithm is used to cluster these attacks in different groups. Each group is used to train a new classifier based RBF neural network. These RBF classifiers represent the misuse detection layers. To handle the problem of the influence of upstream errors on the downstream PHIDS [12] is proposed. The latter contains three levels. The first level is an anomaly detection classifier. The second one is a misuse detection classifier that identifies the main group of intrusion. The third level has four classifiers relative to the four categories of attacks: DOS, PROB, R2L, and U2R.

HPCANN [13] is a hierarchical principal component analysis neural network composed of two levels. The top level is a classifier for anomaly detection. The bottom level is serial or parallel ways for misuse detection.

Toosi and Kahani [14] have proposed a hierarchical IDS based neuro-fuzzy networks, fuzzy inference approach and genetic algorithms. At first, a set of parallel neuro-fuzzy classifiers are used to perform the initial classification. Then, the fuzzy inference system based on the outputs of neuro-fuzzy makes the final decision of classification, where the genetic algorithm optimizes the structure of the fuzzy decision engine.

Tong *et al.*, [15] have proposed a hybrid IDS based RBF/Elman neural networks. The RBF neural network is employed as a real-time pattern classification, and Elman neural network is employed to restore the memory of past events.

FC-ANN [16] is hierarchical IDS based neural network and fuzzy clustering. It is composed of three layers. The first layer is a fuzzy clustering that generates the different training subsets. The second layer represents the different neural networks that are trained to formulate different base models. The last layer is a fuzzy aggregation module, which is employed to aggregate these results and reduce the detected errors.

Horng *et al.*, [17] have proposed an IDS that combines a hierarchical clustering algorithm, a simple feature selection procedure, and the SVM technique. At first, the hierarchical clustering algorithm is used to generate training instances. Then, the simple feature selection procedure was applied to eliminate unimportant features from the training set. Finally, the obtained SVM model classifies the network traffic data.

These related works have shown a good global performance, but possess some inconvenient like the long training time and the low detection rate of the low frequent attack like U2R.

4. Our Approach

Our work aims to build a high performance IDS that better detects the low-frequent attacks without losing their high performance on the detection of frequent attacks and normal behavior. Unlike some related works, the built model must give a high performance and train in a very short time.

4.1. NFPHIDS

In this section, we present the different components of NFPHIDS and their usefulness. As shown in the following Figure 1, our model contains two levels:

- The first level: this level contains the different types of classifiers. These latter are selected for their short training time and their highest performance on the detection of one or more class of connection. As illustrated in Figure 1, each classifier gives five predictions relative to the four categories of attacks, and normal behavior. We maintain only the predictions of classes for which the classifiers are selected. These five predictions are used as inputs of the second level.
- The second level: this level contains one classifier used for their high performance and short training time as final classifier. It analyses the selected predictions of the different classifiers of the first level and takes the final decision. The latter can be attack or normal behavior.

4.2. The Operation Mode of NFPHIDS

The operating mode of NFPHIDS is composed of three stages: select the different classifiers of the first level, training stage and test stage.

4.2.1. Select the Different Classifiers of the First Level

In the aim to select the best classifiers for NFPHIDS, we perform two comparative studies between different types of classifiers. In the first one, we compare the different classifiers relative to their training time and their performance on the classification of connection in one of the five classes (DOS, Probe, R2L, U2R, and normal behavior). We select only the five classifiers those give a short training time, a good global true positive rate, and the highest true positive for at least one of the five classes. To perform the second comparative study, we generate a new data set from the five selected predictions of the first level. Then, we use the new training data set to compare the different classifiers relative to the classification of network connections on attack class or normal behavior class. We select the classifier that gives a short training time, a highest true positive rate and the least false alarm rate.

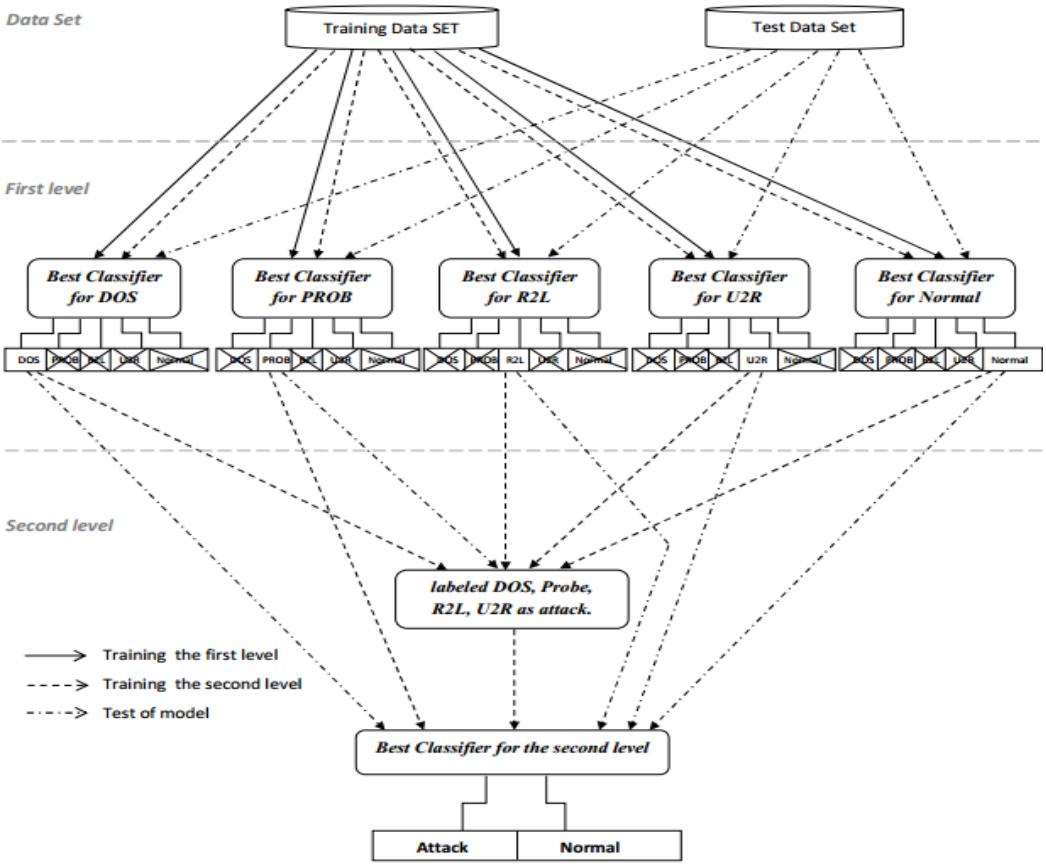

Figure 1. General Structure of NFPHIDS

4.2.2. Training Stage

In this stage, we train our model with the aim to prepare it for the test stage. This stage is composed of two steps:

- Train the first level: we train the different classifiers of the first level with the training data set, where each feature of the training data set represents an input for the classifier.
- Train the second level: a new data set is created from the predictions of the classifiers of the first level. To generate this new Training data set, we associate the selected prediction's results with the correct label as in the following Table 1. The new training data set is used to train the selected classifier of the second level.

Table 1. The New Training Data Set

DOS Prediction	Probe prediction	U2R prediction	R2L prediction	Normal prediction	Label
0.94	0.25	0.17	0.38	0.18	Attack
0.15	0.34	0.18	0.36	0.94	Normal
0.28	0.28	0.89	0.22	0.15	Attack
0.35	0.99	0.38	0.14	0.36	Attack
0.16	0.13	0.25	0.89	0.32	Attack

4.2.3. Test Stage

In this stage, we test the performance of our model after the achievement of the training stage, where we use the test data set. We process each record of the test data set by the different classifier of the first level. Then, we use the selected prediction outputs of the different classifiers of the first level as an input of the classifier of the second level.

4.2.4. Optimization of Training and Test Time

To optimize the training and test time of NFPHIDS, we proposed the distributed architecture detailed in the following Figure 2.

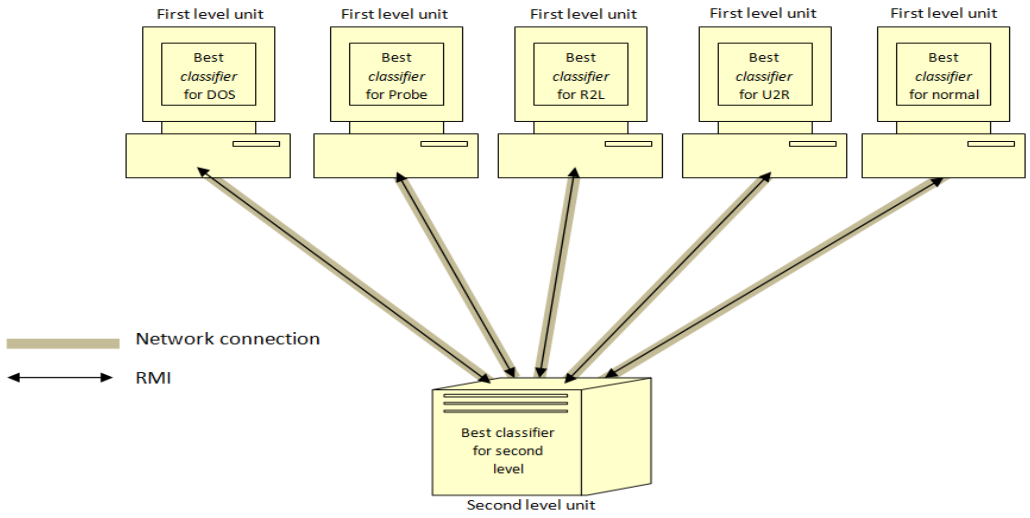

Figure 2. Distributed Architecture of NFPHIDS

This distributed architecture contains two types of units, where all first level units are connected to the second level unit. Each first level unit contains the training data set and one of the selected classifiers. All first level units train their classifier simultaneous by their local training data set. The second level unit generates the new training data set, where it uses the remote method invocation (RMI) to request the predictions of the different first level units for each record of the training data set. Then, it uses the new training data set to train its classifier. This architecture reduces the training time, where it becomes the training time of the slowest classifier of first level in addition to the training time of the second level.

For the test step, the second level unit requests simultaneous all first level units using RMI. Then, it processes their answers by its classifier. The simultaneous request reduces the test time of NFPHIDS.

5. Experiments

This section is divided into three parts. In the first one, we detail the training and test data set. The second part represents a comparative study between eight classifiers, this comparison aim to select the faster and high performance classifiers for the first and second level. The third part represents the comparative study between our new hierarchical model and other well known classifiers.

We have performed a set of experience on KDD99 Cup [18], which represents the most used data set for intrusion detection in the last decade [19, 5]. Weka Data Mining Tools [20]

is used for the implementation of the different classifiers. The results are obtained on a Windows PC with Core 2 Duo 2.0 GHz CPU and 2 GB RAM.

The performance of an IDS is measured by its ability to classify each connection in the right category. Table 2 well known as the confusion matrix shows the four possible cases. True Negative is the correct classification of Negative Class (Normal). False Negative is the wrong classification of Positive Class (Attack) as Negative Class (Normal). False Positive is the wrong classification of a Negative Class (Normal) as Positive Class (Attack). True Positive is the correct classification of Positive Class (Attack).

Table 2. Confusion Matrix

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

The most used performance metrics for the intrusion detection system are:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- $Detection\ Rate\ (DR\ ) = \frac{TP}{TP+FN}$
- $False\ Alarm\ Rate\ (FAR\ ) = \frac{FP}{TN+FP}$

5.1. Training and Test Data Set

The KDD99 Cup is derived in 1999 from DARPA-Lincoln98 [21] data set that is collected by MIT's Lincoln laboratory. KDD99 contains 39 attacks and normal behavior classified into five classes: DOS attack, U2R attack, PROB attack, R2L attack and Normal behavior [18]. Table 3 shows the classification of all attacks that exist in KDD99.

Table 3. Classification of KDD99 Attacks

Category of attack	Attack type		
	Exist in training and test data set	Exist only in Training data set	Exist only in the test data set
DOS	back, land, neptune, pod, smurf, teardrop		apache2, mailbomb, processtable, udpstorm
Probe	ipsweep, nmap, portsweep, satan		mscan, saint
R2L	ftp_write, guess_passwd, imap, multihop, phf, warezmaster	spy, warezclient	named, sendmail, snmpgetattack, snmpguess, worm, xlock, xsnoop
U2R	buffer_overflow, loadmodule, perl, rootkit		httptunnel, ps, sqlattack, xterm

Each record of KDD99 has 41 features where 34 are numeric and 7 are symbolic. The KDD99 Training Data Set [22] covers normal behavior and 22 attacks with 4,940,000 data records. The Test Data Set contains 311,029 data records. It covers normal behavior and 37 attacks, where 17 attacks don't exist in the Training Data Set. KDD99_10% represents 10% of KDD99 Training Data set with the same distribution of attacks and normal behavior. The following table 4 shows the distribution of attacks and normal behavior in the KDD99 training_10% and KDD99 test.

Table 4. Distribution of Attacks and Normal Behavior in KDD99 Training 10% and KDD99 Test

Number of records / Category of connection	KDD99_10% Training data set		KDD99 Test data set	
	All	Distinct	All	Distinct
Normal	97278	87832	60593	47913
DOS	391458	54572	229853	23568
Probe	4107	2130	4166	2678
R2L	1126	999	16189	2913
U2R	52	52	228	215
All	494021	145585	311029	77287

Due to the large-size of KDD99_10%, we created our training data set that contains 30,000 records. To reduce the size of KDD99_10% all redundancy records are removed. Then, the random selection is used to select Normal and DOS (Neptune) records. Table 5 summarizes the distribution of attack and normal behavior records of our training data set. The KDD99 test data set [22] is used to evaluate the performance of our models. Two features (num_outbound_cmds, is_host_login) are removed because of their identical values in the training data set. To normalize the data sets, the ASCII encoding method is used to convert the symbolic data to numerical values. Then, each data x_i of the feature j is normalized based on the following equation:

$$\overline{x_i(j)} = \frac{x_i(j) - min(x(j))}{max(x(j)) - min(x(j))}$$

Table 5. Distribution of Attacks and Normal Behavior of our Training Data Set

Connection type	Record number	Description	Proportion
Normal	8,000	8,000 distinct random selected Normal records extracted from Kdd99 10%	26.67%
DOS	18,819	All distinct Pod, Land, Back, Teardrop, Smurf records plus 16067 distinct and random selected from Neptune. All records extracted from Kdd99 10%.	62.73%
Probe	2,130	All distinct Probe records extracted from Kdd99 10%	7.10%
R2L	999	All distinct R2L records extracted from Kdd99 10%	3.33%
U2R	52	All distinct U2R records extracted from Kdd99 10%	0.17%

5.2. Comparative Study of Classifiers

In the aim to select the best classifier for the two levels of the new hierarchical IDS, we have performed two comparative studies. The first one is to select the different classifiers of the first level, which give a short training time and the best true positive rate for DOS, Probe, U2R, R2L and Normal behavior. The second one is to select the classifier of the second level that gives a short training time and the best true positive rate on the classification of connection in attacks and Normal behavior. The eight classifiers compared are as follows: Multilayer Perceptrons (MLP), Naïve Bayes (NB), C4.5 Decision Tree (DT), Support Vector Machine (SVM), Simple Cart (SC), Random Forest (RF), Best First Decision tree (BFTree), Repeated Incremental Pruning to Produce Error Reduction (RIPPER). In this comparative study, we have used the training data set detailed in above table 5 as training data set, and all KDD99 Data Test [22] as test data set.

5.1.1. Comparative Study between the Eight Classifiers Relative to the First Level

To compare the different classifier relative to the first level, we have performed a set of experiments, where each classifier has 39 inputs that represent the 41 features of KDD9 without num_outbound_cmds and is_host_login. Each classifier gives their predictions for the four categories of attacks (DOS, PROBE, U2L, R2L), and normal behavior. Table 6 summarizes the correct classification of the eight classifiers, and the time need to train these classifiers.

Table 6. Comparative Study between the Eight Classifiers Relative to the First Level

	DOS	Normal	Probe	R2L	U2R	Training Time
NB	90,9%	94,3%	89,6%	0,7%	21,9%	1,62
SVM	97,1%	98,5%	79,6%	9,8%	8,8%	61,62
MLP	97,5%	98,5%	76,9%	6,9%	6,6%	1074,77
BFTree	97,3%	77,9%	90,1%	62,3%	5,3%	28,79
DT	97,4%	87,2%	85,3%	8,5%	7,0%	12.32
RIPPER	98,1%	98,8%	83,1%	13,7%	15,8%	196,31
RF	97,5%	99,3%	84,1%	7,0%	11,4%	14,85
SC	97,4%	99,2%	90,4%	29,1%	8,8%	29,21

RIPPER and MLP are not selected because of their long training time. As illustrated in Figure 3, the best classifier on the detection of Normal behavior and DOS, Probe, R2L, U2R are respectively Random Forests, Simple Carte, Best First decision tree, Naïve Bayes.

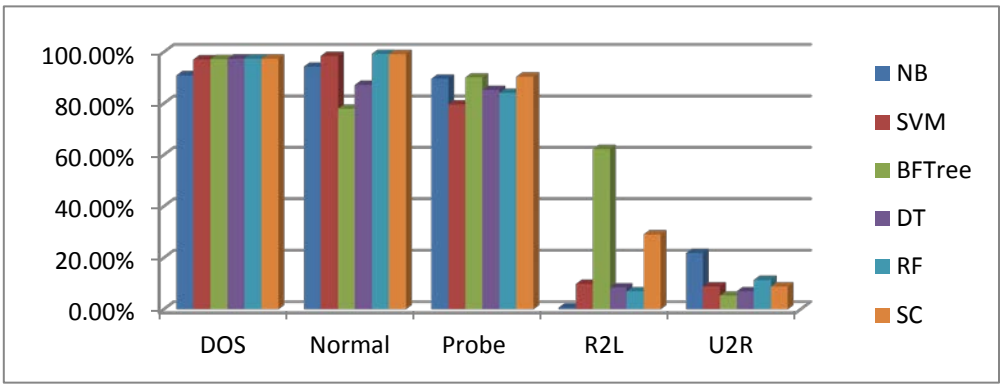

Figure 3. Performance of the Eight Classifiers Relative to the First Level

5.1.2. Comparative Study between the Eight Classifiers Relative to the Second Level

To select the best classifier for the second level, we have made a set of experiments, where each classifier has 5 inputs that represent the five selected outputs of the first level. Each classifier gives their prediction for the two classes attack and normal behavior. Table 7 summarizes the correct classification of the eight classifiers.

Table 7. Comparative Study between the Eight Classifiers Relative to the Second Level

	Attack	Normal	Overall	Training Time
NB	94,3%	98,7%	95,1%	0,22s
SVM	92,2%	99,2%	93,5%	1,28s
MLP	92,0%	99,1%	93,4%	70,49s
BFTree	92,1%	99,1%	93,5%	0,88s
DT	92,1%	99,1%	93,5%	0,3s
JRIP	92,1%	99,1%	93,5%	0,7s
RF	92,0%	99,1%	93,4%	2,41s
SC	92,1%	99,1%	93,5%	1,54s

As illustrated in Figure 4, the best classifier on the detection of Normal behavior and attacks is Naïve Bayes that gives the best performance and the shortest training time.

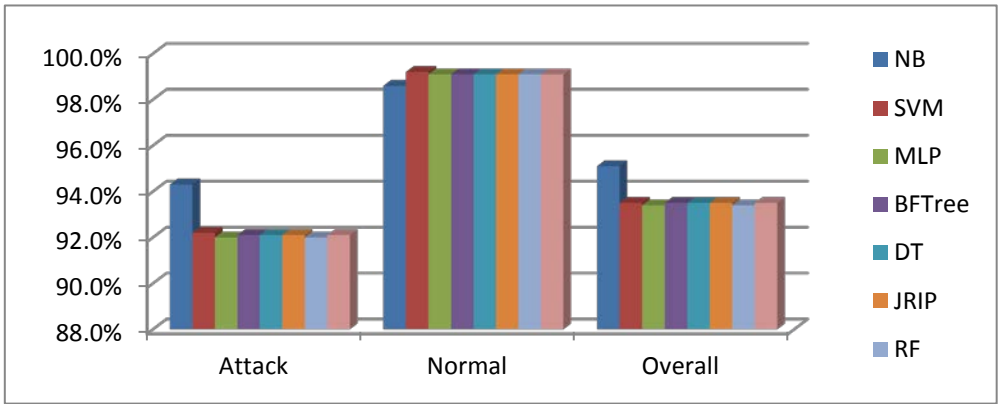

Figure 4. Classifiers Performance Relative to the Second Level

5.3. Evaluation of the New Hierarchical IDS

After analyzing the performance of the different types of classifiers, we exploited their strong points in order to achieve our goal. In the first level, we have selected Random Forest, Simple Cart, Best First Tree, Naïve Bayes for their highest true positive rate in the detection of respectively Normal behavior and DOS, Probe, R2L, and U2R. In the second level, we have selected Naïve that gives the best performance and the shortest training time. Figure 5 shows the practice structure of NFPHIDS.

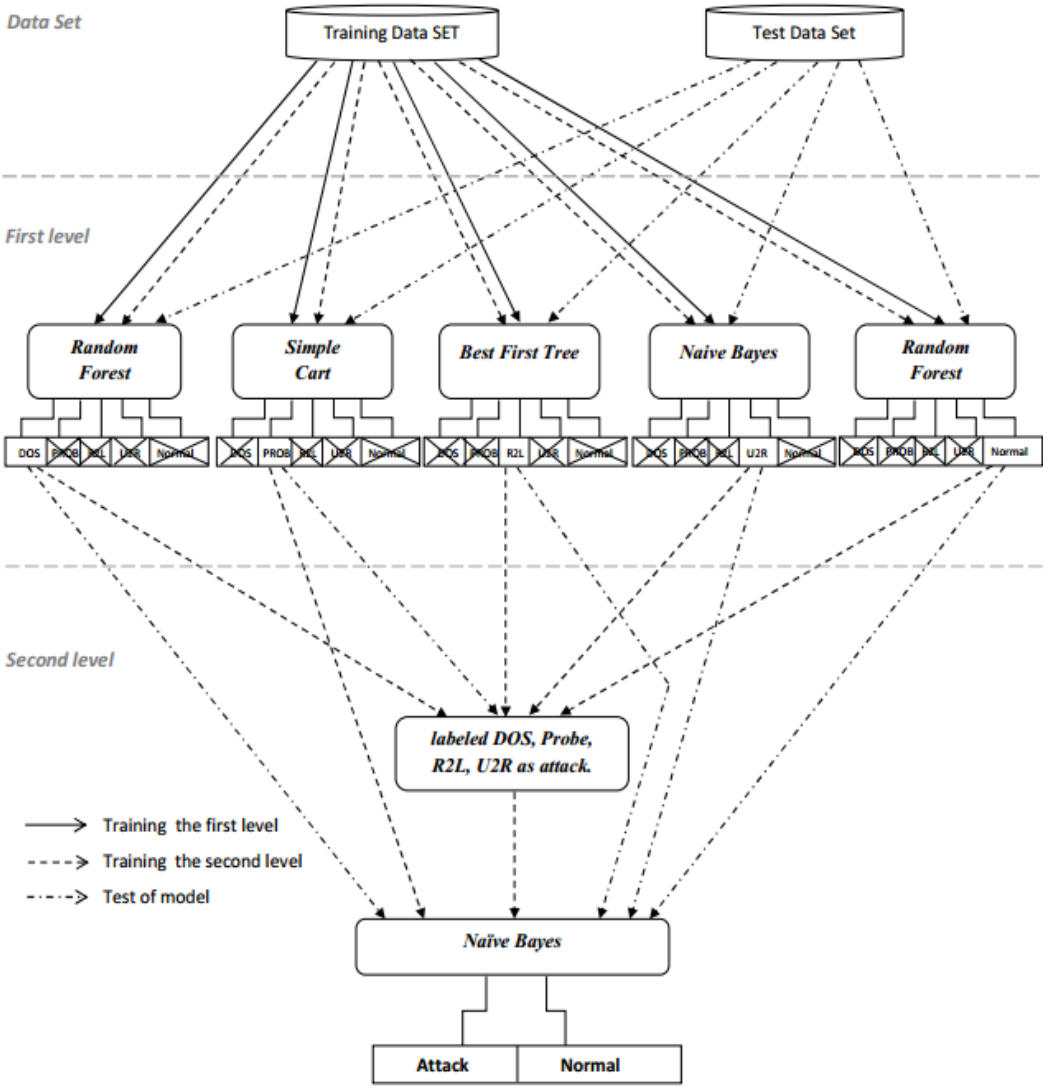

Figure 5. Practice Structure of the NFPHIDS

To evaluate the performance of NFPHIDS, we compared it with some well known classifiers such as: SVM, MLP neural network, RIPPER, C4.5 decision tree. In this comparison, we have used the training data set detailed in above Table 5 as a training data set and all KDD99 Test data set [22] as a test data set. The result of this comparison is shown in the following Table 8.

Table 8. Comparative Study between NFPHIDS and other Well Known Classifiers

		Our model		MLP NN		SeVM		RIPPER		DT	
		Attack	Normal	Attack	Normal	Attack	Normal	Attack	Normal	Attack	Normal
Normal		1,35%	98,65%	0,93%	99,07%	3,75%	96,25%	7,71%	92,29%	2,11%	97,89%
Attack	DOS	97,81%	2,19%	97,40%	2,60%	97,10%	2,90%	97,46%	2,54%	97,64%	2,36%
	PROB	98,13%	1,87%	88,24%	11,76%	84,40%	15,60%	88,36%	11,64%	98,42%	1,58%
	R2L	43,15%	56,85%	8,61%	91,39%	5,09%	94,91%	43,50%	56,50%	12,50%	87,50%
	U2R	72,81%	27,19%	65,35%	34,65%	56,14%	43,86%	78,51%	21,49%	85,96%	14,04%
	ALL	94,26%	5,74%	91,48%	8,52%	90,90%	9,10%	93,80%	6,20%	92,14%	7,86%
DR		94,26%		91,48%		90,90%		93,80%		92,14%	
FAR		1,35%		0,93%		3,75%		7,71%		2,11%	
Accuracy		95,12%		92,96%		91,95%		93,51%		93,26%	
Training Time		29,43		839.89		55.49		106.88		9.41	

As illustrated in figure 6, NFPHIDS gives the highest detection rate, highest Accuracy and second least false alarm rate. NFPHIDS shows its ability to better detect the low frequent attacks as U2R and R2L without losing their high true positive rate on the detection of normal behavior and other frequent attacks, which represents a great advantage. The improvement of the accuracy represents 4,992 records correctly classified more than the best of classifiers used in this comparative study. The training time of our model represents the second shortest training time. Moreover, the time needs to test all KDD99 Test data set is 15 seconds, which means that the time need to test one record is 48 Microseconds. The test time of our model is very short that represents another major advantage.

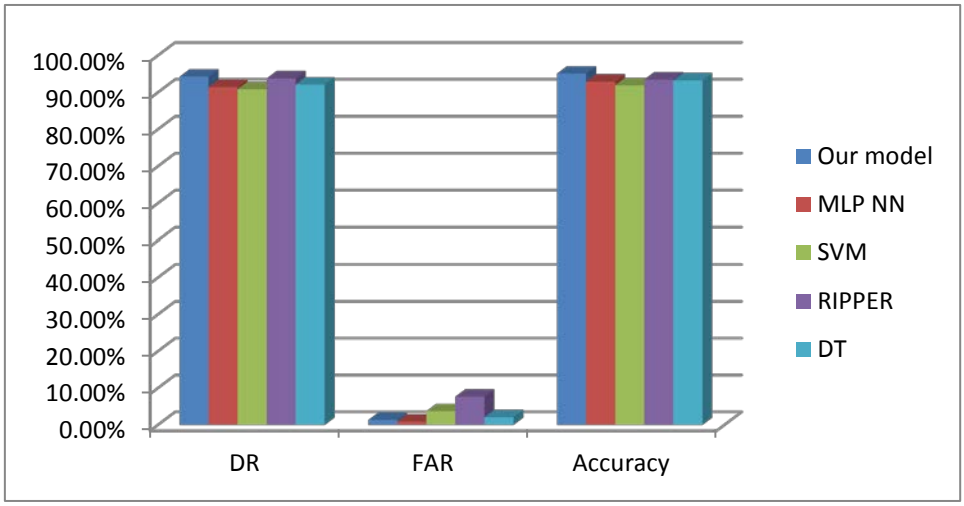

Figure 6. Comparative Study between NFPHIDS and other Well Known Classifiers

6. Conclusion

In this article, we have proposed a new fast and high performance hybrid hierarchical intrusion detection system called NFPHIDS. Our model is based on the combination of different fast classifiers, where we find two levels. The first one contains four fast classifiers

Random Forest, Simple Cart, Best first decision tree, Naive Bayes used for their excellent performance on the detection of respectively Normal behavior and DOS, Probe, R2L, U2R. The second level contains Naïve Bayes as final classifier, which represent a speed classifier, and gives a high performance. The experiments on KDD99 show that NFPHIDS is a very fast model and gives a good performance compared to other intrusion detection techniques, where it gives the highest detection rate, the highest accuracy. Moreover, NFPHIDS has shown their ability to better detect the low frequent attacks and keep their high performance on the detection of frequent attacks and normal behavior.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] E. Cole, R. Krutz and J. Conley, "Network Security Bible", Wiley Publishing, Inc, **(2005)**.
[2] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)", NIST Special Publication, Gaithersburg, **(2007)**, pp. 800-94.
[3] S. Axelsson, "Intrusion detection systems: A survey and taxonomy", Technical Report 99-15, Chalmers University, Goteborg, **(2000)**.
[4] H. Debar, M. Dacier and A. Wespi, "A Revised Taxonomy for Intrusion Detection Systems", Annals of Telecommunications, vol. 55, no. 7-8, **(2000)**, pp. 361-378.
[5] S. Wu. Xiaonan and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review", Applied Soft Computing, vol. 10, no. 1, **(2010)**, pp. 1-35.
[6] W. Chimphlee, A. H. Addullah, M. N. M. Sap, S. Srinoy and S. Chimphlee, "Anomaly-based intrusion detection using fuzzy rough clustering", Proceedings of the 2006 International Conference on Hybrid Information Technology, IEEE Computer Society Washington, vol. 01, **(2006)**, pp. 329-334.
[7] L. S. Scott, "A Bayesian paradigm for designing intrusion detection systems", Computational Statistics and Data Analysis, vol. 45, no. 1, **(2004)**, pp. 69-83.
[8] W. Fan, M. Miller, S. Stolfo, W. Lee and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions", Knowledge and Information Systems, vol. 6, no. 5, **(2004)**, pp. 507-527.
[9] S. Paek, Y. Oh and D. Lee, "sIDMG: Small-Size Intrusion Detection Model Generation of Complimenting Decision Tree Classification Algorithm", Proceedings of the 7th International Workshop, WISA 2006, Jeju Island, Korea, Springer Berlin Heidelberg, **(2006)** August 28-30, pp. 83-99.
[10] Z. Zhang and H. Shen, "Application of online-training SVMs for real-time intrusion detection with different considerations", Computer Communications, vol. 28, no. 12, **(2005)**, pp. 1428-1442.
[11] J. Cannady, "Artificial neural networks for misuse detection", in Proceedings of the 21st National Information Systems Security Conference, Arlington, VA, USA, **(1998)**, pp. 368-381.
[12] C. Zhang, J. Jiang and M. Kamel, "Intrusion detection using hierarchical neural networks", Pattern Recognition Letters, vol. 26, no. 6, **(2005)**, pp. 779-791.
[13] G. Liu, Z. Yi and S. Yang, "A hierarchical intrusion detection model based on the PCA neural network", Neurocomputing, vol. 70, no. 7-9, **(2007)**, pp. 1561-1568.
[14] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers", Computer Communications, vol. 30, no. 10, **(2007)**, pp. 2201-2212.
[15] X. Tong, Z. Wang and H. Yu, "A research using hybrid RBF/Elman neural networks for intrusion detection system secure model", Computer Physics Communications, vol. 180, no. 10, **(2009)**, pp. 1795-1801.
[16] G. Wanga, J. Hao, J. Mab and L. Huanga, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering", Expert Systems with Applications, vol. 37, no. 9, **(2010)**, pp. 6225-6232.
[17] S. Horng, M. Su, Y. Chen, T. Kao, R. Chen, J. Lai and C. D, Perkasa, "A novel intrusion detection system based on hierarchical clustering and support vector machines", Expert Systems with Applications, vol. 38, no. 1, **(2011)**, pp. 306-313.
[18] S. Chaudhuri, D. Madigan and U. Fayyad, "KDD-99", Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining, ACM SIGKDD Explorations Newsletter, vol. 1, no. 2, **(2000)** January 2, pp. 49-51.
[19] C. Tsaia, Y. Hsub, C. Linc and W. Lin, "Intrusion detection by machine learning: A review", Expert Systems with Applications, vol. 36, no. 10, **(2009)**, pp. 11994-12000.

[20] I. Witten, E. Frank and M. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Elsevier Inc, **(2011)**.
[21] The DARPA Intrusion Detection Data Sets, available at: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/index.html (accessed June 2012), **(1998)**.
[22] The KDD CUP 1999 Data, available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed June 2012), **(1999)**.

Authors

Ahmed AHMIM is a PhD student in Badji Mokhtar University, Annaba, Algeria. He is a member of the Laboratory of Computer Networks and Systems, His research interests include computer security, intrusion detection system, neural networks, genetic algorithms, data mining. Ahmed Ahmim is the corresponding author and can be contacted at: ahmed.ahmim@lrs-annaba.net.

Nacira Ghoualmi-Zine is a Professor in Computer Sciences and has been a lecturer in the Department of Computer Science at Badji Mokhtar University, Annaba, Algeria since 1985. She is head of the Master and Doctoral option entitled Network and Computer Security, and head of a Laboratory of Networks and Systems. Her research includes cryptography, computer security, intrusion detection system, wireless networks, distributed multimedia applications, quality of service, security in the protocol, and optimization in networks.