Journal of Applied Logic 24 (2017) 109–118

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Comparison of classification techniques applied for network intrusion detection and classification

Amira Sayed A. Aziz [a,c,*], Sanaa EL-Ola Hanafi [b], Aboul Ella Hassanien [b,c]

[a] *Future University in Egypt (FUE), Cairo, Egypt*
[b] *Faculty of Computers and Information, Cairo University, Egypt*
[c] *Scientific Research Group in Egypt (SRGE), Egypt*

ARTICLE INFO

Article history:
Available online 14 November 2016

Keywords:
Intrusion Detection
Artificial Immune Systems
Intrusion Classification
Machine Learning

ABSTRACT

In a previous research, a multi-agent artificial immune system for network intrusion detection and classification was proposed and tested, where a multi-layer detection and classification process was executed on each agent, for each host in the network. In this paper, we show the experiments that were held to chose the appropriate classifiers by testing different classifiers and comparing them to increase the detection accuracy and obtain more information on the detected anomalies. It will be shown that no single classifier should be used for all types of attacks, due to different classification rates obtained. This is due to attacks representations in the train set and dependency between features used to detect them. It will also be shown that a basic and simple classifier such as Naive Bayes has better classification results in the case of low-represented attacks, and the basic decision trees such as Naive-Bayes Tree and Best-First Tree give very good results compared to well-known J48 (Weka implementation of C4.5) and Random Forest decision trees. Based on these experiments and their results, Naive Bayes and Best-First tree classifiers were selected to classify the anomaly-detected traffic. It was shown that in the detection phase, 90% of anomalies were detected, and in the classification phase, 88% of false positives were successfully labeled as normal traffic connections, and 79% of DoS and Probe attacks were labeled correctly, mostly by NB, NBTree, and BFTree classifiers.

1. Introduction

Computer systems today are usually interconnected, where they are networked into large distributed systems which are essential in industrial computing world. Today's software systems require more trends such as interconnection, intelligence, and ubiquity. This all led to the arise of multi-agent systems. The multi-agent systems mimic human intelligent behavior, and the way humans interact with each other and towards their environment to achieve certain goals. One of the systems that can be implemented successfully and efficiently as a multi-agent system is Artificial Immune Systems (AIS). Artificial Immune System [1,11]

* Corresponding author.
E-mail address: amira.aly.fci@gmail.com (A.S. A. Aziz).

http://dx.doi.org/10.1016/j.jal.2016.11.018
1570-8683/

is a research area that involves immunology, computer science, and engineering. Inspired by natural immune systems, three main fields of research lie under AIS: immune modeling, theoretical AISs, and applied AISs. Immune modeling is concerned about immunity models and immune systems simulations. Theoretical AISs research is about explaining and digging into the theoretical aspects of AIS algorithms, their mathematical models, and their performance and complexity analysis. Finally, applied AISs research is about developing and implementing algorithms and computers systems inspired by immune systems, applying them to a variety of real world applications.

Looking into computer systems, we can find that the self/non-self concept applies in the form of normal/anomalous activities and elements. Intrusion Detection Systems (IDS) are powerful security systems that have a variety of types for all protection purposes. They do not replace a certain security tool, but instead they add a defense line against intrusions and threats from inside the system (a computer or a network) as well as from outside – just like and immune system. IDSs can be categorized in different ways, but basically they can be classified into Misuse-based and Anomaly-based IDSs. Misuse IDS, which is also known as signature-based or knowledge-based, depends on detecting intrusions using patterns representing known attacks. These patterns or signatures are compared to captured events to find possible intrusions. Anomaly IDS, also known as behavior-based, depends on building a profile that represents normal behavior of a system by monitoring its activities over time. Then any deviation from that profile is considered an anomaly. Profiles can be static or dynamic, and are developed using many attributes of the system [13,17].

With the diversity and complexity of attacking techniques, many issues are common with IDSs that need to be considered while building an IDS [17]. The most common issue is deriving an expert rule set, which in most cases is updated through a human expert, especially with misuse IDS. With anomaly detection, such an issue is solved using proper representation and definition of attacks such that different types of attacks can be detected using a limited set of rules with variations of these rules. Another issue is the training of behavioral models, where usually normal data only is used for training or two sets of normal and anomalous data. Machine learning techniques are mostly used for training and learning.

In an attempt to overcome the mentioned issues above, we suggested a multi-agent, two-layer classification algorithm, that detects and classifies anomalies in a network. The suggested system IDS combines Genetic Algorithm with Negative Selection Approach as a first layer of anomalies detection. Then selected classifiers are trained and applied to label the detected anomalies in both the normal and anomalous traffic. The immune system is a distributed system composed of different specialized cells with high interactivity between its components to give a coordinated response. Taking this into consideration, many approaches adopted the implementation of an AIS as a multi-agent system.

In previous researches [14,16,19,22] that are similar some way to the suggested system, either specific classifiers are used for each attack type, or classifiers – which were trained using the labeled data – are applied directly to the data set for intrusion detection and labeling unlabeled attacks. The contribution of this paper is to do some comparative analysis to answer questions about how the system will act to different parameters, and which techniques to use for best results and why. It will also investigate whether it is rewarding to feed normal traffic into the classifiers or not.

In a previously published work [8], a multi-agent artificial immune system was implemented, for network intrusion detection and classification. The algorithm applied as an artificial immune system technique is Negative Selection Approach, using Genetic Algorithm. As an intelligent system, data mining is applied throughout the process for best results. Two classifiers were used for anomalies classification, Naive Bayes and Best-First Tree classifiers. Naive Bayes classifier was used for attacks that have low representation in the training data set as it has proven to give better results than other classifiers in a previous experiment. The BFTree classifier was used for the remaining attacks classification, as it also proved to give better results than other more complex classifiers in the experiments shown in this paper.

The paper is organized as follows. Section 2 presents a background about the basics of the classification techniques. In section 3, the proposed approach and system model is explained with its different components

and phases. Section 4 includes the experiment details and settings. The results are presented and discussed in section 5. Finally, conclusion is given with future work in section 6.

2. Data classification techniques

Classification is the process of assigning a class label to unclassified object(s) based on a set of defined features. A classifier should first get that knowledge by learning the representation of classes using a given set of pre-classified samples. A classifier would act as a predictor for unclassified objects or a descriptor for classified objects. Many approaches exist such as Decision Trees, Rule-based Approaches, Bayesian Classifiers, Neural Networks, Genetic Classifiers, Support Vector Machines, and many others. A classifier is evaluated by its prediction accuracy, speed, robustness, scalability, interpretability, simplicity, and domain-dependent quality indicators [12,23].

2.1. Naive Bayes classifier

It applies the Bayes Theorem with strong independence assumptions as a simple probabilistic classifier. It assumes the presence or absence of a feature is independent from the presence or absence of other features – features are unrelated. It is simple, optimal, and straightforward to apply. NB classifier can be used if we have some knowledge and training data, and we need to estimate probabilities from limited data. Its main advantage is that it does not take long time for training [2,9,10].

2.2. Decision trees

Decision Trees are structured representation of a dataset. A node marks a decision to make from a number of alternatives, and each terminal node indicates a certain classification. DTs are very powerful tools because they are fast and give reasonable performance. The DTs applied in this research are BFTree, NBTree, J48, and RFT. BFTree (Best-First Tree) expands its nodes in best-first order, unlike standard DTs that expand in depth-first order. The best node that is always expanded first is the one that leads to maximum reduction of impurity. NBTree (Naive-Bayes Tree) uses a mapping function to index high-dimensional data, and it should be a light and low computational function. So, those values can be ordered and used later in the resulting one-dimensional structure. J48 (Weka representation of C4.5 DT) expand its nodes in depth-first order. It is a supervised learning algorithm, where it learns the mapping from attribute values to classes, and then this mapping applied to new unknown samples. RFT (Random Forrest Trees) involve building a set of classification trees which are calculated on random subsets of data. This is done using randomly selected predictors for each split in each tree [2,9].

2.3. Multi-layer perceptrons

Multi-layer Perceptrons (MLP) are used if the provided instances can not be separated linearly – input instances can not be mapped to their correct categories linearly. MLPs are feed-forward neural networks that consist of a large number of connected neurons which are classified into: input units, output units, and hidden units in between. The weights assigned to connections are estimated using the Back Propagation (BP) algorithm. The weights values define the performance of the neural network [20].

3. The two-layer detection and classification system

The previously proposed multi-layer system is composed of two layers, preceded by a pre-processing phase. The first layer – detection layer – applies the Negative Selection Approach using Genetic Algorithm

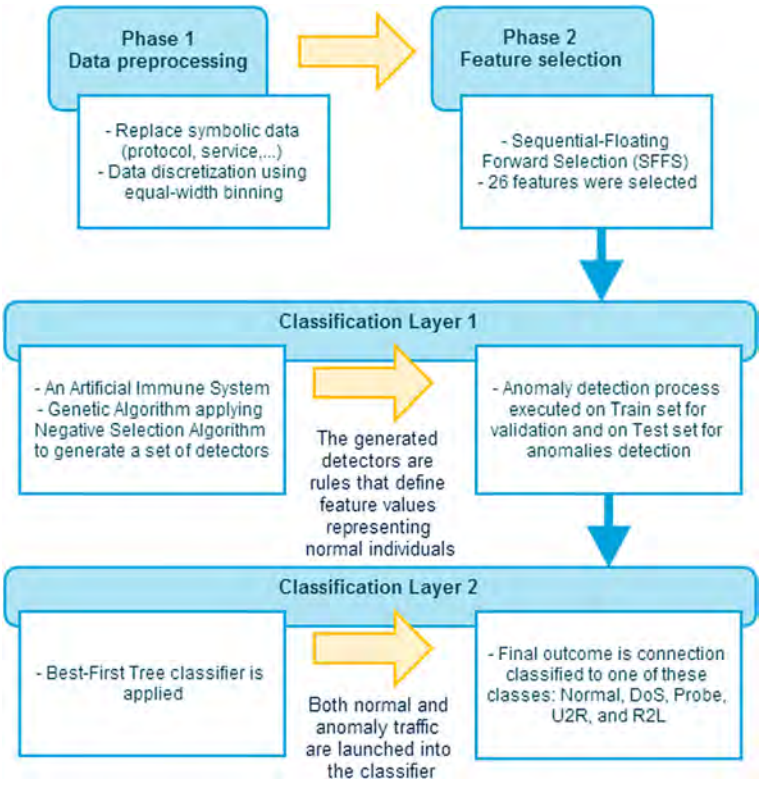


Fig. 1. Two-layer anomalies detection and classification model.

for anomaly intrusion detection, where detectors are trained to identify normal activity so that they would detect any differences (over a certain threshold) as an anomaly. The second layer – classification layer – uses a classifier to label the traffic with its proper class to define attack classes and minimize false alarms. Both normal and anomalous traffic are fed into the classifier. The process shown in Fig. 1.

3.1. Data preprocessing phase

Data preprocessing involves replacing symbolic data – such as protocol and service – with discrete/integral values, and discretization of values to homogeneous and in a limited range. Equal Width Binning algorithm was used, and the original values were replaced with bin numbers corresponding to their range. More details can be found in a previous publication [3].

3.2. Feature selection phase

Sequential Floating Forward Selection (SFFS) technique was applied and 26 features were selected, as stated in a previous publication of our work [4], which was concerned to find the feature selection algorithm with the best results. Features selected by SFFS gave the best accuracy results.

3.3. Layer I: detectors generation and anomaly detection

This is the GADG (Genetic Algorithm for Detectors Generation) – also applied in [4] – to generate the detectors for the anomaly intrusion detection process. In this process, anomaly detectors are generated using the GA applying the NSA concept – generating detectors that are familiar with the normal connections representation, hence they are able to discriminate between normal and anomalous. The generated detectors

Table 1
Distributions of NSL-KDD records.

	Total records	Normal	DoS	Probe	U2R	R2L
Train_20%	25192	13449 53.39%	9234 36.65%	2289 9.09%	11 0.04%	209 0.83%
Train_All	125973	67343 53.46%	45927 36.456%	11656 9.25%	52 0.04%	995 0.79%
Test+	22544	9711 43.08%	7458 33.08%	2421 10.74%	200 0.89%	2754 12.22%

or rules are basically values of the selected features that properly represent normal connections. So, initially the population used to generate these detectors are filled with randomly selected from the normal connections in the data set. Two distance measurements were applied separately – the Euclidean and Minkowski – and detectors group generated by each distance measure are tested and compared.

At the end of the detectors generation process, mature detectors are released to the system to start the discrimination process of self and nonself. Detectors are composed of values for features selected to represent self elements. The detectors generated from the previous step are run through the Test data set to start the anomaly detection process. The results obtained in [4] were fed in the next layer – classification layer – to classify the detected anomalies, as explained in details below.

3.4. Layer II: applying classifiers for attacks classification

After the anomaly detection phase, we have connections tagged as either normal or anomalous, no attack type specified. A classifier should be used to label attacks with their specific class. Many classifiers were tested and compared. Those classifiers are: Naive Bayes, Decision Trees – NBTree, BFTree, J48, and RFT – and Multi-Layer Perceptron neural network. In the classification layer, both the normal and anomalous traffic go through classification – separately. For the anomalous traffic, the anomalies should be labeled as one of the attack classes existing in the data set, or as normal if it was a false positive. For the normal traffic, it goes through the classifier so that if an anomaly was detected as a false negative, it was be correctly labeled as an attack, with the right class label.

The proposed model was implemented as a multi-agent system in [8].

4. Experiments

The experiment was executed using the NSL-KDD [21] IDS evaluation data set. In older researches, the KDD Cup 99 [15] data set was the most used benchmark data set for performance evaluation for network-based intrusion detection systems. It was found that it has some problems that cause the learning algorithm to be biased and the results to be inaccurate due to duplications of its records in both the training and testing data sets. The KDD Cup data set is also very large to be used for evaluation, so usually small portions of the data are used for evaluation, which would cause inaccurate results as well. So, the researchers group removed the redundant records, and sampled the data set proportionally to make it balanced, and this resulted in the NSL-KDD data set. It has been used by many researchers, as it contains much less number of records for both train and test data, so the whole sets can be used in the experiments. There are four general types of attacks in the data set: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). Table 1 shows the distributions of normal and attacks records in the NSL-KDD data set.

Some of the steps for the preparation of the process needed some tuning for their settings in the main agent, in order to give the best results in the detector agents. These values gave the best results with the

features selected by SFFS in a previous experiment illustrated in [5]. In the experiment, 26 features were selected by SFFS. For the classifiers, the Train_20percent data was used for the training, as the classifiers proved to give very good results without having to use the whole train data records, as proved in [6,7]. The classifiers applied in these experiments were used through the Weka tools [24].

5. Results and discussion

We have two result sets from the first phase, one set obtained with the detectors generated using the Euclidean distance measures, and the other set obtained by the detectors generated using the Minkowski distance measure – both for the Train Set and the Test Set. Only the Test Set results are run through the classifiers.

For the Euclidean results, 12889 were detected as anomalies and their actual classes are: 1567 normal connections, 6894 DoS attack connections, 2295 Probe attack connections, 159 U2R attack connections, and 1974 R2L attack connections. On the other hand, 9655 were detected as normal and their actual classes are: 8144 normal connections, 564 DoS attack connections, 126 Probe attack connections, 41 U2R attack connections, and 780 R2L attack connections.

For the Minkowski results, 13239 were detected as anomalies and their actual classes are: 1718 normal connections, 6939 DoS attack connections, 2303 Probe attack connections, 159 U2R attack connections, and 2120 R2L attack connections. On the other hand, 9305 were detected as normal and their actual classes are: 7993 normal connections, 519 DoS attack connections, 118 Probe attack connections, 41 U2R attack connections, and 634 R2L attack connections.

Since the Minkowski detectors gave better detection rates, the experiment was continued using these detection outcomes. Evaluation metrics used to better evaluate the IDS are Precision, Recall, and F-Score [18]. Higher precision means that the system was able to detect only attacks, higher recall indicates that the system was able to detect all attacks, and higher F-Score indicates higher accuracy of detecting attacks – based on precision and recall values. They are calculated as following.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

Where TP (True Positives) is the number of correctly detected/classified attacks, FP (False Positives) is the number of normal connections falsely labeled as attacks, and FN (False Negatives) is the number of anomalous connections falsely labeled as normal.

5.1. Classifiers results

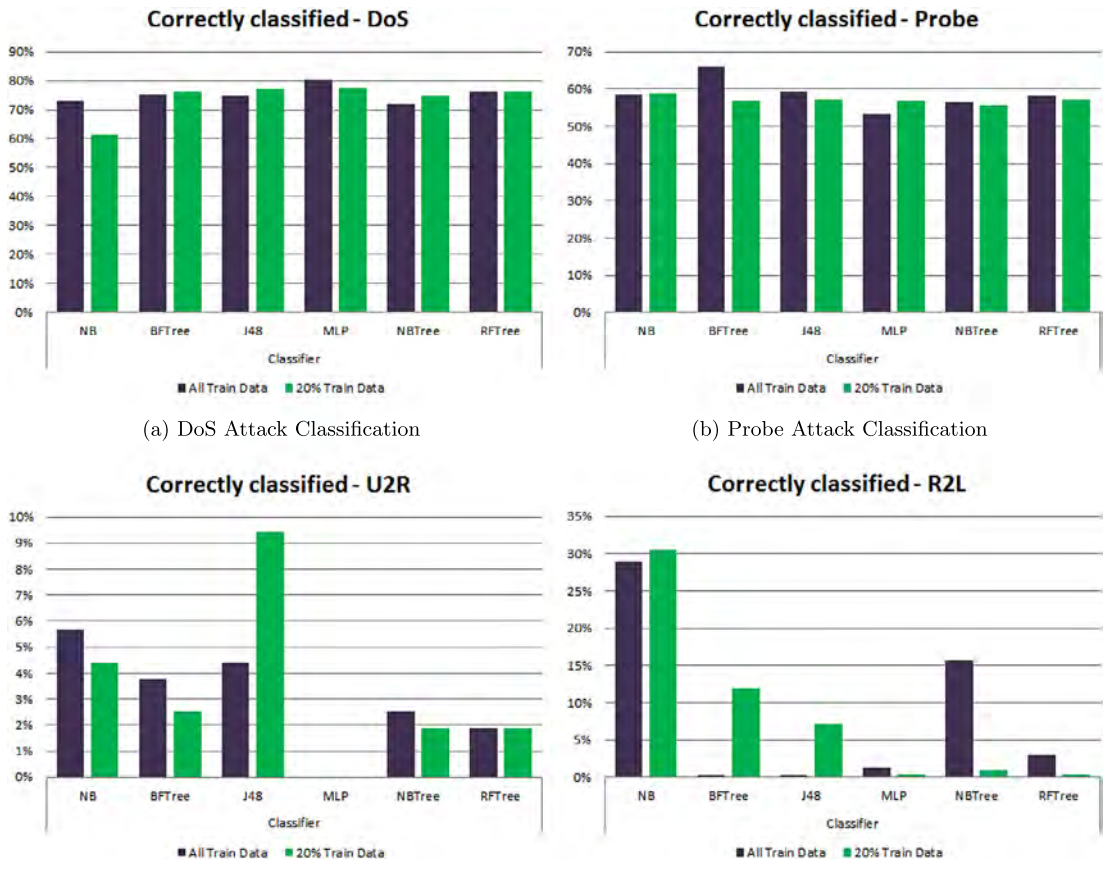
The classifiers were trained once using only 20% of the train set and another time using all the train set. Table 2 shows the precision and recall values for each classifier, on the anomalous traffic obtained by the Minkowski detectors.

We can realize that precision values are high, which indicates that most detected attacks are actually anomalous connections, while recall values are not that high, due to large number of attacks detected as normal connections. Fig. 2 shows the details of the classification results of the anomalous traffic for Minkowski detectors, for each attack class.

As the results show, DoS is best classified with MLP. Probe attacks are best classified by BFTree with all train data, and NB with 20% of the train data. NB is the best classifier for R2L attacks, and for U2R

Table 2
Precision and Recall values obtained by the classification of anomalous traffic detected by Minkowski detectors.

Classifier	TP	FP	FN	Precision	Recall	F-Score
Using all of the train data records						
NB	7027	1264	2534	84.75%	73.50%	78.73%
BFTree	6744	121	4454	98.24%	60.23%	74.67%
J48	6580	156	4493	97.68%	59.42%	73.90%
MLP	6842	102	4322	98.53%	61.29%	75.57%
NBTree	6627	105	4309	98.44%	60.60%	75.02%
RFT	6690	113	4663	98.34%	58.93%	73.69%
Using 20% of the train data records						
NB	6271	1297	2677	82.86%	70.08%	75.94%
BFTree	6874	136	3954	98.06%	63.48%	77.07%
J48	6822	105	4351	98.48%	61.06%	75.38%
MLP	6691	123	4260	98.19%	61.10%	75.33%
NBTree	6506	114	4392	98.28%	59.70%	74.28%
RFT	6608	100	4730	98.51%	58.28%	73.24%



(a) DoS Attack Classification

(b) Probe Attack Classification

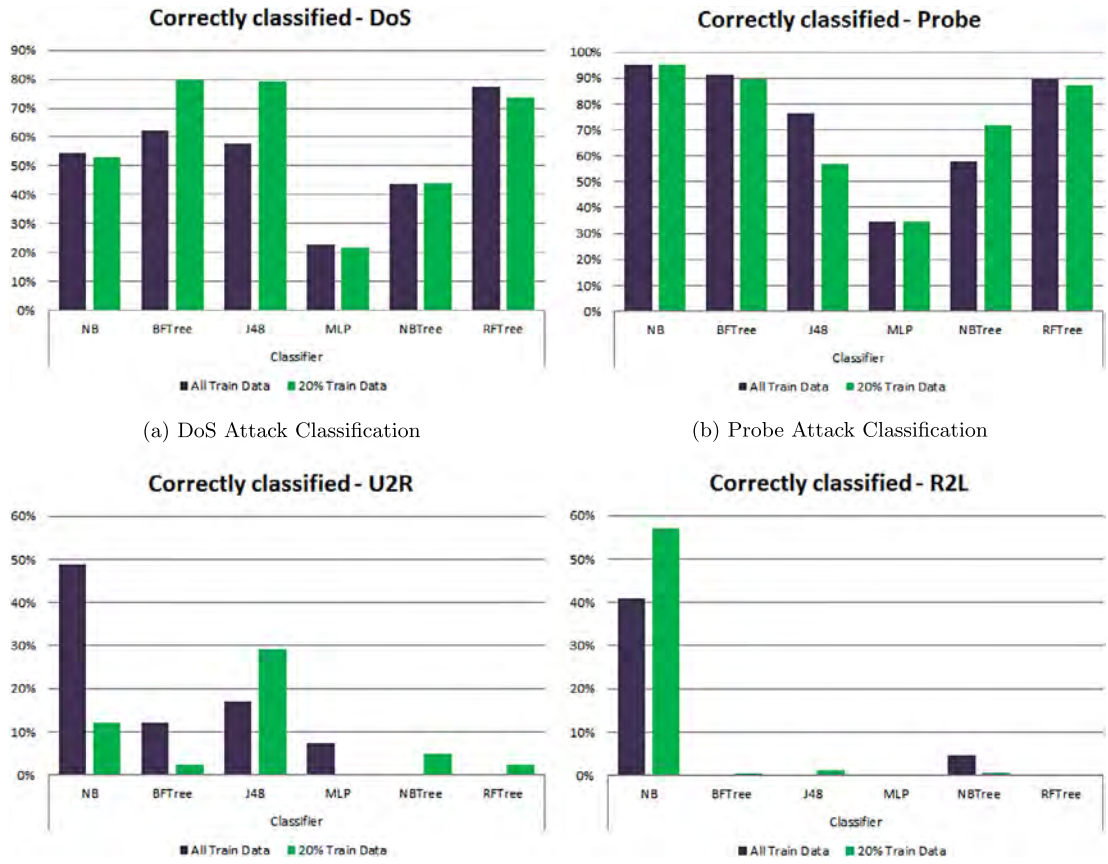
(c) U2R Attack Classification

(d) R2L Attack Classification

Fig. 2. Classification results in anomalous traffic.

attacks using all train data with along with the J48 using 20% of train data. Fig. 3 shows the classification results of the normal traffic, that's where normal traffic goes under classification to find false negatives and label them with their correct attack class.

We can realize through the results that for attacks that were wrongfully detected as normal, they were classified as follows: DoS are best classified with RFTree using all train data, and J48 and BFTree using 20% of train data, Probe attacks are best classified by NB then by BFtree and RFTree, U2R attacks are



(a) DoS Attack Classification

(b) Probe Attack Classification

(c) U2R Attack Classification

(d) R2L Attack Classification

Fig. 3. Classification results in normal traffic.

Table 3
Precision and Recall values obtained by the classification of anomalous and normal traffic detected by Minkowski detectors.

Classifier	TP	FP	FN	Precision	Recall	F-Score
Using all of the train data records						
NB	7702	1264	1859	85.90%	80.56%	83.14%
BFTree	7180	121	4018	98.34%	64.12%	77.63%
J48	6976	156	4097	97.81%	63.00%	76.64%
MLP	7004	102	4160	98.56%	62.74%	76.67%
NBTree	6950	105	3986	98.51%	63.55%	77.26%
RFT	7199	113	4154	98.45%	63.41%	77.14%
Using 20% of the train data records						
NB	7025	1297	1923	84.41%	78.51%	81.35%
BFTree	7398	136	3430	98.19%	68.32%	80.58%
J48	7320	105	3853	98.59%	65.52%	78.72%
MLP	6846	123	4105	98.24%	62.51%	76.41%
NBTree	6826	114	4072	98.36%	62.64%	76.53%
RFT	7095	100	4243	98.61%	62.58%	76.57%

best classified by NB using all train data and J48 using 20% of train data, and R2L attacks best classified by NB too.

Adding up the attacks classified correctly from the normal-detected traffic to the original TP detected in the anomalous traffic (FN turned into TP), the precision and recall values will be as following – Table 3.

NBtree and BFTree classifiers have the highest precision values, while NB and BFTree classifiers have the highest recall values. Which means that NB and BFTree were able to detect more attacks but NBTree

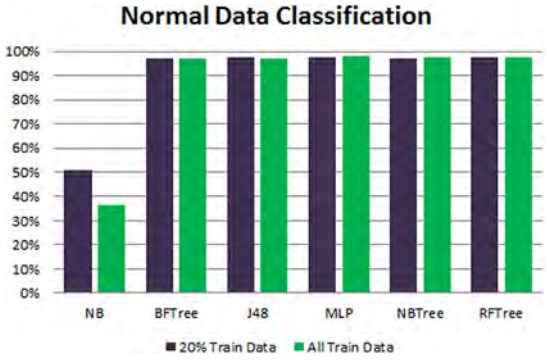


Fig. 4. The overall classification results – Minkowski detectors.

and BFTree have less number of false positives and most of what they detected are really anomalous. Hence, we see that the highest F-Score values belong to NB classifier then to BFTree classifier. This is due to the ability of the NB classifier to detect R2L and U2R attacks that were almost not detected by other classifiers, in both anomalous and normal traffic. The NB classifier was also able to label more Probe attacks that were falsely labeled as normal than other classifiers. Based on how much of the data used for training, NB and NBTree classifiers had better results when all of the training data set was used, while BFTree and J48 had better results when only 20% of the training data were used. RFTree and MLP results had close results in both cases.

Normal Data was highly detected and classified (almost all normal traffic) by all classifiers except NB classifier. For the anomalous connections some were highly detected and some were not, dependent on their representation in the dataset and the dependency between features.

6. Conclusion

In this paper, a hybrid and hierarchical intrusion detection system that is inspired by immunity concepts was presented. Different classifiers were tested and compared in order to find the best classifier to give more information of known attacks and predict the class of unknown attacks. We can realize that in general, decision trees give the best results, but looking into each class, the classifiers have different results based on representation of these classes in the data set and the dependency between features. The NB classifier have the best results concerning low presented attacks such as R2L and U2R. FT gave very good results but they take a long time to train to build the random trees for classification decision. NBTree and BFTree gave higher scores than J48 and RFTree; they are better to use as they take less time to train, and in cases of high representation of classes only 20% of the train set is enough to give very good results. MLP gave best scores in Normal And DoS classification for their high representation in the train set, but in cases of U2R and R2L attacks it failed to classify them – 0% were correctly classified in most cases.

We can conclude that we don't have to use complex classifiers to get high classification scores – NB gave high detection rates in the cases of low-represented attack classes. Also, a single classifier is not enough to use to label and predict the attacks classes, multiple classifiers should be involved to increase the classification accuracy. If 20% of the train data is enough to give high scores – even if slightly less than the case of using all train data – then having less train time in this case is better. The same feature set was used through the whole process and resulted in satisfactory results, so no need for a feature set specific for each attack class. Maybe this would be needed for the classification of specific attack type under the big label of the attack class, which would be the future work to enhance results of the system. The classification of traffic detected as normal enhanced the overall classification results, as shown in Fig. 4. This could be considered for future research, to detect behavioral attacks specifically in the traffic detected as normal.

References

[1] U. Aickelin, J. Greensmith, J. Twycross, Immune system approaches to intrusion detection – a review, in: Artificial Immune Systems, Springer, Berlin, Heidelberg, 2004, pp. 316–329.

[2] J.R. Anderson, Knowledge compilation: the general learning mechanism, in: R.S. Michalski, R.S. Michalski, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach 2, Morgan Kaufmann, 1986.

[3] A.S.A. Aziz, A.T. Azar, A.E. Hassanien, S.E.O. Hanafi, Continuous features discretizaion for anomaly intrusion detectors generation, in: WSC17 2012 Online Conference on Soft Computing in Industrial Applications, 2012.

[4] A.S.A. Aziz, A.T. Azar, A.E. Hassanien, S.E.O. Hanafy, Genetic algorithm with different feature selection techniques for anomaly detectors generation, in: 2013 Federated Conference on Computer Science and Information Systems, FedCSIS, IEEE, 2013, pp. 769–774.

[5] A.S.A. Aziz, A.T. Azar, A.E. Hassanien, S.E.O. Hanafy, Genetic algorithm with different feature selection techniques for anomaly detectors generation, in: 2013 Federated Conference on Computer Science and Information Systems, FedCSIS, IEEE, 2013, pp. 769–774.

[6] A.S.A. Aziz, A.E. Hassanien, A.T. Azar, S.E.O. Hanafi, Machine learning techniques for anomalies detection and classification, in: Advances in Security of Information and Communication Networks, Springer, Berlin, Heidelberg, 2013, pp. 219–229.

[7] A.S.A. Aziz, A.E. Hassanien, S.E.O. Hanafy, M.F. Tolba, Multi-layer hybrid machine learning techniques for anomalies detection and classification approach, in: 2013 13th International Conference on Hybrid Intelligent Systems, HIS, IEEE, 2013, pp. 216–221.

[8] A.S.A. Aziz, S.E.O. Hanafi, A.E. Hassanien, Multi-agent artificial immune system for network intrusion detection and classification, in: International Joint Conference SOCO'14–CISIS'14–ICEUTE'14, in: Advances in Intelligent Systems and Computing, vol. 299, Springer International Publishing, 2014, pp. 145–154.

[9] R. Caruana, A. Niculescu-mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 161–168.

[10] H. Chauhan, V. Kumar, S. Pundir, E.S. Pilli, A comparative study of classification techniques for intrusion detection, in: 2013 International Symposium on Computational and Business Intelligence, ISCBI, IEEE, 2013, pp. 40–43.

[11] D. Dasgupta, S. Yu, F. Nino, Advances in artificial immune systems: models and applications, Appl. Soft Comput. 11 (2) (2011) 1574–1587.

[12] S. Ganapathy, K. Kulothungan, S. Muthurajkumar, M. Vijayalakshmi, P. Yogesh, A. Kannan, Intelligent feature selection and classification techniques for intrusion detection in networks: a survey, EURASIP J. Wirel. Commun. Netw. 2013 (1) (2013) 1–16.

[13] P. Garcia-Teodora, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, Anomaly-based network intrusion detection: techniques, systems and challenges, Comput. Secur. 28 (1–2) (2009) 18–28.

[14] M.S. Hoque, Md. Mukit, Md. Bikas, A. Naser, An implementation of intrusion detection system using genetic algorithm, In arXiv preprint arXiv:1204.1336, 2012.

[15] KDD Cup'99 intrusion detection data set, Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, October 2007.

[16] G. Kim, S. Lee, S. Kim, A novel hybrid intrusion detection method integrating anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) (2014) 1690–1700.

[17] H.J. Liao, K.Y. Tung, C.H. Richard Lin, C.Y. Lin, Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (2012) 16–24.

[18] S. Misra, S.C. Misra, I. Woungang, Selected Topics in Communication Networks and Distributed Systems, World Scientific, 2010.

[19] Z. Muda, W. Yassin, M.N. Sulaiman, N.I. Udzir, Intrusion detection based on K-means clustering and Naïve Bayes classification, in: 2011 7th International Conference on Information Technology in Asia, CITA 11, IEEE, 2011, pp. 1–6.

[20] L. Noriega, Multilayer Perceptron Tutorial, School of Computing, Staffordshire University, 2005.

[21] NSL-KDD intrusion detection data set, Available on: http://iscx.ca/NSL-KDD/, March 2009.

[22] S.K. Sharma, P. Pandey, S.K. Tiwari, M.S. Sisodia, An improved network intrusion detection technique based on k-means clustering via Naïve Bayes classification, in: 2012 International Conference on Advances in Engineering, Science and Management, ICAESM, IEEE, 2012, pp. 417–422.

[23] C. So-In, N. Mongkonchai, P. Aimtongkham, N. Wijitsopon, K. Rujirakul, An evaluation of data mining classification models for network intrusion detection, in: 2014 Fourth International Conference on Digital Information and Communication Technology and Its Applications, DICTAP, IEEE, 2014, pp. 90–94.

[24] Weka 3: data mining software in Java, Available on http://www.cs.waikato.ac.nz/~ml/weka/.