World Applied Sciences Journal 9 (4): 393-397, 2010
ISSN 1818-4952
© IDOSI Publications, 2010

Upgrading Intrusion Detection Systems Through Hybrid Algorithms*

Mohammad Ghasemzadeh and Mehdi Sarram

Computer Engineering Department, University of Yazd, P.O. Box 89195-741, Yazd, Iran

Abstract: Two types of algorithms are realized which have been utilized within the supervised of model of intrusion detection systems. These algorithms are either of type eager or lazy as far as their performance is concerned. At the leaning phase, the lazy algorithms are fairly simple however the eager algorithms are highly effective. The classification phase on the other hand is in at most contrast with learning phase. This research work, is aim at taking the advantages of both lazy and eager algorithms to achieve a hybrid algorithm. This approach necessitates employing an eager algorithm of rule induction on the training set, which is led to creation of a set of rules. Then this set of rules is applied on training set, which results in having a set of binary vectors. In order to enhance the training set these binary vectors were added as new attributes. Then with the lazy algorithm of nearest neighbors, we have classified the samples. The outcome of test results from existing algorithms has been compared with our proposed algorithm. The results shows that the proposed algorithm out performs where the volume of samples are high and their attributes are less. The performance of the hybrid algorithm is also remarkable within platforms, with limited processing resources.

Key words: Intrusion Detection System · Machine Learning · Classification · Hybrid algorithms

INTRODUCTION

The growth of network intrusions on large enterprise networks continues to increase. Thousands of hackers probe and attack computer networks each day. These attacks range from relatively benign ping sweeps to sophisticated techniques exploiting security vulnerabilities [1].

Intrusion detection is the task of detecting and responding to this kind of computer misuse, by detecting unauthorized access to a computer network [2]. Intrusion detection systems are "systems that collect information from a variety of system and network sources and then analyze the information for signs of intrusion and misuse" [2]. In other words, an IDS is a device, typically a computer system, that monitors activity to identify malicious or suspicious alerts. An IDS can be compared with a spam filter, that raises an alarm if specific things occur [3].

Machine Learning: The field of machine learning is concerned with the higher-level question of how to construct computer programs that automatically learn with experience [4]. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [4]. Thus, machine learning algorithms automatically extract knowledge from machine readable information [5].

Two categories of machine learning are supervised learning and unsupervised learning. Supervised learning uses labeled training data, whereas unsupervised learning uses unlabeled training data. Supervised algorithms classify examples into known classes, whereas clustering algorithms first discover the classes and then categorize them [6].

Eager Learning: Eager learning is a form of supervised learning, which means that there is a learning module, a model and a classification module, as shown in Figure 3.1. Eager learning algorithms invest most of their effort in the learning phase. They construct a compact representation of the target function by generalizing from the training instances. Classification of new instances is usually a straightforward application of simple learned classification rules that employ the eager learner's model [7].

Lazy Learning: Next to eager learning, there is also lazy learning as a form/variant of supervised learning. In different contexts, memory-based learning algorithms

*This research work is partially supported by Iranian Telecommunication Research Center (ITRC)

Corresponding Author: Mohammad Ghasemzadeh, Computer Engineering Department, University of Yazd, P.O. Box 89195-741, Yazd, Iran, Tel: +98 (0)351 812 2359, E-mail: m.ghasemzadeh@yazduni.ac.ir.

have been named lazy, instance-based, exemplar based, memory-based, case-based learning or reasoning [8]. The reason for calling certain machine learning methods lazy, is because they defer the decision of how to generalize beyond the training data until each new query instance is encountered [10].

A key feature of lazy learning is that during the learning phase, all examples are stored in memory and no attempt is made to simplify the model by eliminating noise, low frequency events, or exceptions [10]. The search for the optimal hypothesis takes place during the classification phase [7].

Decision Tree: A Decision Tree is a representation of how to make a decision according to a particular attribute set [6]. Any given Decision Tree is completely deterministic, although some algorithms can alter their trees given additional knowledge [11]. Each node of a decision tree is some attribute, with the branches representing alternative values of that attribute. Leaves represent the class to place an example in. To make a decision using a decision tree, select a set of values for an attribute and start at the root of the tree Building Decision Trees is a form of supervised learning.

Naive Bayes: The naive Bayes model is a heavily simplified Bayesian probability model [6]. In this model, consider the probability of an end result given several related evidence variables. The probability of the end result is encoded in the model along with the probability of the evidence variables occurring given that the end result occurs. The probability of an evidence variable given that the end result occurs is assumed to be independent of the probability of other evidence variables given that the end result occurs.

Rule Induction: Rule induction is a form of eager learning. During the learning phase, rules are induced from the training sample, based on the features and class labels of the training samples. The goal of rule induction is generally to induce a set of rules from data that captures all generalizable knowledge within that data and that is as small as possible at the same time [12]. The rules that are extracted during the learning phase can easily be applied during the classification phase when new unseen test data is classified.

K-Nearest Neighbors: There are several instance-based learning algorithms. One of the best known is k-Nearest Neighbor (k-NN), which will be used here. Historically, memory-based learning algorithms are even descendants of the k-nearest neighbor (henceforth k-NN) algorithm [13]. The learning phase of k-NN is simply a storage step. As was described earlier, the most important phase for a lazy learner is the classification phase. 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 [7].

Hybrids: Next to lazy and eager learning algorithms, hybrids of both types were evaluated as well. The hybrids are mixtures of the k-NN classifier and rule induction. The reason for constructing hybrids is the contrast between memory-based learning and eager learning. Memory based learners put time in the classification phase, whereas eager learners invest their time in the learning phase. Combining eager and lazy learners into hybrids will produce machine learners that put effort in both the learning phase and the classification phase. This leads to the expectation that this double effort will be repaid with improved performance. The hybrid will use both the global hypothesis as induced by rule induction, as well as the local hypothesis created during memory-based learning [7].

During this research, k-NN will be combined with rule-induction. Although rules appear quite different object from instances as used in k-nearest neighbors classification or instance-based learning, there is a continuum between them. Rules can be seen as generalized instances; they represent a subset of training instances that match on the conditions on the left-hand side of the rule. Therefore, k-NN classification can naturally be applied to rules [12].

RBR: In order to create a hybrid, based on Rules induction, rules are induced from the training set using RIPPER. These rules are then transformed into vectors, representing the binary rule-features. The vectors that are produced can be used as instances of a training set. The test set is converted to a vector of binary rule-features as well and classified using k-NN classification [12].

When using Rules-R-H, the binary rule-features replace the original features in the instances. This hybrid is referred to as Rules-R-H, because the middle R denotes "replace" [7]. From the k-NN perspective, Rules-R-H attempts to repair k-NNs sensitivity to noise and irrelevant features, since induced rules will typically not cover low-frequency noise and will not test on irrelevant features [7]. Replacing the original features of the instances by rule features can thus be considered as a compression and noise filtering step in which the rule induction algorithm has grouped interacting feature values together, which the normal k-NN algorithm is incapable of [7].

ABR: Next to Rules-R-H, there is a second type of hybrid. When using this hybrid, the initial features of the instance are not replaced by the binary rule-features. These binary rule-features are added to these initial features. This is also the reason why this hybrid is called Rules-A-H. Thus, this hybrid is a k-NN classifier with extra added features that represent the per-instance firing patterns of the induced rule set, the same way as described for Rules-R-H. In this case the rule-features can not be considered as a compression and noise filtering step, but it is expected that the rule-features can repair k-NNs sensitivity to noise in a more implicit way [7]. As was described in Section 3.4.2, feature weighting in k-NN gives a higher weight to more important features. As many of the created rule-features will have a strong predictive power, they are likely to receive high feature weights, enabling them to influence the distance calculation. It is expected that the extra rule-features can overrule the influence of noise and irrelevant features [7].

Dataset Details: The data files used are from the University of California, Irvine Knowledge Discovery and Data Mining (UCI KDD) website [14]. The data files give the necessary information to create and train the algorithms. The kddcup.data file lists the value of the class and the value of the attributes. It should be noted that neither [14] nor [15], the main references in this research for this dataset, mention what a "hot" indicator is. The testing data for the 10 percent data set contains 311,029 examples. These examples contain 60,593 normal items and 250,436 attacks. Therefore, this data is most likely atypical because it contains more attacks than normal data. The training dataset contains 494,020 items. There are 97,277 normal connections and 39,6743 attack connections. The attacks make up 80.31% of the dataset.

Evaluation Methods: In order to evaluate the general performance of different classifiers, different metrics were used. These metrics were calculated using the confusion matrix, which shows the predicted and actual classifications.

There are two types of errors that can be made; false positive and false negative. The number in "b" in the confusion matrix corresponds with the type I errors. The number in "c" equals the number of type II errors made. Both false positives and false negatives have to be reduced [16], which means that the accuracy has to be as close to 1 as possible. The accuracy is the proportion of the total number of predictions that were correct and is measured by:

$Accuracy = \frac{a+d}{a+b+c+d}$. Furthermore, the precision is the proportion of the predicted positive cases that were correct. The precision of the classification is calculated by: $Precision = \frac{d}{d+b}$. Another interesting metric for evaluating classification methods is recall (or true positive rate), which is the proportion of alerts that were correctly identified, using the following equation:

$Recall = \frac{d}{d+c}$. Finally, since the equation mentioned above for accuracy may not be an adequate performance measure, the F-Measure was used. Sparseness of the positive examples could cause the average classification accuracy on the testing set to be unreliable [17]. For this reason, the weighted harmonic mean of precision and recall, with β given a value of 1, can be calculated as: $F = \frac{(\beta^2+1)}{\beta^2 \times precision + recall}$

RESULTS

The results of the experiments are summarized in Table 2.

It can be conclude that the results indicates the RBR, ABR and Rule Induction algorithms have far better performance in comparison with other algorithms. In order to find out which of these three algorithms are superior to others, their F-measure parameters are compare as is depicted in Figure 1.

For the next stage of analysis the three parameters of F-measure, Precision and Recall are compared for the RBR, ABR and Rule Induction algorithms. This comparison is done under the condition where few attribute are available. For example when the attributes are filtered with Infogian-2 method, Figures 2, 3, 4".

Figure 5, shows the F-measure for RBR, ABR and Rule Induction algorithms where all attributes have been filtered with CFS method.

Table 1: Confusion matrix

	Predicted Negative	Predicted Positive
Actual Negative	a	b
Actual Positive	c	d

Table 2: Experiments summarized results

		Precision	Recall	F-Measure
RIPPER	Full	0.985	1	0.992
	CFS	0.977	0.992	0.984
	IG-10	0.994	0.999	0.996
	IG-6	0.990	0.998	0.994
	IG-2	0.990	0.993	0.991
IB1-G	Full	0.983	0.998	0.990
	CFS	0.973	0.979	0.976
	IG-10	0.984	0.997	0.990
	IG-6	0.994	0.999	0.996
	IG-2	0.984	0.990	0.987
Naive Bayes	Full	0.965	0.998	0.981
	CFS	0.976	0.928	0.951
	IG-10	0.976	0.988	0.982
	IG-6	0.976	0.989	0.982
	IG-2	0.972	0.937	0.954
Decision Tree	Full	0.970	0.993	0.982
	CFS	0.971	0.993	0.982
	IG-10	0.994	0.999	0.996
	IG-6	0.971	0.993	0.982
	IG-2	0.972	0.937	0.954
RBR	Full	0.986	0.998	0.992
	CFS	0.975	0.998	0.986
	IG-10	0.997	0.999	0.998
	IG-6	0.997	0.999	0.998
	IG-2	0.993	0.996	0.994
ABR	Full	0.987	0.994	0.990
	CFS	0.978	0.999	0.988
	IG-10	0.991	0.997	0.994
	IG-6	0.996	0.998	0.997
	IG-2	0.989	0.986	0.996

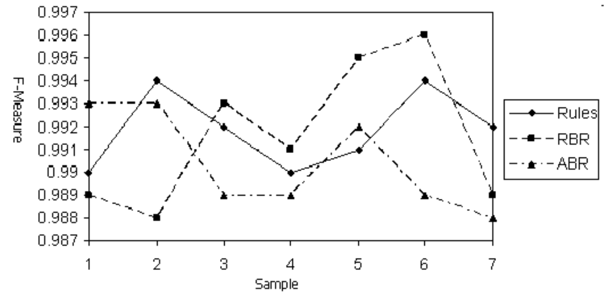

Fig. 1: F-measure of 3 algorithms

Eventually it can be concluded that the Rule Induction algorithm in most instances has far superior performance under the condition where there are high volume attributes, high processing capability and highly susceptible to any attacks. Under circumstances where the volume of training set is small but the processing capability is high, the Rule Induction algorithm is suggested. However when there are limitations on any or some parameters, either we are forced to select some of more effective attributes or to have high volume of training set or high processing capability is available,

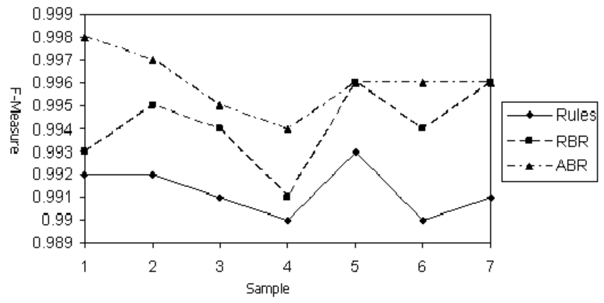

Fig. 2: F-measure of 3 algorithms - Info-2

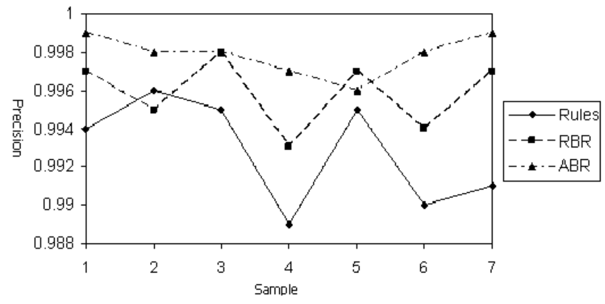

Fig. 3: Precision of 3 algorithms - Info-2

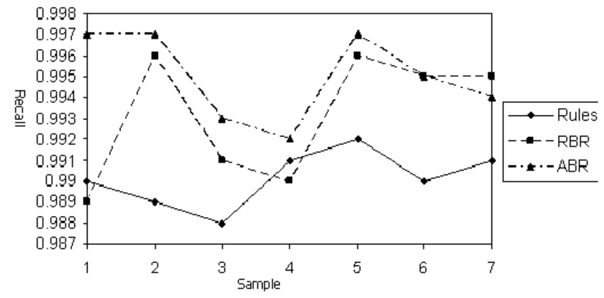

Fig. 4: Recall of 3 algorithms - Info-2

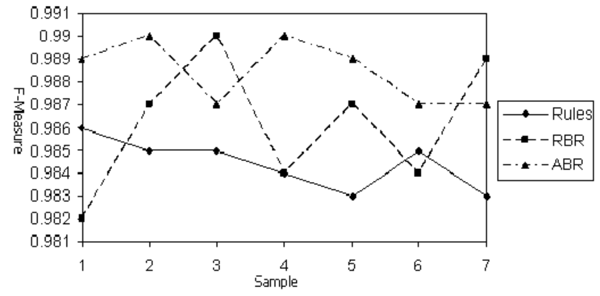

Fig. 5: F-measure of 3 algorithms - CFS

the ABR algorithm is preferable. Finally when there is limitation on processing power RBR algorithm performs better than ABR algorithm.

REFERENCES

1. Jackson, T., J. Levine, J. Grizzard and H. Owen, 2004. An investigation of a compromised host on a honeynet being used to increase the security of a large enterprise network. In Proceedings of the 2004 IEEE Workshop on Information Assurance and Security IEEE.
2. Proctor, P., 2001. The practical Intrusion Detection Handbook. Prentice Hall.
3. Pfleeger, C. and S. Pfleeger, 2003. Security in computing. Prentice Hall.
4. Mitchell, T., 1997a. Does machine learning really work? In AI Magazine, pp: 11-20.
5. Hall, M. and L. Smith, 1996. Practical feature subset selection for machine learning. In Proceedings of the Australian Computer Science Conference (University of Western Australia). University of Waikato.
6. Russell, S.J. and P. Norvig, 2002. Artificial Intelligence: A Modern Approach (International Edition). Pearson US Imports & PHIPEs.
7. Hendrickx, I., 2005. Local Classification and Global Estimation. Koninklijke drukkerij Broese & Peereboom.
8. Van den Bosch, A. and W. Daelemans, 1998. Do not forget: full memory in memory-based learning of ord pronunciation. In NeMLaP3/CoNLL98, pp: 195-204.
9. Mitchell, T., 1997b. Machine Learning. McGraw-Hill International Editions.
10. Daelemans, W., A. Van Den Bosch and J. Zavrel, 1999. Forgetting exceptions is harmful in language learning. In Machine Learning, 34: 11-41.
11. Fisher, D., L. Xu, J. Carnes, Y. Reich, S. Fenves, J. Chen, R. Shiavi, G. Biswas and J. Weinberg, 1993. Applying ai clustering to engineering tasks, pp: 51-60.
12. Van Den Bosch, A., 2004. Feature transformation through rule induction, a case study with the k-nn classifier. In Proceedings on the workshop on advances in Inductive rule learning ath the ECML/PKDD, pp: 1-15.
13. Daelemans, W., J. Zavrel, K. Van Der Sloot and A. Van Den Bosch, 2005. TiMBL: Tilburg Memory-Based Learning Reference.
14. Hettich, S. and S.D. Bay, 1999. Kdd cup data. http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html.irvine, ca: University of california, department of information and computer science.
15. P.A.C.P., J. Stolfo, W. Fan and W. Lee, 2000. Cost-based modeling for fraud and intrusion detection: results from the jam project, in DARPA Information Survivability Conference and Exposition, of DISCEX, 2: 130-144.
16. Gowadia, V., C. Farkas and M. Valtorta, 2005. Paid: A probabilistic agent-based intrusion detection system. In Computers & Security, 24: 529-545.
17. Kubat, M. and S. Matwin, 1997. Addressing the curse of imbalanced training sets: one-sided selection. In Proceedings of the 14th International Conference on Machine Learning, pp: 179-186. Morgan Kaufmann.