Classifying Network Attack Data Using Random Forest

Tonya Fields, Jonathan Graham
Department of Computer Science, Norfolk State University
Norfolk, Virginia 23504
(tlfields,jmgraham,)@nsu.edu

Abstract

Network intrusion Detection Systems (NIDS) have become an important component in protecting industry and government network infrastructure. Various approaches to intrusion detection are currently being used, however most are rule based systems such as SNORT™ whose performance depend on their rule sets. While these rule based systems are highly effective in detecting known intrusions, they are less effective at discovering novel attacks for which no signatures exist. This paper explores the use of the Random Forest (RF) algorithm to build an anomaly based NIDS which can detect novel attacks. Utilizing the KDD'99 data mining training set, we built predictive classification models using the RF algorithm to evaluate the ability to identify attacks from a supplied test set. We explored how the time to build models vary with the number of trees in the forest, learned the optimal number of decision trees for the forest, looked at the role the number of features with respect to the number of decision trees played in the accuracy and also explored the role feature selection had on the accuracy of the trees. Our results indicate adequate results can be obtained using a reduced feature set. This is significant in that a reduced feature set may improve computation cost and may enhance the prediction accuracy by improving the signal to noise ratio.

1 Introduction

Cybercriminals use information technology to exploit, corrupt and harm the computer systems of private citizens, corporate organizations and all levels of government [7][8][9]. Network security technology, particularly Network Intrusion Detection System (NIDS) has become a crucial component in the network security defense infrastructure. NIDS are able to detect attacks in progress and interrupt them, alert appropriate personnel and determine the identity and location of the perpetrators. There have been many approaches to building NIDS. The majority of these have been very successful in detecting previously known attacks; however those approaches have had little success in identifying and preventing novel attacks. To overcome these shortcomings, a growing number of research projects have used various data mining (also known as machine learning) algorithms to build NIDS [1][2][12][13][14]. This approach holds the promise of being able to recognize novel attacks. Our research evaluated the performance of classifiers built using the Random Forest algorithm. We compared the accuracy and the speed when using all features of the dataset to the accuracy and speed when a reduced feature set is used and found the reduced dataset performed comparatively well.

As attacks continue to escalate against sensitive U.S.A. installations, a reliable NIDS to detect novel attacks is urgently needed [6][7][8]. Our research investigates the Random Forest approach to developing a NIDS and demonstrates its accuracy for both full and reduced data sets. Our results show that Random Forest classifier can accurately identify novel attacks.

2 Background

The most widely used dataset for test and evaluation of intrusion detection research, is the dataset of the Third International Knowledge Discovery and Data Mining Tools Competition held in 1999 (KDD'99) [11].
The KDD'99 dataset made available by Sal Stolfo and Wenke Lee [11] is a modified version of the 1998 MIT Lincoln Lab DARPA Intrusion Detection Evaluation Program dataset, prepared and managed by MIT Lincoln Labs [16]. The dataset has been used as the bench mark for many researchers to compare their results with the original results as well as ongoing experiments and research. The task of the KDD'99 Classifier Learning Contest was to build a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" or normal connections in a computer network. Each record of this dataset has 41 features (including duration, protocol, type, flag, etc.) and labeled as either normal, or labeled as one of the attack type (such as Smurf, Pearl, etc.). Attacks fall into four main categories:

Denial of Service (DoS): whereby an attacker overwhelms the victim host with a large number of bogus requests which overwhelms the host and prevents legitimate request, (e.g., syn flood).
Remote to User (R2L): intruder tries to access the system from a remote machine to exploit the system vulnerabilities in order to control the remote machine; often gains access by guessing password.

CATA 2016, April 4-6, 2016, Las Vegas, Nevada, USA
978-1-943436-02-6 / Copyright ISCA

User to Root (U2R): an attacker tries to get access rights from a normal host in order to gain the root access to the system (e.g., various "buffer overflow" attacks).
Probing: an attacker employs surveillance and other information gathering attempts to discover services available on the network in order to look for exploits (e.g., port scanning).

The full KDD'99 dataset consists of nearly 5 million connections (4,893,430) . The standard training set that was extracted from the full set contains 494,020 connections. Within the training set attacks are categorized into either one of four attack types or as normal. The four attack types are (1) probing, (2) denial of service, (3) User-to-root (U2R), and (4) Remote-to-Local (R2L). The task of the original KDD'99 contest was to build a classifier capable of distinguishing between four kinds of attack types and the normal traffic. The original training set (10% of the full dataset) is imbalanced in that certain connections such as DoS has 391, 458 connections but the U2R has only 52 connections.

3 Related Work

When comparing our research to others we looked at researchers that used the Random Forest algorithm and the KDD'99 Dataset and explored how others approached the, preprocessing of the dataset, feature selection of the dataset and various parameter setting of the RF algorithm and compared the performance and accuracy of our classifiers.

Zhang and Zulkernine [21] used the dataset from the 1999 (KDD'99) and the Random Forest algorithm to build a classifier. Their work focused on (1) feature selection (2) preprocessing the training set and adjusting the parameters of the RF algorithm. For feature selection they employed variable importance as calculated by the RF Algorithm and used the top 38 features for their experiments. When preprocessing the training set, they balanced the training set by down-sampling the Normal and DoS classes by randomly selecting 10% of connecting belongs to Normal and DoS from the original dataset, They oversampled the minority attacks of U2R and R2L by replicating their connections. Their balanced training set consisted of 62,620 connections as was much smaller than the original 494,020. They also experimented with parameter optimizing for RF running experiments on 10 trees using the reduced dataset as well as a reduced feature set. Their results show better performance over the original results of the KDD'99 data sets. Our reduced feature results share 12 of the top 20 features in common. Our approach was different in that we did not use any balancing techniques of the training set based on type of attack, rather we preprocessed the dataset by removing the duplicate records and compared the results with duplicates records included. We also looked at feature selection by using the Attribute Selection Information Gain Ranker filter within WEKA to rank the importance of our 41 features. We then created a dataset that contained only the top 25 features as determined by the Information Gain Ranker Filter.

Pundir and Amrita [18] aim was to find an optimal feature subset of the KDD'99 dataset using a regularized random forest (RRF) package of r-tool to rank the features of the dataset by importance then used the RF classifier of WEKA to classify the feature set and check their performance against the results when all 41 features were used. They concluded the results of the smaller feature set consisting of 15 features was comparable in accuracy as when using the full 41 feature set. As mentioned above we also experimented with a reduced feature set and found subsets that were comparable in performance to the full feature set, but required less time to build the model .

Tesfahun, and Bhaskari [19] used a Synthetic Minority Oversampling Technique (SMOTE) to deal with the class imbalance of the KDD'99 training set and Information Gain to accomplish feature reduction of the dataset. SMOTE is an oversampling approach developed by Chawla et al. [5] in which the minority class is over-sampled by performing some operation on the original data for generating new synthetic samples. Their results show that the RF classifier with SMOTE and Information Gain based feature selection performed better than when using the Random Forest without SMOTE. Specifically the detection rate for the minority class "User to Root" (U2R) attack increased from 0.596 to 0.962. Our experiments shared 16 of the top 20 features in common with their work although our accuracy detection rate was not as high.

4 Methodology

We used the KDD'99 Dataset for the experiments. We used three distinct subsets for our experiments as follows:

- **Dataset 1**:10 Percent of the full training set consisting of 494,019 records.
- **Dataset2**: Removal of duplicates from the 10% training set resulting in 142,999 records.
- **Dataset3**: Reduced feature set of relevant features of the 494,019 records.

To analyze the data we used the Random Forest [3][4]]algorithm from Waikato Environment for Knowledge Analysis (WEKA) data mining software. WEKA is an open source collection of machine learning algorithms for data mining tasks written in Java [20]. WEKA is recognized as a landmark system in data mining and machine learning and has achieved widespread acceptance within academia and business circles. The work of Mitkut and Reischl [16] compared the performance of both commercial and open source data mining tools and found WEKA to be one of the

top 10 open source tools for data mining. Using WEKA allowed us to focus on our experiments rather than on developing algorithms.

In our research we sought to answer the following questions:

1. How does the accuracy of the RF classifier vary as the number of trees are increased?
2. Are there an optimal number of decision trees for the random forest?
3. What role does the number of features with respect to the number of decisions trees play in the accuracy of the Random Forest algorithm?
4. What role does feature selection play in the accuracy of the Random Forest algorithm?

5 Experiments and Results

5.1 Experiment 1: How Does the Accuracy of the Random Forest Classifier Vary as the Number of Trees are Increased?

We first built the classifiers with dataset 1 as explained above no duplicates removed, using the entire 10% training set 494,019 records to train the classifiers) Each tree was built using six randomly selected features. Six was chosen since the literature suggests that $\log_2 F$ (where F= [total number of features]) would produce the best results. Our dataset contains 41 features, ($\log_2 41$) thus we chose six random features to build each tree in the forest. These results are shown in Table 1. We then built the classifiers with dataset 2 as explained above (duplicates removed, resulting in 142,999 records to train the classifiers).

Table 1: Results of Random Forest Classifier Using 6 Random Features with Duplicates Removed vs Duplicates Included

# Trees	Detection Accuracy Dataset #1	Time to build model	Detection Accuracy Dataset #2	Time to build model
	Duplicates Included		Duplicates Removed	
1	91.8461	8.44	90.276	9.91
2	91.612	26.97	90.273	17.52
3	92.0329	37.91	90.66048	27.77
4	92.0069	45.55	90.89842	40.39
5	92.0396	61.32	90.97476	44.98
6	92.0406	71.89	91.01823	54.51
7	92.0438	100.68	91.0568	62.49
8	91.9329	95.04	91.15593	75.08
9	91.9326	107.19	91.1991	83.74
10	91.9329	118.84	91.2607	94.18
11	91.9356	163.73	91.1806	101.74
12	91.9348	216.17	91.2434	114.69
13	91.9358	190.76	91.24995	122.23
14	91.9332	160.11	91.27506	126.53
15	91.9368	178.12	91.30709	140.82
16	91.9364	190.32	91.32706	157.22
17	91.9374	248.91	91.31814	175.19
18	91.939	278.34	91.35269	170.79
19	91.9403	249.14	91.33973	181.58
20	91.9403	231.57	91.34673	199.12
Average	**91.9394**	**139.05**	90.276	9.91

Table1 shows result of our experiments to determine how the accuracy of the classifier varied as the number of tree increased. The experiment was ran for 20 trees as shown in Column 1. Column 2 shows the accuracy obtained using the full 10% dataset with duplicates included. Column 3 shows the time in seconds taken to build the model, column 4 shows the accuracy of when using data set 2 (duplicates removed) and column 5 is the time taken to build the model using dataset 2. Overall there is a general increase in accuracy as the number of trees are increased when using dataset 1. When using dataset 2 the accuracy increased for tress 1-7 then there is a slight decrease for tress 8-20. So for the second dataset it was inconclusive as to whether the accuracy increased with the number of trees

5.1.2 Experiment 2: Is There an Optimal Number of Decision Trees for the Random Forest?

From experiment 1, using dataset 1, the algorithm peaks at 7 trees suggesting 7 might be the optimum number of trees for the random forest. To further investigae we chose to look at the performance of the RF with as many as 50 trees. The time to buid the models dramically increased, however the accuracy did not improve beyond the previous optium obtained using 7 trees. As shown in Figure 1.

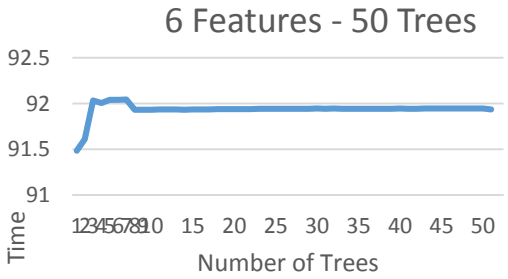
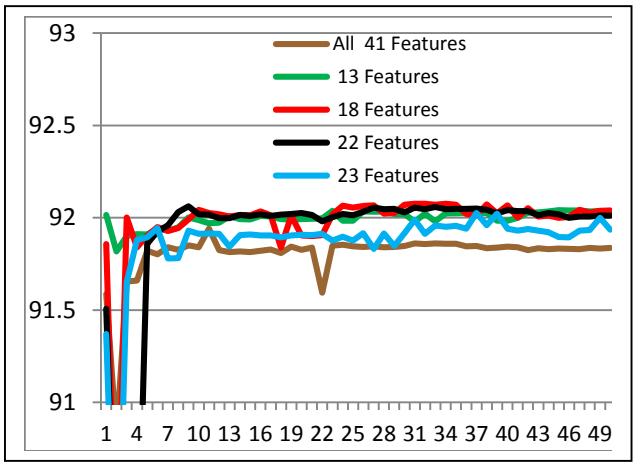


Figure 1: Performance of Random Forest –Analysis of 50 Trees

5.1.3 Experiment 3: What Role does the Number of Features with Respect to the Number of Decision Trees Play in the Accuracy of the Random Forest Algorithm?

We varied the number of trees from 1 to 50 and varied the number of features from 1 to 50 for a total of 2050 experiments in an attempt to determine which of the 2050 combinations of features and trees would produce the highest detection accuracy. Results show that the highest accuracy was obtained when using 13, 18, 22, and 23 random features. The accuracy was independent of the number of trees once the number of trees is greater than 10. See figure 2.

Figure 2 Random Forest Accurancy Analysis of 50 trees using 41 Features



5.1.4 Experiment 4: What Role Does Feature Selection Play in the Accuracy of the Random Forest Algorithm?

Since the classification is a direct result of the set of features, that are used, feature selection plays a critical role in the performance of the classifier. Determining which of the 41 features plays the greatest role in the classifying normal vs attack connections may result in a reduced feature set which could improve computations cost and y enhance the prediction accuracy by improving the signal to noise ratio.

Various approaches at reduced feature selection of the KDD 99 dataset produced interesting results[18][19][21]. We used the Attribute Selection Information Gain Ranker filter within WEKA to rank features by importance. We compared our feature ranking with the ranked feature results of Zang and Zulkernine who used the RF algorithm to rank the importance of features [21].

We noticed similarities in the top 20 highest ranked features of both approaches. In particular we noticed the top 20 features of both algorithms contained 12 features in common. These common features are indicated by asterisk in Table 2. We ran an experiment using the dataset 3 which included the 25 most relevant features .Our results show an accuracy of 92.0728 when using all 41 features as compared to an accuracy of 91.364 when using the reduced dataset.

Table 2: Top Feature Ranking Comparison

Feature Ranking Comparison				
WEKA Info Gain			**Zang and Zulkernine**	
Feature #	**Feature Name**	**Rank**	**Feature #**	**Feature Name**
***5**	**src_bytes**	1	***3**	**service**
***23**	**count**	2	***23**	**count**
***3**	**service**	3	10	hot
***24**	**srv_count**	4	***35**	**dst_host_diff_srv_rate**
***36**	**dst_host_same_src_port_rate**	5	***33**	**dst_host_srv_count**
2	protocol_type	6	17	num_file_creations
***33**	**dst_host_srv_count**	7	8	wrong_fragment
***35**	**dst_host_diff_srv_rate**	8	6	dst_bytes
34	dst_host_same_srv_rate	9	32	dst_host_count
30	diff_srv_rate	10	14	root_shell
29	same_srv_rate	11	***24**	**srv_count**
4	flag	12	***5**	**src_bytes**
6	dst_bytes	13	***36**	**dst_host_same_src_port_rate**

6 Conclusion and Future Work

Our first experiment showed minimal gain by increasing the number of trees. Our data set with duplicates include showed gains in detection accuracy of less than 2 percent. As far as the optimal number of decision trees for our forest, ten appears to produce the best results. This number is of course expected to vary from one dataset to another and is certainly not a universal result. Complex feature interaction and the relative importance of the different features would indicate that there would be no pattern on the number of features needed to produce the best results. This was borne out in our research which showed 13, 18, 22 and 23 random features produced the best results when 10 trees were used. In any data set, one would expect that all features would not have equal importance, and this is true for the KDD'99 dataset. Our list of important features have elements in common with other lists reported in the literature, however there was not a clearly identified universal "best set" of features.

In conclusion it appears that for our Random Forest algorithm, 90 – 92 % appears to be as good as it gets in detecting the accuracy of attack vs normal connections. Our results are comparable to the top performers of the original KDD'99 Classifier Learning Contests that correctly classified 92.92% of the test samples. The good news is that this maximum detection accuracy can be obtained with a small number of trees and a reduced set of features. A deep analysis of the results of any data mining algorithm will always be challenging because of the complex interaction of features and the large number of possible tuning parameters.

In the future we plan to examine other more recent datasets such as the 2015 Microsoft "Kaggle" dataset [15] which is 500 GB(s) of data of known malware files representing a mix of 9 families in 2 datasets: train and test; 10868 malwares in the training set and 10783 in the test set. This should result in more relevant finding.

In addition to the Random Forest algorithm, we will explore other algorithms such as Naïve Bayes, Support Vector Machines and Neural Networks to investigate whether or not we could improve upon our results. Additionally, we want to run the experiments on a distributed "big data" platform such as Hadoop using Spark and "R". We would expect our algorithms to execute more quickly, but will be interested in the exact way in which the running time varies as we increase the number of computational noise.

References

[1] Amor Nahla., Benferhat Salem., Elouedi Zied. Naïve Bayes vs. Decision trees in Intrusion Detection Systems. In *Proceedings of the ACM SAC* 2004, pages 420-424, New York, NY, March 2004.

[2] Barbara, Daniel, et al. ADAM: Detecting intrusions by data mining. *In Proceedings of the IEEE Workshop on Information Assurance and Security*, pages 11-16, 2001.

[3] Breiman Leo, "Random Forest:, *Machine Learning*, (45): 5-32, 2001

[4] Breiman Leo and Cutler, Adele. "Random Forest" [Online] Available; http://www.stat.berkeley.edu/~breiman/RandomForests/ [Accessed Nov, 12, 2015]

[5] Chawla Nitesh, Bowyer Kevin, Hall Lawrence, and Kegelmeyer W., SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16:321-357, 2002.

[6] Defense in Depth Strategy. [Online]Available: http://www.nsa.gov/ia/_files/support/defenseindepth.pdf [Accessed Oct 30, 2015]

[7] Federal Aviation Administration. Report on Review of Web application Security and Intrusion Detection in Air Traffic Control Systems. Federal Aviation Administration Report: FI-2009-049 (May 2009). [Online] Available; https://www.oig.dot.gov/sites/default/files/ATC_Web_Report.pdf [Accessed Oct 30, 2015]

[8] Federal Bureau of Investigation (FBI) and National White Collar Crime Center. Internet Crime Complaint Center, 2003 Internet Fraud Report; January 1, 2003 – December 21, 2003: [Online] Available: http://www.ic3.gov/media/annualreport/2003_IC3Report.pdf [Accessed Oct 24 14, 2015]

[9] Government Accountability Office (GAO). Public and Private Entities Face Challenges in Addressing Cyber Threats. Government Accounting Office GAO03-852T, 2007. [Online] Available: http://www.gao.gov/new.items/d07705.pdf [Accessed November, 10, 2015]

[10] Han Jiawei., Kamber Micheline., *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2nd edition, San Francisco, CA, 2006.

[11] KDD dataset, 1999; [Online], Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed Feb. 9 2012]

[12] Kumar V., Parallel and Distributed Computing for Cyber security. *IEEE Distributed* Systems, Online, vol. 6, no. 10, 2005.

[13] Lee Wenke. and Stolfo Sal, Combining Knowledge Discovery and Knowledge Engineering to Build IDSs, *International Symposium on Recent Advances in Intrusion Detection (RAID).*1999.

[14] Lee Wenke, Stolfo Sal. and Mok Kui. Adaptive Intrusion Detection: A Data Mining Approach. *Artificial Intelligence Review*, 14(6):533-57, 2000.

[15] Microsoft Malware Classification Challenge (BIG 2015). [Online].Available: https://www.kaggle.com/c/malware-classification. [Accessed Nov 8, 2015]

[16] Mikut, Ralf and Reischl, Markus. "Data mining tools". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5): (201) 431–443.

[17] MIT Lincoln Laboratory, "DARPA Intrusion Detection Evaluation", [Online] Available https://www.ll.mit.edu/ideval/docs/attackDB.html [Accessed Nov 8, 2015]

[18] Pundir, Sneah Lata and Amrita. Feature Selection Using Random Forest in Intrusion Detections System. *International Journal of Advances in Engineering & Technology*,6(3), 2013

[19] Tesfahun, A. and Bhaskari, DL. Intrusion Detection Using Random Forest Classifier with SMOTE and Feature Reduction. *International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, Visakhapatnam, AP, India, pp. 127-132,2013

[20] A WEKA software, Machine Learning, [Online] Available http://www.cs.waikato,ac.nz/ml/weka/, The

University of Waikato, Hamilton, New Zealand. [Accessed Jan 14, 2011]

[21] J. Zhang, M. Zulkernine and A. Haque.Random-Forests-Based Network Intrusion. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 38(5) :649-659. 2008