Network Attribute Selection, Classification and Accuracy (NASCA) Procedure for Intrusion Detection Systems

Zheni Stefanova
Department of Mathematics and Statistics
University of South Florida
Tampa, Fl 33620-5700, USA
stefanova@mail.usf.edu

Kandethody Ramachandran
Department of Mathematics and Statistics
University of South Florida
Tampa, Fl 33620-5700, USA
ram@usf.edu

***Abstract*— With the progressive development of network applications and software dependency, we need to discover more advanced methods for protecting our systems. Each industry is equally affected, and regardless of whether we consider the vulnerability of the government or each individual household or company, we have to find a sophisticated and secure way to defend our systems. The starting point is to create a reliable intrusion detection mechanism that will help us to identify the attack at a very early stage; otherwise in the cyber security space the intrusion can affect the system negatively, which can cause enormous consequences and damage the system's privacy, security or financial stability. This paper proposes a concise, and easy to use statistical learning procedure, abbreviated NASCA, which is a four-stage intrusion detection method that can successfully detect unwanted intrusion to our systems. The model is static, but it can be adapted to a dynamic set up.**

Keywords*— *cybersecurity; network; vulnerability; NASCA; Intrusion Detection.

I. INTRODUCTION

The Network Attribute Selection, Classification and Accuracy (NASCA) procedure is a four step procedure for intrusion detection systems. The model first begins by extracting the information from the Transmission Control Protocol (TCP) and then ranking the relevant information that characterizes the network. During the second stage, it classifies the network as being under *attack* or *not*. Moreover, if the model detects that the system is under attack, it executes another level of classification approach to identify the type of attack that is occurring. The third stage provides us with results about the accuracy of the performed analytical estimate. The attribute selection method begins with choosing a subset of appropriate information by removing redundant, unrelated, and noisy data from the original dataset. The classification method starts initially with an existing set of labeled networks; consequently, it learns the dependency between the content of the network and its corresponding label and then predicts the label of a set of unlabeled networks as accurately as possible using a decision tree type of analysis.

The common classification procedures are based on familiar machine learning models such as Naive Bayes [12] or discriminant models such as Support Vector Machines [11] and [22].Those machine learning approaches study primarily a set of network characteristics (where there is no order or sequence in the specifics of the network), and their objective is typically to calculate a category score. Furthermore, the question that arises, is how properly they will perform in network classification, especially when we observe more noise and the information structure is not homogenous [11]. We need to test how accurate are these methods for predicting categories for large networks, where the information is concentrated in only few characteristics. The goals of this paper will be the following; (1) To propose a NASCA procedure for intrusion detection systems; (2) To test it on a real data set; (3) To report the obtained results; and (4) To provide evidence of why this procedure is able to outperform the existing ranking and classification techniques.

II. RELATED WORK

Some of the earlier works on network classification were made by Cannady [5] and [17]. Cannady indicates that neural networks are reasonable solutions when they are trained for a specific problem domain with representative sets of training data. The model is incapable of adapting to streaming and especially new data. Therefore, it is necessary for the individual protecting our system, to take off-line the data every time when he needs to train the model and to run it to the updated set of representative data. Furthermore, the authors employed a three-layer control feedforward mechanism intended to yield a series of input-output mappings. The particular Intrusion Detection System (IDS) agent, consequently, learns how to spot flood-based Denial of Service attacks based on Internet Control Message Protocol (ICMP) together with the User Datagram Protocol (UDP). The method initially studies how to detect the ICMP attacks and as a result, updates and retrains the model frequently. Therefore, it is capable of learning about how to recognize new attacks based on the UDP protocol.

One approach, used to find breaches in host-based intrusion detection systems is established by observing sequences of system calls. These calls are initiated by a process running on the host, and they are grouped in sets of

Interdisciplinary Data Science Consortium (IDSC), https://idscbigdata.com/

traces. Each trace contains a list of system calls generated from the beginning to end. To apply machine learning techniques using sequences of system calls, scientists commonly construct a transition model using labelled examples of normal and attack activity. The states of the model are defined by short sequences of system calls in a single trace. Xu et al. [24] applied Hidden Markov Models (HMM) and Reinforcement Learning (RL) to detect host intrusion by learning the state transition probabilities. Consequently, they argued that there are uncertainties in modelling the state transition on IDS, therefore HMM are capable to offer an appropriate alternative solution to the problem.

In [19] the researchers employed k-means clustering technique to find the accuracy of intrusion detection. Nguyen et al. [16] employed Principal Component Analysis (PCA) for outlier and anomaly detection in IDS. Shilpa et al. [1] compared different methods for attribute selection and examination of abnormality detection. It is an important task to select a proper model, which will assist the further analysis and outline an optimal solution in dealing with the tradeoff between accuracy and complexity.

III. Data Description

A network is structured by Transmission Control Protocol (TCP) packets starting and ending at some well-defined time between which data streams to and from one source IP address to another target IP address under a determined protocol. Each network is labelled as either normal or as an attack with exactly one specific attack type. The data that are used in this paper are ISCX NSL-KDD Data Set[1] that are an improved version of the KDD CUP 99, DARPA[2], conducted by MIT Lincoln Labs. Lincoln Labs simulated an environment to obtain nine weeks of raw TCP dump data for a local-area network (LAN), pretending a typical United States Air Force network. Moreover, they operated the LAN as if it were a true Air Force atmosphere, however simulated numerous attacks. Even though the contribution of DARPA and KDD (University of California) dataset is remarkable [21], their ability to reflect real-world situations has been widely questioned, McHugh (2000) and Brown et al. (2009). Therefore, in order to conduct a meaningful research, in the current paper, we will use the ISCX NSL-KDD dataset, provided by The Information Security Centre of Excellence (ISCX) within the Faculty of Computer Science, University of New Brunswick, Canada. The data in the NSL-KDD dataset is either labeled as normal or as one of the 24 different kinds of attack. Additionally, these 24 attacks are clustered into four groups: Denial of Service (DoS), Probing (Probe), Remote to Local (R2L), and User to Root (U2R). The attack types according to the dataset are summarized in Table 1.

There are 42 variables, one of them represents the condition of the network, labeled as being under a specific type of attack or normal.

[1] "Nsl-kdd data set for network-based intrusion detection systems." http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.

[2] KDD Cup 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.htm

TABLE I. Types of Labels of the Network

Types of network			Number of instances
Normal			67,343
Attacks (37)	DoS (11)	Land, Pod, Smurf, Teardrop, Mailbomb, Processtable, Neptune Udpstorm, Apache2, Worm, Back	45,927
	Probe (6)	Satan, IPsweep, Nmap, Portsweep, Mscan, Saint	11,656
	R2L (13)	Guess_password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named	52
	U2R (7)	Buffer_overflow, Loadmodule, Rootkit,Perl, Sqlattack,Xterm,Ps	2,756
Total			127,734

They are summarized in three categories. (1) Basic features: this group contains all the characteristics that are collected from a TCP/IP. (2) Traffic features: this class outlines the features which are measured with regards to a period of time and they are divided into two clusters: same host and same service features. (3) Content features: in order to recognize the suspicious behavior, we may evaluate features like the number of failed login attempts.

IV. Used Methodology

A. *Network Intrusion Detection - Outline of the Procedure.*

The classification agent is capable of making decisions in a constantly changing environment and therefore testing the model, while evaluating the network. As an addition, an essential advantage is the fact that only relevant network information will be used before the particular classification decision is accomplished. Under those circumstances, the agent is able to learn how to classify the network promptly, accurately and efficiently.

The steps of the procedure are presented on Figure1:

1) Network Data Collection: Collect the network data from the TCP/IP, using for any raw socket communication instrument for reading the data;

2) Attribute Selection: Apply an Information Gain Attribute Evaluation Approach, which is a tool for feature reduction and attribute ranking for classification purposes (Mitchell, 1997);

3) Classification: Provide a classification procedure, which will learn how to label the network precisely and promptly. In this work, we will use a two-stage procedure:

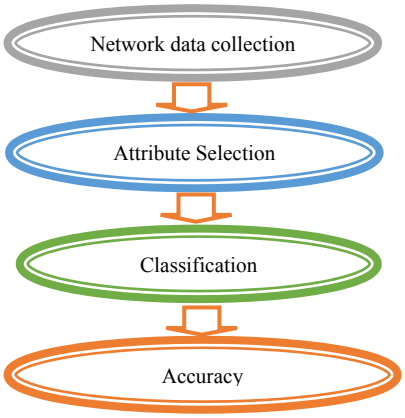
a) Classify the network as being under attack or not: In the first step of the classification, we will use a Random Forest (RF) to label the network as being under attack or not; if Step (a) labels the network as being is under attack, then use step (b).

b) Classify the network further only if an attack is occurring: We use a partial decision tree (PART) method to

further classify possible estimate for the type of the attack that is undergoing.

4) Accuracy: Here, we will demonstrate the accuracy of the estimate for the specific dataset used. Once the data is trained and a model is built, accuracy factors are reported.

Fig. 1. NASCA Procedure



B. *Attribute Selection, using the Information Gain Attribute Evaluation Approach.*

Attribute selection is the method of recognizing and eliminating the irrelevant and redundant information from an evaluated system. If we are able to eliminate some of the unrelated features, we can moderate the complexity, by removing irrelevant dimensions and to enhance the performance of the prospective classification procedure. The decision agent is capable of operating not only quicker and with less information, but also to improve the classification accuracy process, [9].

There exist a variety of different proposed attribute selection methods. For example, Hall and Holmes et al. analyzed number of these attributes selection techniques and outlined the ones that achieved noticeable results, namely [10] "Correlation-Based Feature Selection", "Information Gain Correlation", "Wrapper Subset Evaluation"[14], "Recursive Elimination of Features" [13], and "Consistency-Based Subset Evaluation" [15]. The main idea of the attribute selection procedure is to rank the relevant variables and henceforth to use only the appropriate information in order to perform the classification of the network. Attribute reduction is the process of mapping the existing high-dimensional data onto a lower-dimensional space. For example, for a given dataset points of n variables $\{x_1,x_2,\ldots,x_n\}$, we need to compute their dimensional representation $x_i \in R^d \rightarrow y_i \in R^p$ $(p<<d)$. The criterion for feature reduction can be different based on diverse problem settings. In this paper, we will test different ranking algorithms, consequently we will provide results, so that we can demonstrate the outperformance of the Information Gain Ranking filter in comparison to other algorithms for attribute selection. Information gain (IG) quantifies the volume of information in bits about the class estimate. It measures the expected decrease in entropy of the class variable after the value for the feature is observed (uncertainty associated with a random feature) (Mitchell, 1997). It is an entropy based filter that categorizes the gain of the attributes. For example, an Entropy for i classes can be defined as:

$$H(X) = -\sum_i P(x_i)\ log_2(P(x_i)) \quad (1)$$

Entropy signifies the level of insecurity in the system. In the equation (1), $P(x_i)$ is the marginal probability density function for the random variable X, which is obtained by integrating the joint probability density function. First, we observe the values of X in the training data set S and separate them according to the values of a second feature Y. Correspondingly, we measure the entropy of X with respect to the partitions induced by Y. In the case that the measure of the entropy is a smaller than the entropy of X prior to partitioning, we say that there is a relation between features X and Y.

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j)\ log_2(P(x_i)) \quad (2)$$

$P(x_i|y_i)$ is the conditional probability of X given Y. Having in mind the fact that the entropy is a condition of impurity in a training set S, we can describe a measure reflecting additional information about X provided by Y, which represents the amount by which the entropy of X decreases. This measure is known as information gain and it is given by:

$$IG\ (X|Y) = H(X) - H(X|Y) \quad (3)$$

The larger the value of the informational gain (IG), the further the attribute contributes to the data set. Although, a disadvantage of the IG criterion is that it will favor attributes with more values, because it's biased towards choosing attributes with larger number of values that produce higher IG. However, given the characteristics of our particular data set, IG is a preferred instrument for attribute selection.

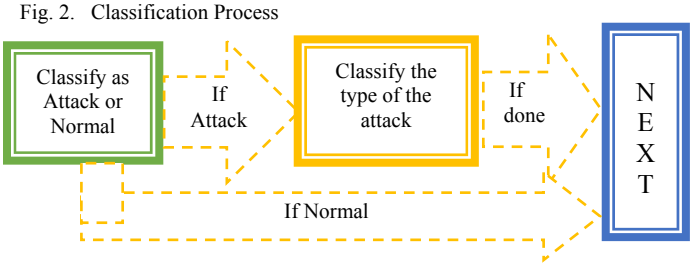
C. *Classification Process*

The classification method is an essential step of the proposed procedure. It is a two-stage process with objectives – accuracy, precision and faster classification. Therefore, in order to accelerate the procedure, supplementary analyses will be completed, only whenever the network is classified as being under attack. The proposed procedure combines two major classification techniques, namely Random Forest and consequently partial decision tree (PART). In Fig. 2 we can observe the main idea of this stage and how the classification process is analyzed.

Random forest is introduced by Breiman [2], it is a cooperative learning method that produces various classifiers and summarizes the outcomes. As an addition, it can be executed if needed with two major procedures in order to perform the classification or prediction analysis, namely boosting and bagging. On one hand, in boosting, the succeeding trees assign additional weight to instances that were incorrectly classified by earlier trials and at the end, a

weighted score is calculated for the classification purposes. On the other hand, in bagging, the succeeding trees are independent from the previous trees, moreover every tree is grown by means of a bootstrap sample.

Fig. 2. Classification Process



The classification is accomplished based on the so called majority score split (Liaw and Wiener 2002). Random forest grows multiple trees and each of them produces a classification with an assigned score for the specific class. As a result, the forest indicates the classification with the highest score. The term originated from random decision forests that was first proposed by Tin Kam Ho by Bell Labs in 1995. Random forest (RF) is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution of all trees in the forest. The error of a forest of tree classifiers, depends on the score of the distinct trees in the forest and the correlation among them.

The Random Forest process in our analysis starts with the creation of many trees. It introduces randomness into trees such that each tree has minimum correlation with the other trees. Each tree in the collection is formed by first selecting at random, at each node, a small group of the input characteristics to split and, secondly, by calculating the best split based on these features in the training set. The method that we will use in the splitting process is a two-step randomization technique. Initially the tree is grown by using bootstrap sample and then we introduce another stage of randomization, using random feature selection approach. In a summary, instead of splitting the node of the tree using all k features, we will randomly select at each node of each tree a subset of m-tries, where $m \epsilon [1,k]$ for splitting the node.

Brieman et al., 1984 outlines some of the splitting suggestions and development of the tree. The method suggests a subspace randomization structure that is combined with bagging, Buehlmann and Yu, 2002 [3]. The idea is to re-sample, with replacement, the training data set every time when a new individual tree is built. Biau and Devroye (2010) as well as Meinshausen (2006) studied the consistency of random forests in the setting of conditional quantile prediction [6].

The detailed procedure of RF starts with the creation of a new random vector Θ_n for each n-th tree that is independent from the previous random vectors $\Theta_1, \Theta_2 \ldots$. Moreover, it is generated from the same distribution and based on a training set Θ_n, a tree is grown. Consequently, the tree organization of RF is based on classifiers $\{h(\mathbf{X}, \Theta_n\}, n=1,2,3\ldots\}$, where $\{\Theta_n\}$ are *i.i.d.* random vectors. Respectively, each tree is assigned a score as described above and an input vector **X**.

If the network is under attack, then we will employ the partial decision tree PART [7] and [8] method for classification purposes has several advantages compared to the other methods. PART will label the network with the type of the 37 types of attack and therefore it will perform the process faster rather than the RF. The reason of why we will employ PART algorithm instead of RF again will be because it is a rule based method, which does not need to achieve a global optimization to produce accurate results, which will speed up the classification process. It knows how to label new occurrences quickly and possesses an outstanding accuracy and precision. Moreover, it adopts the separate-and-conquer approach, and consequently, once it constructs a rule, it eliminates the covered alternatives. Hence, it keeps creating repeatedly rules for the residual instances until it executes all possible outcomes. In essence, to create a single rule a pruned decision tree is built for the current set of instances, as a result the leaf with the greatest coverage is converted into a rule, and afterwards the tree is discarded.

The idea of recurrently building decision trees only to reject the majority of them in PART is not as unusual as it seems. A pruned tree can be employed to obtain a rule, instead of constructing it incrementally by adding combinations one by one. PART is capable of avoiding the over-pruning problem of the rule learner - separate-and-conquer [4]. The model performs with an improved speed, although the above advantages are still achieved. The key idea is to build a "partial" decision tree instead of a fully explored one. A "partial" decision tree is a regular decision tree which builds divisions to unknown subtrees. Once a "partial" tree has been built, a single rule is produced based on it [4]. The aim is to find the supreme general rule by choosing the leaf that covers the greatest number of instances or the leaf with the lowest error rate.

D. Accuracy

The Accuracy step is related to testing the performance of the model based on two main criteria: accuracy and complexity. A too complex model will take longer time to classify the network, as an addition, there is a tradeoff between complexity and accuracy. Different classification and attribute selection models are performed, and a ranking comparison is done based on those two criteria. The results are reported with regards to the cost analysis of the complexity, Kappa statistic and the error rate for accuracy. The data set is tested on a basis of 80-20% split, and the results are compared and reported, depending on the applied classification methodology.

V. Results

The primary objective of the initial step is to eliminate the redundant variables and the features which do not contribute to the classification process. The data set contains 127,734 observations and the test is done based on the 80-20% split. A variety of attribute selection methods are tested before selecting an approach. This model is chosen in terms of

number of eliminated variables and the time to perform the elimination.

The nominated method for attribute selection is the Information Gain Ranking Filter, where in total eleven variables are excluded from the further analysis, since they do not contribute to the classification process, "Table II". Additionally, in the next step of the procedure, we will analyze how the model increases its performance, based on the attribute selection step. The top three variables are likewise essential to an appropriate evaluation, especially whenever the administrator is concerned with the vulnerability of the network. However, the objective of this article is to present how to effectively classify the network based on the information structure that we can obtain from the TCP packets. Therefore, our intent is to reduce the dimensions of the data set, consequently abandon the features that do not contribute to the classification process. The Information Gain Ranking filter is an appropriate method for attribute selection, established with the assistance of our data set. It completes a prompt and accurate analysis, and eliminates a significant number of variables, which are not influencing the ranking process and the performance of the classification step.

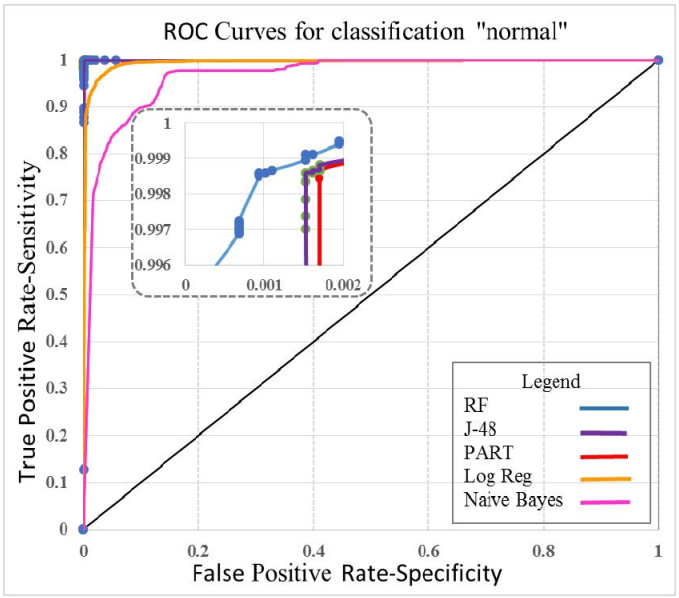
TABLE II. ATTRIBUTE SELECTION RESULTS

Select attribute method	Eliminated variables number	Total	Top 3 variables	
			Name	Number
Information Gain Ranking filter	20,22,18,17, 19,7,9,15,11,16 ,21	11	src bytes, service, flag	5,3,4
Correlation Attribute evaluation	18,20,7,21	4	same serv rate, srv serror rate, serror rate	29,26,25
Gain Ratio Feature evaluator	15,20,22,11, 19,18,7,16,9,17, 21	11	wrong fragment, root shell, dst host serror rate	8,14, 40
Relief Attribute Evaluation	18,20,7,21,5	5	flag,serror rate, srv serror rate	4,25, 26
Symmetrical Uncertainty Ranking Filter	20,22,18,17,19, 7,9,15,11,16,21	11	src bytes, flag, diff srv rate	5,4,30

The classification process is the next phase of our proposed procedure as described above. After we eliminate the variables that don't contribute to the model, the objective here is to classify the network as being under attack or not. Several discrimination models are tested and compared, and the best in terms of cost structure and accuracy is selected. Based on the Random Forrest (RF) classification method, subsequently, the suggested first stage is accomplished. The splitting criteria used at this stage is mainly based on reduction of the Mean Squared Error as the method is described above in part IV. Moreover, our investigation suggests that RF outperforms the other classification approaches. The results for the top six models on the first step are presented in Table III and the corresponding Receiver operating Characteristics Curves (ROC) are illustrated on Fig 3. The ROC curves assist us to recognize the tradeoff between sensitivity and specificity. The slope of the tangent line at the threshold points represents the likelihood ratio for that value of the test, whereas the area under the curve is a measure of accuracy. Out of the 127,734 networks, there are 23 attacks that the model is not able to recognize and classified them as normal and 8 normal connections that the model classifies as attack, but they are normal, in total 31 out of 127,734 instances were wrongly classified. We can observe that Random Forrest outperforms the other methods and the area under the ROC curve is almost 1, which signifies an excellent classification. The other two approaches, PART and J-48 also give exceptional results and we are able to employ them in the analysis. Support Vector Machines, with area under the ROC curve of .9746 and Logistic regression decreases the capacity of the classification process, although they exceed 95% accuracy. Only the Naïve Bayes is around 90%, which categorizes it as the least desired approach from the top six, but also a reasonable one.

J-48 and PART provide worthy results and take a reasonable amount of time (Fig 4), however, the results suggest that it's better to select RF at this stage as a classification method, because the accuracy is higher, the time to perform the classification is shorter and there is less chance of overfitting in its algorithm. Real world data inevitably contain noise - in either the feature values, the class labels, or both. ".

Fig. 3 Receiver Operating Characteristic Functions



The models are compared with regards to time and complexity. The results in seconds are presented on "Fig. 4" and Fig. 5 respectively. Although RF takes third place in the assessment of the time, it is still the preferred classifying technique that handles the tradeoff between accuracy and complexity. The complexity is a measure that is related to the informational structure of the data set and how long does it take for the model in seconds in order to evaluate the information. The complexity is measured in bits per second and implies computational effort.

We will apply the PART algorithm on the second stage of the classification process only if the network is under attack.

PART model on this step achieves superior results in comparison to RF in terms of accuracy and the model was able to perform better the classification process.

TABLE III. ACCURACY CLASSIFICATION PROCESS STEP 1

Model	RF	J-48	PART	SVM	Logistic Regression	Naïve Bayes
Correctly classified	99.88%	99.86%	99.85%	97.42%	96.76%	90.63%
Kappa Statistic	0.998	0.9971	0.9970	0.948	0.9348	0.8111
Mean absolute Error	0.0029	0.0019	0.0016	0.0258	0.0433	0.0939
Root mean squared Error	0.0325	0.0372	0.0384	0.1606	0.1553	0.3024
Relative absolute error	0.59%	0.39%	0.33%	5.18%	8.70%	18.87%
Root relative squared error	6.52%	7.46%	7.69%	32.19%	31.12%	60.61%

Fig. 4 Time in Seconds to Build a Model

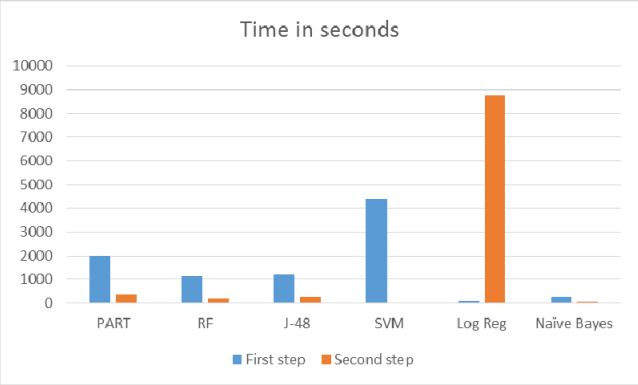
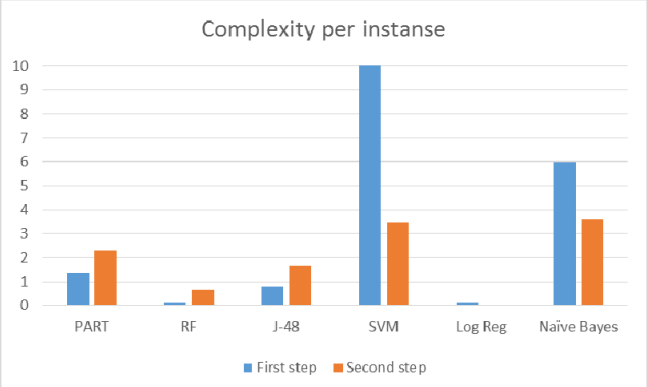


Fig. 5 Complexity Bits per Instance



Unlike PART, which is able to accomplish the classification process with an incomplete tree, RF needs additional resources for creating and evaluating an entire tree. PART employs pruning in the evaluation process. Therefore, PART benefits the reduction of the complexity and decreases the error. Compared to the J-48, the risk of overfitting the data in the evaluation process is minor.

TABLE IV. ACCURACY CLASSIFICATION PROCESS STEP 2

Model	PART	RF	J-48	SVM	Log Reg	Naïve Bayes
Correctly classified	99.75%	99.72%	99.72%	99.59%	99.51%	83.17%
Kappa Statistic	0.995	0.994	0.994	0.9923	0.9906	0.70559
Mean absolute Error	0.0002	0.0006	0.0003	0.0826	0.0011	0.0151
Root mean squared Error	0.0149	0.0153	0.0153	0.1999	0.0199	0.1186
Relative absolute error	0.524%	1.421%	0.678%	183.6%	2.41%	33.48%
Root relative squared error	9.862%	10.142%	10.18%	132.7%	13.19%	78.76%

The results for the second stage of the classification process are presented in Table IV. All of the methods slightly decrease their accuracy in terms of correctly classified type of attacks. However, once the "normal" networks are not taken under consideration, the SVM approach increases the number of the correctly classified instances. Regardless of the similar accuracy of the top six methods, we can observe that the Mean Absolute Error and especially the Relative Absolute Error are lesser for the PART model. Additionally, the accuracy for the PART model is higher and the time for the classification process of PART is reasonable.

VI. CONCLUSION

NASCA is a four-step Intrusion Detection procedure, which is created using machine learning techniques. Moreover, it is a superior method compared to the existing data mining models in network security. The time for performing the analysis is relatively short, and the accuracy is remarkable. The suggested classification process is capable to serve as a convenient and efficient protection tool for detecting an intrusion in our network. In order to reduce the vulnerability, it is essential to discover a sophisticated approaches such as discussed in [18] to defend our systems

VII. REFERENCES

[1] Bahl, Shilpa, and Sudhir Kumar Sharma. "Improving Classification Accuracy of Intrusion Detection System Using Feature Subset Selection." *2015 Fifth International Conference on Advanced Computing &Amp; Communication Technologies*, 2015.

[2] Breiman, Leo. "Machine Learning." *Machine Learning*, vol. 45, no. 1, 2001, pp. 5–32.

[3] Buehlmann, Peter, and Bin Yu. "Analyzing Bagging." *The Annals of Statistics*, vol. 30, no. 4, 2002, pp. 927–961.

[4] C, Lakshmi Devasena. "Effectiveness Evaluation of Rule Based Classifiers for the Classification of Iris Data Set." *Bonfring International Journal of Man Machine Interface*, vol. 1, no. 1, 2012, pp. 05–09.

[5] Cannady, James. “Distributed Detection of Attacks in Mobile Ad Hoc Networks Using Learning Vector Quantization.” *2009 Third International Conference on Network and System Security*, 2009.

[6] Forman, George. “Choose Your Words Carefully: An Empirical Study of Feature Selection Metrics for Text Classification.” *Principles of Data Mining and Knowledge Discovery Lecture Notes in Computer Science*, 2002, pp. 150–162.

[7] Frank, Eibe, and Ian H Witten. “Generating Accurate Rule Sets Without Global Optimization.” *Proceeding ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 144–151.

[8] Frank, Eibe et al. “Machine Learning.” *Machine Learning*, vol. 32, no. 1, 1998, pp. 63–76.

[9] Guyon, Isabelle, and André Elisseeff. “An Introduction to Feature Extraction.” *Feature Extraction Studies in Fuzziness and Soft Computing*, pp. 1–25.

[10] Hall, M.a., and G. Holmes. “Benchmarking Attribute Selection Techniques for Discrete Class Data Mining.” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, 2003, pp. 1437–1447.

[11] Joachims, Thorsten. “Text Classification.” *Learning to Classify Text Using Support Vector Machines*, 2002, pp. 7–33.

[12] John, George H., and Pat Langley. “Estimating Continuous Distributions in Bayesian Classifiers.” *Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 338–345.

[13] Kira, Kenji, and Larry A. Rendell. “A Practical Approach to Feature Selection.” *Machine Learning Proceedings 1992*, 1992, pp. 249–256.

[14] Kohavi, Ron, and George H. John. “Wrappers for Feature Subset Selection.” *Artificial Intelligence*, vol. 97, no. 1-2, 1997, pp. 273–324.

[15] Liu, Huan, and R. Setiono. “Feature Selection via Discretization.” *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, no. 4, 1997, pp. 642–645.

[16] Nguyen, David T. et al. “A Reconfigurable Architecture for Network Intrusion Detection Using Principal Component Analysis.” *Proceedings of the Internation Symposium on Field Programmable Gate Arrays - FPGA'06*, 2006.

[17] Nziga, Jean-Pierre, and James Cannady. “Minimal Dataset for Network Intrusion Detection Systems via MID-PCA: A Hybrid Approach.” *2012 6th IEEE INTERNATIONAL CONFERENCE INTELLIGENT SYSTEMS*, 2012.

[18] Ramachandran, Kandethody, and Zheni Stefanova. “Dynamic Game Theories in Cyber Security.” *Proceedings of Dynamic Systems and Applications*, vol. 7, 2016, pp. 303–310.

[19] Rong, Yang, and Zheng. “Classification and Regression Trees, Random Forest Algorithm.” *Machine Learning Approaches to Bioinformatics Science, Engineering, and Biology Informatics*, 2010, pp. 120–132.

[20] Sahu, Santosh Kumar. “A Study of K-Means and C-Means Clustering Algorithms for Intrusion Detection Product Development.” *International Journal of Innovation, Management and Technology IJIMT*, 2014.

[21] Tavallaee, Mahbod et al. “A Detailed Analysis of the KDD CUP 99 Data Set.” *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[22] Vapnik, Vladimir N. “Constructing Learning Algorithms”, *The Nature of Statistical Learning Theory, 1995,* pp.119-166.

[23] Wohlrab, Lars, and Johannes Fürnkranz. “A Review and Comparison of Strategies for Handling Missing Values in Separate-and-Conquer Rule Learning.” *Journal of Intelligent Information Systems*, vol. 36, no. 1, 2010, pp. 73–98.

[24] Xu, Xin, and Tao Xie. “A Reinforcement Learning Approach for Host-Based Intrusion Detection Using Sequences of System Calls.” *Lecture Notes in Computer Science Advances in Intelligent Computing*, 2005, pp. 995–1003.