Evaluation of Ensemble Design For Intrusion Detection

Assumani Shabani Serge\textsuperscript{1} and Ms. Shrutika Suri\textsuperscript{2}

\textsuperscript{1}Research Scholar, CSE, FET, MRIU, Faridabad
\textsuperscript{2}Assistant Professor, CSE, FET, MRIU, Faridabad

Abstract—Intrusion Detection System (IDS) examine large quantity of data with the intent to find malicious event among them. A misclassification of what is examine can lead to a higher rate false positive or false negative, thus any model built for the task must be as accurate as possible. The purpose of this study was observe how applying ensemble method can affect the performance of a classifier. We worked with: Adaptive Boosting (AdaBoost) and Bagging Classifier as ensembles on different classifiers, namely: Naïve Bayes and Decision Tree (CART).

Algorithm were trained and test on the NSL-KDD dataset and the F-score, precision and recall were taken as base metric to evaluate how well algorithms performed.

We notice that while ensemble method in fact improve the performance of week classifier, this sometimes comes with little cost. When looking at multiple metrics for improvement, one metric might get improved while the other one decreases.

Index Terms—Ensemble method, classification, attack, Intrusion Detection, NSL-KDD

I. INTRODUCTION

The increase in communication and data exchange has generates a proportional growth of data storage, and requirement for faster network to do the bridge between end-systems. All those interconnections, data exchange and remote access are convenient not only to authorized users but also for attackers that might be interested in any kind of malicious activities. In other world being open to the outside brings advantages that comes with vulnerabilities. Among the solution to mitigate that problem, Intrusion Detection Systems (IDS) play a big role stated as second line of defense. IDS are mainly classified into knowledge based and anomaly based. Both knowledge based and anomaly based are elaborated using different approaches\cite{1}, \cite{2}. Knowledge based IDS relies on pattern called signature that is match against a packet in order to detect an intrusion, while anomaly based IDS depend on building a profile from which any deviation is taken as anomaly.

Anomaly detection is done through different techniques, namely: statistical, classification, clustering and information theory \cite{3}. Among those technique used for anomaly detection, classification is one where labeled data are provided to a classifier, so that it can build boundaries between the different classes. The only issue here is that an IDS build around a classification model can only be as good as the capability of the classifier model to correctly perform the classification. A model with poor performance increases the rate of false positive and false negative.

Multiple researches in the area have opted for different technique based on preprocessing and algorithms, such as feature selection, resampling, parameter tuning and ensemble learning.

This study takes the ensemble learning approach to classification optimization in consideration.

II. LITERATURE REVIEW

Multiple researches on intrusion detection systems using different algorithms associated with various methods with the intent to improve accuracy and reduce false positive rate had been proposed. Ensemble learning is one of the method proposed by those researches.

Sornsawit & Jaiyen \cite{4} had use correlation based feature selection in conjunction with Ada Boost algorithm applied to Decision Tree, Naïve Bayes, SVM, and MPL classifier in order to improve detection accuracy.

Experiment was performed on KDD CUP 99 with the help of WEKA mining tool and MATLAB. The feature selection processes had led to 5 relevant features selected, namely: dst_byte, root shell, num_file_creations and is_host_login.

The experiment consisted of two stage; in first stages the individual classifiers and respective ensembles were trained and tested using the entire set of features.
while in the next stage only the selected features were considered.

The experiment as deducted that AdaBoost used we feature selection did in fact improve the result of the different classifiers. The observations were that Naïve Bayes and MPL had the higher sensitivity, however, Decision Tree offered the least performance because both attack types contained small data and had long period time pattern.

Woźniak et al[5] had summarized the main research stream on ensemble classifiers. They exposed different concept that are ensemble design, fuser design, and concept drift, to then state the applications of ensemble classifier. The key issue related to the problem was under the consideration of classifiers diversity and method of combination. According to their observations multiple classifiers systems show in most instances improved performance, resiliency and robustness to high data dimensionality and diverse form of noise, such as labeling noise.

Mukkamala et al. [6] showed that an ensemble composed of different types of ANN, SVM with radial basis function (RBF) kernel and MARS combined with the bagging techniques produce better performances than should be expected with a classifier taken alone. The variants of ANN were: resilient back propagation (RP), scaled conjugate gradient algorithm (SCG) and one-step-secant algorithm (OSS).

A subset of DARPA 1998 with 11,982 observations randomly selected from the original dataset, in which representations of each class was proportional to the size of the class except for the smallest class which was included entirely. The extracted subset was further divided into training and testing set, having respectively 5,092 and 6,890 observations each. This dataset was then divided into a training set of 5,092 examples and a test set of 6,890 examples.

Five binary classifiers of type SVM and MARS were assigned to perform the classification task with the actual target being taken as normal against all the attacks labeled as rest.

The result obtained by each individual algorithm was compared to those obtained respective ensembles. The study as exposed that SVM used alone outperforms the other single algorithm but is totally outperformed by the ensemble of ANN, SVM and MARS. This ensemble surprisingly obtained a 100% accuracy on the test set for the R2L class. However, the researchers warn that some of these results might not be statistically significant because of the unbalanced dataset used.

Zainal et al[7] used Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF) for ensemble models to show an empirical improvement in decision accuracy for the different attacks classes: Probe, DoS, U2R, R2L. The experiment was performed on KDD Cup 99. Only a subset of the KDD Cup 99 with 11,982 observation divided into training and testing data with respectively of 5,092 and 6,890 records respectively was taken into account. As for feature selection, Rough-Discrete Particle Swarm Optimization (RoughBPSO) was used. After the selection process, 15 features were selected.

The experiment showed that by assigning proper weight to in ensembles, deduction accuracy was improved. While Linear Genetic Programming had performed well in all classes except for U2R, Random Forest had better true positive rate for U2R.

Chebrolu et al[8] investigated the performance of two feature selection algorithms involving Bayesian networks (BN) and Classification and Regression Trees (CART) and an ensemble of BN and CART. Experiment was done on a subset of KDD Cup 99 that resulted into training and testing set, having respectively 5,092 and 6,890 observations each.

The initial experiments using Principal Component Analysis and Independent Component Analysis to compress data did not result in satisfactory feature reduction regarding intrusion detection. But the feature selection method using Markov blanket model and decision tree analysis gave better results. The ensemble approach was used for the 12, 17 and 41 variable data sets. The final output were decided on basis of given weight to respective classifier depending on generalization accuracy.

The proposed ensemble of BN and CART combined the complementary features of the base classifiers. The result showed that using the hybrid model Normal, probe and DOS could be detected with 100% accuracy and U2R and R2L with 84% and 99% accuracies.

III. CLASSIFICATION

Classification is the process of assigning a class label to unclassified object(s) based on a set of defined
The class having the highest posterior probability will be the designated one for the observation.

b) Classification And Regression Tree

Decision tree algorithms work using the “divide-to-conquer” approach. The tree representation is issue from the splitting based on observed features. While some decision tree algorithms compare the predictors with constant, other compare them to each other or use some function of one or more attributes[11]. The term CART stands for Classification And Regression Tree and was introduced in [12] to refer to decision trees. CART uses a binary decision tree representation. With \( \hat{y} \) being the outcome of the classification and \( x_1,x_2,...,x_n \) the predictors that form a vector \( X \), CART algorithm proceed by selection one predictor \( x_a \) from \( X \) and a value \( v \) taken by the selected predictor in the observations. In case the value is of type continuous, the algorithm might simply split the current node into two subsequent node, where, on one side \( x_a < v \) and the other \( x_a \geq v \). For a categorical attribute, let say \( x_b \in \{a,b,c,d\} \), the split is done by dividing the values into two subset: \( a \) and \( b,c,d \), \( b \) and \( a,c,d \), \( c \) and \( a,b,d \) \( d \) and \( a,b,c \). This partitioning process is done reclusively until the tree is built.

The Gini index is the name of the cost function used to evaluate splits in the dataset, in other words it tells how good a split is. A perfect separation results in a Gini score of 0, whereas the worst case split that results in 50/50 classes in each group results in a Gini score of 1.0 (for a 2 class problem).

The Gini index impurity is calculated as following:

\[
G_i(t) = 1 - p(t)^2 - (1 - p(t))^2
\]

Where \( p(t) \) is the (possibly weighted) relative frequency of the first class in the node. The gain generated by the split of the parent node \( P \) into left and right children is:

\[
I_C(P) = G_i(P) - qG_i(L) - (1-q)G_i(R)
\]

Where \( q \) is the fraction of the instance going left[10].

IV. ENSEMBLE METHODS

Ensemble method make a combination of trained classifiers to improve the stability and predictive power of the model.
a) Bagging
Bootstrap aggregation state as bagging was introduced by Leo Bierman. The starting point of bagging is to choose the base learner. Bagging works by implementing similar learner (copies of the base learner) on randomly selected elements from a data set to the then estimate the mean of all the predictions. The sets randomly selected element are referred to as bootstrap samples. Bootstrap samples are generated by uniformly sampling m instances from the provided training set with replacement [13]–[15].
The main aim for bagging is to reduce overfitting. However, bagging methods such as random subspace manipulates the feature space. This method randomly selects feature subsets from the feature space and trains each classifier on a different feature subset.
Bagging adopts the most popular strategies for aggregating the outputs of base learners, i.e. voting for classification and averaging for regression. To predict a test instance, taking classification, for example, bagging feeds the instance to its base classifiers and collects their outputs, and then votes the labels and take winner label as prediction, where ties are broken arbitrarily.

b) Boosting
Boosting is an iterative technique which adjusts the weight of an observation based on the earlier classification. It tries to increase the weight of an observation in the case where this last was misclassified and vice versa. Generally speaking, boosting decreases the bias error and builds strong predictive models. However, they may sometimes overfit the training data.
In simple words, the overall aim of boosting method is to produce classifiers that are able to correctly classify observations for which the previous classifiers predictions were poor.
In theory, Bagging is good for reducing variance (Over-fitting) whereas Boosting helps to reduce both Bias and Variance[16].

c) Stacked generalization
Stacked generalization is also referred to as blending. It defers from the traditional ensemble learning techniques in a way that it does not used methods like voting, averaging, voting, weighted voting, etc.
In stacked generalization, the final output is found by taking the output of the base classifiers as training data for another classifier to approximate the same target function.

V. NSL-KDD
The DARPA Dataset for Intrusion Detection Evaluation (KDD Cup 99) has appeared to be one of the widely used dataset for evaluation of IDSs among the few publically available dataset for the purpose.
The NSL-KDD data set is a suggested data set that solves the inherent issues in KDD Cup 99. All the redundant record had been removed and it comes split into training and testing set. The reduced amount of record makes it useable with less memory requirement.
The database contains a wide range of attacks that are aggregated into four main categories [17]:
- Denial-of-Service attacks (DoS): attacks attempt to interrupt or degrade a service that a system provides to its intended users by overloading the system resources.
- Probe: occurs when an attacker with the tries to gather information about the system or network. Probe attacks scan for system vulnerabilities which are usually exploited for other further attacks.
- Remote-to-Local (R2L): occurs when an attacker who obtained access to the network try to gain user privileges through any account in the network or system.
- User-to-Root (U2R): occurs when a person with user privileges tries to acquire root user privileges. This is basically some privilege escalation.

Each observation in the dataset in made of 41 features.

<table>
<thead>
<tr>
<th>No</th>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>duration</td>
<td>Continuous</td>
</tr>
<tr>
<td>2</td>
<td>protocol_type</td>
<td>Discrete</td>
</tr>
<tr>
<td>3</td>
<td>service</td>
<td>Discrete</td>
</tr>
<tr>
<td>4</td>
<td>flag</td>
<td>Discrete</td>
</tr>
<tr>
<td>5</td>
<td>src_bytes</td>
<td>Continuous</td>
</tr>
<tr>
<td>6</td>
<td>dst_bytes</td>
<td>Continuous</td>
</tr>
<tr>
<td>7</td>
<td>land</td>
<td>Discrete</td>
</tr>
<tr>
<td>8</td>
<td>wrongFragment</td>
<td>Continuous</td>
</tr>
<tr>
<td>9</td>
<td>urgent</td>
<td>Continuous</td>
</tr>
<tr>
<td>10</td>
<td>hot</td>
<td>Continuous</td>
</tr>
<tr>
<td>11</td>
<td>numFailedLogins</td>
<td>Continuous</td>
</tr>
<tr>
<td>12</td>
<td>loggedIn</td>
<td>Discrete</td>
</tr>
<tr>
<td></td>
<td>Feature</td>
<td>Type</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>13</td>
<td>num_compromised</td>
<td>Continuous</td>
</tr>
<tr>
<td>14</td>
<td>root_shell</td>
<td>Discrete</td>
</tr>
<tr>
<td>15</td>
<td>su_attempted</td>
<td>Discrete</td>
</tr>
<tr>
<td>16</td>
<td>Num_root</td>
<td>Continuous</td>
</tr>
<tr>
<td>17</td>
<td>num_file_creations</td>
<td>Continuous</td>
</tr>
<tr>
<td>18</td>
<td>num_shell</td>
<td>Continuous</td>
</tr>
<tr>
<td>19</td>
<td>num_access_file</td>
<td>Continuous</td>
</tr>
<tr>
<td>20</td>
<td>num_outbound_cmds</td>
<td>Continuous</td>
</tr>
<tr>
<td>21</td>
<td>is_host_login</td>
<td>Discrete</td>
</tr>
<tr>
<td>22</td>
<td>is_guest_login</td>
<td>Discrete</td>
</tr>
<tr>
<td>23</td>
<td>count</td>
<td>Continuous</td>
</tr>
<tr>
<td>24</td>
<td>srv_count</td>
<td>Continuous</td>
</tr>
<tr>
<td>25</td>
<td>serror_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>26</td>
<td>srv_serror_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>27</td>
<td>srv_error_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>28</td>
<td>rerror_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>29</td>
<td>same_srv_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>30</td>
<td>diff_srv_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>31</td>
<td>srv_diff_host_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>32</td>
<td>dst_host_count</td>
<td>Continuous</td>
</tr>
<tr>
<td>33</td>
<td>dst_host_srv_count</td>
<td>Continuous</td>
</tr>
<tr>
<td>34</td>
<td>dst_host_same_srv_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>35</td>
<td>dst_host_diff_srv_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>36</td>
<td>dst_host_src_port_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>37</td>
<td>dst_host_srv_diff_host_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>38</td>
<td>dst_host_error_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>39</td>
<td>dst_host_srv_serror_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>40</td>
<td>dst_host_reerror_rate</td>
<td>Continuous</td>
</tr>
<tr>
<td>41</td>
<td>dst_host_srv_reerror_rate</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

Table I. Features of NSL-KDD Dataset

VI. EXPERIMENTATION SETUP

Classifier had been trained on KDDTrain+ and tested on KDDTest+. The two data sets are from different distributions and highly imbalanced, this makes the classification harder.

As said before, multiples attacks are represented in the datasets, we further agrrated into them into 4 classes for analysis purpose.

A two by two correlation based approach with a threshold of 0.95 had been used to find highly correlated features. From those features, a selection based on the best correlation with the target had been used to drop features introducing duplication of information.

The nominal columns, namely, protocol type, service, and flag had their values transformed into dummy data and additional care had been taken to make the possible values taken by the protocol type in the train dataset correspond with those taken in the test dataset.

The encoding made the number of features increase to 116 features, and from those features sequential forward feature selection had been applied to reduce the feature space to 30 features. Only those 30 features had been used through all the experiment.

In total 38 attacks are represented with only 24 in the training set and additional 14 attacks in the testing set only.

VII. RESULTS AND DISCUSSION

a) Naïve Bayes

From the false negative rate perspective, Naïve Bayes had offer the lowest with 0.17, with a slight difference over its bagging version that has 0.22, and the boosting version that has the worst rate of 0.31. Nevertheless, the false positive rate was decreased by both bagging and boosting, and the F-score is increased.
In an overall perspective, an improvement had been introduced by the ensemble method. Bagging on Naïve Bayes provides the best result with an f-score of 0.74.

b) Cart
CART algorithm performances were improved by both Bagging and AdaBoost for all the metrics. AdaBoost had shown the best improvements by increasing the precision and recall by respectively 0.06 and 0.07 which made the f-score raise to 0.76 as compared to 0.69 provided by CART used alone. Bagging applied to CART had shown an improvement of 0.05 on the precision to offer a score 0.70. The CART algorithm is stable as all the metric are either improved or kept at the same level.

c) Discussion
From a single metric point of view, it is easy to figure out if we have really obtained an improvement after ensemble method was applied. But when more than one metric is taken, it becomes a little complex. For instance, applying bagging and AdaBoost to Naïve Bayes had resulted in an increase of precision and f-score but a decrease in recall due to higher false negative rate. Hence, it might good to set some priorities on the chosen metrics and then make the deduction in a balanced manner.

We have noticed in most of the case, precision is higher than recall, this is due to the fact that the imbalanced attacks distribution in KDDTrain+ and KDDTest+ hardens the correct classification of classes that are less represented. During the training phase, the learners sometimes only see the most represented classed and tend to ignore the others. Among all base learners used, CART algorithm has proven to be the most stable for all the metrics and the one that has provided the best f-score value of 0.76 when combined it with AdaBoost.

VIII. CONCLUSION
We had investigated how boosting and bagging could improve the performance of weak classifiers as well as how those classifiers performance used alone. The algorithms chosen were Classification and Regression Tree and Naïve Bayes as weak classifiers; Ada Boost and Bagging algorithms as ensemble method.

We noticed in most of the case that some improvement had been introduced by ensembles, but a cost was also introduced in the overall result. Bagging and Boosting applied on logistic regression had a negative effect on the performance. From the result obtained in each part of the experiment, it could be concluded that the best result was provided by applying Ada Boost to CART with an f-score of 0.76. Ensemble methods are not an ultimate solution, even if they improve in fact the quality of weak classifiers, the best way to get more accurate result is to combine them with different techniques such as feature selection in order to produce a more robust machine learning classifier.

REFERENCES
[9] Ming-Syan Chen, Jiawei Han, and P. S. Yu, “Data mining: an overview from a database


