Anomaly-Based Network Intrusion Detection: An Approach Using Ensemble-Based Machine Learning Algorithm

Kashif Gul Chachar[1†] and Syed Nadeem Ahsan[2††],

Department of Computer Science IQRA University, Karachi Sindh, Pakistan

Abstract
With the seamless growth of the technology, network usage requirements are expanding day by day. The majority of electronic devices are capable of communication, which strongly requires a secure and reliable network. Network-based intrusion detection systems (NIDS) is a new method for preventing and alerting computers and networks from attacks. Machine Learning is an emerging field that provides a variety of ways to implement effective network intrusion detection systems (NIDS). Bagging and Boosting are two ensemble ML techniques, renowned for better performance in the learning and classification process. In this paper, the study provides a detailed literature review of the past work done and proposed a novel ensemble approach to develop a NIDS system based on the voting method using bagging and boosting ensemble techniques. The test results demonstrate that the ensemble of bagging and boosting through voting exhibits the highest classification accuracy of 99.98% and a minimum false positive rate (FPR) on both datasets. Although the model building time is average which can be a tradeoff by processor speed.

Keywords:
Seamless, Intrusion detection, Anomaly, Ensemble, J48, Reptree, Naïve-byes, Support Vector Machine (SVM).

1. Introduction

With the advancement of Information and Technology, Network usage has become a key necessity for everyone thus making network security a critical issue for network users as well as service providers, thus everyone is trying to secure their data while it's traveling on the network in the form of packets or on their computer in the form of data. The financial losses caused by cybercrimes are increasing rapidly. The most catastrophic digital crimes are those caused by malicious attackers, denial of services, and web-based attacks. Organizations can lose their intellectual property with such malevolent software intruded into their network, which may lead to intrusion in the country's critical system. One of the focused techniques to avoid cybercrime is to detect the attack process early [1]. Cyber-crimes can be due to malicious stuff users, or in the form of denial of services, or a web-based attack through IP spoofing or phishing. In this connection organization and enterprises implements firewalls, anti-virus software, and Intrusion detection systems. An intrusion can be characterized as "any collection of activities that endeavor to bargain the uprightness, privacy or accessibility of network assets" [2]. IDS plays an essential role in the computer and network data security field.

Network Intrusion Detectors are designed to distinguish the malicious use of computer networks, violation of security rules, intruders, and initiate the necessary action against them. The network Intrusion detection systems are partitioned into two different categories [3]. Signature (misuse) detection is the type of NIDS which searches for the pre-defined attack patterns in network data, and when the pattern matches, the system flags it as an attack or intrusion. It maximizes the speed of attack detection and minimizes the rate of false alarms. However, it is unable to find non-predesignated or new attacks. Hence unable to detect the new attacks [4]. And the second type is an anomaly, which is a situation that is suspicious from the perspective of security [5]. Anomaly-based NIDS stores the patterns and features of the user's normal network usage into the database and then keeps on comparing the user's usage with those patterns stored in the library and if any irregular behavior or anomaly is detected in network traffic then the alarm is generated. It maximizes the possibility of detecting newly created attacks that were never detected before but increases the percentage of false-positive alarms (FPR) due to the versatility of the network traffic and the behavior of the user keeps on changing. The anomaly-based NIDS first learns the characteristics of normal activities, and when it detects traffic that deviates from the normal activities [6]. Current anomaly detection methods are mainly classified as statistical-based, cognition-based, and Machine Learning based [7]. Three main ensemble methods are Bagging, Boosting, and Stacking. In this paper, we propose an effectual and impressive NIDS architecture that is built on a novel ML approach called "ensemble of ensembles", by applying voting on very famous ensemble techniques called bagging and boosting to maximize the classifier's detection accuracy and minimize the number of false positives. For the improvement of an exact and powerful anomaly identification model, we used several statistical and graphical data analytical methods on our selected data sets. The results of our experiments drove us to choose the ensemble technique using voting with bagging and boosting and some non-ensemble algorithms like j48, Support Vector

Manuscript received January 5, 2024
Manuscript revised January 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.1.12

Machine (SVM), Naïve Byes, and Reptree due to the non-linear nature of the selected dataset.

The performance of our proposed methodology is assessed and contrasted with the traditional classification approaches by performing various experiments with the KDD Cup 99 and NSL-KDD data sets. The benchmark datasets KDD Cup 99 (10%) by DARPA and NSL-KDD are the most common source of training and testing NIDS models. Both datasets contain 41 features. NSL-KDD comprises chosen records of the total KDD dataset and does not suffer from any deficiencies like redundant records and duplication [8]. Table 1 specifies the details of the KDD Cup 99 and NSL-KDD dataset, concerning the number of instances per class belonging to each type of attack or normal. No previous work has introduced such a system to our knowledge framework that combines bagging and boosting using a voting technique using this dataset.

This proposed NIDS model is a combination of bagging and boosting using a voting technique that finds classes with rare instances in the dataset and it can detect newly designed attacks easily. Our model forms two sub-ensembles bagging+ and boosting+ which are made up of a combination of four base classifiers such as J48, Naïve Byes, Reptree, and SVM. The efficacy and usefulness of the proposed approach are measured using different metrics such as accuracy, precision, recall, F-measure, and false alarm rate. The paper's key contributions are:

- A review of the actions and characteristics of the KDD Cup 99 and NSL-KDD dataset.
- A proposal for using synthetic minority oversampling technique to generate synthetic data samples for minority classes in the dataset.
- In our understanding, the first study on the ensemble of ensembles using the voting technique on bagging and boosting classification model for anomaly detection using two benchmark computer networking datasets.

The results achieved from sub-ensembles are further processed to form the final classifiers VBB+. The experiment results in achieving the reduction in FP rate up to 0.001% and maximizing the accuracy of NIDS up to 99.98% using the proposed model VBB+. The research adds its part in the field of cybersecurity with the proven results achieved.

The rest of this paper is structured as follows: Section II provides a brief literature review of research work that has been done so far. Section III begins with a brief discussion about Machine Learning and usage of the ensemble of learners (bagging and boosting techniques). Section IV presents the methodology and experimental setup of the proposed NIDS model. Section V presents the experimental results and briefly concludes in section VI.

Table 1:Class-wise details of datasets for each attack type

Datasets		***KDD cup 99 10 percent corrected***	***NSL-KDD Dataset***
Class	Total instances	494025	125973
	Feature	41	41
	Type	23	23
Normal	Normal	97278	67343
DOS	land	21	18
	back	2203	956
	pod	264	201
	smurf	280790	2646
	neptune	107201	41214
	teardrop	979	892
Probe	ipsweep	1247	3599
	portsweep	1040	2931
	satan	1589	3633
	nmap	231	1493
R2L	ftp_write	8	8
	guess_passwd	53	53
	imap	12	11
	multihop	7	7
	phf	4	4
	spy	6	2
	warezclient	1020	890
	warezmaster	20	20
U2R	buffer_overflow	30	30
	loadmodule	9	9
	perl	3	3
	rootkit	10	10

2. Literature Review

In this section, the study surveyed some related work, concentrating on anomaly-based machine learning methods for network intrusion detection.

Recent research carried on real credit risk applications reveals that conventional single-prediction models are less predictive and less robust than ensemble models, especially in large or high-dimensional data sets [9]. Machine Learning's wide variety of algorithms such as the Genetic Algorithm, Bayesian Belief Network, and Hidden

Markov Model, Artificial Neural Network, and Clustering method had been used for the implementation of the intrusion detection system. The coalescence of various base ML classifiers is known as the ensemble method.

The literature review observed that the ensemble method proves to be better for the reduction of false-positive alarm in an anomaly-based network intrusion detection system [10]. In [11] the authors have reviewed various recent works on machine learning (ML) methods that leverage SDN to implement NIDS. More specifically, they evaluated the techniques of deep learning in developing SDN based NIDS and concluded that random tree model NIDS holds high accuracy and low false alarm rate. In [12] the authors have proposed an unsupervised Machine Learning strategy to characterize traffic and distinguish application in the system. They have utilized an element choice procedure to discover the ideal arrangement of stream qualities. This measurable property of stream is utilized for characterization and distinguishing proof of packets in the network. The impact of various properties on the learning is additionally decided. In [13] the authors introduced a novel intrusion detection technique. The authors utilized the Bagging technique along with REPTree as the base classifier to execute the interruption discovery framework. In [5] a survey of the most notable irregularity-based system interruption discovery methods and enrolled the present stages and frameworks a work in progress and illustrated primary difficulties to be managed. In [14] authors have utilized Association Based Classification for planning the interruption identification framework. The speed of Apriori calculation is expanded by decreasing things included. In guideline enlistment with no data misfortune. The fluffy affiliation rules are utilized to manufacture expressive models of various classes. The proposed classifier is effective for grouping of extensive data set and can handle the representative qualities. In [15] Authors utilize a shrewd framework to boost the acknowledgment rate of system assaults by inserting the fleeting conduct of the assaults into a TDNN neural organized structure. The proposed framework comprises of a few modules for packet catching, pre-processing, design acknowledgment, characterization, and checking and cautioning and tried the framework and appeared great outcomes. In [16] Authors have presented a detailed survey of intrusion detection based on various techniques and are classified as follows into neural networks, support vector machine, K-means classifier, and hybrid technique. For a comprehensive analysis, detection rate, time, and false alarm rate from various research papers. In [17] have proposed a framework for test assessment of classifier for interruption recognition framework. Authors have tended to one class v-SVM classifier utilizing RBF kernel. They have proposed the inclusion of the reproduced tests into the preparation information to enhance the security of discriminative SVM classifiers. In [18] Authors, IDS solution using ensemble learning is presented. Authors executed a Bayesian system and Random Tree as base classifiers alongside meta-learning calculations Random Committee and vote. To assess the model's execution, the KDDcup99 dataset is utilized. In [19] authors presented a NIDS combined with a supervised and unsupervised clustering method.

The solution bunches a named preparing informational index into various groups to exhibit the profile of typical and abnormality parcels. The directed grouping is utilized to test the object with the bunch profiles for marking them. The Authors of [20] have developed the intrusion detection system using the min-max normalization method. The interruption information of KDD 99 is standardized before going to the SVM. They found that standardization can accelerate computation and get a decent execution classifier. They have broken down and recreated a portion of the standardization techniques in this paper. They decide that min-max standardization has better precision, great execution in speed than other standardization strategies. The Authors of [21] have proposed decision tree induction is one of the classification algorithms in data mining. The Classification calculation is inductively figured out how to develop a model from the pre-ordered informational index. Every datum thing is characterized by estimations of the properties. The arrangement might be seen as the mapping from a lot of ascribes to a specific class. The decision tree orders the given information thing utilizing the estimations of its traits. In [22] authors have utilized a gathering boosting approach with a versatile sliding window for interruption discovery. Authors of [23] have executed an upgraded steady HMM stochastic process for interruption recognition framework. In [24] authors utilized a few group classifiers from the regulated learning classification. They assessed Bagged trees, AdaBoost, RUSBoost, LogitBoost, and GentleBoost calculations, and gave an investigation of the execution of the classifiers, and looked at their learning abilities, taking for the reference UNSW-NB15 dataset. In [25] the performance of the approach is evaluated and compared to the traditional classification approach by conducting different experiments with the Kyoto 2006+ dataset that was built of network traffic collection from honeypots in Kyoto University. The proposed framework combines unsupervised and supervised machine learning approaches tested on the Kyoto dataset and results in the quadratic discriminant analysis (QDA) as the most effective among support vector machines (SVM), k nearest neighbors (kNN), random forests (RF), with the accuracy of 94% on training and 82% on testing.

3. Machine Learning and Ensemble of Classifiers

The term machine learning explains itself as a mechanism to teach and train machines. The area of Machine learning (ML) is committed to creating frameworks that can consequently gain from the datasets [24] and distinguish hidden anomalies and patterns without being expressly modified to do so. ML algorithms are classified by the learning style they utilize and by the useful likeness of how they function [26]. Machine learning systems are viewed as proficient techniques to enhance identification rate, diminish the rate of false alerts, and meanwhile, reduce calculation and correspondence cost [27]. The machine learning methodologies can be classified into administered, unsupervised learning, and semi-supervised learning [28].

In supervised learning, the machine is first trained on the labeled input data to learn the patterns against various classes, and then it is implemented to predict the unknown patterns. Commonly used algorithms are support vector machine (SVM) and Random Forest. In unsupervised learning, the machine is trained on unlabeled data, the model learns the patterns from the structure and representation of the data. The objective of an unsupervised learning calculation is to demonstrate the crucial structure or conveyance in the information to anticipate obscure information [24]. Calculations utilized are bunching methods like essential part examination (PCA) and self-sorting out guide (SOM). Semi-Supervised is the type of learning that the machine is trained on a major portion of unlabeled data and a small portion of labeled data. Spectral Graph Transducer and Spectral Graph Transducer and Gaussian Fields approach, used to identify obscure attacks, and one semi-regulated grouping technique MPCK-implies used to enhance the execution of the recognition framework [29].

3.1 Ensemble of Classifiers

The proposed model is built on an ensemble of learners which allows utilizing a similar learning algorithm multiple times however train every leaner on the various arrangement of the data selected from an original set of data with replacement, this technique is known as bootstrap aggregation or bagging [30]. Given a standard training set *D* of size *n*, bagging generates m new training sets *Di*, each of size n', by sampling from *D* uniformly and with replacement. By sampling with replacement, some observations may be repeated in each *Di*. If $n' = n$, then for large *n* the set *Di* is expected to have the fraction *(1 - 1/e) (*63.2%)* of the unique examples of *D*, the rest being duplicates [31]. After the subsets *Di* formation, each subset is used to train a different model *M* using the same learning algorithm. We will have *Mi* different models each one trained on the different subset of data *Di* learning algorithm. Finally, we will query each *Mi* with *X* and collect the all *Yi* output of each model, and calculate *Y(mean)*. The bagging technique is suitable for bias reduction by selecting random data samples with the replacement but there are chances that some data samples are selected multiple times and some are never selected.

$$\overline{f}(x) = sign(\sum_{i=1}^{T} f_i(x)) \quad (1)$$

On the other hand, the boosting ensemble technique tries to improve learner's performance by focusing on the areas where the system is not performing well by assigning weights to those instances which are classified wrong by the previous learning model. AdaBoost is a famous algorithm using this approach. Given a standard training set *D* of size *n*, boosting generates new training set *D1*, of size n', by sampling from *D* uniformly and with replacement were $n' < n$ and train model *m1*. Then complete training set *D* is used to test the model *m1*, and we will discover that some of the instances are misclassified with some significant error. Now another subset *D2* is built from training set *D* of size n' but each instance is weighted according to the error rate of the previously misclassified instances so that those instances previously misclassified have more probability to be selected in *D2* and again a model *m2* is trained, tested and continues until it reaches subset m. Boosting is very suitable for reducing biases. Figure: 1 provides the graphical representation.

$$\overline{f}(x) = sign(\sum_{i=1}^{T} \alpha_i f_i(x)) \quad (2)$$

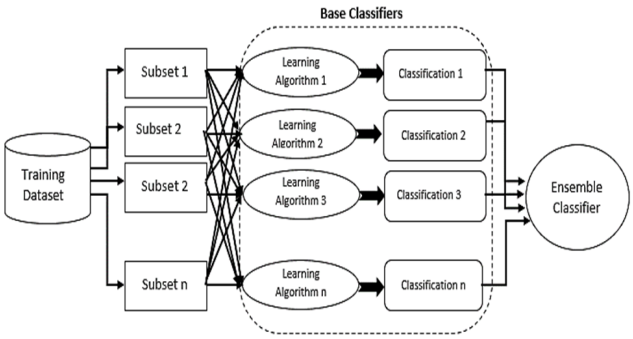

Figure 1: Ensemble of Learners

3.2 Classifier Performance Measures

The four most important statistical measures used in the study are the number of true positive instances (TP), which represents the number of correctly classified positive instances on the test set. False Positive (FP), number of instances misclassified as the positive class. Precision is the number of instances retrieved correctly belonging to any

class out of the total instances retrieved and ROC values that are considered as more important than accuracy rate. ROC is the most widely recognized assessment technique for giving an extensive estimation of the classifier exhibitions by comparing the detection performance of different algorithms. The classification accuracy can be achieved from the confusion matrix that represents the number of correctly and incorrectly classified instances for each class of the dataset.

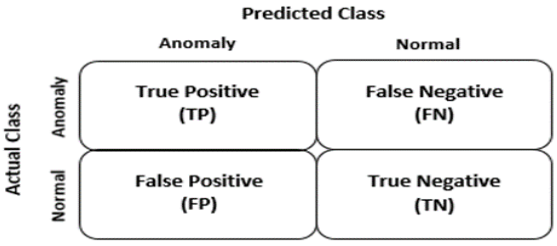

Figure 2: Confusion Matrix

Confusion Matrix provides the important values which can be used for calculating the performance of the classifier, such as accuracy rate (AC), sensitivity, specificity, precision, and Receiver Operating Characteristics (ROC). We will evaluate our proposed model based on accuracy rate (percentage of correctly classified instances), minimum numbers of false-positive instances, and ROC curve.

Dealing with the Imbalance Dataset refers to the situation when the numbers of instances in one class are much lesser than that of others than we can apply SMOTE (synthetic minority oversampling technique) to balance the classes and for these kinds of datasets classifiers, ROC values are considered as more important than accuracy rate. ROC is the most widely recognized assessment technique for giving an extensive estimation of the classifier exhibitions by comparing the detection performance of different algorithms. ROC graph is developed by plotting the true positive rate against the false-positive rate at various threshold settings. The sensitivity or recall (TPR) is the probability of the instances belonging to any particular class detected correctly while specificity is the number of instances belonging to a negative class detected as negative by the classifier (TNR). The positive predicted value (Precision) for any particular class is the percentage of the instances that were classified as a specific class and that are truly correctly classified.

- Precision = (Number of documents retrieved that are relevant) / (Total number of documents that are retrieved)
- Recall = (Number of documents retrieved that are relevant)/ (Total number of relevant documents)
- F-Measure = 2 × Recall × Precision / (Recall + Precision))

$$Sensitivity = TPR = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{4}$$

$$Pr ecission = PPV = \frac{TP}{TP + FP} \tag{5}$$

ROC is plotted as TPR on one axis *(y-axis)* and *FPR = 1 – TNR* on the other *(x-axis)*. Each instance prediction plots a point on the ROC graph.

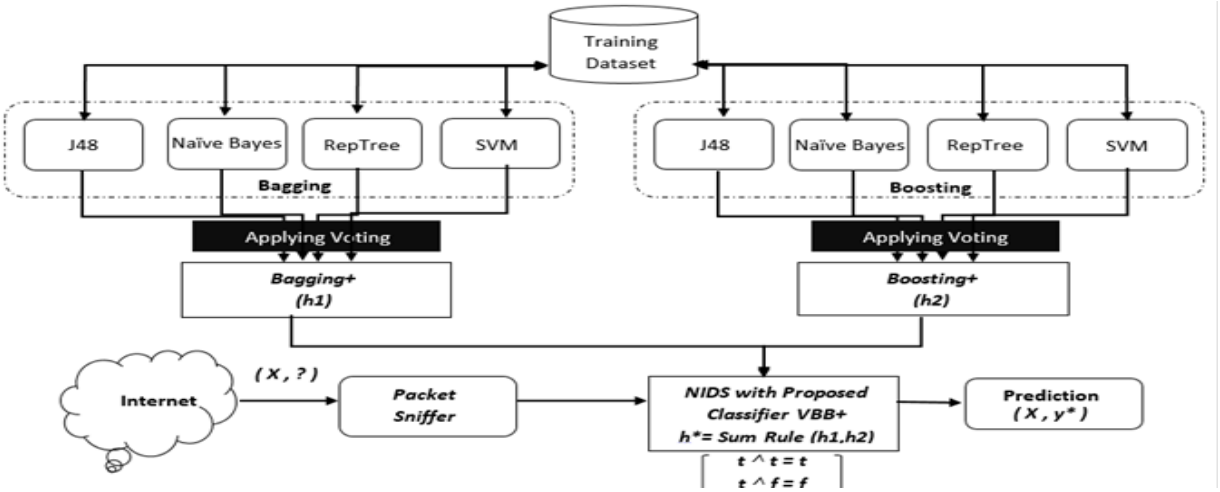

Figure 3: Proposed Architecture of NIDS

$$ROC = \frac{TPR}{FPR} \tag{6}$$

4. Proposed Methodology and Experimental Setup

Bagging is always good for majority class instances as it works on the voting mechanism. If the majority base classifiers give the wrong prediction for an instance then the final result will be a misclassification. This problem can be solved by assigning small weights to majority classes and higher weights to minority classes in the training subset or those instances generating errors. The upside of the boosting over bagging is because it works straight for the reduction of error cases, while bagging works indirectly. For improving the accuracy and reduction of FPR we suggest combining bagging and boosting methodology with sum rule voting (VBB+). When the sum rule is used each sub-ensemble has to give a confidence value for each candidate. In our model, each sub-ensemble expresses the degree of their preferences against each candidate using as the confidence score the probabilities of sub ensemble prediction. Next, the confidence values for each prediction are added for each candidate and the candidate with the highest confidence score will be the final prediction. The weight of misclassified instances generated by the sub ensembles is increased by 0.1% of the total number of instances in the dataset and added to the original dataset to increase the classification accuracy. The proposed model is an ensemble of ensembles and it is schematically presented in Fig. 3. The experiment is divided into two stages. Initially, the imbalance datasets are pre-processed using SMOTE (synthetic minority oversampling technique) to increase the weight of minority class by generating synthetic instances. The NSL-KDD dataset has suggested 41 features to implement an intrusion detection system [23]. All the experiments are carried on a 3.10GHz Intel(R) Core(TM) i5-2400 CPU with 4GB of RAM. Weka data mining tool version 3.8 is used to measure the performance of the proposed intrusion detection model, in the Windows environment. To overcome the overfitting and under-fitting issues, the standardized 10 fold cross-validation technique is applied among the train and test model.

4.1 Architecture of Proposed IDS

Following the properties of an intrusion detection system, the proposed architecture is composed of two phases, and each phase consisting of multiple modules. The two phases are the training phase and the application phase. The training Phase is based on Data Pre-processing, Formation of Sub-Ensembles (bagging+ and boosting+) using four base classifiers (Naïve Byes, SVM, j48, and Reptree), Weight Promotion of the misclassified instances by 0.1% and finally the construction of proposed classifier. The application phase consists of a packet sniffer and a detector. The data pre-processing module of the training phase is used to increase the weight of minority classes for improving classification accuracy and the reduction of FP alarms. Formation of the sub ensembles (bagging+ and boosting+) uses four different base classifiers in each. The proposed classifier is constructed on the voting method using bagging and boosting with j48 as the base classifier. In the training phase, the system builds the proposed classifier. After the completion of this phase, the application phase captures the online packets from the network through a packet sniffer and processes them through the proposed classifier. The proposed classifier classifies the packet into two categories, intruder packet, and normal packet. Normal packets are allowed to pass into the network while the intruder generates the alarm, additionally, the unknown packets will be logged so that their behavior can be studied later and can be classified respectively. Figure 3 represents the architecture of the proposed network intrusion detection system graphically.

4.2 Proposed Algorithm VBB+

The upside of boosting over bagging is that boosting acts straightforwardly to diminish errors, though boosting works indirectly. From the literature review, it has been observed that for bagging and boosting, the increase in the number of sub classifiers leads to an increase in classification accuracy and a decrease in the number of prediction errors, but the relative effect of each consecutive classifier is constantly decreasing. Most of the effect of each technique is obtained by the first few classifiers [30], [32], and [33]. We used 4 different base classifiers for building each sub-ensemble of the proposed system. To bring further improvements in the classification accuracy and decrease the number of misclassified instances we suggest the combination of bagging and boosting technique with sum rule VBB+.

Algorithm: Algorithm for proposed VBB+

Input: Dataset
Output: Trained Model VBB+.

Begin:

Step 1: Arrange both datasets 10% of KDD Cup 99 and NSL-KDD in .ARFF file format.

Step 2: Pre-process ← Datasets.

Step 3: Sub-ensemble (bagging+) ← voting (Bagging (j48, SVM, Naïve Byes, and Reptree) using a 10-fold cross-validation test.

Step 4: Sub-ensemble (boosting+) ← voting (AdaBoostM1 (j48, SVM, Naïve Byes and Reptree) using 10-fold cross-validation test.

Step 5: New Dataset ← misclassified instances (bagging+ and boosting+) increase their weight with a ratio of 0.1%.

Step 6: VBB+ ← voting (bagging and boosting (New Dataset)) and test it using 10-fold cross-validation.

Step 7: Calibrate the results and implement the model.

End.

When using the sum rule, each sub-ensemble must offer each candidate a confidence value. In our algorithm, voters use the probabilities of the sub-ensemble forecast as the confidence to convey the degree of their choice. Next, for each candidate, all confidence values are added and the candidate with the largest amount wins the election. The proposed ensemble classifier technique for NIDS is presented in Fig. 3, where *h1* and *h2* are the produced hypothesis of each sub ensemble, *x* denotes the instance for classification, and *y** is the final prediction of the proposed classifier.

5. Results and Discussion

In this section, the study provides the experimental outcomes of the proposed method in terms of model building time, number and type of misclassified instances, classification accuracy, and the number of false positives. The study also provides a comparison of the proposed method with other existing machine learning techniques.

The purpose of this work is to assess the efficacy of the VBB+ model in the detection of misclassified trends in the dataset. However, a critical initial step to better apprehend the dataset is to study its behavior to know more about it. The study has used the benchmark datasets KDD Cup 99 and NSL-KDD for calibrating the said parameters for performance evaluation. From the dataset description in Table. I, It can be observed that the 10% corrected dataset of KDD Cup 99 has 396,747 types of attack instances belonging to four major classes of attack and 97,278 instances of the normal class. While NSL-KDD has 58,630 instances of attacks and 67.343 instances of the normal class. The Denial of services (DOS), In a DOS attack, and the attacker targets the server by flooding a large number of legitimate-looking requests to the server in a way that the server cannot differentiate between a valid and non-valid request. It will overwhelm the system to a point that the server cannot handle the capacity anymore. The point of such kind of attack is to overload the targets bandwidth and other resources making it unavailable to other clients. Distributed DOS (DDOS) is a new type of attack. Probe, In Probe an attack is intentionally created by an attacker to identify its purpose and record it in the report with an unmistakable "fingerprint".

Table 2: Classification Accuracy and Number of Misclassified Instances Using NSL-KDD Dataset.

Classifier	***Model Building Time (sec)***	***Classifier Accuracy in percent***	***The misclassified instance in Percent***
Naïve Byes	10.8	48.38%	51.61% (65025 instances)
SVM	152.23	98.88%	1.11% (1406 instances)
Reptree	150.88	99.42%	0.58% (719 instances)
J48	131.45	99.65%	0.35% (432 instances)
bagging+	738.13	99.71%	0.29% (359 instances)
boosting+	318.44	99.75%	0.25% (314 instances)
Proposed VBB+	442.29	99.86%	0.14% (177 instances)

Table 3: Classification Accuracy and Number of Misclassified Instances Using KDD Cup 99 Dataset.

Classifier	***Model Building Time (sec)***	***Classifier Accuracy in Percent***	***The misclassified instance in Percent***
Naïve Byes	12.89	93.69%	6.31% (31161 instances)
SVM	290.96	99.82%	0.18% (866 instances)
Reptree	231.43	99.84%	0.16% (755 instances)
J48	248.52	99.86%	0.14% (687 instances)
bagging +	911.39	99.86%	0.14% (669 instances)
boosting +	724.84	99.87%	0.13% (594 instances)
Proposed VBB+	983.43	99.98%	0.02% (103 instances)

The attacker at that point utilizes the community-oriented framework to gain proficiency with the locator's area and defensive capabilities from the reports. In a remote to local (r2l) attack the attacker tries to gain unauthorized access to the victim's machine and pretend to be a local network user. And in User to root (u2r) attack, the attacker legally accesses the local network machine to illegally gain the root privileges. The attack types in the dataset are further divided into subclasses, few subclasses like "land, ftp_write, imap, multihop, spy, warezmaster, loadmodule, perl, and rootkit" has less number of instances. Synthetic Minority Oversampling is the technique used to generate synthetic samples instances of the data to overcome the problem of imbalance in the subclasses.

Table 2 and 3 demonstrate the extensive comparison between the proposed technique and other classic approaches of machine learning and it can be observed that the proposed technique serves the purpose of this research by achieving the targeted results. Although the model building time is more than others the minimum number of misclassified instances makes the model ideal for the intrusion detection system. The model building time can be reduced to a minimum by increasing the processor speed and memory. According to data in the tables, it is very clear that the performance of Naïve byes is very poor while among all non-meta classifiers j48 is much better. This is because of the Naive Bayes approach's premise that all parameters are autonomous. That's not always the situation, though many parameters of safety are interdependent. As a consequence, Naive Bayes classifier requires less memory and less computation time resulting in poor outcomes. Through the literature review and experiment, it has been observed that J48 results outstanding with high classification accuracy and less false positives. Authors of the Weka machine learning software described the C4.5 (J48) algorithm as "a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date" [32]. This inspired property of J48 becomes the reason for using J48 as a base classifier in the proposed ensemble technique and the results show that the proposed technique enhances the results of the J48 algorithm. On the other hand, the Reptree and SVM also work fine but still, certain subclasses remained ambiguous using these techniques. The fact that SVM does not possess such high accuracy is because SVM always requires ready real-valued vectors as features which might cause the system in generating false alarms due to diverse kind of traffic on the network. It requires the system to possess the ability to identify new kinds of traffic and attacks.

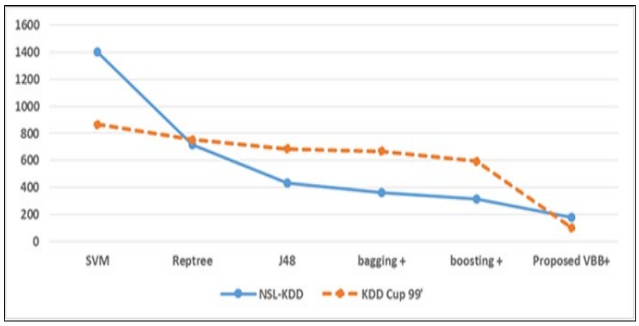

Figure 4: Number of Misclassified Instances

The line graph in Fig: 4, presents the trend that how ensemble classifiers lower the number of misclassified instances on both datasets as it's the most important criteria for NIDS. Misclassification may lead to network intrusion, as the system considering attack-type of data as normal or vice versa. The proposed model misclassifies 177 number of instances out of 125973 total number of instances which is equal to 0.014% of total instances on the NSL-KDD dataset while on the KDD cup 99 datasets the VBB+ misclassifies 103 instances out of 494025 equals to 0.022% of the total instances.

The bar graph in fig. 5 depicts a picture of per class reduction in the number of misclassified instances, either they belong to DOS, Probe, U2R, or R2L type of attack or they belongs to the Normal class because, in anomaly-based NIDS, the traffic belonging to 'normal' class is always allowed to move forward to the network so if the 'normal' data packets are classified accordingly then remains very fewer chances of false-positive alarms. The 'Probe' class of attack is misclassified the most, among all the tested methods except the proposed VBB+ because these type of attacks are used to detect the network configuration to find the loopholes in the systems and they do not exhibit any harmful data but blank or normal traffic crafted by the attacker to detect the network setting so it's difficult to classify as it's very closer to normal packet but the proposed model successfully works on it too.

Figure 5: Class Wise Distribution of Misclassified Instances

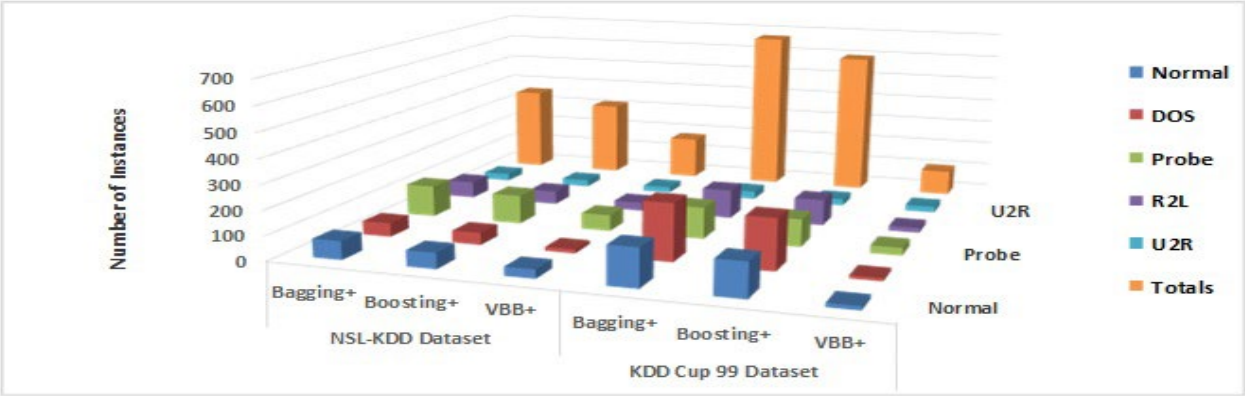

Figure 6: Class Wise Distribution of Misclassified Instances

The class-wise results achieved from the proposed model are presented in Table IV, following the above stated important evaluation parameters, such as True Positive rate (TP), False Positive rate (FP), Precision, and ROC values. True positive is the number of instances correctly classified for a particular class, while false positive is the number of adverse instances categorized as positive wrongly for a particular class. Statistics in table IV, clearly show that the percentage of true positives for each class for each attack type is very high while the percentage of false positives is very low. Precision defines the accuracy of the model for each class.

The characteristics of the receiver operator curve (ROC) illustrate the trade between sensitivity and specificity. ROC is a graphical plot illustrating the efficiency of a binary classification scheme as its limit for discrimination is diverse. ROC curves plot the true positive vs. the false positive at the varying threshold. The achieved ROC values, plots almost an accurate curve for each class. The class "spy", (R2L type of attack) contains a very less number of instances which becomes the cause of less accuracy, this can be overcome by generating synthetic samples using SMOTE. While the other classes "ftp_write, imap and phf" of the same attack type are detected with high accuracy.

Type of Attack	***Class***	***NSL-KDD Dataset***				***KDD Cup 99 Dataset***			
		TP Rate	***FP Rate***	***Precision***	***ROC value***	***TP Rate***	***FP Rate***	***Precision***	***ROC value***
NORMAL	Normal	99.80%	0.00%	99.80%	1	99.99%	0.00%	99.90%	1
DOS Attack	Land	61.10%	0.02%	61.10%	1	85.70%	0.002%	94.70%	1
	Back	99.70%	0.001%	99.80%	1	99.90%	0.00%	100.00%	1
	Pod	100.00%	0.00%	99.50%	1	99.60%	0.00%	99.60%	1
	Smurf	100.00%	0.00%	100.00%	1	100.00%	0.00%	100.00%	1
	Neptune	100.00%	0.00%	100.00%	1	100.00%	0.00%	100.00%	1
	Teardrop	100.00%	0.00%	100.00%	1	99.90%	0.00%	100.00%	1
PROBE Attack	Ipsweep	99.70%	0.00%	99.50%	1	99.70%	0.00%	99.60%	1
	Portsweep	99.40%	0.01%	99.90%	1	99.00%	0.00%	99.70%	1
	Satan	99.40%	0.00%	99.80%	1	99.40%	0.00%	99.70%	1
	Nmap	99.00%	0.00%	99.30%	1	97.40%	0.00%	98.70%	1
R2L Attack	ftp_write	37.50%	0.001%	100.00%	0.828	37.50%	0.001%	60.00%	0.866
	guess_passwd	94.30%	0.00%	98.00%	0.986	96.20%	0.00%	100.00%	1
	Imap	63.60%	0.001%	100.00%	0.881	83.30%	0.001%	100.00%	1
	Multihop	42.90%	0.002%	75.00%	0.93	42.90%	0.002%	50.00%	0.985
	Phf	100.00%	0.00%	100.00%	1	100.00%	0.00%	100.00%	1
	Spy	40.00%	0.001%	40.00%	0.694	83.30%	0.001%	100.00%	0.995
	Warezclient	98.70%	0.00%	98.90%	1	99.70%	0.001%	100.00%	1
	Warezmaster	75.00%	0.001%	88.20%	0.982	80.00%	0.002%	88.90%	0.995
U2R Attack	bufferoverflow	76.70%	0.001%	95.80%	0.999	76.70%	0.001%	82.10%	0.995
	Loadmodule	11.10%	0.00%	50.00%	0.907	22.20%	0.001%	33.30%	0.965
	Perl	100.00%	0.00%	75.00%	1	100.00%	0.00%	100.00%	1
	Rootkit	10.00%	0.00%	33.30%	0.766	0.00%	0.002%	0.00%	0.937

6. Conclusion and Future Work

The primary goal of any IDS is to recognize normal activities from strange ones and raise an alarm when intrusions are identified. Even though the intrusion detection literature is extremely bothering, there are no ideal IDS that can always appropriately differentiate among intrusion and normal activities with 100% accuracy rates. Machine Learning-based IDS has been an effective answer for secure systems against intrusion attacks [35]. In this paper, the study analyzed both NSL-KDD and KDD Cup 99 datasets renowned as a benchmark for NIDS and proposed a model for anomaly-based network intrusion detection using ensemble machine learning techniques. Building excellent ensemble classifiers has become an active study area in supervised machine learning, and the findings have shown that ensemble techniques are much more precise than individual classifiers. The reason behind the better performance of the ensemble is that it tries to find the global optima instead of getting stuck in local optima. In this proposed model the study built an ensemble of sub-ensembles using the voting methodology of bagging and boosting, using a four base classification algorithm (Reptree, Naive byes, SVM, and J48).

The experimental findings indicate that the proposed VBB+ method ensures maximum precision in the classification of 99.86% on the NSL-KDD dataset and 99.98% on the KDD Cup 99 dataset. Finally, the priority for future research is to implement the proposed model in a real-time environment and further investigate more ensemble techniques to a range of different intrusions. ML algorithms have their vulnerabilities, thus with the advancements of their research, the attackers will also develop new methods of intrusion that are more dynamic and capable of bypassing IDS and other network security measures. In recent years, security is gaining a high focus in network deployment, especially in the connected vehicle stream [36] [37]. However, Hackers have access to an advanced set of tools and have established professional skills that allow them to conduct the hacking process from a far distance. This technology is evolving, thereby overcoming the fear of the cybersecurity threat and sooner or later affecting every aspect of our lives. [38].The day to day increasingly new type of attacks remains a big challenge and elusive goal for research. Our future work directions are defined towards network security and the development of ML algorithm variations with an awareness of all relevant security issues.

Acknowledgments I would like to express my special thanks to my supervisor Dr. Syed Nadeem Ahsan as well as our Dean FEST Dr. Syed Kamran Raza, who gave me the golden opportunity to do this research, and I am very thankful specially to my parents and family members who always supported me.

References

[1] Ponemon Institute and Hewlett Packard Enterprise "2015 Cost of Cyber Crime Study: Global" Research Department 2308 US 31 North Traverse City, Michigan 49629 USA 2015. Retrieved from http://www8. hp.com/us/en/software-solutions/ponemon-cyber-security-report/ (accessed 26 June, 2017)

[2] Heady, Richard, George Luger, Arthur Maccabe, and Mark Sevilla. "The architecture of a network-level intrusion detection system". No. LA-SUB-93-219. Los Alamos National Lab., NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science, 1990. syst.

[3] M. N. Mohammad, N. Sulaiman, and O. A. Muhsin, "A novel intrusion detection system by using intelligent data mining in weka environment," Procedia Computer Science, vol. 3, pp. 1237-1242, 2011.

[4] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," Information Sciences, vol. 177, no. 18, pp. 3799-3821, 2007.

[5] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems, and challenges," computers and security, vol. 28, no. 1-2, pp. 18-28, 2009.

[6] Y. Bai and H. Kobayashi, "Intrusion detection systems: technology and development," in 17th International Conference on Advanced Information Networking and Applications, 2003. AINA 2003., 2003: IEEE, pp. 710-715.

[7] V. Jyothsna, V. R. Prasad, and K. M. Prasad, "A review of anomaly-based intrusion detection systems," International Journal of Computer Applications, vol. 28, no. 7, pp. 26-35, 2011.

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009: IEEE, pp. 1-6.

[9] C.-H. Su, F. Tu, X. Zhang, B.-C. Shia, and T.-S. Lee, "A ensemble machine learning based system for merchant credit risk detection in merchant mcc misuse," Journal of Data Science, vol. 17, no. 1, 2019.

[10] Dietterich, Thomas G. "Ensemble methods in machine learning." In International workshop on multiple classifier systems, pp. 1-15. Springer, Berlin, Heidelberg, 2000.

[11] N. Sultana, N. Chilamkurti, W. Peng, and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," Peer-to-Peer Networking and Applications, vol. 12, no. 2, pp. 493-501, 2019.

[12] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l, 2005: IEEE, pp. 250-257.

[13] D. Gaikwad and R. C. Thool, "Intrusion detection system using bagging ensemble method of machine learning," in 2015 International Conference on Computing Communication Control and Automation, 2015: IEEE, pp. 291-295.
[14] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, "Intrusion detection using fuzzy association rules," Applied Soft Computing, vol. 9, no. 2, pp. 462-469, 2009.
[15] O. Al-Jarrah and A. Arafat, "Network Intrusion Detection System using attack behavior classification," in 2014 5th International Conference on Information and Communication Systems (ICICS), 2014: IEEE, pp. 1-6.
[16] S. K. Jonnalagadda and R. P. Reddy, "A literature survey and comprehensive study of intrusion detection," International Journal of Computer Applications, vol. 81, no. 16, pp. 40-47, 2013.
[17] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," IEEE transactions on knowledge and data engineering, vol. 26, no. 4, pp. 984-996, 2013.
[18] Y. Wang, Y. Shen, and G. Zhang, "Research on Intrusion Detection Model using ensemble learning methods," in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2016: IEEE, pp. 422-425.
[19] P. Gogoi, B. Borah, and D. K. Bhattacharyya, "Network Anomaly identification using supervised classifier," Informatica, vol. 37, no. 1, 2013.
[20] Z. Liu, "A method of SVM with normalization in intrusion detection," Procedia Environmental Sciences, vol. 11, pp. 256-262, 2011.
[21] S. Peddabachigari, A. Abraham, and J. Thomas, "Intrusion detection systems using decision trees and support vector machines," International Journal of Applied Science and Computations, USA, vol. 11, no. 3, pp. 118-134, 2004.
[22] S. S. Dongre and K. K. Wankhade, "Intrusion detection system using new ensemble boosting approach," International Journal of Modeling and Optimization, vol. 2, no. 4, p. 488, 2012.
[23] J. Hu, X. Yu, D. Qiu, and H.-H. Chen, "A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection," IEEE network, vol. 23, no. 1, pp. 42-47, 2009.
[24] V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," in the 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2017: IEEE, pp. 13-19.
[25] Salo, Fadi, Mohammad Noor Injadat, Abdallah Moubayed, Ali Bou Nassif, and Aleksander Essex. "Clustering Enabled Classification using Ensemble Feature Selection for Intrusion Detection." In 2019 International Conference on Computing, Networking and Communications (ICNC), pp. 276-281. IEEE, 2019.
[26] J. Brownlee. "Supervised and Unsupervised Machine Learning Algorithms." https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (accessed 20 June 2017).
[27] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," arXiv preprint arXiv:1701.02145, 2017.
[28] M. Zamani and M. Movahedi, "Machine learning techniques for intrusion detection," arXiv preprint arXiv:1312.2177, 2013.
[29] C. Chen, Y. Gong, and Y. Tian, "Semi-supervised learning methods for network intrusion detection," in 2008 IEEE International Conference on Systems, Man and Cybernetics, 2008: IEEE, pp. 2603-2608.
[30] L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123-140, 1996.
[31] J. A. Aslam, R. A. Popa, and R. L. Rivest, "On Estimating the Size and Confidence of a Statistical Audit," EVT, vol. 7, p. 8, 2007.
[32] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in icml, 1996, vol. 96: Citeseer, pp. 148-156.
[33] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," Machine learning, vol. 36, no. 1-2, pp. 105-139, 1999.
[34] Witten, Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
[35] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," Ad Hoc Networks, vol. 90, p. 101842, 2019.
[36] Amoozadeh, Mani, Arun Raghuramu, Chen-Nee Chuah, Dipak Ghosal, H. Michael Zhang, Jeff Rowe, and Karl Levitt. "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving." IEEE Communications Magazine 53, no. 6 (2015): 126-132.
[37] Sharma, Prinkle, Hong Liu, Honggang Wang, and Shelley Zhang. "Securing wireless communications of connected vehicles with artificial intelligence." In 2017 IEEE international symposium on technologies for homeland security (HST), pp. 1-7. IEEE, 2017.
[38] Rivoirard, Lucas, Martine Wahl, Patrick Sondi, Marion Berbineau, and Dominique Gruyer. "Chain-Branch-Leaf: A clustering scheme for vehicular networks using only V2V communications." Ad Hoc Networks 68 (2018): 70-84.

Kashif Gul received his BS degree in computer science from Shah Abdul Latif University in 2010. He is currently enrolled in, the master's degree program at IQRA University Karachi, Pakistan. In 2015, he joined the IBA Community College, Naushehro Froze, Pakistan, as a lecturer. His current research interests include machine learning, network security and implementation, and cloud computing. He is very interested in machine learning techniques for pattern recognition and detection researches.

Dr. Syed Nadeem Ahsan

He is a Ph.D. from the Institute for Software Technology, Graz University of Technology, Austria. he has worked as a senior scientist and performed research in software engineering, and supervised multiple projects related to software engineering. He has completed several professional courses in Microsoft and Oracle. His main research area is focused around the automatic fault prediction in source code, multi-label classification of software change requests (SCR), and impact analysis of SCR. His interest is in the application of artificial intelligence and machine learning techniques in software engineering. He has published more than 12 research articles for the journal, conferences, and workshops and has been a member of the program committee for the 5th ICSEA 2010 (Fifth International Conference on Software Engineering Advance, Nice, France, 2010) and also chairs a session in the 4th ICSEA 2009. He is a member of the IEEE Computer Society. He is currently working as Assistant Professor at IQRA University Karachi, Pakistan.