A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning

Saikat Das[1], Deepak Venugopal[1] and Sajjan Shiva[1]

[1] The University of Memphis, Memphis, TN 38152, USA
{sdas1, dvngopal, sshiva}@memphis.edu

Abstract. Distributed Denial of Service (DDoS) has been the most prominent attack in cyber-physical system over the last decade. Defending against DDoS attack is not only challenging but also strategic. Tons of new strategies and approaches have been proposed to defend against different types of DDoS attacks. The ongoing battle between the attackers and defenders is full-fledged due to its newest strategies and techniques. Machine learning (ML) has promising outcomes in different research fields including cybersecurity. In this paper, ensemble unsupervised ML approach is used to implement an intrusion detection system which has the noteworthy accuracy to detect DDoS attacks. The goal of this research is to increase the DDoS attack detection accuracy while decreasing the false positive rate. The NSL-KDD dataset and twelve feature sets from existing research are used for experimentation to compare our ensemble results with those of our individual and other existing models.

Keywords: Unsupervised Machine Learning Ensemble, Novelty and Outlier Detection, DDoS Detection, Accuracy, IDS, and False Positive Rate.

1 Introduction

From the beginning of the architectural evolution of the Internet, the proper way to transmit a packet, and process reduction were the major concerns. Cyber attackers easily exploit the existing limitations of the Internet protocols (TCP, UDP, etc.) and the readily available attack tools. A Distributed Denial of Service (DDoS) attack is mostly a network attack that causes bandwidth overloading due to the use of immense inbound or outbound traffic over the network, resulting in disruption of the normal operation. The first well-documented DDoS attack appears to have occurred on August 1999, when a DDoS tool called ‘Trinoo’ was deployed in at least 227 systems, to flood a single University of Minnesota computer, which was knocked down for more than 2 days. In recent years, attacks on financial systems, broadcast systems, and Internet-based services have grown exponentially [1]. Moreover, those attacks are devastating, wide-ranging, easy to implement, and difficult to detect and defend, posing a major threat to Internet privacy and security. Today’s Internet is badly plagued by DDoS attack and the attack has been escalated drastically over the last decade. In the last couple of years, the giants such as GitHub, Amazon, Cloudflare, Facebook, Instagram, etc. had their service disruption by DDoS attack. According to the World Infrastructure Security Report 2018 [2], for the first time ever, a DDoS attack reached 1 Tbps (Terabyte per Second in size and the Internet has officially entered the terabit attack era. The largest attack was recorded as 1.7 Tbps. In the report, 16,794 DDoS attacks occurred per day have been mentioned which is equal to 700 attacks per hour or 12 attacks per minute and it predicts that this number has been growing rapidly day by day.

To keep alive in the competition, defenders are developing the newest technologies and mechanisms against those attacks. The existing DDoS attacks have been scrutinized and it is found that the attack can be mitigated by one of these three approaches or defense mechanisms, namely attacker-end approach, victim-end approach, and in-network approach, depending on their locality of deployment. Though attacker-end detection approach is much more challenging than the victim-end detection approach, solutions exist. On the other hand, victim-end detection is easier to implement compared to the other two types of detection approaches. The existing detection approaches can be categorized into statistical, soft computing, clustering, knowledge-based, and hybrid. These approaches can also be classified as supervised or unsupervised based on the type of dataset [3].

In the evolution of Intrusion Detection Systems (IDS), anomaly-based detection is more popular than signature-based detection. Machine learning (ML) has promising outcomes in detecting cyber-physical attacks including DDoS. Many researchers have already used ML classifiers to build IDSs in defending against DDoS attacks. In Machine learning, supervised, semi-supervised, and unsupervised are three basic ways to classify anomalous packets from normal packets. Supervised methods have the privilege of differentiating anomalous and normal data from a tagged dataset. Unsupervised methods, on the other hand, cluster dataset into different clusters where the strength of the clustering lies within the algorithm itself. Among those, novelty and outlier detection strategies are the unsupervised methods that have significant outcomes in detecting the unseen anomaly. One class SVM (Support Vector Machine), Local Outlier Factor, Elliptic Envelope, Isolation Forest, etc. are the most well-known novelty and outlier detection classifiers.

Both supervised and unsupervised classifiers are being used in developing IDS. However, the majority of these approaches have focused on learning a single model for intrusions. Moreover, due to the varied nature of intrusions, it may be hard to learn a single model that generalizes to all types. For example, some types of intrusions can be modeled using a simple linear model (e.g. logistic regression) while others may require more complex non-linear models (e.g. support vector machines with kernels). Therefore, the main idea of this paper is to train several models that can identify DDoS intrusions and then combine these into a unified system based on different mechanisms.

The benefits of ensemble learning, i.e., combining multiple classifiers to form a more powerful classifier have been well-studied in the ML community. Dietterich et al. [4] mentioned that ensembles can perform better than a single classifier and many classification problems have benefited from the idea of combining multiple classifiers. In general, there are two ways to ensemble the classifiers: homogeneous and heterogeneous. When similar types of classifiers are used to build a training model, it is called a homogeneous ensemble (e.g.; bagging and boosting), whereas combining different types of classifiers is called a heterogeneous ensemble (e.g.; stacking). Both homogeneous and heterogeneous ensembles are being used to build IDS. Aburomman et al. [5] mentioned a wide range of ensemble ML techniques and methods used to detect network intrusion. However, there are several drawbacks with existing ML approaches. First, existing techniques do not use relevant domain knowledge in constructing the classifier. This means that they end up using a lot of irrelevant features which results in the so-called "curse-of-dimensionality", i.e., the accuracy and generalization reduce as the number of features increase. Next, most existing methods focus on supervised ML models which is problematic since it requires a large amount of labeled data. Finally, even in existing methods that use unsupervised methods to detect network intrusions, there is no systematic approach that has been developed that can combine different unsupervised models, which is particularly important since learning an ensemble model is more robust as compared to learning a single model. In this work, our goal is to address all these three issues.

Here, twelve feature sets [6, 7, 8, 9, 10, 11, 12] that produce higher accuracy are considered for experimentation. The novelty of this research is to ensemble 'novelty and outlier detection' type unsupervised classifiers for better detection accuracy and lower false positive alarm. We show that this ensemble approach outperforms existing research methods as well and empirically show that generalization over new attacks is significantly improved when we combine different approaches as compared to using any one single approach.

The rest of the paper is organized as follows: In Section 2, we discuss the state of the art of recent IDS that use ensemble learning and how this research contribution is different and better from other approaches. An ensemble-based IDS framework is proposed in Section 3 and in Section 4, experimentation and observations are shown. Finally, in Section 5, the pros and cons of the proposed model are discussed and the conclusion of the paper with future research direction is drawn.

2 Literature Review

DDoS attacks have become a weapon of choice for hackers as well as for cyber terrorists and used as a form of protest in a politically unstable society. Various detection techniques are being improvised by researchers to defend against DDoS attacks over the year. To evade the existing DDoS attack detection solutions, the attack itself changes frequently. Based on the various techniques such as cloud computing, software defined networking (SDN), backbone web traffic, and big data strategies, the DDoS attack detection can be categorized into filtering mechanism, routers function, network flow, statistical analysis, and learning machine.

A comprehensive survey of Machine learning intrusion detection [13], systematic literature review and taxonomy of DDoS attack [14] are necessary to know the state of art of ML approaches for both IDS and DDoS. Ahmad Riza'ain [14] et al. performed an in-depth analysis on DDoS attack types as well as on existing DDoS detection and attack prediction techniques by characterizing the attacks. Also, they identified the factors behind those attacks. Moreover, they have classified and ranked at least 53 articles from different digital libraries such as Science Direct, ACM Digital Library, IEEE Xplore, Springer, and Web of Science related to DDoS detection and prevention and found 30% of them using ML techniques as their detection or prevention strategy.

To detect the DDoS attack, supervised [15], semi-supervised [16], and unsupervised methods are being used to build the training model. A combination of supervised and unsupervised ML model to detect anomaly can also be found in [17]. Neural Network and SVM for supervised modeling, KNN for unsupervised modeling, and Principal Component Analysis (PCA) and Gradual Feature Reduction (GFR) for feature selection with NSL-KDD dataset are used there. However, the reason behind using the combination of the supervised and unsupervised method in their research is ambiguous.

Ensemble is the way of combining multiple classifiers for better performance over single classifier and many classification problems have benefited from this idea [4]. Homogeneous (combination of similar types of classifiers) and heterogeneous (combination of different types of classifiers) are the two major ensemble types. A detailed survey of ensemble and hybrid classifiers [5] helps in understanding the usage and shortfalls of ensemble ML in network security.

Outlier and novelty detection techniques are more efficient in detecting unknown attacks as they use unsupervised ML models. An unsupervised ML model is used [18] to detect a high-volume DDoS attack using in-memory distributed graph. Jabez et al. [19] mentioned an outlier detection mechanism NOF (Neighborhood Outlier Factor) to detect the anomaly. But there could be a high chance for a single classifier to predict incorrectly compared to multiple classifiers' prediction. Therefore, an ensemble classifier would be a perfect fit for predicting anomalous behavior precisely. Smyth et al. [20] showed that stacked density estimation outperforms a single best model which could be chosen based on cross-validation, combining with uniform weights, or even through bias. A few hybrid supervised learning models [21] are used to detect DDoS attack but realistically for a zero-day attack or unknown attacks, an unsupervised hybrid model has better detection accuracy.

Though most of the researchers have chosen a single classifier to train their model in detecting DDoS attack, a combination of classifiers has better accuracy compared to a stand-alone model. Moreover, none of them have focused on either unsupervised ensemble or outlier and novelty detection ensemble. In addition, their works didn't guide properly in detecting unseen attacks. Therefore, our motivation and the goal of this paper is to build an unsupervised ensemble model which combines five different 'outlier and novelty detection' classifiers, resulting in the detection of unseen DDoS attacks. Using unsupervised ensemble model is the novelty of this research, and the better detection accuracy with lower false positive rates are obtained with this model.

3 Proposed Method

As DDoS has the devastating damaging effects on organizations' assets, a comprehensive defense mechanism is required to protect assets. Anomaly-based IDS over signature-based has a better detection accuracy privileging in detecting unseen attacks but at the expense of a lot of false identification of unusual activities as anomalous. Traditional IDSs are defending and upgrading their strategies to cope up with new attack types and patterns. However, with the change of attackers' motive and intention, an adaptive IDS is most demanding in the cyber world. Here, in this paper, an ML based IDS is proposed that has the novelty to ensemble unsupervised classifiers based on outlier detection approach, and which gives a better detection accuracy with a lower false positive rate in detecting DDoS.

3.1 Dataset

In this research, NSL-KDD [22] dataset is used for training and testing purposes. NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set which are mentioned in [23]. Although, McHugh discussed some problems that are suffered by this new version of the KDD data set and may not be a perfect representative of existing real networks due to the lack of public data sets for network-based IDSs. It can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods. NSL-KDD has some major improvements over the original KDD'99 dataset [22]:

1. No redundant records in the train data, so classifiers will not be biased towards more frequent records;
2. No duplicate records in the test data, so the performance of the learners is not biased by the methods which have better detection rates;
3. The number of selected records from each difficulty group is inversely proportional to the percentage of records in the original KDD'99 dataset;
4. The number of records in the train and test sets are reasonable that makes the dataset affordable to run the experiments, etc.

The dataset contains eight data files of different formats that are compatible with most experimental platforms. Table 1. shows a summary of the testing and training data record.

Table 1. Summary of training and testing data records.

Class	Training Set	Occurrence %	Testing Set	Occurrence %
Normal	67343	53.46%	9711	43.08%
DDoS	45927	36.46%	7460	33.085%
Other	12703	10.08%	5373	23.85%
Total	125973	100%	22544	100%

3.2 Data Preprocessing

A Machine learning classifier trains and tests it's model by using a dataset that contains numeric values. So, it is necessary to convert any non-numeric data content to numeric data before it can be fed to a classifier. After analyzing the whole dataset, categorical values have been found in features, namely protocol type, service, and flag, whereas all other features contain numeric values. In the preparation of feature selections step, to select the most important features, the categorical values are needed to convert into numeric values. In the data preprocessing phase, for each such feature, the distinct values are identified for all entries in that column and replaced with numeric values using simple integer assignment starting from 1. The reference for this conversion is shown in Table 2.

Table 2. Conversion table for categorical variables to integer values

Feature Name	Integer Conversions
protocol type	'tcp': 1, 'udp': 2, 'icmp': 3
service	'ftp_data': 1, 'other': 2, 'private': 3, 'http': 4, 'remote_job': 5, 'name': 6, 'netbios_ns': 7, 'eco_i': 8, 'mtp': 9, 'telnet': 10, 'finger': 11, 'domain_u': 12, 'supdup': 13, 'uucp_path': 14, 'Z39_50': 15, 'smtp': 16, 'csnet_ns': 17, 'uucp': 18, 'netbios_dgm': 19, 'urp_i': 20, 'auth': 21, 'domain': 22, 'ftp': 23, 'bgp': 24, 'ldap': 25, 'ecr_i': 26, 'gopher': 27, 'vmnet': 28, 'systat': 29, 'http_443': 30, 'efs': 31, 'whois': 32, 'imap4': 33, 'iso_tsap': 34, 'echo': 35, 'klogin': 36, 'link': 37, 'sunrpc': 38, 'login': 39, 'kshell': 40, 'sql_net': 41, 'time': 42, 'hostnames': 43, 'exec': 44, 'ntp_u': 45, 'discard': 46, 'nntp': 47, 'courier': 48, 'ctf': 49, 'ssh': 50, 'daytime': 51, 'shell': 52, 'netstat': 53, 'pop_3': 54, 'nnsp': 55, 'IRC': 56, 'pop_2': 57, 'printer': 58, 'tim_i': 59, 'pm_dump': 60, 'red_i': 61, 'netbios_ssn': 62, 'rje': 63, 'X11': 64, 'urh_i': 65, 'http_8001': 66, 'aol': 67, 'http_2784': 68, 'tftp_u': 69, 'harvest': 70
flag	'SF': 1, 'S0': 2, 'REJ': 3, 'RSTR': 4, 'SH': 5, 'RSTO': 6, 'S1': 7, 'RSTOS0': 8, 'S3': 9, 'S2': 10, 'OTH': 11

The NSL-KDD dataset that we have used is a tagged dataset. Since the proposed model initially works with unsupervised methods which don't require any class level, 'class' column from the dataset is removed in this phase. On the other hand, to combine the outputs of those unsupervised methods, logistic regression (LR) and naïve Bayes (NB) are used which require a class label. We use the removed class label from this phase to train LR and NB model later.

3.3 Feature Selections

In our earlier research [24], we have built a supervised ensemble model using 24 reduced features [6, 7]. Here, twelve different feature sets from existing research are used for experimentation and analyzed their accuracy. Based on our domain knowledge, we have verified each of the features' relevancy with the DDoS attack. Table 3. shows all feature sets that have been used on data classification phase.

Table 3. Feature sets with features' list from references

Feature Set	References	Feature Count	Features
FS-1	[6,7]	24	2, 3, 4, 5, 7, 8, 10, 13, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 38, 39, 40, 41
FS-2	[8]	13	3, 4, 29, 33, 34, 12, 39, 5, 30, 38, 25, 23, 6
FS-3	[8]	14	5, 3, 6, 4, 30, 29, 33, 34, 35, 38, 12, 39, 25, 23
FS-4	[8]	14	12, 26, 4, 25, 39, 6, 30, 38, 5, 29, 3, 37, 34, 33
FS-5	[8]	14	5, 3, 6, 4, 29, 30, 33, 34, 35, 12, 23, 38, 25, 39
FS-6	[8]	14	3, 29, 4, 32, 38, 33, 39, 12, 36, 23, 26, 34, 40, 31
FS-7	[9]	12	23, 5, 3, 6, 32, 24, 12, 2, 37, 36, 8, 31
FS-8	[10]	16	2, 4, 10, 14, 17, 19, 21, 24, 25, 26, 27, 30, 31, 34, 35, 39

FS-9	[11]	35	9,26,25,4,12,39,30,38,6,29,5,37,11,3,22,35,34,14,33,23,8,10,31,27,28,32,1,36,2,41,40,17,13,16,19
FS-10	[12]	19	3, 4, 5, 6, 12, 23, 24, 25, 26, 29, 30, 32, 33, 34, 35, 36, 37, 38, 39
FS-11	[5]	15	4, 5, 6, 8, 10, 12, 17, 23, 26, 29, 30, 32, 37, 38, 39
FS-12		41	All features

3.4 Data Classification

Data classification section is divided into two major parts: individual classifier and ensemble classifier.

Individual Classifier. To detect a DDoS attack using outlier detection and novelty detection techniques, it is required to be able to decide whether a new observation data belongs to the same distribution as existing observations (can be called an inlier) or should be considered as different (can be called an outlier).

In any training dataset, data could be concentrated into different regions or separated from each other. The observations that are not concentrated and far from any concentrated regions, are defined as outliers. Outlier detection estimators try to fit in those regions and ignore the deviant observations. On the other hand, the training data is not polluted by outliers and a new observation on outlier is known as a novelty. Both outlier and novelty detection techniques are used for anomaly detection where they are interested in detecting abnormal or unusual observations. Outlier detection and novelty detection are also known as unsupervised and semi-supervised anomaly detection respectively.

Here, the proposed method starts with each of the five unsupervised outlier detection classifiers working individually and then observing the performance metrics: accuracy, false positive rate, precision, recall, and F1 scores. Then the Majority Voting, Logistic Regression, and Naïve Bayes ensemble techniques are applied on these five classifiers to get better performance.

One Class SVM. Support vector machines (SVMs), which are the types of supervised learning models are very well-known in the ML environment that analyze data and recognize patterns. It can also be used for both classification and regression tasks. One-class classification (OCC), also known as unary classification or class-modeling, tries to identify objects of a specific class amongst all objects by primarily learning from a training set containing only the objects of that class [25].

Therefore, in anomaly detection, one-class SVM is trained with data that has only one class, which is the "non-anomalous" or "normal" class. It infers the properties of normal classes and using these properties, it can predict which examples are unlike the normal. This is useful for anomaly detection because the scarcity of training examples is what defines anomalies; typically, there are very few examples of the network intrusion, fraud, or other anomalous behavior [26].

Local Outlier Factor. The local outlier factor score is computed by the LOF algorithm which reflects the degree of abnormality of the observations. With respect to its neighbors, LOF measures the local density (obtained from k-nearest neighbors) deviation of a given data point. As a result, it detects the samples that have a substantially lower density than their neighbors. For an observation, the LOF score is equal to the ratio of the average local density of its k-nearest neighbors and its own local density. A "normal" data is expected to have a local density similar to that of its neighbors, while an "abnormal" data is expected to have much smaller local density.

Isolation Forest. Random forest is used to perform the outlier detection efficiently in high-dimensional datasets. The algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected features [27]. Here, recursive portioning is used to split the values. As the recursive patronizing is illustrated by a tree structure, the required number of splitting to isolate a sample is equivalent to the path length from the root node to the leaf node where it terminates. The path length from the root node to the terminating node, averaged over a forest of such random trees, is a measure of normality and the decision function. Random partitioning produces noticeably shorter paths for anomalies. Therefore, the random forest trees that collectively produce shorter path lengths for particular samples are highly likely to be anomalies [28].

Elliptic Envelope. In outlier detection, one common way to detect the outlier is to assume that the regular data comes from a known distribution (e.g.; Gaussian distribution). From this assumption, a 'shape' of the data can be defined, and data points stand far enough from the fit shape are defined as outlying observations. Elliptic Envelope assumes the data as normally distributed and based on that assumption it 'draws' an ellipse around the data, classifying any observation inside the ellipse as an inlier (labeled as (+)1) or "normal" and any observation outside the ellipse as an outlier (labeled as (-)1) or "anomalous".

Ensemble Classifier. All four classifiers defined previously are used to build five training models, where One-class SVM is used twice with different hyperparameters. According to the framework, on top of these five models, different ensemble techniques are applied to combine them. The majority voting mechanism is chosen as a baseline. Then logistic regression (LR) and naïve Bayes (NB) are two supervised models that are applied on top of these five classifiers to ensemble, for the better detection accuracy and lower false positive alarm.

Majority Voting. Majority voting scheme is the very common and basic technique in ML ensemble. Generally, a majority means when the greater part or more than half of the total accumulates. In Machine learning, an output prediction could be '1' or '0'. A majority voting mechanism could be applied on any number of classifiers' output. When the greater part or more than half of the total classifiers' predictions agree with a certain prediction value '1' or '0', that prediction value would be the final output of this majority voting mechanism. For example, for five classifiers 'A, B, C, D, E' that predict a certain data instance '1, 0, 1, 1, 0' respectively, the final output will be the '1' decided by majority voting mechanism. When Majority Voting is used in ML ensemble as a combination rule (which only works with nominal classes), each of these classifiers will predict a nominal class label for a test sample. The label which was predicted the most will then be selected as the output of the voting classifier.

Logistic regression. Logistic regression is the go-to method for binary classification problems (problems with two class values) which is borrowed by ML from the field of statistics. In statistics, the logistic model (or logit model) is used to model the probability of an existing class or event such as normal/abnormal, pass/fail, win/lose, hot/cold, etc. This can be extended to model with several classes of events, and each of the events would be assigned a probability value between 0 to 1, where the sum of all probabilities will be a complete 1. The coefficients (Beta values, b) of the logistic regression algorithm must be estimated from the training data which is done by using maximum-likelihood estimation. The best coefficients would result in a model that would predict a value very close to 1 (e.g. anomalous) for the default class and value very close to 0 (e.g. normal) for the other class. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients (Beta values) that minimize the error in the probabilities predicted by the model to those in the data.

Naïve Bayes. Naïve Bayes is a simple and common classifier used in many ML problems. It is a Bayes theorem based probabilistic classifier which helps to define the probability of an event based on some prior knowledge of certain conditions associated with that event. The goal of any probabilistic classifier (e.g.; X_0, X_1, …. X_n features and C_0, C_1, . . . C_k classes) is to determine the probability of the features occurring in each class and to return the most likely class. Naïve Bayes classifier assumes that the features are independent of each other and thus it is named as "Naïve".

Fig. 1. shows the process flow of the proposed framework from data preprocessing to ensemble classification. In this framework, NSL-KDD dataset contains both training and testing data and is used as an input of Step-1. In Step-2, data are converted into a model readable format with the help of data preprocessing and feature selection. Processed training data is then fed into five different classifiers: One-Class SVM (OCS) with two different hyperparameters, Local Outlier Factor (LOF), Isolation Forest (ISOF), and Elliptic Envelope (ELE). At the end of this step, all five different classifiers have built their models using training data. Then test data is transferred from the raw dataset through feature selection and data preprocessing phase to Step-4 and evaluates the training models that are built on Step-3 to predict outcomes. In Step-5, all the predictions coming from different training models in the last step generate a vector for a single data instance which is then carried to each of the three ensemble classifiers: Majority Voting (MV), Logistic Regression (LR), and Naïve Bayes (NB). MV, LR, and NB have different performance measures for that prediction vector. Based on the higher detection accuracy, precision, recall, F-1 score, and lower false positive rate, Step-6 decides the best ensemble algorithm which is finally chosen for DDoS attack detection.

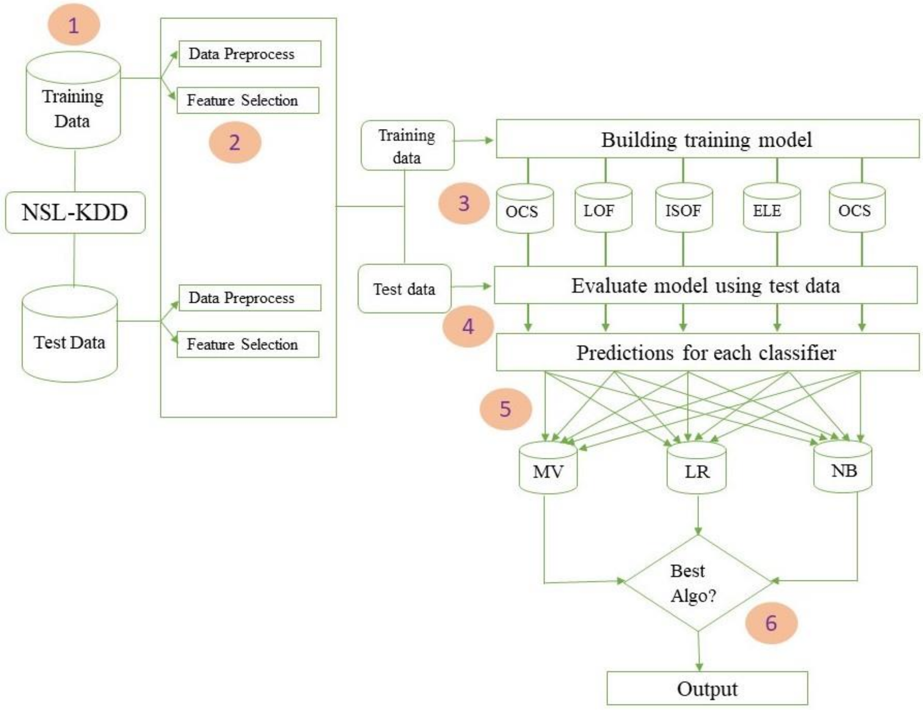

Fig. 1. Process flow of the proposed framework to detect DDoS attacks

4 Experimental Results

The proposed model has been implemented in python and the ML tool scikit-learn [29], a python library to model the training data and to evaluate that model using test data. In experimentation, raw data was extracted from the NSL-KDD website and then converted into a classifier readable format. The classifier readable format came from data preprocessing phase (described earlier in 'Proposed Method' section). The basic idea of data preprocessing is the conversion of non-numeric data content to the corresponding assigned numeric value. Since unsupervised outlier detection models were used in this experiment, the training data should contain the majority of 'normal' or 'non-anomalous' type data instances for better predictions. From the whole training dataset, normal and anomalous data were separated, and a new training dataset was created that contained 99% of normal data and 1% of anomalous data. The reason behind using 1% anomalous data in the training dataset was to make the model run efficiently and accurately in detecting anomaly by learning from normal behavior. The additional percentage of noise (anomalous data) was added later to measure the framework's efficiency. After separating normal and anomalous data from the training data, 67343 data were found as normal (See Table 1.). 1% of this normal data which is 673, was added with anomalous data to create a new or modified training dataset. On the other hand, test dataset wasn't separated but only 1000 amount of random data instances were chosen from test data and created a modified test dataset to evaluate the training model. For both cases, the 'class' column from the dataset was removed as we used unsupervised methods in this framework. However, the 'class' column from the test dataset is preserved to use it later in determining the accuracy of all three ensemble classifiers, and training as well as testing purposes for logistic regression and naïve Bayes models.

In the early phase of the experimental section, five different classifiers were used to build five different training models by using training dataset. As the goal of this research is to detect existing and new DDoS attacks pattern, outlier and novelty detection type classifier was the highest priority on selecting classifiers. Here, four different types of outlier and novelty detection classifiers were chosen, namely One-Class SVM, Local Outlier Factor, Isolation Forest, and Elliptic Envelope. As four is the even number and there might be a chance of a tie situation while choosing an outcome using majority voting ensemble, the next odd number five was chosen as the classifier count in this experiment. Initially, we have experimented with different hyper-parameter values of these classifiers and based on the accuracy, we have selected the best five hyperparameter combinations. The details of the classifiers with hyperparameter combinations used in our experiment are listed in Table 4.

Table 4. Outlier and novelty detection classifier details

Classifier Name	Short Code	Hyperparameters
One Class SVM-Poly kernel	OCSVM-1 or OCSVM- Poly	nu=0.2, kernel="poly", gamma=0.1
One Class SVM-Linear kernel	OCSVM-2 or OCSVM- Linear	nu=0.2, kernel="linear", gamma=0.1
Local Outlier Factor	LOF	n_neighbors=20, contamination=0.22, novelty=True
Isolation Forest	ISOF	behaviour='new', max_samples=100, random_state= RandomState(listLength), contamination=0.2
Elliptic Envelope	ELE	support_fraction=1, contamination=0.2, random_state = RandomState(listLength)

By using different combinations from Table 4., five different training models were built from the modified training set. Then test dataset was used to evaluate each model.

In terms of performance evaluation, Confusion Matrix, True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR), Precision, Recall, and F-Measure are used very frequently. Confusion Matrix has three main terms: Sensitivity, Specificity, and Accuracy which can be defined as follows:

$$Sensitivity = \frac{TPR}{(TPR+FNR)} \quad (1)$$

$$Specificity = \frac{TNR}{(TNR+FPR)} \quad (2)$$

$$Accuracy = \frac{(TPR+TNR)}{(TPR+TNR+FPR+FNR)} \quad (3)$$

Also, Precision, Recall, and F-Measure are another three important performance metrics that are used to evaluate a model. Those terms can be defined in terms of TP, TN, FP, FN from equations (4), (5) and (6).

$$Precision\ (P) = \frac{TP}{(TP+FP)} \tag{4}$$

$$Recall\ (R) = \frac{TP}{(TP+FN)} \tag{5}$$

$$F - Score = \frac{2PR}{PR} \tag{6}$$

Table 5 shows the details of performance metrics for these classifiers when they trained their model by using a single classifier.

After training with the single classifier, a top label classifier or technique was used to ensemble these five classifiers. Majority voting was considered as a baseline, then logistic regression (LR) and naïve Bayes (NB) were used to ensemble these classifiers.

Table 5. Performance metrics when single classifier used.

Classifier	OC SVM-Poly kernel	OC SVM-Linear kernel	LOF	Isolation Forest	Elliptic Envelope
Accuracy	0.863	0.863	0.570	0.893	0.900
FPR	0.075	0.077	0.332	0.079	0.092
Precision	0.866	0.884	0.496	0.890	0.878
Recall	0.780	0.782	0.438	0.855	0.890
F-1 Score	0.829	0.830	0.465	0.872	0.884

In majority voting, the nominal class label which was predicted the most from the unsupervised outputs will then be selected as the final output. On the other hand, LR and NB are two supervised models that require a class label to learn. Here, we used those two models to combine the outputs of five unsupervised models. To train those models, a tagged dataset was required to create using the unsupervised models' outputs as features and the class label that was removed in the data preprocessing phase. Maximum-likelihood estimation is the key mechanism for both logistic regression and naïve Bayes to predict the ensemble outputs. Fig. 2. shows the graphical representation of the comparison of ensemble classifiers' performance metrics with single classifier models.

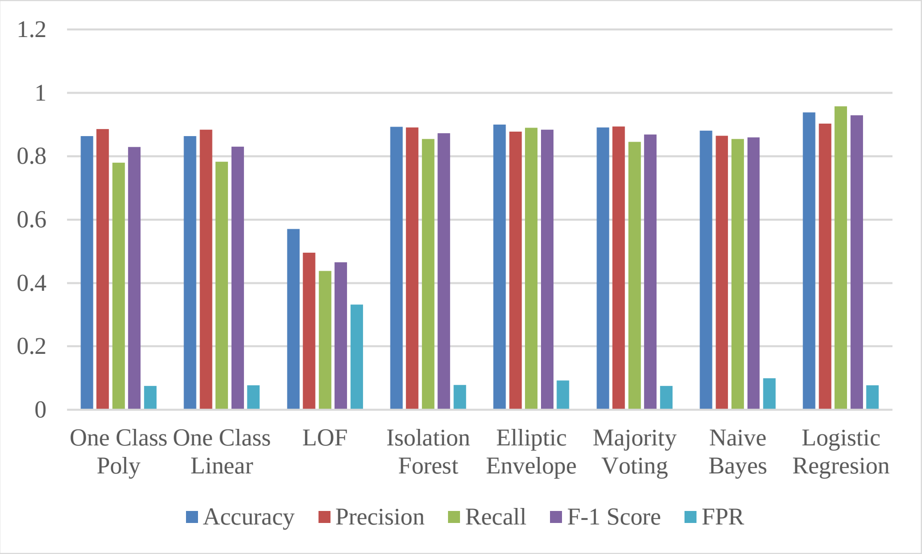

Fig. 2. Performance metrics of five single classifiers compared to ensemble classifiers.

As mentioned in Section 3.3, twelve different feature sets where each of the features is relevant for DDoS attack were considered to build training models. Fig. 3. shows the accuracy of three different types of ensemble techniques with respect to twelve different feature sets and we found that feature set 4 (FS-4) has the best accuracy when Logistic Regression was used to ensemble.

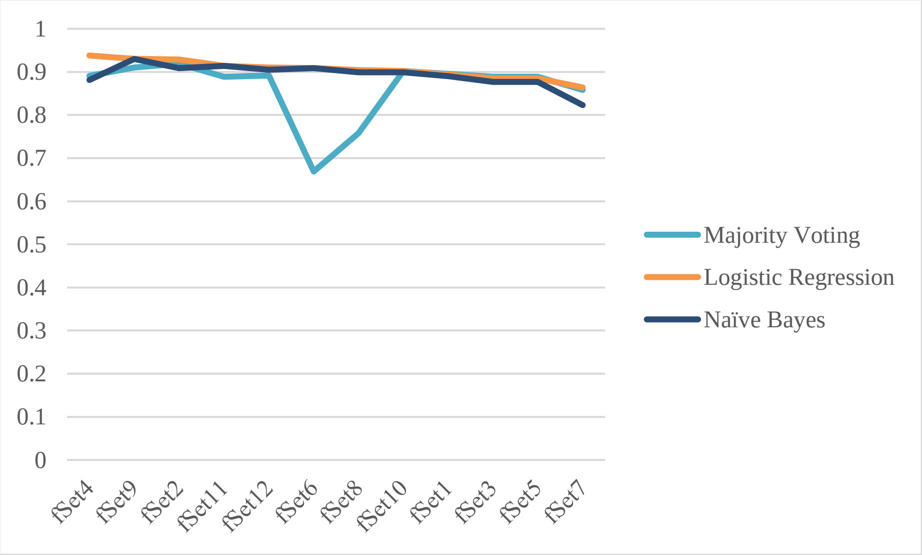

Fig. 3. Accuracy of ensemble classifiers with respect to different feature sets.

As mentioned earlier, the majority of the data instances were normal data and a very few amounts (1%) of noise (abnormal data) mixture was added to build a modified training dataset. To verify our framework's stability and efficiency with the increase of noise (adding more anomalous data into dataset) added with the training dataset, we varied the noise amounts from 1% to 5% and measured the performance metrics. Fig. 4. shows the deviation of performance by adding noise with the training data.

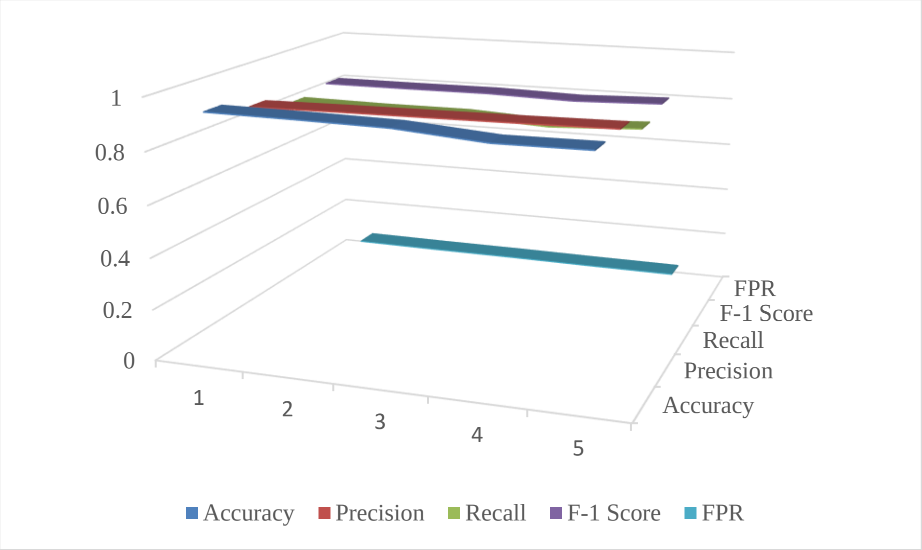

Fig. 4. Performance measurement with respect to adding noise

The proposed framework was tested and verified with both single classifiers and ensemble classifiers to get a higher detection accuracy with a lower false positive rate. Logistic Regression based ensemble model was found with the highest detection accuracy and lowest false positive rate among all five different classifiers: One-Class SVM, Local Outlier Factor, Isolation Forest, and Elliptic Envelope. Table 6. shows the overall comparison among all classifiers including single classifiers, ensemble classifiers as well as some existing research outcomes. 'N/A' refers to the results that are not mentioned in those existing researches.

Table 6. Performance metrics comparison of single classifiers, ensemble classifiers, and existing research

Classifier	Type	Accuracy	FPR	Precision	Recall	F-1 Score
OCSVM-1	Single	0.863	0.075	0.886	0.780	0.829
OCSVM-2	Single	0.863	0.077	0.884	0.782	0.830
LOF	Single	0.57	0.332	0.496	0.438	0.465
ISOF	Single	0.893	0.079	0.890	0.855	0.872
ELEC	Single	0.90	0.092	0.878	0.890	0.884
MV	Ensemble	0.891	0.075	0.894	0.845	0.869

LR	Ensemble	0.938	0.077	0.903	0.958	0.930
NB	Ensemble	0.881	0.099	0.865	0.855	0.860
GAR [30]	Single	0.773	N/A	N/A	N/A	N/A
IG-GAR [30]	Ensemble	0.789	N/A	N/A	N/A	N/A
SU-GAR [30]	Ensemble	0.776	N/A	N/A	N/A	N/A
LDA-NB-kNNCF [31]	Ensemble	0.82	N/A	N/A	N/A	N/A
LOO-OAR-SVM [32]	Ensemble	0.827	N/A	N/A	N/A	N/A

ROC curve (Receiver Operating Characteristics) is a probability curve. In anomaly detection, higher the ROC, better the model is at distinguishing the anomalous traffic from the normal one. The ROC curve is plotted with True Positive Rate (TPR) against the False Positive Rate (FPR) where TPR is on y-axis and FPR is on the x-axis. Fig. 5. shows the ROC curve for DDoS classification in this experiment.

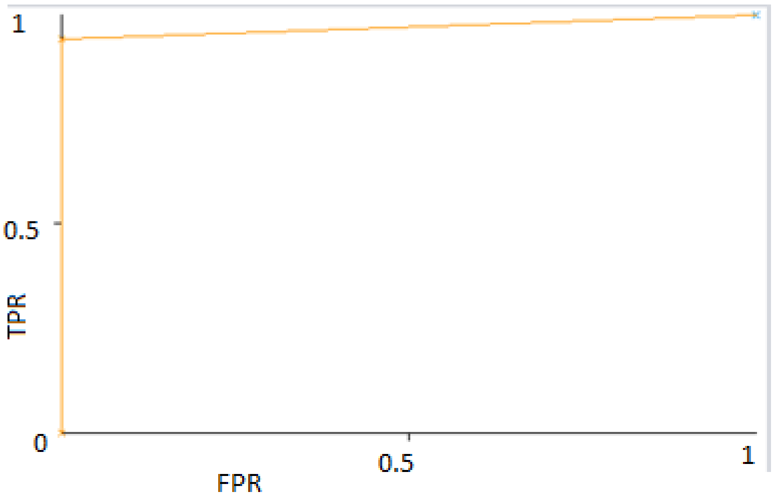

Fig. 5. ROC curve for DDoS classification

5 Conclusion

In this research, the goal was to detect DDoS attacks using an unsupervised ML ensemble. Classifiers from various classifier families of outlier and novelty detection type have been chosen to build the proposed framework. Initially, single classifiers were used to measure the performance metrics in detecting DDoS attacks. On top of these five individual classifiers (One class SVM: two different hyperparameters, Local outlier factor, Elliptic envelope, and Isolation forest), an ensemble with majority voting was applied as a baseline. Naïve Bayes and logistic regression were then used to ensemble these five classifiers again to get a better detection accuracy. In our experiment, Logistic regression based ensemble has the best performance measures that are not only compared to baseline majority voting and Naïve Bayes ensemble but also with the models where the single classifiers were used. In addition, it was also observed from the experimental results and compared to existing research that logistic regression based ensemble while using feature set 4 (FSet-4) has the best detection accuracy, high precision and recall, F1 score, and low false positive rate. The proposed model is not only capable of detecting existing DDoS attacks, but also using outlier detection classifiers, it has the capability to detect unseen or new DDoS attack.

In this research, only one dataset was considered for experimentation and we plan to continue our experiments with other different datasets using the proposed framework. The dataset that we have used was an offline data, hence we have the limitation of experimenting with online data. Twelve different feature sets have been chosen from existing research for experimentation. In the future, we plan to reduce the features on our own using different feature reduction techniques and domain knowledge. With this research as the base, we will consider deep learning methods and software agents [33] in detecting DDoS attacks more accurately.

References

1. Lee, Y.-J.; Baik, N.-K.; Kim, C.; Yang, C.-N. (2018): Study of detection method for spoofed ip against DDoS attacks. Personal and Ubiquitous Computing, vol. 22, no. 1, pp. 35-44.
2. NETSCOUT Report, https://www.netscout.com/report/, last accessed 2019/7/10.
3. Specht, S. M. and Ruby B. L. 2004. Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures. In Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems.
4. Dietterich, Thomas G. "Ensemble methods in machine learning." International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000
5. Aburomman, Abdulla Amin, and Mamun Bin Ibne Reaz. "A survey of intrusion detection systems based on ensemble and hybrid classifiers." Computers & Security 65 (2017): 135-152.
6. Noureldien, N. A., and I. M. Yousif. "Accuracy of machine learning algorithms in detecting DoS attacks types." Science and Technology 6.4 (2016): 89-92.
7. Olusola, Adetunmbi A., Adeola S. Oladele, and Daramola O. Abosede. "Analysis of KDD'99 intrusion detection dataset for selection of relevance features." Proceedings of the World Congress on Engineering and Computer Science. Vol. 1. WCECS, 2010.
8. Osanaiye, Opeyemi, et al. "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing." EURASIP Journal on Wireless Communications and Networking 2016.1 (2016): 130.
9. Ambusaidi, Mohammed A., et al. "Building an intrusion detection system using a filter-based feature selection algorithm." IEEE transactions on computers 65.10 (2016): 2986-2998.
10. Gaikwad, D. P., and Ravindra C. Thool. "Intrusion detection system using bagging ensemble method of machine learning." 2015 International Conference on Computing Communication Control and Automation. IEEE, 2015.

11. Shrivas, Akhilesh Kumar, and Amit Kumar Dewangan. "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set." International Journal of Computer Applications 99.15 (2014): 8-13.
12. Tesfahun, Abebe, and D. Lalitha Bhaskari. "Intrusion detection using random forests classifier with SMOTE and feature reduction." 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies. IEEE, 2013.
13. Haq, Nutan Farah, et al. "Application of machine learning approaches in intrusion detection system: a survey." IJARAI-International Journal of Advanced Research in Artificial Intelligence 4.3 (2015): 9-18.
14. Yusof, Ahmad Riza'ain, Nur Izura Udzir, and Ali Selamat. "Systematic literature review and taxonomy for DDoS attack detection and prediction." International Journal of Digital Enterprise Technology 1.3 (2019): 292-315.
15. Belavagi, Manjula C., and Balachandra Muniyal. "Performance evaluation of supervised machine learning algorithms for intrusion detection." Procedia Computer Science 89 (2016): 117-123.
16. Ashfaq, Rana Aamir Raza, et al. "Fuzziness based semi-supervised learning approach for intrusion detection system." Information Sciences 378 (2017): 484-497.
17. Perez, Deyban, et al. "Intrusion detection in computer networks using hybrid machine learning techniques." 2017 XLIII Latin American Computer Conference (CLEI). IEEE, 2017.
18. Villalobos, Juan J., Ivan Rodero, and Manish Parashar. "An unsupervised approach for online detection and mitigation of high-rate DDoS attacks based on an in-memory distributed graph using streaming data and analytics." Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies. ACM, 2017.
19. Jabez, Ja, and B. Muthukumar. "Intrusion detection system (IDS): anomaly detection using outlier detection approach." Procedia Computer Science 48 (2015): 338-346.
20. Smyth, Padhraic, and David Wolpert. "Stacked density estimation." Advances in neural information processing systems. 1998.
21. Hosseini, Soodeh, and Mehrdad Azizi. "The hybrid technique for DDoS detection with supervised learning algorithms." Computer Networks 158 (2019): 35-45.
22. Canadian Institute for Cybersecurity, Datasets/ NSL-KDD, https://www.unb.ca/cic/datasets/nsl.html, last accessed 2019/7/10
23. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.
24. Das, Saikat, Ahmed Mahfouz, Deepak Venugopal, and Sajjan Shiva. "DDoS Intrusion Detection through Machine Learning Ensemble." 2019 IEEE International Conference on Software Quality, Reliability and Security (QRS). IEEE Computer Society, 2019
25. One-Class classification, https://en.wikipedia.org/wiki/One-class_classification, last accessed 2019/7/10
26. Microsoft, One-Class Support Vector Machine, https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/one-class-support-vector-machine, last accessed 2019/7/10
27. Scikit learn, Novelty and Outlier Detection, https://scikit-learn.org/stable/modules/outlier_detection.html, last accessed 2019/7/10
28. Scikit learn, Isolation Forest, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html, last accessed 2019/7/10

29. Scikit learn, Homepage, https://scikit-learn.org, last accessed 2019/7/10
30. Kanakarajan, Navaneeth Kumar, and Kandasamy Muniasamy. "Improving the accuracy of intrusion detection using GAR-Forest with feature selection." Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015. Springer, New Delhi, 2016.
31. Pajouh, Hamed Haddad, GholamHossein Dastghaibyfard, and Sattar Hashemi. "Two-tier network anomaly detection model: a machine learning approach." Journal of Intelligent Information Systems 48.1 (2017): 61-74.
32. Pervez, Muhammad Shakil, and Dewan Md Farid. "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs." The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014). IEEE, 2014.
33. Das, Saikat, and Sajjan Shiva. "CoRuM: Collaborative Runtime Monitor Framework for Application Security." 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). IEEE, 2018.