2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)

Empirical Evaluation of the Ensemble Framework for Feature Selection in DDoS Attack

Saikat Das, Deepak Venugopal, and Sajjan Shiva
Department of Computer Scinece
The University of Memphis
Memphis, TN, USA
{sdas1, dvngopal, sshiva}@memphis.edu

Frederick T. Sheldon
Department of Computer Science
University of Idaho
Moscow, ID, USA
sheldon@uidaho.edu

Abstract— **Over the past two decades, Distributed Denial of Service (DDoS) attacks have been responsible for most of the catastrophic failures in the Internet causing a huge amount of disruption of services across all sectors of the economy. Almost every year this attack scores top among all other attacks in terms of the cost to the overall global economy. Machine Learning (ML)-based Intrusion Detection Systems (IDSs) heal the global economy with the goal of reducing the prevalence of cyber incidents, such as DDoS. In an ML classification problem, the feature selection process, aka feature engineering, is treated as a mandatory pre-processing phase that potentially reduces the computational complexity by identifying important or relevant features from the original dataset and results in the overall improvement of classification accuracy. In this paper, we propose an ensemble framework for feature selection methods (EnFS) that combines the outputs of seven well-known feature selection methods using the majority voting (MV) technique and produces an optimal set of features. In the evaluation of the proposed framework, an extensive experiment was performed using the intrusion detection benchmark dataset NSL-KDD [1]. Furthermore, using the optimal feature set, we have experimented with ensemble supervised ML framework [2] for the same dataset that demonstrated the efficacy of our approach by producing greater accuracy and negligible false alarms compared to existing approaches.**

Keywords—Feature selection, Ensemble Feature Selection, EnFS, DDoS, Intrusion Detection System, Machining Learning, Ensemble Machine Learning.

I. INTRODUCTION

Cybersecurity has become a very serious problem not only for national security but also for organizations. Traditionally, attackers leverage the vulnerabilities to exploit the new and unprecedented opportunities that are available to them to profit from and/or disrupt e-commerce. For example, Cybersecurity Ventures estimates that the economic loss due to cybercrime, will soon reach the level of $6 trillion annually by 2021 [3]. Distributed Denial of Service (DDoS) is one of the infamous cyber-attacks through which intruders render the victim server's bandwidth, services, and resources unavailable to a legitimate user. According to the Worldwide Infrastructure Security Report [4], the DDoS attack has already reached 1.7 Terabyte per second in 2018, and is dominating the cyber-attack arena.

In response, to mitigate the severity of these attacks, Intrusion Detection Systems (IDSs) are being used in ways to scrutinize attempted attacks anticipating that there will be follow on attempts. Machine Learning (ML) based techniques are being employed thereby to incorporate active intelligence and make detection more effective in countering such attacks. Combining ML into IDS can improve the accuracy, reliability, and resiliency of networked public facing information infrastructures compared with standard signature-based IDSs. Feature Selection (FS) methods used in the pre-processing phase have the greatest potential to improve the performance of ML classifications when combined with IDSs. FS uses various techniques to extract a subset of features within the data to better discriminate between classes resulting in needing fewer features and less processing time. Thus, features that do not help to discriminate the class are eliminated because they do not contribute to the models' prediction. FS methods can be classified into three categories, namely (1) filter-based, (2) wrapper-based, and (3) embedded methods. Here, we subsume all three types of methods within our proposed framework by combining them in a way that eliminates the inherent bias and drawbacks when used individually.

The contribution of this study includes the stepwise process as well as an empirical validation using the NSL-KDD dataset to evaluate our approach in the case of DDoS attack detection. Recent studies show that ensemble technique for feature selection improves the performance of models in several ways by i) removing non-discriminating features, ii) identifying important features which have a high correspondence with the target class [5], iii) finding some features that produce weak performance individually, and strong performance when used in a group [6], etc. In this research, we propose an ensemble framework EnFS that combines seven FS methods using the majority voting (MV) technique. The EnFS framework codifies a systematic and repeatable method that provides better results (prediction accuracy) in less computational time (more efficient), and maintains such benefits as reducing overfitting, reducing classification and training time, etc. Furthermore, an extensive set of experiments have been conducted with fifteen different FS methods. Using a grid-search algorithm, we chose the best seven methods from the various selection method categories. Then, by using the reduced feature set obtained from the EnFS framework, we performed data classification with our previous supervised ensemble ML framework [2] to identify the best performances. Consequently, using the well-known NSL-KDD dataset, we could clearly demonstrate that the subset of features produced by our ensemble approach (i.e., EnFS) yields better accurate results. This was true for several different classification methods as compared to using a single FS criterion or without using any FS method.

978-1-7281-6550-9/20/$31.00 ©2020 IEEE
DOI 10.1109/CSCloud-EdgeCom49738.2020.00019

The remainder of the article is organized as follows: In Section II, we review several FS methods within this problem domain. Section III briefly describes the FS methods used in the proposed framework. The EnFS framework is presented in Section IV. The EnFS validation experiments are presented with the ensemble supervised classification model using the NSL-KDD dataset in Section V. The results of the experiments are further discussed in Section VI. In Section VII, we present some conclusions and future aspirations.

II. Literature Review

Machine Learning (ML) algorithms focus on the development of computer programs where they provide the systems with the ability to automatically learn and improve from experience without the intervention of humans and without being explicitly programmed. The feature selection (FS) process is one of the vital ML pre-processing phases where it removes unwanted and irrelevant features with the goal of improving prediction (i.e.; detection) accuracy and reducing computational complexity.

Dash and Liu [7] mentioned four basic procedures in a FS method. The procedures are generation, evaluation, stopping, and validation. Various support vector machine (SVM) models with NSL-KDD dataset [8], genetic-fuzzy rule mining approach [9], genetic algorithm approach [10], mutual information-based [11] techniques, filter-based methods [12], etc. were used in feature selection process for intrusion detection systems. Several FS methods are also found in detecting DDoS attacks such as detecting DDoS in cloud computing [8][13], detecting robust backscatter DDoS [14], chi-square and information gain FS methods [15] in detecting general DDoS attacks, etc. In addition, supervised [2] and unsupervised [16] ensemble frameworks were also used to detect DDoS attacks with better accuracy.

A significant number of surveys and taxonomies of FS methods are found from the recent research. Chandrashekar and Sahin [17] conducted a detailed survey on various FS methods using the DARPA dataset. A taxonomy and survey on semi-supervised FS methods were accomplished by Sheikhpour, Razieh, et al. [18] using several datasets. Khalid, Samina et al. [19] performed a brief survey on well-known FS methods to check the suitability of different FS and feature extraction techniques in certain situations based on experiments. A survey of various selection algorithms that helps decide which algorithm to use in certain situation [20], a FS survey for gaussian mixture models and hidden Markov models [21], taxonomy of FS algorithms in intrusion detection systems [22], etc. are found to depict the state-of-the-art of FS methods.

From the above studies, most of the researchers other than Osanaiye, Opeyemi, et al. [5], provided either a detailed survey of FS methods in general and/ or specific research areas, or implemented various FS methods with several types of datasets. None of them mentioned combining several selection methods and demonstrated their outcomes. Osanaiye, Opeyemi, et al. [5] used an ensemble based multi-filter (only filter-based) selection method although, they did not consider the other two types of selection methods (i.e.; wrapper-based and embedded). In this research, we propose an ensemble framework for feature selection methods (EnFS) where all three types of methods are used and combined using a majority voting technique to extract a valid minimal subset of features that improves the performance of DDoS detection problem.

III. Feature Selection Methods

We consider a high dimensional dataset with n data instances and m columns (e.g.; features) i.e.; the data matrix is $X \in \mathbb{R}^{n \times m}$, and a target variable (level) is y. A target variable can be either continuous or discrete. A feature selection (FS) algorithm selects a subset of $p \ll m$ features i.e.; $Xs \in \mathbb{R}^{n \times p}$, where p features are most relevant to the target variable y [23]. The subsequent sections discuss briefly three major categories of FS methods and the corresponding FS methods that fall under each category.

A. Filter-based Methods

Filter-based methods utilize the underlying statistical characteristics of the input data during ML model training time. A correlation value between the feature and the target variable is calculated for each feature. A general filter-based FS process can be accomplished by selecting the features for which the correlation value exceeds a threshold value [23].

a) Pearson's Correlation: Pearson correlation coefficient (r) is a statistical measurement that calculates the linear correlation between two random variables x and y using the formula in (1). The value of Pearson's r can be +1, 0, or −1; where +1 denotes a positive linear correlation, 0 denotes no linear correlation, and −1 denotes a negative linear correlation [24].

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2+\sum(y-\bar{y})^2}} \quad (1)$$

b) Chi-Square: In a contingency table, Chi-Square test determines the relationship between two or more random variables i.e.; tells how much difference exists between observed frequencies and expected frequencies, while assuming no relationship among the data instances using the formula in (2). The test statistic is computed from a ${\chi_c}^2$ distribution in order to make the null hypothesis true by evaluating how close the observed and expected frequency values are [25].

$${\chi_c}^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

where, ${\chi_c}^2$ is the chi-square distribution with c degrees of freedom, and O and E are the observed and the expected values, respectively. If the chi square test statistic is very small, it means that the observed data fit very well with the expected data i.e.; both data have a relationship. Otherwise, the observed data don't fit well with the expected data i.e.; there is no relationship between these two datasets.

c) Mutual Information: The measurement of the mutual information between two random variables X and Y can be obtained by doing the reduction in uncertainty for one random variable, given that the other random variable's value is already known using the formula in (3).

$$I(X;Y) = \int_X \int_Y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \qquad (3)$$

where, $p(x, y)$ denotes the joint probability density function of two random variables X and Y. The marginal density functions of two random variables X and Y are $p(x)$ and $p(y)$, respectively. When two random variables X and Y are independent, the joint probability density function is equal to the product of two marginal density functions, i.e.; $p(x, y) = p(x)\ p(y)$ which results in the value of the integration (equation (3)) to become zero. So, the stronger relationship between two random variables is determined by the larger value of the integration.

B. *Wrapper-based Methods*

Wrapper-based methods exploit an ML algorithm to evaluate the goodness of features, and the FS process is accomplished by the means of a search problem where different combinations are exhaustively prepared, evaluated, and compared with other combinations.

a) Recursive Feature Elimination: In Recursive Feature Elimination (RFE), the process starts with initializing the predictors with a rank that comes from an initial measure of importance. The very first model is built using the complete set of predictors. Then a smaller set of predictors is used to build the next model, where the smaller set is obtained by removing the least important ones. This process (extracting a smaller set of predictors and building a model) continues recursively to a defined way until a minimum number of predictors are remained.

C. *Embedded Methods*

Several algorithms are used in embedded methods, and they have built-in mechanisms for selecting certain features which are executed during model training time i.e.; the FS process can be completed within the construction of ML algorithms. With its own variable, an ML model performs feature selection as well as classification/regression at the same time.

a) LASSO Regression: Least Absolute Shrinkage and Selection Operator (LASSO) is a regression analysis that is often used as a FS method. To accomplish the FS process, LASSO method performs L1 regularization through which it assigns a constraint on the sum of absolute values of the model parameters and penalizes the regression variable's coefficient by shrinking some of the variables towards zero. After the regularization process, the features having zero values on their regression coefficient are eliminated. Then, a new subset of features can be constructed with the features having non-zero regression coefficients which have strong association with the target variable [26].

b) Logistic Regression (LR) with L1 Penalty: From the statistical point of view, LR models are used to model the probability of an existing class or event, such as normal/abnormal, pass/fail, win/lose, hot/cold, etc. Using L1 regularization in LR, each non-zero coefficient is added as a penalty that forces weak feature coefficients to have a zero value. Here, FS is performed by producing sparse solutions.

c) Random Forests: Random Forests are formed with four to twelve hundred decision trees where each of the trees is built over a random extraction of the observations from the dataset and a random extraction of the features. These trees are uncorrelated since they can't access all features or all observations and therefore less prone to overfitting. Each of the trees is constructed by a sequence of simple yes/no questions based on a single or combination of features. Based on the answers (yes/no), the tree divides the dataset into two buckets; observations that are most likely similar among themselves are put into one bucket, whereas the dissimilar ones are put into another bucket. The importance of each feature is measured based on the purity of each bucket [27].

IV. Proposed Framework

This section provides an overview of our proposed ensemble framework EnFS. The detailed architectural diagram depicting the process flow is given in Fig. 1. It shows the processing phases, namely a) data preprocessing, b) feature selection, c) ensemble selection methods and d) model classification with performance analysis of the detection.

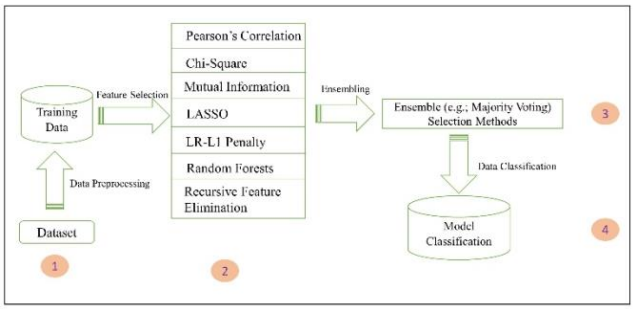

Fig. 1. Process flow of the Ensemble Feature Selection Methods Framework (EnFS)

A. *Data Preprocessing*

There are various subtasks that must be done in a data preprocessing phase, like removing unwanted data, data conversion, scaling, removing invalid data, etc. The detail of data preprocessing is described in Section V-B.

B. *Feature Selection using Individual Methods*

Selecting the right feature and right number of features could lead the classification model to its cherished goal. Feature selection (FS) phase is one of the crucial phases of model classification which can be done by various inbuilt mechanisms or by using domain knowledge. In this proposed framework, we have used seven FS methods (described in Section III) individually to experiment with NSL-KDD dataset and to extract a minimal number of features from each of the methods.

C. *Ensemble Selection Methods*

The goal of the FS methods is to extract a minimal set of features, and using that feature set, ML models can produce better outcomes in different types of classification problems. Using a single selection method may not always produce valid or an accurate number of features, therefore, outcomes from multiple selection methods are likely more trustworthy. Ensembling (combining multiple selection methods) is the primary goal of this work depicted here in this phase to obtain

valid, a more accurate feature set and to produce a higher accuracy and detection rate in DDoS attack detection problems using the extracted feature set.

D. Model Classification

In this phase several ML models are considered and analyzed to evaluate the accuracy of the feature set that are obtained from the previous section. The detail of model classification is described in Section V-D. In addition, we experimented with full features (i.e.; no FS methods were applied) to compare the results and to validate that the FS process is necessary.

V. Experiments

This section presents a set of experiments and their details using our proposed EnFS framework.

A. Dataset

The NSL-KDD dataset is used in this experiment which was created by curating the well-known KDD'99 dataset [28]. The dataset consists of 41 predictor attributes and 1 target attribute which indicates that if the corresponding set of predictor attributes can be any of 39 attacks [1]. These attacks fall under four main attack categories: DDoS, U2R, R2L or probe type attacks. Out of these 39 attacks, 10 of them are DDoS type attacks, namely back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, and udpstorm. To experiment with the NSL KDD dataset, we used 113268 data instances for training purpose and 17164 data instances for testing purpose.

B. Data Preprocessing

In the NSL-KDD dataset, there are some text-based categorical variables, namely protocol type, service, and flag data. For data classification, these variables are converted into numeric values by label-encoding, i.e., by converting to integer based categorical variables. This creates a binary column for each category and returns a sparse matrix or dense array. Since the data in each column are varied within a different range, we used a couple of scaling mechanisms (e.g.; Standard and Min-Max) to normalize the data. The standard scaling standardizes features by removing the mean and scaling to unit variance. On the other hand, Min-Max scaling transforms features by scaling each feature to a given range(e.g.; 0 to 1).

C. Feature Selection

In this section, we have performed two layers of experiments. Initially, seven selection methods are chosen from among fifteen selection methods based on accuracy, performance, and other metrics using a manual grid search algorithm. The search algorithm selected all three types (filter-based, wrapper-based and embedded methods) of selection methods. We have selected the top seven FS methods from among fifteen that produced the best results. After completing method selection, seven FS methods were used individually to extract the features, where each of the methods selected a different subset of features. Subsequently, the majority voting (MV) technique is used to ensemble all seven methods. Finally, a combined subset of features is extracted that was further used in data classification.

D. Model Classification

We utilized our ensemble supervised classification framework (from previous research [2]) to evaluate the models' performances using the feature set obtained from the previous sub-section V-C. For the supervised ensemble framework, Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), Neural Network (NN), and Logistic Regression (LR) were used for individual data classification, and on top of those classifications another layer of classification was performed to ensemble them. We have used various ensemble techniques, like Majority Voting (Ens_MV), Logistic Regression (Ens_LR), Naive Bayes (Ens_NB), Neural Network (Ens_NN), Decision Tree (Ens_DT), and Support Vector Machine (Ens_SVM). All these methods and the framework were used here to analyze the efficacy of the DDoS classification problem using the feature set obtained from this research.

VI. Results and Discussion

In this section, we describe evaluation metrics used to evaluate the accuracy of the framework. In addition, the results obtained from several experiments are illustrated in detail.

A. Evaluation Metrics

Accuracy, Precision, Recall, F-1 Score, and False Positive Rate are the evaluation metrics that we have used to measure the performance for the classification models. These metrics are defined by four measurements: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In addition, Receiver Operating Characteristics (ROC) curve is a probability curve that was used to evaluate the models based on TPR and FPR. Generally, a ROC curve is plotted with True Positive Rate (TPR) (in y-axis) against the False Positive Rate (FPR) (in x-axis). In anomaly detection, the higher the ROC, the better the model is at distinguishing anomalous traffic.

B. Discussion of Results

Experimental results performed by the EnFS along with seven FS methods are analyzed here. The goal of this experiment is to extract important features using several selection methods as well as find the exact number of features after combining all these methods using ensemble technique (i.e.; MV). Table I shows the features that were extracted from seven FS methods.

TABLE I. Extracted features from seven FS methods

F#	Method	Extracted Features
F#1	Pearson Correlation	['dst_host_rerror_rate', 'dst_host_srv_diff_host_rate', 'srv_diff_host_rate', 'service', 'dst_host_count', 'flag', 'logged_in', 'count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'serror_rate', 'srv_serror_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'same_srv_rate']
F#2	Chi-Square	['service', 'flag', 'logged_in', 'count', 'serror_rate', 'srv_serror_rate', 'srv_rerror_rate', 'same_srv_rate', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'dst_host_rerror_rate']
F#3	Mutual Information	['service', 'flag', 'src_bytes', 'dst_bytes', 'same_srv_rate', 'diff_srv_rate']
F#4	LASSO	['duration', 'protocol_type', 'wrong_fragment', 'logged_in', 'srv_count', 'srv_serror_rate',

		'srv_rerror_rate', 'same_srv_rate', 'diff_srv_rate', 'dst_host_srv_diff_host_rate']
F#5	Logistic Regression with L1 Penalty	['duration', 'protocol_type', 'flag', 'wrong_fragment', 'hot', 'root_shell', 'num_file_creations', 'is_guest_login', 'count', 'srv_count', 'srv_serror_rate', 'srv_rerror_rate', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_srv_diff_host_rate']
F#6	Random Forests	['service', 'flag', 'dst_bytes', 'count', 'serror_rate', 'srv_serror_rate', 'same_srv_rate', 'diff_srv_rate', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate']
F#7	Recursive Feature Elimination	['duration', 'protocol_type', 'flag', 'wrong_fragment', 'hot', 'logged_in', 'is_guest_login', 'count', 'srv_count', 'srv_serror_rate', 'srv_rerror_rate', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_srv_diff_host_rate', 'dst_host_serror_rate']

The majority voting (MV) technique (i.e.; a feature can be selected if more than half of the methods select it) is used here to ensemble all seven selection methods' output. Since seven selection methods are used in this research, ensemble framework selects those features who have been selected by any four of the seven methods.

Table II enumerates that EnFS selects 11 features, and each of the features is demonstrated in a form of ✓ or ✕ mark that shows the selection by individual methods. In addition, a score card counter is added to support the MV technique (i.e.; Total Count)

TABLE II. SCORE CARD: EXTRACTED FEATURES USING ENSEMBLE FEATURE SELECTION FRAMEWORK (*ENFS*).

Feature Name	Pearson	Ch-Square	Mutual Info	LASSO	LR-L1	RF	FRE	Total Count
srv_serror_rate	✓	✓	✕	✓	✓	✓	✓	6
flag	✓	✓	✓	✕	✓	✓	✓	6
same_srv_rate	✓	✓	✓	✓	✕	✓	✕	5
count	✓	✓	✕	✕	✓	✓	✓	5
dst_host_srv_count	✓	✓	✕	✕	✓	✓	✓	5
dst_host_serror_rate	✓	✓	✕	✕	✕	✓	✓	4
logged_in	✓	✓	✕	✓	✕	✕	✓	4
service	✓	✓	✓	✕	✕	✓	✕	4
dst_host_same_srv_rate	✓	✓	✕	✕	✓	✕	✓	4
dst_host_srv_diff_host_rate	✓	✕	✕	✓	✓	✕	✓	4
srv_rerror_rate	✕	✓	✕	✓	✓	✕	✓	4

To evaluate those selections (both EnFS and the individual seven methods), we performed three types of experiments using the ensemble supervised model. Initially, we used the full feature set (i.e.; no FS method was applied), then seven feature sets obtained from seven selection methods, and finally we used feature set obtained from EnFS for model classification. Table III shows the best performed experiments, whereas the full experimental results are available in https://github.com/simplysaikat/EnFS/ . From Table III, it is obvious that the features obtained from EnFS perform better than all other FS methods. In addition, full feature set (i.e.; without applying any selection method) was used in another experimentation to compare and evaluate the necessity of FS methods or framework like EnFS.

TABLE III. BEST PERFORMED CLASSIFICATION RESULTS USING FULL FEATURES, SEVEN FEATURES AND THE FEATURES OBTAINED FROM ENFS.

Method	Model Name	F-1 Score	Accuracy	Precision	Recall	FPR
No FS	Ens_DT	0.884	0.900	0.878	0.890	0.011
PEARSON	Ens_DT	0.882	0.904	0.941	0.830	0.040
CHI2	Ens_DT	0.925	0.936	0.941	0.909	0.043
MUTINFO	Ens_DT	0.869	0.895	0.950	0.801	0.032
LASSO	Ens_NN	0.921	0.936	0.989	0.862	0.007
LRL1	Ens_NB	0.888	0.912	0.982	0.811	0.011
RF	Ens_DT	0.898	0.918	0.977	0.831	0.015
RFE	Ens_SVM	0.893	0.916	0.990	0.814	0.006
EnFS	Ens_DT	0.971	0.975	0.991	0.952	0.006

The ROC curve performance analysis for EnFS framework is shown in Fig. 2. Rest of the ROC curves for seven selection methods are available in https://github.com/simplysaikat/EnFS/tree/master/ROC_AUC/

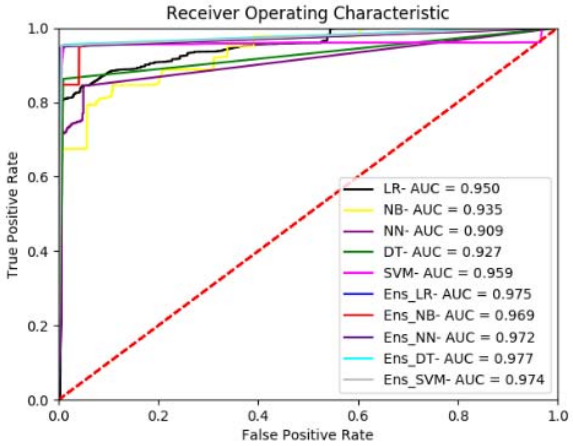

Fig. 2. ROC curve using EnFS framework

Using Table III, a bar chart can be plotted as shown in Fig. 3. It shows the comparison of performances using features from seven methods, from EnFS, and appling no FS method.

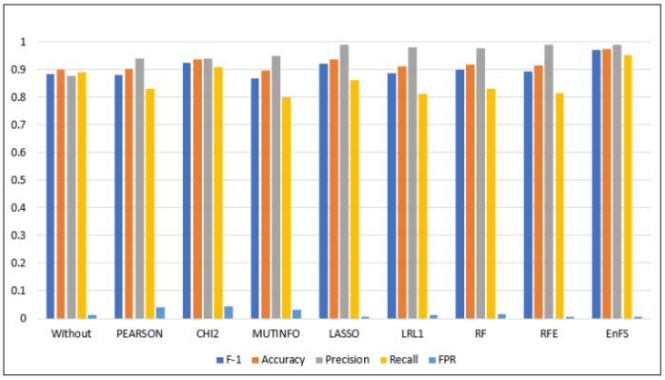

Fig. 3. Performance comparison of EnFS with other seven selection methods and using no selection method.

From all the above figures and comparisons, it is obvious that our ensemble framework for FS methods (EnFS) outperforms any other single selection methods.

VII. Conclusion

Feature selection is a vital part of any classification problem. In this research, we have proposed an ensemble framework for feature selection (EnFS) which combined seven well-known selection methods. The goal of combining these methods is to extract the most accurate set of features that produces better outcomes in detecting DDoS attacks. We have performed three experiments using the i) full feature set initially, then ii) seven feature sets obtained from the seven selection methods, and iii) finally the resultant feature set obtained from our EnFS that used the majority voting technique. The NSL-KDD dataset provided the basis for validating EnFS that reduced the number of features from 41 to 11. Subsequently, we performed an extensive set of experiments using our ensemble supervised ML framework [2] to evaluate the performance of the resulting feature set. As a result of this extensive experimentation, we were able to demonstrate, in this case, that a better performance measurement is achieved in terms of the f-1 score, accuracy, precision, recall, and the false positive rate which is minimized.

On the basis of using these results as a baseline, we plan to expand this approach using interpretable ML with smart agent simulation [29]:

- to better understand why certain features are more relevant than others,
- to gain greater confidence in the conclusions that are key to early detection and prevention of DDoS attacks, and
- to show that the EnFS can play a significant role in providing a frontline defense for these types of attacks and persuasive argument to pursue this approach in other types of intrusion analyses.

References

[1] 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.

[2] Das, Saikat, et al. "DDoS Intrusion Detection Through Machine Learning Ensemble." 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2019.

[3] S. Morgan, "2017 cybercrime report," Cybersecurity Ventures, 2017, last accessed 2020/05/05.

[4] NETSCOUT Report, https://www.netscout.com/report/, last accessed 2020/05/05.

[5] 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.

[6] Bolon-Canedo, Veronica, Noelia Sanchez-Marono, and Amparo Alonso-Betanzos. "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset." Expert Systems with Applications 38.5 (2011): 5947-5957.

[7] Dash, Manoranjan, and Huan Liu. "Feature selection for classification." Intelligent data analysis 1.3 (1997): 131-156.

[8] 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.

[9] Tsang, Chi-Ho, Sam Kwong, and Hanli Wang. "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection." Pattern Recognition 40.9 (2007): 2373-2391.

[10] Stein, Gary, et al. "Decision tree classifier for network intrusion detection with GA-based feature selection." Proceedings of the 43rd annual Southeast regional conference-Volume 2. 2005.

[11] Amiri, Fatemeh, et al. "Mutual information-based feature selection for intrusion detection systems." Journal of Network and Computer Applications 34.4 (2011): 1184-1199.

[12] 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.

[13] Das, Saikat, Ahmed M. Mahfouz, and Sajjan Shiva. "A Stealth Migration Approach to Moving Target Defense in Cloud Computing." Proceedings of the Future Technologies Conference. Springer, Cham, 2019.

[14] Balkanli, Eray, A. Nur Zincir-Heywood, and Malcolm I. Heywood. "Feature selection for robust backscatter DDoS detection." 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops). IEEE, 2015.

[15] Suresh, Manjula, and R. Anitha. "Evaluating machine learning algorithms for detecting DDoS attacks." International Conference on Network Security and Applications. Springer, Berlin, Heidelberg, 2011.

[16] Das, Saikat, Deepak Venugopal, and Sajjan Shiva. "A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning." Future of Information and Communication Conference. Springer, Cham, 2020.

[17] Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40.1 (2014): 16-28.

[18] Sheikhpour, Razieh, et al. "A survey on semi-supervised feature selection methods." Pattern Recognition 64 (2017): 141-158.

[19] Khalid, Samina, Tehmina Khalil, and Shamila Nasreen. "A survey of feature selection and feature extraction techniques in machine learning." 2014 Science and Information Conference. IEEE, 2014.

[20] Molina, Luis Carlos, Lluís Belanche, and Àngela Nebot. "Feature selection algorithms: A survey and experimental evaluation." 2002 IEEE International Conference on Data Mining, 2002. Proceedings. IEEE, 2002.

[21] Adams, Stephen, and Peter A. Beling. "A survey of feature selection methods for Gaussian mixture models and hidden Markov models." Artificial Intelligence Review 52.3 (2019): 1739-1779.

[22] Chen, You, et al. "Survey and taxonomy of feature selection algorithms in intrusion detection system." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2006.

[23] Thomas Huijskens, Feature selection, https://thuijskens.github.io/2017/10/07/feature-selection/, last accessed 2020/05/05.

[24] Wikipedia, Pearson Correlation, https://en.wikipedia.org/wiki/Pearson_correlation_coefficient, last accessed 2020/05/05.

[25] Wikipedia, Chi Squared Test, https://en.wikipedia.org/wiki/Chi-squared_test, last accessed 2020/05/05.

[26] Fonti, Valeria, and Eduard Belitser. "Feature selection using lasso." VU Amsterdam Research Paper in Business Analytics (2017): 1-25.

[27] Towards Data Science, Random Forests, https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f, last accessed 2020/05/05.

[28] Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[29] 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.