Evaluation of Correlation Feature Selection and Random Forest for Network Intrusion Detection

[1] Lubna Mohammed Kunhi, [2] Mustafa Basthikodi,[3] Ahmed Rimaz Faizabadi, [4]Ayshathul Thuhara,[5]Safina Banu,
Dept. of CSE, Bearys Institute of Technology Mangalore.

***Abstract* — Network and system security are of much importance in the present data communication world. The rapid development of the internet in the last few decades has created many security problems related to intrusions on computer and network systems. To detect intrusions using machine learning, an efficient classifier has to be established. To have better predictive accuracy, various feature selection methods are used. Correlation Feature Selection (CFS) and Random Forest techniques are discussed in this paper. Both these methods follow different approaches, the former being a filter method and the latter being an embedded method. A comparison shows that even though Random Forest has higher predictive accuracy than CFS based classifier, it is computationally expensive. Both of these techniques are suitable with their own merits and demerits.**

***Keywords* - CFS, Random Forest, NSL KDD, Network Intrusion**

I. INTRODUCTION

Recent hackers have been inventing new techniques on a daily basis to bypass security layers and to avoid detection. Thus it is the time that we figure out new techniques to defend against such attacks. With the tremendous growth of the internet, attack cases are increasing each day along with modern attack methods. So, it is necessary that we implement an Intrusion Detection technique so that it can detect normal and attack data. Due to increasing incidents of cyber attacks and heightened concerns for cyber terrorism, implementing effective Intrusion Detection mechanisms is an essential task. These techniques automate the process of monitoring and analyzing the events that occur in a computer network, to detect malicious activities.

In order to have a good Intrusion Detection System, the performance measures of the suggested model must be high. This is in fact affected by one of the steps before classification, which is Feature Selection. Feature Selection is the process of selecting those features which contribute most to the output of the model and discarding the less important ones. One reason why feature selection is important is that some features which do not contribute much (or no contribution) may act as noise. This, in turn, affects the accuracy and performance of the model. Also, using only selected features makes the machine learning algorithms to run faster. It reduces complexity and most importantly, overfitting.

This paper presents the different types of feature selection techniques: filter method, wrapper method, and embedded method. A comparison is made among the two most commonly used methods based on accuracy, the number of features selected, the training and testing time required, and performance. All feature selection methods discussed here are suitable to be used in NSL-KDD dataset.

The remainder of the paper is organized as follows. Section II briefly outlines the types of feature selection techniques, mentioned earlier. Section III presents the methodology of two of the feature selection methods in detail. Section IV gives a comparison between the two methods for feature selection. Section V concludes the paper.

II. FEATURE SELECTION METHODS

There are three types of feature selection methods which are categorized as follows:

1. Filter methods: These methods do not depend on the learning algorithm. They select features based on the contribution made by the features to the determination of class output. Thus features are selected on the basis of some scores or statistical measures. It is simple and fast, and more important, feature selection needs to be implemented only once, and different classifiers can be used.
2. Wrapper methods: Unlike filter methods, this class of method requires learning algorithm which is predetermined in order to evaluate each candidate feature selection. It gives better results compared to filter methods however, it is time-consuming, much slower, computationally expensive, and has a high risk of overfitting and is specific to a classifier.
3. Embedded methods: These Methods searches for an optimal subset of features in the classifier construction. It is specific to the learning algorithm but is less computationally intensive than the wrapper method.

III. METHODOLOGY

1. *Correlation Feature Selection:*

Correlation Feature Selection (CFS) is one of the filter methods. In this method, the statistical characteristics are taken into accounts such as the correlation between a feature and the output class or inter-correlation between features, without involving a learning algorithm. Correlated features are those which are influenced by some similar mechanisms and they vary together. It may be a positive correlation (+1), a negative correlation (-1) or no correlation (0). If a small change in a feature reflects on the output class, as well as if the change is proportional and very high, we can say that it is a good idea to keep it around. The bias of this approach is towards subsets that contain features that are highly correlated with the class and uncorrelated with each other. Irrelevant features can be discarded because they have low correlation with the class. The
merit of the feature subset is calculated in (1).

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} \qquad (1)$$

where M_s is the the "merit" , S is the subset and k is the number of features, $\overline{r_{cf}}$ denotes the mean feature-class correlation, and $\overline{r_{ff}}$ denotes the average feature-feature correlation which is calculated as follows:

$$\overline{r_{cf}} = \frac{r_{cf1}+r_{cf2}+\cdots+r_{cfk}}{k} \qquad (2)$$

$$\overline{r_{ff}} = \frac{r_{f1f2}+r_{f1f3}+\cdots+r_{fkf1}}{\frac{k(k-1)}{2}} \qquad (3)$$

1. *Random Forest:*

Random Forest is often used as a feature selection method. This is because the strategies used by random forests naturally rank by how well they improve the purity of the node. Nodes having the impurity occur at the end of the tree. Thus, by pruning trees below a particular node, we can create a subset of the most relevant features. Nodes with the greatest decrease in impurity will appear at the start of the tree, while nodes with the least decrease will be in the end. One advantage of this method is, it is easy to compute, it tells how much each feature is contributing to the decision. Feature selection using Random Forest falls under the category of embedded methods. It combines the qualities of filter and wrapper methods. These are implemented by algorithms that have feature selection methods inbuilt. This increases their accuracy.

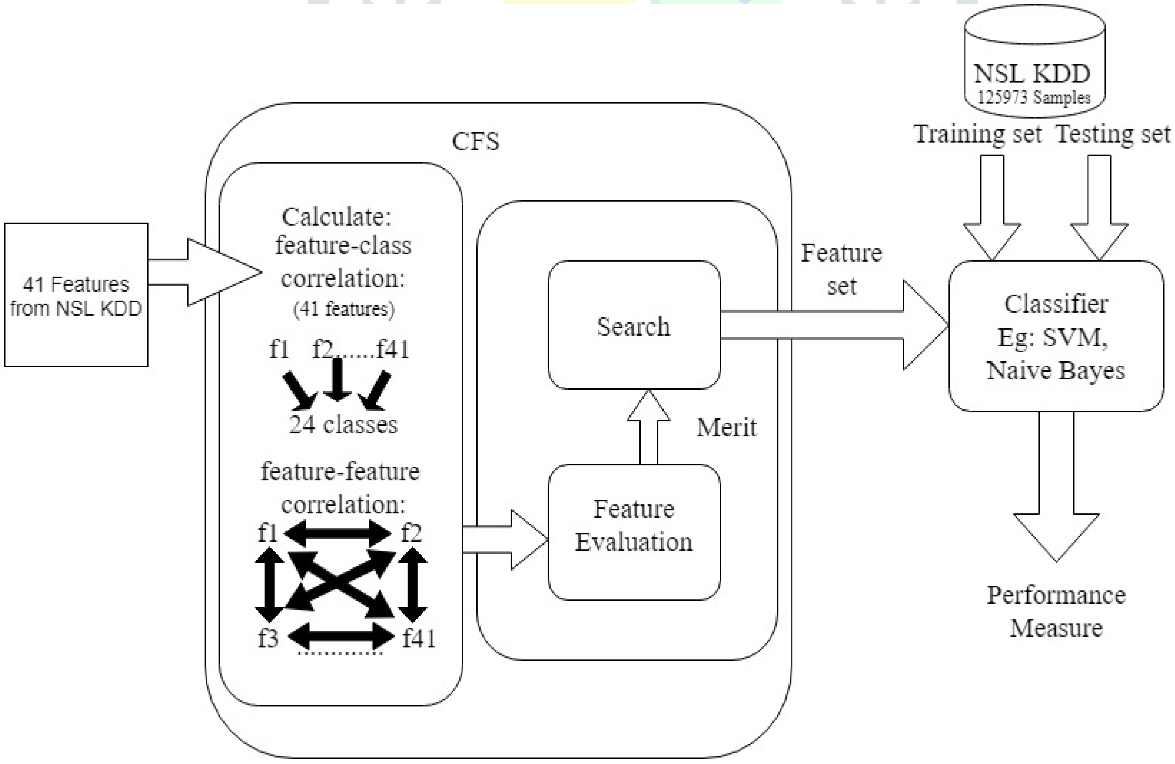

Figure 1: Mechanism of CFS

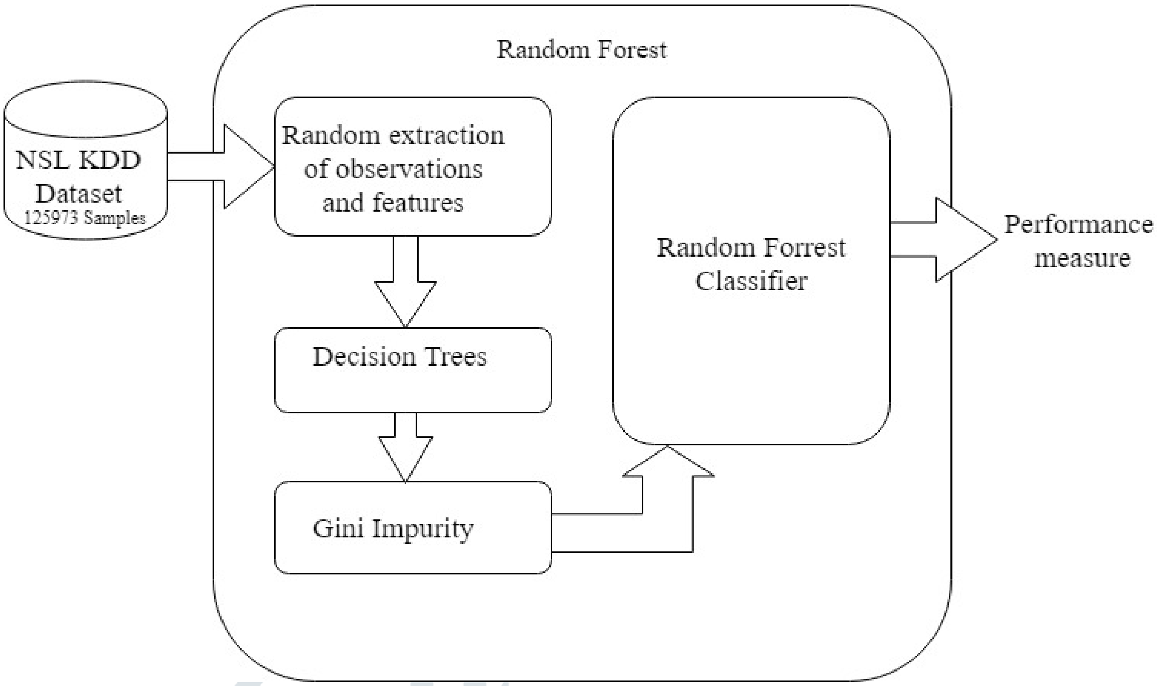

Figure 2: Mechanism of Random Forest

Random Forests consists of a number of decisio trees, which is built over a random extraction c. observations from the NSL KDD dataset and a random extraction of features. All the trees do not see all the observations or all the 41 features. Thus, trees are not correlated and thus are less prone to overfitting. Each tree is also a sequence of yes/no questions based on a single or combination of features. At each node, the tree divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket. Therefore, the relevance of each feature is derived from how "pure" each of these buckets is. As the feature decreases the impurity, the more important the feature is. In random forest, the impurity decrease from each feature can be averaged across trees to determine the final importance of the feature. Once the tree is created, the importance of each feature is checked by looking at differences in measures, such as Gini Gain. Equation (4) gives the formula for calculating Gini Gain:

$$Gini(D) = 1 - \sum_{i=1}^{m} {p_i}^2 \qquad (4)$$

$$Gini_A(D) = \frac{|D1|}{|D|} Gini(D1) + \frac{|D2|}{|D|} Gini(D2) \qquad (5)$$

$$GiniGain(A) = Gini(D) - Gini_A(D) \qquad (6)$$

where $p(i)$ is the probability of certain classification i, per the training dataset. This is calculated starting from the root node in a greedy recursive procedure until it reaches maximum depth, or each node contains samples only from one class. Figure 2 shows how the random forest works as a feature selection method. The disadvantage of this method is that once features are selected, it can be used only for the specific classifier.

IV. COMPARISON OF METHODS

CFS and Random Forest Feature Selection are evaluated on the NSL-KDD dataset. It has 41 features and 22 different attack types which fall under four main categories in the training dataset: Denial of Service (DoS), Probe attacks, U2R (User to Root) and R2L (Remote to Local). The most common attack is DoS followed by Probe. Correlation Feature Selection and Random Forests vary in the fact that the former is a filter method and the latter is an embedded method. This brings the first advantage of CFS, that is, once feature selection is done, a number of different classifiers can be used in predicting the accuracy, in contrast to Random Forests. Also, the computational cost of CFS is less when compared to Random Forests. Table 1 summarizes the findings of this research.

Table 1: Comparison between CFS and Random Forest

Criteria	**Correlation Feature Selection**	**Random Forest**
Training and Testing Time	Comparatively less	Comparatively more
Predictive Accuracy	95.43%	98.94%
Computational Cost	Lesser	Higher
Bias	Correlated features	Features with more levels
Approach	Filter based	Embedded

V. CONCLUSION

A Network Intrusion Detection System is a must-have. In order to have an efficient system, there must be a suitable classifier. The classifier must operate on a selected set of features in order to have higher accuracy. Two of the feature selection methods are discussed here: CFS and Random Forest. It is seen that both methods have their own advantages and disadvantages. CFS achieves a maximum accuracy of 95.43% whereas Random Forest achieves an accuracy of 98.84%. There are other areas of comparison among the two methods which includes the number of features selected, the time taken for training and testing, bias towards certain features and other performance measures.

REFERENCES

[1] Ahmad, M. Basheri, M.J. Iqbal, A. Raheem, “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection”, IEEE 2018

[2] Amrita & P Ahmed vol.2, issue 3, Sep 2012 1-25, “A study of feature selection methods in intrusion detection system: a survey”.

[3] Florian Gottwalt, Elizabeth Chang, and Tharam Dillon, “Analysis of feature selection Techniques for correlation-based Network Anomaly detection”, Springer 2018.

[4] H.P Vinutha and B. Poornima, “An Ensemble Classifier Approach on Different Feature Selection Methods for Intrusion Detection”, Springer 2018

[5] Hai Nguyen, Katrin Franke and Slobodan Petrovic, “Improving Effectiveness of Intrusion Detection by Correlation Feature Selection”, 2010 International Conference on Availability, Reliability ,and Security

[6] Kazi Abu Taher, Bilal Mohammed, Md. Mahbubur Rahman, “ Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection”, IEEE Feb 2019.

[7] Kritika Singh and Bharti Nagpal, “Random Forest Algorithm in Intrusion Detection System: A Survey”, IJSRCSEIT 2017

[8] M. Hall, “Correlation-Based Feature Selection for Machine Learning”, Doctoral Dissertation, University of Waikato, Department of Computer Science, 1999.

[9] Md. Zainal Abedin et. Al, “Performance Analysis of Anomaly-Based Network Intrusion Detection Systems”, IEEE 2018

[10] R. Vijayanand, D. Devaraj and B.Kannapiran, “Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with the genetic algorithm based feature selection” Elsevier – August 2018

[11] Rajesh Thomas and Deepa Pavithran, ”A Survey of Intrusion Detection Models based on NSL-KDD Data Set”, IEEE 2018

[12] Rania A. Ghazy, El-Sayed M. EL-Rabaie, Moawad I. Dessouky, Nawal A. El-Fishawy, Fathi E. Abd El-Samie, “Efficient Techniques for Attack Detection Using Different Features Selection Algorithms and Classifiers”, Springer 2018

[13] Ripon Patgiri, Udit Varshney, Tanya Akutota and Rakesh Kunde, “An Investigation on Intrusion Detection System Using Machine Learning”, IEEE 2018

[14] Sneh Lata Pundir and Amrita, “Feature Selection using random forest in Intrusion Detection System”, International Journal of Advances in Engineering & Technology, July 2013

[15] Srinivas Mukkamala and Andrew H.Sung, “A Comparative study of techniques for intrusion detection”, IEEE 2003

[16] Xin Zhang, Li Jia , Hongyan Shi, Zhongbin Tang, Xiaoling Wang, ”The application of machine learning methods to Intrusion Detection”, IEEE 2012