# Detecting Denial of Service Attack Using Principal Component Analysis with Random Forest Classifier

S. Revathi *
Ph.D. Research Scholar, PG and Research, Department of Computer Science,
Government Arts College, Coimbatore-18
revathisujendran86@gmail.com

Dr. A. Malathi**
Assistant Professor, PG and Research, Department of Computer Science
Government Arts College, Coimbatore-18
malathi.arunachalam@yahoo.com

**Abstract---Nowadays, computer network systems plays gradually an important role in our society and economy. It became a targets of a wide array of malicious attacks that invariably turn into actual intrusions. This is the reason that computer security has become an essential concern for network administrators. In this paper, an exploration of anomaly detection method has been presented. The proposed system uses a statistical method called principal component analysis to filter the attributes and random forest classifier is used to detect various attack present in Denial of Service using NSL-KDD dataset. The principal component Analysis filters attributes drastically to improve classification performance. Regarding to the task of intrusion detection a new method of random forest classifier is used to improve accuracy. Experimental result shows that the proposed method can achieve high detection rate than other existing machine learning techniques. This approach is dynamic in the sense that the model is updated based on the variations of its input. Our experiments revealed relevant results that can effectively be used to classify Denial of Service attacks.**

**Keyword--**Intrusion Detection, Principal component analysis, Random Forest, NSL-KDD dataset

## 1. INTRODUCTION

With the growing rate of interconnections among computer systems, network security is turning into a major challenge, so as to satisfy this challenge, Intrusion Detection Systems (IDS) are being designed to guard the availability, confidentiality and integrity of important networked information systems. Machine-driven detection and immediate coverage of intrusion events are required so as to produce a timely response to attacks.

Early within the analysis into IDS, two major approaches referred to as anomaly detection and signature detection were received. The former relies on flagging behaviors that are abnormal and the later flagging behaviors that are near to some antecedently outlined pattern signature of a known intrusion [1]. This paper describes a network-based anomaly detection methodology for detecting Denial of Service attacks. The detection of intrusions or system abuses assumes the existence of a model [2]. In signature detection, additionally brought up as misuse detection, the glorious attack patterns area unit sculptured through the development of a library of attack signatures. Incoming patterns that match associate degree element of the library are labeled as attacks. If only actual matching is allowed, misuse detectors operate with no false alarms. By permitting some tolerance in attack matching, there's a risk of false alarms, however the detector is predicted to be able to sight bound categories of unknown attacks that don't deviate abundant from the attacks listed in the library. Such attacks are known as immediate attacks.

In anomaly detection, the conventional behavior of the system is sculptured. Incoming patterns that deviate considerably from traditional behavior area unit labeled as attacks. The premise that malicious activity could be a set of abnormal activity implies that the abnormal patterns may be used to indicate attacks. The presence of false alarms is expected during this case in exchange for the hope of detecting unknown attacks, which can be substantially totally different from neighboring attacks. These are known as novel attacks. Detecting novel attacks whereas keeping low rates of warning, is probably the foremost challenging and vital drawback in Intrusion Detection.

IDSs may also be categorized as either network-based or host-based. The main difference between network-based and host based IDSs is that a network-based IDS, although run on a single host, is responsible for an entire network, or some network segment, while a host-based IDS is only responsible for the host on which it resides [3].

In this paper, a new method for detecting Denial of service attack is presented. The method use Principal Component Analysis to reduce the dimensionality of the feature vectors to enable better visualization and analysis of the data. The data for both normal and attack types are extracted from the NSL-KDD Intrusion Detection Evaluation data sets [4]. The feature analyzed using PCA generated various statistics to detect intrusion with relatively low false alarm rate. The Random Forest classifier is used to split data which increase accuracy of the proposed system. Rest of the paper are structured as follows. In section II Detailed study of NSL-KDD dataset with DOS attacks are explained. In section III proposed architecture as PCA and Random Forest algorithm is descripted. Section IV shows experimental result and analysis and section V draws some conclusion.

## II. DATASET DESCRIPTION

In earlier days DARPA 98 [15] and later KDDcup99 [14] dataset has been used for analysis intrusion behavior, but there are various statistical degradation in the dataset which result in poor evaluation of anomaly detection. The inherent problem of KDD dataset leads to new version of NSL KDD dataset that are mentioned in [4, 5]. It is very difficult to signify existing original networks, but still it can be applied as an effective benchmark data set for researchers to compare different intrusion detection methods [6].

The statistical examination exposed that there are essential issues in the data set which highly affects the performance of the systems, and results in a very poor estimation of anomaly detection approaches. To solve these issues, a new data set as, NSL-KDD [6] is proposed, which consists of selected records of the complete KDD data set. The advantage of NSL KDD dataset are

- No redundant records in the train set, so the classifier will not produce any biased result.
- No duplicate record in the test set which have better reduction rates.
- The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set.

The training dataset is made up of 21 different attacks out of the 37 present in the test dataset. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test dataset i.e. not available in the training datasets. The attack types are grouped into four categories: DoS, Probe, U2R and R2L. Table 1 shows the major attacks in Denial of Service in both training and testing dataset.

Table 1 Attacks in DOS

| DOS Attack Name | Training Attack | Testing Attack |
|---|---|---|
| Neptune | 41214 | 4657 |
| Smurf | 2646 | 665 |
| Pod | 201 | 41 |
| Teardrop | 892 | 12 |
| Land | 18 | 7 |
| Back | 956 | 359 |
| apache2 | 0 | 737 |
| Processtable | 0 | 685 |
| mailbomb | 0 | 293 |
| **Total** | **45927** | **7456** |

In DOS the attacker make use of computing resource or memory too busy to handle legitimate user request. The Neptune attacks can make memory resources too full for a target by sending a TCP packet requesting to initiate a TCP session. This packet is part of a three-way handshake that is needed to establish a TCP connection between two hosts. The SYN flag on this packet is set to indicate that a new connection is to be established. The Smurf attacks, also known as directed broadcast attacks, is a popular DoS packet floods. It attacks rely on directed broadcast to create a flood of traffic for a victim. The attacker sends a ping packet to the broadcast address form some network on the Internet that will accept and respond to directed broadcast messages, known as the Smurf amplifier [7]. The total number of training attack be 45927 and the testing attack be 7456, the attack such as mailbomb, apache, and processtable are present only in testing phase detection.

## III. PROPOSED ARCHITECTURE

### *A. Feature Reduction*

In the proposed method NSL-KDD dataset is used to detect intrusion. The dataset consist of 41 attribute out of which some may be unwanted and irrelevant that leads to dimensionality problem when dataset size is huge. To reduce the overwhelming size the attribute are reduced using principal component analysis [8]. PCA mainly used to reduce dimensionality and multivariate analysis technique in data compression, image processing, and pattern recognition and time series prediction. The three important feature of PCA are first, it is the optimal (in terms of mean squared error) linear scheme for compressing a set of high dimensional vectors into lower

dimensional vectors and then restructuring it. Second, the model parameters can be computed directly from the data – for example sample covariance. Third, compression and decompression are easy processes to perform the model parameters [9].

**Steps for feature reduction PCA:**

Step 1. Normalize the data with respect to their means.

Step 2. Establish the covariance matrix V.

Step 3. Find the eigen values of V. Let $\lambda_1 > \lambda_2 > \ldots > \lambda_P$ be the distinct eigen values of V.

Step 4. Let $D_1, D_2, \ldots, D_P$ be the eigen vectors corresponding to $\lambda_1, \lambda_2, \ldots, \lambda_p$ respectively, and $\|D_i\| = 1$ for i = 1, …, P.

Step 5. The set of eigen vectors $D_1, D_2, \ldots, D_{N'}$ form the N'-space onto which we can project our samples.

Step 6: Dimensionality reduction as keep only the terms corresponding to the *K* largest eigen values:

$$x - \bar{x} = \sum_{i=1}^{k} b_i u_i$$ where (x be any actually vector with linear combination of $b_1u_1, b_2u_2 \ldots\ldots b_nu_n$

The total variance of points projected onto this N'-space is equal to $\lambda_1 + \lambda_2 + \ldots + \lambda_{N'}$. $D_1$ is called the first principal component, the attribute reduce to 14 which are then used for random forest classifier to detect accuracy. Table 2 shows the reduced attribute.

Table 2: Reduced Attribute List

| | |
|---|---|
| Protocol Type | Discrete |
| Service Name | Discrete |
| Flag status | Discrete |
| Size of Source in Bytes | Continuous |
| Size of Destination in Bytes | Continuous |
| Emergency Flag | Discrete |
| Connection Status | Continuous |
| Node Logged in status | Discrete |
| Shell Access Count | Continuous |
| Number of Files Created | Continuous |
| Number of Outbound commands | Continuous |
| Accessed Files Count | Continuous |
| Host Login Status | Continuous |
| Guest Login Status | Continuous |

*B. Random Forest Classifier*

Random Forest is a moderately new algorithm for classification developed by Leo Breiman [10] that uses an ensemble of classification trees [11]. Each classification tree is built using a bootstrap sample of the data, and at each split the candidate set of variables is a random subset of the variables. Thus, random forest uses both bagging and boosting as successful approach [12], and random variable selection for tree building. Each tree is unpruned, so as to obtain low-bias trees. The algorithm yields an ensemble that can achieve both low bias and low variance.

More specifically, a forest is grown by using *mtree* bootstrapped samples each of size *m* randomly drawn from the original data of *m* points with replacement. This first type of randomization helps in building an ensemble of trees and in reducing dependence among the trees. These data are used to obtain unbiased estimates of correct classification rates and variable importance. The second type of randomness is used during building classification trees. For each node of a tree, Random Forest randomly selects *mtry* variables and uses only them to determine the best possible split using certain splitting criterion. This algorithm is fairly robust to the choice of the number *mtry*, the value of which is usually taken to be the square root of the total number of variables. Random forest trees are built without pruning. Predictions for test samples are accepted either by the majority vote of classification trees in the forest or based on a threshold value selected by the user [13]. It has excellent performance in pattern recognition tasks for detecting attacks.

## IV. EXPERIMENTAL ANALYSIS

The experimental analysis of the proposed work has been performed on Weka tool using NSL-KDD dataset. The network has to discriminate the different kinds of anomaly –based intrusions. We used 45927 training samples and 7456 testing samples with 41featuers of data set. After Feature, reduction 41 features are reduced to 14 features. First training and testing is applied on all 41 features and the results of classification are calculated. After that training and test is done with reduced features and the results of, classification is calculated.

The experimental result shows the out of bag error as 0.0013 and the mean square error be 0.0002 which show approximately 99.9% accuracy. The precision, recall and f-value of individual attack are shown in following table.

Table 3: Performance of proposed system

| Precision | Recall | F-Measure | Attack |
|---|---|---|---|
| 1.000 | 1.000 | 1.000 | Neptune |
| 1.000 | 1.000 | 1.000 | Teardrop |
| 1.000 | 1.000 | 1.000 | Smurf |
| 0.999 | 0.999 | 0.999 | Back |
| 1.000 | 0.998 | 0.998 | Pod |
| 0.944 | 0.971 | 0.972 | land |

The error visualiziation of various denial of service attacks are shown in figure1. From the above table and figure its clar that the proposed system detect attacks more accurately than other existing methods.

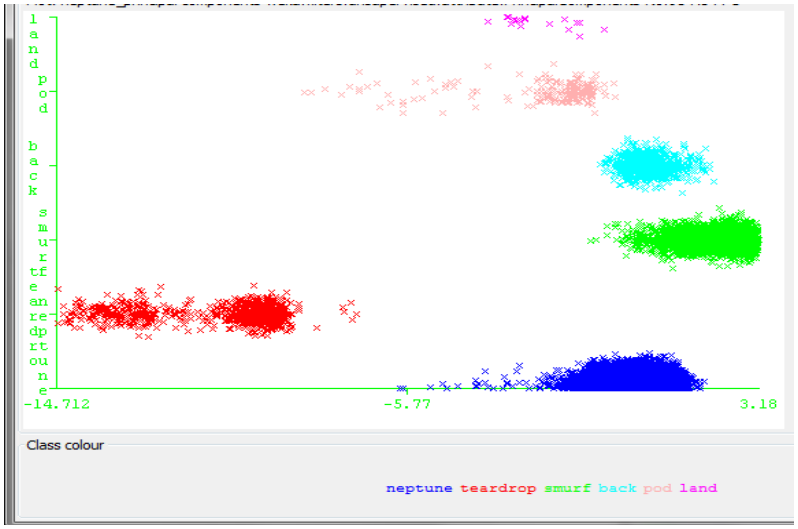


Figure1. Visualization of various DOS attack.

## V. CONCLUSION

This paper presents a method for detecting Denial-of-Service attacks using Principal Component Analysis with random forest classifier as multivariate statistical tool. The paper described the nature various attacks in DOS, using Principal Component Analysis of using it for detecting intrusions. The paper used PCA to reduce attribute to avoid dimensionality problem and random forest classifier is used to split data and to classify accuracy of the proposed system. It also discussed the results obtained using a proposed criterion for detecting intrusion and leads to approximately 99 % detection rate. The paper also shows the visualization of various attacks using graphical representation. Future work includes testing other attacks as Probe, U2R and R2L and also work on other real time environment

## REFERENCES

[1] Axelsson S., “Intrusion Detection Systems: A Survey and Taxonomy”. Technical report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, March 2000.
[2] Cabrera J., Ravichandran B., Mehra R., “Statistical Traffic Modeling for Network Intrusion Detection”
[3] Shah H., Undercoffer J., Joshi A., “Fuzzy Clustering for Intrusion Detection”
[4] “Nsl-kdd data set for network-based intrusion detection systems.” Available on: http://nsl.cs.unb.ca/KDD/NSLKDD. html, March 2009.
[5] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set”, In the Proc. Of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009), pp. 1-6, 2009.
[6] J. McHugh, “Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory,” ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.
[7] Skoudis E., “Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses”. Prentice Hall Inc., 2002

[8] Hotelling H., "Analysis of a Complex of Statistical Variables into Principal Components". Journal of Educational Psychology, 24:417–441, 1933.
[9] Lindsay I Smith A tutorial on Principal Components Analysis February 26,2002.
[10] Breiman L: Random Forests. Machine Learning, 45:5-32 (2001)
[11] Liaw A, Wiener M: Classification and Regression by Random Forest. Rnews 2:18-22 (2002)
[12] Breiman L: Bagging Predictors. Machine Learning 24:123-140 (1996)
[13] Svetnik V, Liaw A, Tong C, Wang T: Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. Multiple Classier Systems, Fifth International Workshop, MCS 2004, Proceedings, 9¨C11 June 2004, Cagliari, Italy. Lecture Notes in Computer Science, Springer, 3077:334-343 (2004).
[14] KDDCUP 99 dataset, available at: http://kdd.ics.uci.edu/dataset/kddcup99/kddcup99.html.
[15] MIT Lincoln Labs, 1998 DARPA Intrusion Detection Evaluation. Available on: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/ index.html, February 2008.