Network Anomaly Detection Using a Random Forest Classifier

Conference Paper · November 2023

DOI: 10.52305/XUNI4482

CITATION
1

READS
93

2 authors:

Naresh Kumar Voruganti
CMR Technical Campus
62 PUBLICATIONS **76** CITATIONS
SEE PROFILE

Srujan kotagiri Raju
CMR Technical Campus
214 PUBLICATIONS **883** CITATIONS
SEE PROFILE

Manaswini Pradhan
and Satchidananda Dehurl
Editors

Information and Knowledge Systems

Library of Congress Cataloging-in-Publication Data

ISBN: 979-8-89113-303-7

Published by Nova Science Publishers, Inc. † New York

Chapter 4

Network Anomaly Detection Using a Random Forest Classifier

T. Subburaj[1]
K. Srujan Raju[2]
N. M. Sinchana[3]
K. Suthendran[4]
and Voruganti Naresh Kumar[5]

[1]Department of Master of Computer Applications, RajaRajeswari College of Engineering, Bangalore, India
[2]Department of Computer Science and Engineering, CMR Technical Campus, Hydrapad, India
[3]Department of Information Science and Engineering, RajaRajeswari College of Engineering, Bangalore, India
[4]Department of Information Technology, Kalasalingam Academy of Research and Education, Tamilnadu, India
[5]Department of Computer Science and Engineering, CMR Technical Campus, Hydrapad, India

Abstract

Machine Learning (ML), which is a sub-section of Artificial Intelligence (AI) which lets in all kinds of programs to emerge as greater way at predicting consequences without being explicitly programmed to do so. ML algorithms use historic records to predict new outputs. Classical device getting to know is frequently categorized with the aid of using how a set of rules learns to grow to be extra correct in its predictions. Network-attacks are looking to be more complex, displaying more problems in accurately recognizing anomalies, and the inability to avert these anomalies may compromise security services' credibility.

In: Information and Knowledge Systems
Editors: Manaswini Pradhan and Satchidananda Dehurl
ISBN: 979-8-89113-303-7

Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS) are the two types of Intrusion Detection Systems (IDS). The framework is sorely tested with the new Test and Train data set. There are many applications for random forests. Probability estimation and prediction have been done using it. Infiltrator detection has yet to be automated with the technique, though. Detecting anomalies with Random Forests Algorithm is a key component of our proposed system.

Keywords: machine learning, decision tree, random forest, anomaly, SIDS, AIDS

Introduction

Cyber-attacks are getting more complex day by day, posing more hurdles in detecting intrusions effectively, and failure to prevent intrusions could jeopardise security services cred- ibility. An Intrusion Detection System (IDS) is a device or software program which looks onto malicious acts or policy breaking on a network system. By making use of a confidential information and event management system, any malicious acts or violation can be reported or collected on a central network (Yasir Hamid et al., 2016).

Computer safety is to be considered as a crucial issue which is a result of the continuous development of business in agile manner, and the growth of the cyber communities. Network intrusion detection techniques are critical for preventing ma- liciousbehaviour in our system and network. An Intrusion Detection System is a collection of both types of program that examines a whether there are any malicious activities present in the system. The security breaches are identified frequently by safety and event management. There are many IDS that can identify the intrusions very quickly.

Signature-based IDS looks for the things and a form of pattern in computer system, for example byte sequences, or known dangerous attacks patterns that the malware will use, to find out the possible threats. The term “signature” was actually formed by an antivirus software, which is predominantly recognized as signatures. The rather usage in signature-based IDS is that it has the ability to quickly identify and recognize pre-defined attacks, new intrusions are hard to detect because it contains new pattern that cannot be understood by the network system (Praneet Singh et al., 2021).

Many IDSs nowadays are rule-based systems, indicating that their performance is heavily reliant based on the regulations under-lined by security

experts. The process of encoding rules is expensive and lengthy due to the large amount of network traffic. Using a rule-driven language that is specified, security personnel must manually alter or deploy new rules. The identification of these intrusions in real time is a significant topic in the networking field in order to maintain user anonymity and trust.

Problem Statement

Attacks are just a few of the sorts of threats that damage ahuge number of computers on a daily basis. Few of the attacksnamely: Denial of Service, Probe, R2L, U2R are very harmful to computer.

With today's technologies, minimizing security breaches is extremely difficult. As a result, intrusion detection has become a significant issue in network security and computer forensics. The major goal is to make the information system run with minimal traffic error.

The process of encoding rules is expensive and lengthy due to the huge amount of network traffic. Using a specificrule-driven language, security personnel can manually alter ordeploynewrules.

Proposed System

Probing, Denial of Service (DoS), Remote to User (R2L), and User to Root (U2R) attacks are just a few of the threats that might harm your system. To detect network breaches, a lot of techniques are used. The Random Forest Algorithm is used to do this. Aim was to use the Train-Test data set to detect network breaches as they happened.

Train-data and Test-data are the two datasets used, each of which comprises fifteen critical attributes for regression. namely: "service," "flag," "src bytes," "dst bytes," "count," "srv count, "serror rate," "same srv rate," "diff srv rate," "dst host diff srv rate," "dst host same src port rate," "dst host srv diff host rate," "dst host serror rate," "dst host srvserror rate," "dst host rerror rate." The Random Forest Classifier (RFC) is applied to the dataset, which is split into training and testing samples.

Random Forest

Random Forests (RF) were first developed by Breiman. RFis a supervised ML algorithm that is mainly used in classification and prediction problems. It uses ensemble techniques such as bagging and boosting. RF is constructed using multiple Decision Trees(DT). Every bootstrapped sample from the initial data has a tree built for it .Each DT will provide its vote on the classification of the object. Figure 1 shows the work flow diagram of the Random Forest. The majority vote is taken as the final classification. The following is how each tree is grown:

In the beginning we consider that there are N data that exist in the original training data. Looking at the original training data, we create a bootstrap sample which takes the size N. This sample will be used to create a fresh training dataset for the tree. Out-of-bag is the data which are present in the training data that was initially built, but is not used in the bootstrap sample (Abebe Tesfahun et al., 2013).

In the original training data, M seems to be the total input features. Only m attributes will be chosen to construct each tree using this bootstrap sample dataset. For each node of the tree, the characteristics from this collection create the best possible split. During the growth of the forest, the value of m should stay unchanged (Tarun Dhar Diwan et al., 2021).

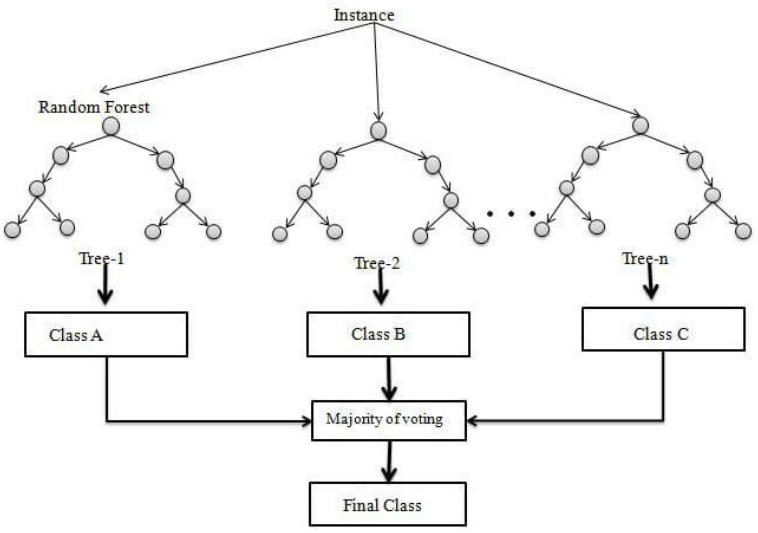

Figure 1. Work flow diagram-RandomForest.

Feature Selection

The Classifier increases the accuracy as well as reduces the data size and improve data comprehension and visualisation. Identifying effective features which have the best ability to discriminate between classes is one of the primary feature limitation concerns. There are two common techniques forminimizing features: Filter and wrapper. Function selection is utilised for Information Gain criterion(IG). Touse Information Gainasa function, each data attribute's entropy value must be calculated (Subburaj et al., 2017) (Subburaj et al., 2019).

In order to determine information gain (IG), entropy prior to and after a transformation is compared. As a result of the application of mutual information, we can calculate the statistical dependence between two variables.

Attack Classification

Experiments with the Train-Test data set are used in the detection of various attacks that can occur in a network Communication (Rahul Vigneswaran et al., 2021). There are namely four types of attacks as follows:

1. *DoS (Distributed Denial of Service):* Itis a type of attackwhere in which the user isdenied access to a perfectly safewebsite. The genuine users are consideredinvalid to enter a website safely.
2. *Remote to User (R2L):* In R2L where in an unidentified person without access to a remote machine transmits packets over a computer networks so that he can exploit that machine and its vulnerability to get access to user's personal computer.
3. *User to Root (U2R):* Here, the hacker will use a normal account of a user to get access to their personal computer andthen exploits confidentiality, trust of the user's system and acts as the administrator of that system.
4. *Probing:* A probing attack occurs when a unidentified person searching a network of computers for information's oral ready present flaws to gain the personal records of the user.

System Architecture

The raw data-set is read into the model where pre-processing and feature selection is carried out. The experimentis carried out on the dataset which are Train.csv and Test.csv. The reason for selecting the csv files uded for the experiment are that the training along with testing datasets in the data-set have a substantial number of redundant number of values in that file. The experiments were carried out on a total of 125973 rows 42 columns. The reason for choosing RandomForest is because it can build multiple decision trees to whichthe random forest and be applied later on, in practice we can conclude that more the trees in the forest the more accurate the prediction will be and hence can achieve higher accuracy. Figure 2 shows the Intrusion detection system architecture.

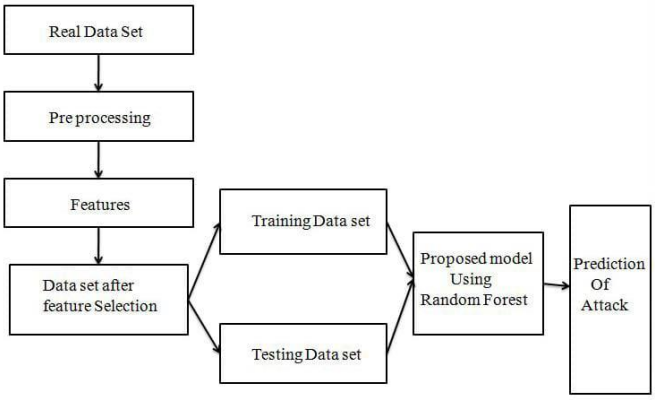

Figure 2. System Architecture for intrusion Detection System.

When the loading of dataset is complete, the first stage is data pre-processing, which involved going the model to see if there are any outliers. Outliers should be removed if they are present. Next, the proposed system is screened for the presence of categorical variables. Extract these categorical data from both test and train data. Later on these categorical data are encoded. Now the target column is separated from these categorical data. The next phase in Intrusion Detection is Feature Selection and Extraction, which is performed out using the RFE (Recursive Feature Elimination) technique. Using this method, only 15 features are determined to represent the assaults category. Following the selection of features using the RFE (Recursive Feature

Elimination) technique, the next step is to categorise the various types of assaults based on the various characteristics across the network and assess their accuracy, precision, recall, and support.

Methodology

Random Forest

Random Forest has two phases, the very first requirement isto construct the trees taking a sample from the data-set thatcontains N decision trees, and the second it is required to detect or predict the outcome from each tree and perform thevoting method (Subburaj et al., 2017) (Subburaj et al., 2021).

- Step 1:From the training set, choose a bootstrapped dataset.
- Step2:Constructingdecisiontreesforthesampleofdata.
  - Choosingmfeaturesrandomlyfrompfeatures.
  - Using the best split point and information gain among the m characteristics, determine the noded.
  - $IG(T, A) = ENTROPY(T) - \sum_{V=0}^{n} \epsilon \, \mathrm{A}\left(\frac{\mathrm{T_V}}{\mathrm{T}}\right) . \, \mathrm{ENTROPY}\,(\mathrm{T_V})$ (1)
  - Splitting nodes in to daughter nodes.
  - Repeating steps I through III until there is a complete DecisionTree.
- Step3:Repeat Steps 1 and 2 for a Nth time to produce a Nth number of trees.
- Step 4: Note down the output for new data points from each decision tree, and make sure to assign the labels that have the most votes.

Algorithm

Algorithm to predict the intrusions in a networking system

- Input: training dataset D=(x1,y1),(x2,y2)...(xm,ym);

- This contains the actual dataset with malicious target class.
- Output: H(x): The result of the vote from x;
- CRF: Random Forest Classifier where Rfii = 1,2,...,N
- Initialize the sampling size to be null as of now CRF=0
- D` is the new generated sample through feature selection.
- Build multiple decision trees
- **while** TRUE **do**
- i=0
- **for** j=1,2,...N **do**
- $D`j \leftarrow Bootstrapsample$
- $\text{Tree}_j \leftarrow Decision\ Tree$
- endfor
- endwhile
- Perform majority voting method for each random trees
- returnH(x)

Here the training data-set is given to the classifier in the beginning that contains x rows and y attributes. X is the potential anomaly data that is present in the train.csv dataset. The Random Forest Classifier will be initialized to null. For every nodes decision tree is constructed and each tree outputsthe classification of object. The majority voting is taken to be the final output (N. Venkateswaran et al., 2020)

Maximum Vote Calculation

$$\mathrm{H(X)} = MAX\{C_{J,}D'jif \sum_{i=1}^{\mathrm{D}} \mathrm{h_i^j(x)} > 0.5 \sum_{i=1}^{\mathrm{D}} \sum_{i=1}^{\mathrm{D}} \mathrm{h_i^k(x)}\} \qquad (2)$$

Here each Decision Tree will output at the classification of the object and the major classification is considered to be the final output which is done through voting mechanism.

C_j is the result of a set of decision trees and D`j contains the value of other decision trees leaving C_j .The maximum of these two is taken as the outcome of the model.

Random Forest vs Other Technologies

- Compared to Decision Trees, Random Forests can provide accurate results without overfitting.
- A logistic regression must have a lower or equal number of noise variables than explanatory variables. The Random forest performance improves with more explanatory variables
- Using ensemble learning, bagging improves Machine Learning accuracy and stability. In contrast, random forest eliminates the complexity of overfitting models and is good in unbalanced and missing data situations.

Result

Table 1. Confusion Matrix

		TrueClass			
Normal	36760	1	0	0	0
DoS	3	53883	0	0	0
Probe	0	2	9312	0	0
R2L	2	8	0	770	0
U2R	0	2	0	0	35

	dst_host_srv_rerror_rate	xAttack	prediction
0	0.00	normal	normal
1	0.00	normal	probe
2	0.00	dos	dos
3	0.01	normal	normal
4	0.00	normal	normal
5	1.00	dos	dos
6	0.00	dos	dos
7	0.00	dos	dos
8	0.00	dos	dos
9	0.00	dos	dos

Figure 3. Represents the predicted classes for the given dataset.

To predict the model's performance, the Confusion Matrix is used which is a Table 1. Each cell of the table represents the correct and incorrect predictions respectively. Here our example contains 5 class classification dataset describing each attacks: Normal, DoS, Probe, R2L, U2R as represented in table. If K = 16 then the mean accuracy of our model is found to be 0.9983627369118869. This indicates the overall accuracy of detecting the intrusions using Random Forest Model is very high. Among the models with k values from 16 to 19, the model with K value 16 holds the highest accuracy of 99%.

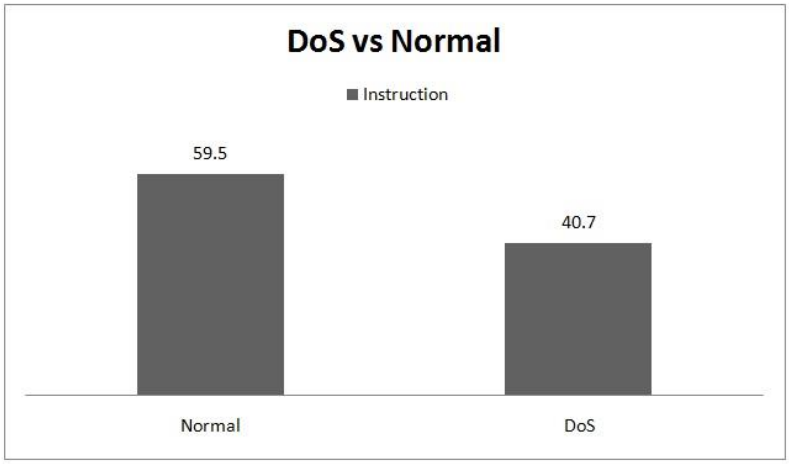

Figure 4. Barchart : Dos vs Normal.

Figure 3 shows the representation of the predicted classes for the given dataset. From the training data set the number of intrusions that occur are counted using value counts () function. The intrusions are classified under the column named "class." There are totally 59.5 percent normal intrusions while 40.7 percent are Denial to service. Figure 4 shows the Bar chart foer comparision of Normal and Denial of Servcie.

Here from the training data set the number of intrusions that occur are counted using value counts () function. The intrusions are classified under the column named "class." There are totally 59.5 percent normal intrusions while 40.7 percent are Probe. Figure 5 shows the Bar chart foer comparision of Normal and Probe.

Here from the training data set the number of intrusion s that occur are counted using value counts () function. The intrusions are classified under the column named “class.” There are totally 59.5 percent normal intrusions while 40.7 percent are R2L. Figure 6 shows the Bar chart foer comparision of Normal and R2L.

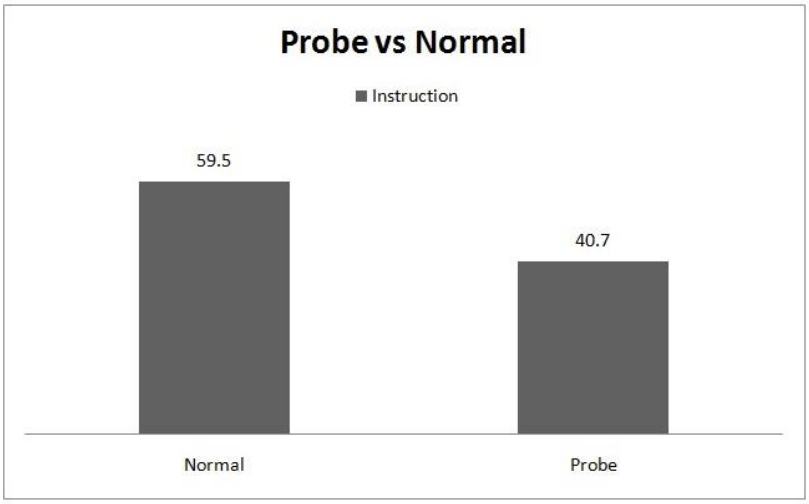

Figure 5. Barchart: Probe vs Normal.

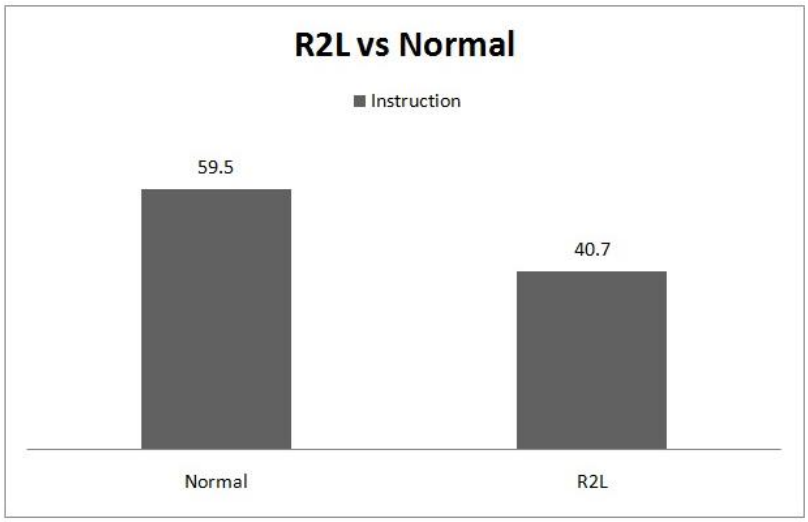

Figure 6. Barchart: R2L vs Normal.

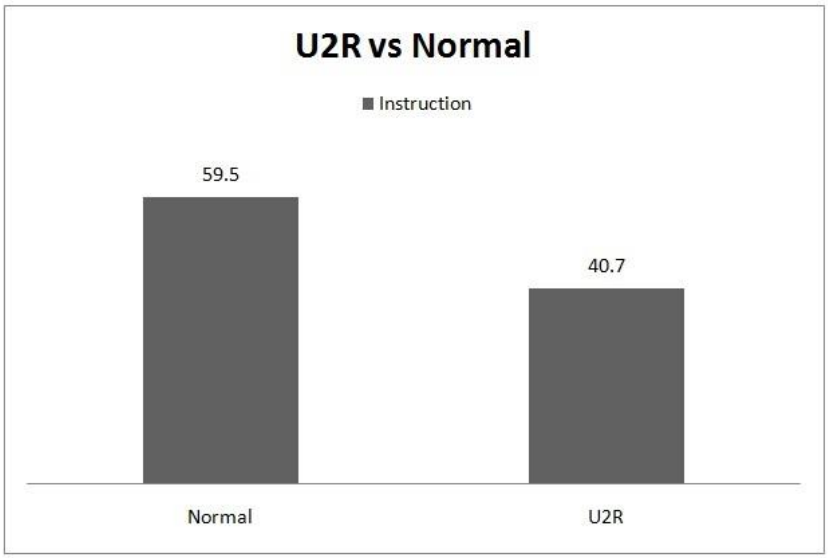

Figure7. Barchart: U2R vs Normal.

Looking at the training dataset the number of intrusions that occur are counted using value counts() function. The intrusions are classified under the column named "class." There are totally 59.5percent normal intrusions while 40.7 percent are U2R. Figure 7 shows the Bar chart foer comparision of Normal and U2R

Table 2. Performance

	Precision	*Recall*	*F1-Score*	*Support*
Normal	1	1	1	53886
DoS	1	1	1	36761
Probe	1	1	1	9314
R2L	1	1	1	780
U2R	1	0.95	0.97	37

Table 2 is a representation of how the model's performance was observed. The table consists of 4 different types that are classified for both actual and predicted cells. The performance table consists of Precision, Recall, F1-Score, Support.

Precision: It gives the percentage of correct predictions were considered to be true.

$$Precision = \frac{True\ Positive}{True\ Positive + FalsePositive} \quad (3)$$

Recall : Using this method, we can determine what percentage of positive samples were correctly predicted as positive by the classifier.

$$Recall = \frac{TruePositive}{True\ Positive + False\ Negative} \quad (4)$$

F1-Score: Recall and precision are combined in one measure. In mathematics, it represents precision and recall.

$$F1 - Score = \frac{2 * TruePositive}{(2 * TruePositive) + FalseNegative} \quad (5)$$

Support: It appears to imply that the number of instances of each class in the true responses is the support. It can be calculated by adding the rows of the contingency table together (Zeeshan Ahmad et al., 2020).

Conclusion

Modern communication networks include intrusion detection systems asstandard equipment. Toprotect their sensitive data from unauthorised persons, business environments demand a high level of security. This project's dataset was gathered from Kaggle by simulating a wide range of incursions in a military distributed system. For each connection row, 41 features were gathered from both categories. There are two types of class variables: normal and anomalous. The attempted superuser and the number of unsuccessful logins. This offers us a rough notion of how superuser access is linked to failed logins.

Future Work

In future we will propose the anomaly detction based on RF with SDN. In this method we will classify the traffic and identify the atatcks efficiently.

References

Abebe T, Lalitha D. Intrusion Detection using Random Forests Classifier with SMOTE and Feature Reduction. *IEEE International Conference on Cloud Ubiquitous Computing Emerging Technologies*: (2013):128- 132.

Praneet S, Jishnu J, Akhil P and Reshmi M. Edge-Detect: Edge-centric Network Intrusion Detection using Deep Neural Network. *IEEE 18th Annual Consumer Communications & Networking Conference:* (2021): 1-6.

Rahul V, Vinayakumar, Soman K P and Prabaharan P. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. *IEEE - 43488* (2018): 1-6.

Subburaj T and Suthendran K. Threat detection on UDP Protocols using Packet rates in IoT. *Machine Intelligence and smart systems, Springer* (2021): 675 – 682.

Subburaj T and Suthendran K. Detection and Trace back of Low and High Volume of DDoS attack based on Statistical Measures. *Concurrency and Computation: Practice and Experience, Wiley Publications* (2019) : 1-22.

Subburaj T, Suthendran K and Arumugam S. Statistical Approach to Trace the Source of Attack Based on the Variability in Data Flows. *Lecture Notes in Computer Science, LNCS 10398, Springer* (2017): 392-400.

Subburaj T and Suthendran K. Detection and Trace Back of DDoS Attack Based on Statistical Approach. *Journal of Advanced Research in Dynamical and Control Systems 13-Special Issue* (2017): 66-74.

Tarun D D, Siddartha C and Hota H.S. A Detailed Analysis on NSL-KDD Dataset using various Machine Learning Techniques for Intrusion Detection. *Turkish Journal of Computer and Mathematics Education* (2021) 12(11) : 2954-2968.

Venkateswaran N and Umadevi K. Hybridized Wrapper Filter Using Deep Neural Network for Intrusion Detection. *Computer Systems Science & Engineering* (2020) 42(1): 1-14.

Yasir H, Sugumaran M and Journaux L. Machine Learning Techniques for Intrusion Detection: A Comparative Analysis. *Proceedings of the International Conference on Informatics and Analytics Article 53*: (2016): 1-6.

Zeeshan A, Adnan S K, Cheah W S, Johari A and Farhan A. Network intrusion detection system: A systematic study of machine learning and deep learning approaches, *Transactions on Emerging Telecommunications Technologies*, *Wiley* (2020) 32(1): 1-29.