iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 9 **Issue:** VI **Month of publication:** June 2021

**DOI:** https://doi.org/10.22214/ijraset.2021.35406

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*
*Volume 9 Issue VI June 2021- Available at www.ijraset.com*

# Random Forest Classifier Based Network Intrusion Detection System

Aadhar Dutta
*Department Information Technology, Delhi Technological University, Delhi, India-110042*

***Abstract:*** ***In today's digital world, we all use the Internet and connect to a network, but all the data we send or receive, is safe? Some kind of attack is present in network packets that might access the computer's private information to the hacker. We cannot see and tell whether a network is safe to connect with or not, so we made a Network Intrusion Detection Model predict whether these network packets are secure or some attack is there on the package. We use Random Forest Classifier to obtain the maximum accuracy. To test our model in real-time, we have created a packet sniffer that would sniff out network packets, convert them into required features, and then try it in our model to predict the legitimacy of the network packet.***
***Keywords: Intrusion Detection System, Network packets, packet sniffer, Random Forest Classifier, Anomaly Detection.***

## I. INTRODUCTION

Due to the web's general uses, electronic attacks on organizations and information systems of the monetary associations, military, and energy areas are expanding. Different intruders and hackers assault enormous sites of any association. The data of government and private associations might be spilled or harmed by unapproved clients. Intruders can have numerous structures, for example, viruses, spyware, worms, malicious logins, spam ware.

Data security is a significant aspect of securing critical information about any association. The associations need security applications that adequately shield their organizations from malicious assaults and abuse. The intrusion detection system can identify the intrusions and protect the data framework from security infringement. The intrusion detection system distinguishes intruders and makes a move against the intruder. It is utilized to recover the data by fixing the association's harm brought about by an unapproved client and recognizing the malicious utilization of P.C. and P.C. organization. It identifies admittance to an un-approved client, the infringement of security, and finds illegal clients.

There are two sorts of intrusion detection strategies, viz., misuse identification and anomaly. Misuse recognition is information or example-based, though anomaly identification is behaviour-based. Misuse location is dependable for identifying known assaults with low false positives.

The misuse recognition method cannot distinguish the new assault. An anomaly identification procedure can recognize new assault with a high positive error. Existing intrusion detection systems have a high detection rate, though they experience high false alerts ill effects. The undertaking of decreasing false positives is very vital for the intrusion detection system.

Since the system has the upside of finding helpful information from datasets, different Machine learning approaches are implemented. These methodologies can diminish false positives. Bayes principle, Artificial Neural Network, Hidden Markov Model, Bayesian Belief Network, Genetic Algorithm, and Association of rules and Bunching strategies for machine learning are generally used to execute an intrusion detection system. The mix of various base Machine learning calculations is called an ensemble technique.

In writing a study, it is discovered that an Ensemble technique for Machine learning assists with diminishing false-positive rates. There are three principal strategies to join fundamental Machine learning classifiers viz., Bagging, Boosting, and Stacking. In this paper, the Bagging group technique for machine learning is proposed to execute an intrusion detection system. NSL_KDD dataset and Defense Advanced Research Projects Agency (DARPA) datasets are broadly utilized as preparing and testing datasets for the intrusion detection system.

The NSL_KDD dataset gives 42 features dataset to train and test the intrusion detection system. However, every 42 elements in the dataset are not essential and needed for training and testing reasons. If all components are utilized to prepare intrusion detection systems, model structure time is expanded.

The critical feature selection is a whole cycle in intrusion detection systems. In light of this examination, significant features of the NSL_KDD dataset are physically chosen, which improves the order exactness.

## II. RELATED WORK

Lee et al.[1] in his paper has explained mature Data mining techniques, which could help significantly finding out the best way to detect Intrusion systematically; thus, we understand his report and further use his research to find much better results in our Random Forest Classifier technique.

Idan et al.[2] helped us interpret the proper understanding of Machine learning in the field of Cyber scrutiny and how we should be processing our datasets in a much better and informative way.

Aggarwal et al.[3] This paper is an essential paper to figure out how to change over TCP dump data to the 42 features in the KDD data set; thus, this paper explains the variable disparities in a very defined manner.

Wu et al.[4] In this research paper, we were able to comprehend how we can work towards building an maximum attack comprehension and detection while maintaining a minute low false alarming rate when worked out against ML concepts.

## III.DATASET

The experimental arrangement is isolated into two stages. In the principal stage, the NSL_KDD dataset is pre-processed. In the pre-processing step, valuable features are chosen.

Feature selection is an essential information pre-preparing step to reduce the component of the dataset. A decrease in the measurement of the dataset prompts a better reasonable model. The NSL-KDD dataset has proposed 41 highlights to implement an intrusion detection system. Each of the 41 highlights is not needed to implement an intrusion detection system.

This paper has included 16 features for experimental work that would give high accuracy with low false-positive rates. These features are: -

Table I. Different features that give high accuracy with low false-positives.

| | |
|---|---|
| Src-bytes | Number of data bytes transferred from source to destination in single connection. |
| count | Number of connections to the same destination host as the current connection in the past two seconds. |
| service | Destination network service used. |
| srv_count | Number of connections to the same service (port number) as the current connection in the past two seconds. |
| protocol_type | Protocol used in the connection. |
| diff_srv_rate | The percentage of connections that were to different services, among the connections aggregated in count. |
| same_srv_rate | The percentage of connections that were to the same service, among the links aggregated in count. |
| flag | Status of the connection – Normal or Error. |

| | |
|---|---|
| dst_bytes | Number of data bytes transferred from destination to source in single connection. |
| srv_serror_rate | The percentage of connections that have activated the flag s0, s1, s2 or s3, among the connections aggregated in srv_count. |
| logged_in | Login Status: 1 if successfully logged in; 0 otherwise. |
| duration | Length of time duration of the connection. |
| lnum_compromised | The number of "compromised" conditions. |
| wrong_fragment | The total number of wrong fragments in this connection. |
| is_guest_login | 1 if the login is a "guest" login; 0 otherwise. |

## IV. METHODOLOGY & MODEL

The Basic Pipeline of our model is as follows: -

1) *Feature Selection:* We used WEKA to obtain the most correlated features. We selected a total of 16 most correlated features.
2) *Tcpdump:* Tcpdump application of Linux is used to dump all the network data transmission on all ports. Wifi is set as the default interface for the data dump, and it output the data in a cap file.
3) *Feature Extractor:* In the pcap file, Pyshark is used to analyze the data. This script basically creates all connection records in the pcap file, and then using pyshark, it scrapes all the connection and creates records according to our intrusion detection model.
4) *Intrusion Detection:* The finalized ML model is then used to predict whether the packet is regular or attack type.
5) *Model:* We divided the KDD dataset into a training and testing set of 65:35, and then we used the Random Forest Classifier of ensemble learning to train the model. We saved the model using a pickle, which we used to predict the packets in testing mode directly.

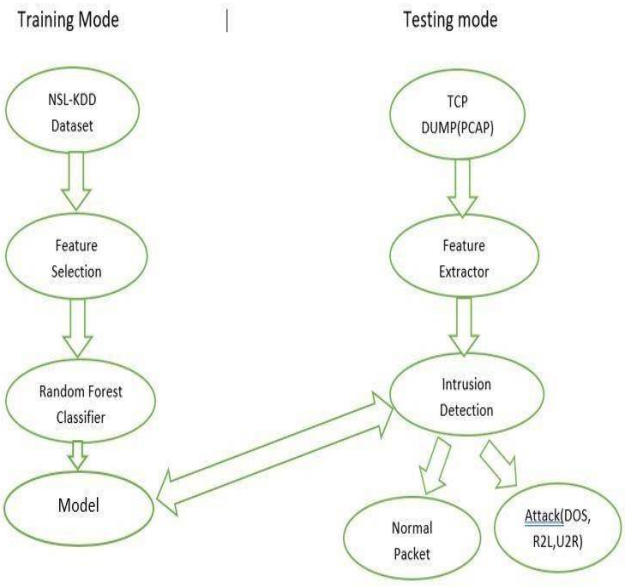



Fig. 1: Flowchart depicting the relationship between both Training Mode & Testing Mode.

The attacks are broadly being classified into four types: -

a) *DOS* – In this attack, the intruder seeks to make a machine or network resource unavailable to its intended users by Temporarily or indefinitely disrupting services of a host connected to the Internet.
b) *R2L* – This type of attack is launched by an attacker to gain unauthorized access to a victim machine in the entire network.
c) *U2R* – This type of attack is launched illegally to obtain the root's privileges when legally accessing a local machine.
d) *Probe* – A probe is an attack that is deliberately crafted so that its target detects and reports it with a recognizable "fingerprint" in the report. The attacker then uses the collaborative infrastructure to learn the detector's location and defensive capabilities from this report.

If the packet does not belong to these four types of attack, then it is classified as a regular or safe packet.

## V. RESULTS

The features were extracted from WEKA, so these features were given to our model to train with the 65% of the KDD dataset; the Random Forest Classifier was used to train our dataset, we used other methods like K-NN and Decision Trees also to train our model, but thRandom Forest Classifier gave the best results Rest 35% of the dataset was tested on the trained model. It gave the best accuracy of 99% and had a shallow rate of false positives.

For real-time analysis of the model, we created a packet sniffer which would sniff real-time packets from the network, and then our feature extractor script decodes it to obtain the required features and then stores the data obtained in the CSV, and then our trained model tests this real-time data and classifies the packets into the five categories which are [Normal, DOS, U2R, R2L, Probe].

The best method to measure the accuracy is to calculate precision, Recall, and f1 score, which are given below: -

TABLE II. Classes to measure the Weighted Accuracy

| Class | value |
|---|---|
| NORMAL | |
| True Positive | 25479 |
| True Negative | 9291 |
| False Positive | 29 |
| False Negative | 17 |
| DOS | |
| True Positive | 8927 |
| True Negative | 26503 |
| False Positive | 11 |
| False Negative | 5 |
| PROBE | |
| True Positive | 229 |
| True Negative | 34566 |
| False Positive | 9 |
| False Negative | 11 |
| R2L | |
| True Positive | 119 |
| True Negative | 34674 |
| False Positive | 5 |
| False Negative | 18 |
| U2R | |
| True Positive | 6 |
| True Negative | 34804 |
| False Positive | 1 |
| False Negative | 5 |

The formula to find the value of Precision we can use the formula given below,

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

In the above formula Precision, the True Positive & False Positive are achieved using Table II.
The weighted accuracy acquired from training the above Precision Model, we get,

TABLE III. Weighted accuracy achieved from the Precision Model

| ATTACK MODEL | WEIGHTED ACCURACY |
|---|---|
| NORMAL | ~0.99 |
| DOS | ~0.99 |
| PROBE | 0.95 |
| R2L | 0.96 |
| U2r | 0.86 |

The formula to find the value of Recall we can use the procedure given below,

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

In the above formula for Recall, the True Positive & False Positive are achieved using Table II.
The weighted accuracy acquired from training the above Recall Model, we get,

TABLE IV. Weighted accuracy achieved from the Precision Model

| ATTACK MODEL | WEIGHTED ACCURACY |
|---|---|
| NORMAL | ~0.99 |
| DOS | ~0.99 |
| PROBE | 0.96 |
| R2L | 0.88 |
| U2r | 0.55 |

After achieving the above formulas for Precision and Recall, we can find out the F1 score by using the below procedure,

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In the above formula for the F1 Score, the True Positive & False Positive are achieved using TABLE III & IV.
The weighted accuracy acquired from training the above F1 Score, we get,

TABLE V. Weighted accuracy achieved from the Precision Model

| ATTACK MODEL | WEIGHTED ACCURACY |
|---|---|
| NORMAL | ~0.99 |
| DOS | ~0.99 |
| PROBE | 0.96 |
| R2L | 0.92 |
| U2r | 0.67 |

## VI. CONCLUSION

Today we live in a digital world, and every day, we tend to use the Internet and establish the connection on the network with the ports and services inside the network, so there are high risks of being attacked by a hacker who uses these kinds of attack to obtain access to monitor, so our network intrusion detector model can be used here to predict the legitimacy of the packets as it has an accuracy of 99% and could be used in an organization also for security purposes.

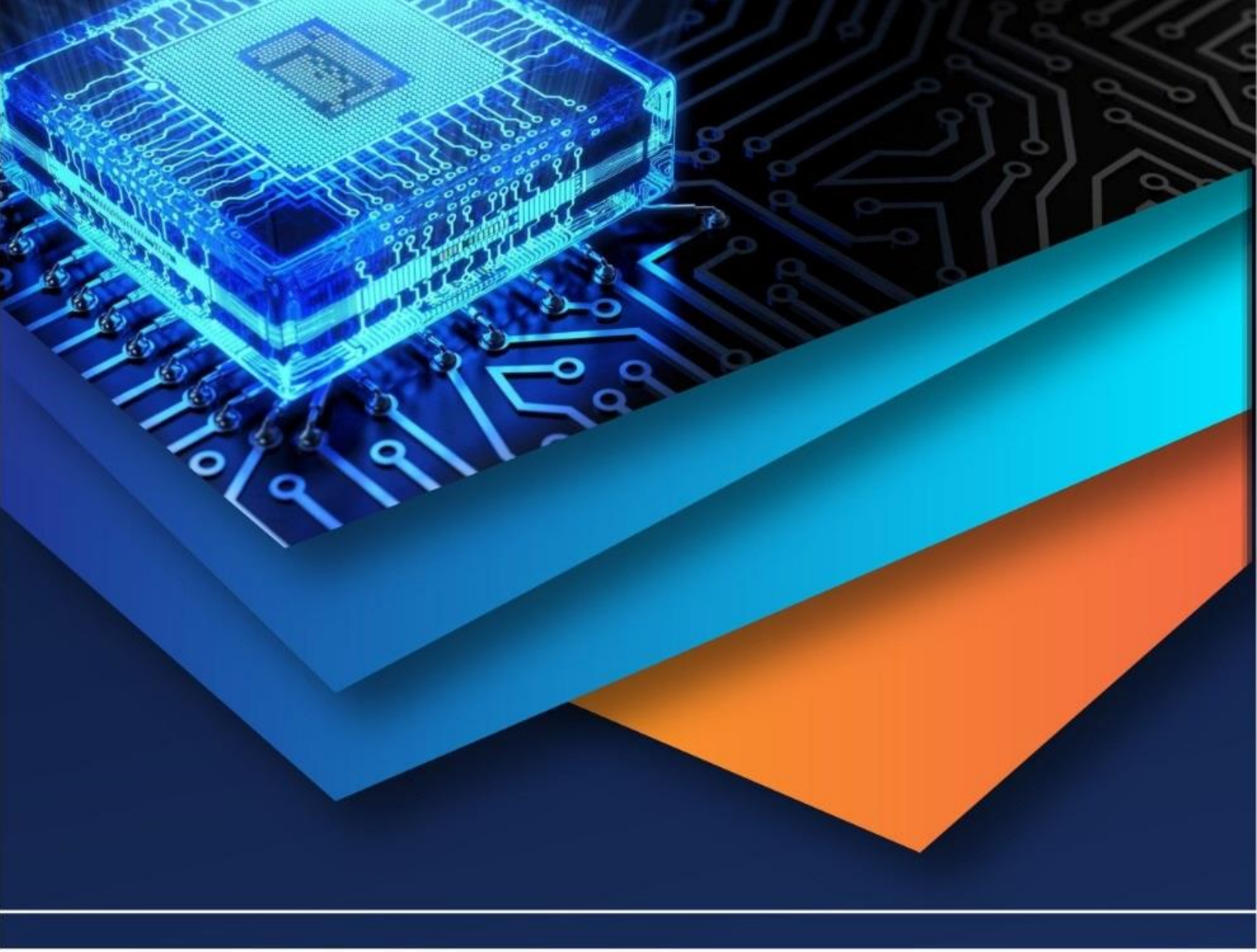
## REFERENCES

[1] L. W and S. S.j, "Data mining approaches for intrusion detection. In: Proceedings of the 7th," *USENIX Secur. Symp. (SECURITY-98*, pp. p79--94, 1998, [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjBho6oo9vdAhVExoUKHdnYDFoQFjAAegQICRAC&url=https%3A%2F%2Fpdfs.semanticscholar.org%2F1a76%2Ff0539a9badf317b0b35ea92f734c62467138.pdf&usg=AOvVaw2lfvokfrG-_FA7LzIoDPHj.

[2] I. Amit, J. Matherly, W. Hewlett, Z. Xu, Y. Meshi, and Y. Weinberger, "Machine Learning in Cyber-Security - Problems, Challenges and Data Sets," 2018, [Online]. Available: http://arxiv.org/abs/1812.07858.

[3] P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," *Procedia Comput. Sci.*, vol. 57, pp. 842–851, 2015, doi: 10.1016/j.procs.2015.07.490.

[4] P. Wu, H. Guo, and N. Moustafa, "Pelican : A Deep Residual Network for Network Intrusion Detection," no. Ml.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY