RESEARCH ARTICLE OPEN ACCESS

Ensemble Learning Vs Ann Approaches to Network Based Intrusion Detection System

Manish Chava [1], Aman Agarwal [2], Dr Radha K [3]
[1] CSE, GITAM University, Hyderabad, Rudraram
[2] CSE, GITAM University, Hyderabad, Rudraram
[3] CSE, GITAM University, Hyderabad, Rudraram

ABSTRACT

Network attacks and breaches have become very common among global companies leading them to substantial revenue losses. It is critical to protect data and networks against malicious assaults. Therefore, it is high time for companies to use a highly dependable Intrusion Detection System (IDS). An Intrusion Detection System (IDS) is a system that monitors network traffic for suspicious activity and sends an alert when such activity is discovered. This project demonstrates the working of an intrusion detection system that scans the network for any malicious activity. An intrusion detection system uses the data to identify highly advanced threats before they wreak the system. Machine learning must be applied to make the IDS more generic and capable of discovering new attack techniques and avoiding such attacks.

Because of the rapid growth in computer network usage and the massive rise in applications that use these computer networks, providing security is becoming significantly vital. The existing security systems include technical and commercial flaws that are difficult for manufacturers to address. As a result, the role of IDS in detecting network threats is becoming increasingly important.

Keywords *:-* Machine learning, deep learning, Artificial Neural Networks, XGBoost, Random Forest, Intrusion Detection System, computer Networks.

I. INTRODUCTION

Although many research papers discuss NIDS implementation, the companies do not use them because of their high fault ratio. The existing security systems include technical and commercial flaws that are difficult for manufacturers to address. It, therefore, resulted in the rise in the need for genuine and highly acceptable NIDS that are reliable. The goal is to develop an algorithm capable of detecting attacks in network traffic by adopting a machine learning-based approach.

First, a highly reliable NIDS is developed using supervised machine learning algorithms. The dataset considered for this project is the NSL KDD dataset.

After proper analysis of the dataset, the appropriate algorithms are chosen. This project adopts classification algorithms to predict the types of network attacks that are possible. The attacks are categorized into five types: Normal, PROBE, R2L, U2R, and DOS. After the NIDS model is developed, it is trained and tested with the data to see its performance. The performance considers how well the model predicts for each attack category. Different classification metrics like train and test data accuracy, precision, recall, and f1 score, ROC AUC score are used.

This project comprises several algorithms designed and developed for potential network attack prediction. The supervised machine learning approach generally consists of well-structured data to feed the machine learning algorithm. The algorithm usually tries to learn from the training data, hidden patterns, and features and then uses the knowledge it gained to predict future outcomes. In the project, the dataset is well structured and labelled with training and testing data separately.

II. BACKGROUND

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. Ensemble Techniques

Ensemble learning is a unique machine learning technique in which multiple weak learners, called base models, are developed. Their results are combined to get one bigger and higher accuracy model rather than a single model with lower accuracy. Ensemble learning is of two types Bagging and Boosting. In this project, one technique from each sub-category is explored. They are Random Forest from Bagging and Extreme Gradient Boosting (Xgboost) [4].

1) Random Forest: Random Forest classifier is also called as Bootstrap Aggregation.

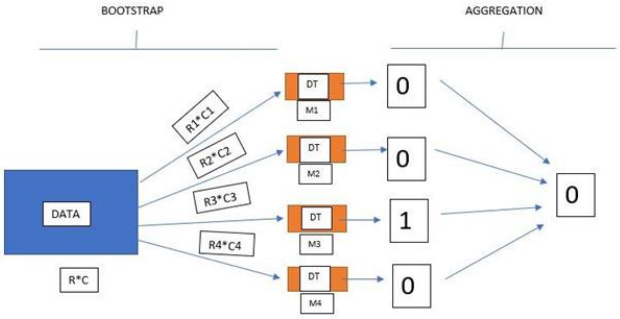

Fig. 1 Random Forest Process

Let us consider a dataset D with dimensionality rxc, where r = no. of rows,

c = No.of columns. In Random Forest, multiple models will be crated based on decision trees(D.T). A subset of the data D is provided individually for each of these models. The technique of dividing the data D into subsets is called row-wise sampling with replacement. Each model considers only a subset of features. After each model makes its prediction, it combines all the results by choosing the majority class as the final prediction class.

2) EXTREME GRADIENT BOOSTING(XGBOOST):

Boosting algorithm is a serial ensemble technique.

Initially, a base model will be constructed. This base model output will be the dependent or target column's average. Later the errors are calculated for each instance called residuals, and the following model (say model 1) will be fitted on these residuals.

A similarity score on the residuals node will be calculated. Then splitting criteria will be selected. After the splitting is done, the similarity score for each sub-branch will be calculated.

The Gain value is generated. Later, using the gamma value provided to the XGBoost algorithm, If the gain value is greater than the gamma value, the split will be considered; otherwise not. This is how the auto pruning for the decision tree will happen [2].

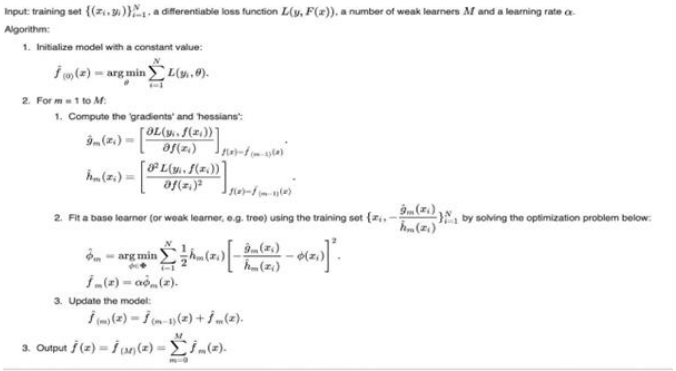

Fig. 2 XGBoost Process [9]

$$Similarity\ Score = \frac{(Sum\ of\ Residuals)^2}{Total\ No.of\ Residuals + Lambda}$$

$$Gain = Similarity\ Score\ after\ Splitting - Similarity\ Score\ Befor$$

Lambda: Regularization parameter

If lambda value is high, it leads to higher pruning of the tree thus avoid overfitting of the tree.

Also, higher lambda value takes care of the effect outliers on the model to some extent.

$New\ prediction = Previous\ prediction + Learning\ rate\ x\ output$New prediction = Previous prediction + Learning rate x output.

B. DEEP LEARNING

Artificial neural networks are used. The input values must be Standardized or Normalized. These methods keep your parameters in a suitable level, allowing your Neural Network to function them more easily. This is essential for the operational capacity of your Neural Network. In a neural network, in order to find the optimal weights, we need to reduce the cost function. One approach for this is the brute force approach. However, this is not feasible when the neural networks are complex. So, the Gradient descent approach was adopted [4].

The GDF(GRADIENT DESCENT FUNCTION) is an iterative algorithm to find the minimum of a function. Here that function is the cost function. Usually, the GDF is applied to smooth and convex cost functions. If the function is not convex, then the algorithm might not find the global minimum but the local minimum. Moreover, if the cost function curve is not smooth but sharp, then it is not differentiable [4].

Fig. 3 Types of Cost Function Curves [10]

The ANN is a sequential neural network. The sequential order of the layers are as follows.
Input layer -> Hidden layers -> Output layer

The No.of neurons in the input layer will equal the no of independent features. For a classification task, the No.of neurons in the output layer will equal the total No.of class labels. For hidden layers, the value is dynamic, meaning the neurons' count depends on how well it performs on the data. Different activation functions are used for each layer. Generally, for ANN, the Rectified/Relu activation function is used [2].

3) Intrusion Detection System: The intrusion detection systems are of four types. They are explained below.

a) Network Intrusion Detection System (NIDS): These IDS are deployed across multiple locations within a network that is unprotected and has a high risk of getting exposed to potential attacks [8].

b) Host Intrusion Detection System (HIDS): They are implemented at multiple host/client machines within a network. Contrary to the NIDS, These HIDS monitors the network packets within the host machine (Internally) for any suspicious activity [8].

c) Anomaly-Based Intrusion Detection System (AIDS): This form of IDS uses a method or strategy to monitor the network traffic and compare it to predefined standards. It then detects and notifies administrators of abnormal activity in the network [8].

d) Signature-based Intrusion Detection System (SIDS): These systems have a database or library of signatures or properties that are present in known intrusion attacks or malicious threats. Signature-based IDS monitors all network packets and detects potential malware by comparing signatures to suspicious activity [8].

III. METHODOLOGY

A. DATA PREPROCESSING

Data pre-processing involves checking the dataset for Null values, data variety, Label encoding, data standardization, and Feature selection. The NSL KDD dataset involves no null values. For Label encoding, the first attack feature is categorized into five attack types, namely 'DOS,' 'PROBE,' 'NORMAL,' 'R2L', and 'U2R'. They are categorized as follows [6] [5].

```
attack_types =
{
    "back":"DOS", "land":"DOS", "neptune":"DOS", "pod":"DOS", "smurf":"DOS", "teardrop":"DOS", "apache2":"DOS",
    "udpstorm":"DOS", "processtable":"DOS", "worm":"DOS", "satan":"PROBE", "ipsweep":"PROBE", "nmap":"PROBE",
    "portsweep":"PROBE", "mscan":"PROBE", "saint":"PROBE", "guess_passwd":"R2L", "ftp_write":"R2L", "imap":"R2L",
    "phf":"R2L", "multihop":"R2L", "warezmaster":"R2L", "warezclient":"R2L", "spy":"R2L", "xlock":"R2L",
    "xsnoop":"R2L", "snmpguess":"R2L", "snmpgetattack":"R2L", "httptunnel":"R2L", "sendmail":"R2L", "named":"R2L",
    "buffer_overflow":"U2R", "loadmodule":"U2R", "rootkit":"U2R", "perl":"U2R", "sqlattack":"U2R", "xterm":"U2R",
    "ps":"U2R","mailbomb":"DOS", "normal":"NORMAL"

}
```

Fig. 4 Categorizing attacks into five categories

The machine understands the data only in numbers, so the label encoder is used to convert the categorical data into numerical data. The categorical features that are present in the data are 'protocal_type', 'flag,' and 'attack.'
The label encoder encodes each category with a value between 0 and n_classes-1, where n is the number of distinct categories.
The Label encoding is done in the following manner.

FEATURE	CATEGORIES	LABEL ENCODED VALUES
flag	OTH REJ RDTO RSTOS0 RSTR S0 S1 S2 S3 SF SH	0 1 2 3 4 5 6 7 8 9 10
attack	DOS NORMAL PROBE R2L U2R	0 1 2 3 4
Protocol type	icmp tcp	0 1

	udp	2

Table. 1 Label Encoded Values Distribution

The data in the 'attack' column is distributed as follows.

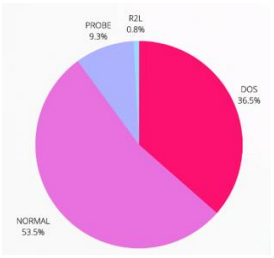

Fig. 5 Training data

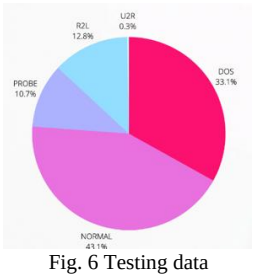

Fig. 6 Testing data

The dataset consists of 43 features. Feeding all this data into the machine learning model can invite much noise to the model and thus make the model inefficient and inaccurate. So we used features selection techniques to extract the best features from the data set that would help predict the attacks accurately and efficiently.

The features selection technique adopted in this project is 'SelectKBest'. This technique generates a score for each feature called 'Feature_score.' These feature scores select the best K features where K is an integer.

In this project, we considered K as 6,10,15. So we generated three separate data frames for top_10, top_15, and top_6 features. Each algorithm is trained on the above-mentioned top_n features. Where n = 10,15,6.

After feature selection, the data is standardized using the standard scalar technique. It converts the data to unit variance and means zero.

The process of data pre-processing and feature selection is made to both training and testing data.

B. Model Building and Training

A model is built using the 'Random Forest,' 'Extreme Gradient Boosting,' and 'ANN' algorithms.

Dealing with such massive data involves trying to obtain the optimal values for all the algorithm's hyperparameters. For Random Forest and XGBOOST, the RandomSearchCV technique is used for hyperparameter tuning. It takes a set of values for each parameter. Usually, the ones we are interested in as input form random combinations of values from those parameter values and find the best combination of hyperparameter values that gives optimal output.

This is performed on the top 10,15 and 6 feature datasets.

For ANN, the Keras Tuner is used for hyper Parameter tuning. This includes finding the optimal No.of hidden layers required to understand the data during training and their activation functions efficiently. It uses RandomSearch for this. The input layer contains neurons equal to the count of features fed into the network. The output layer contains five neurons with the SoftMax activation function, which signifies the five attack classes. The hidden neurons contain neurons in the range of 5>=x<=15, which Keras Tuner determines during the model-building process.

IV. RESULTS AND DISCUSSION

A. Random Forest

For the Random Forest classifier, the highest accuracy of 76% was achieved when the Top 15 features were considered. The ROC AUC score is 94%. A review of the classification report is as below.

CLASS LABEL	PRECISION	RECALL
DOS	0.96	0.79
NORMAL	0.68	0.97
PROBE	0.71	0.61
R2L	0.92	0.15
U2R	0.19	0.10

The reason for the low scores of the fourth category, namely 'U2R', is the fewer data available for this attack class. Less than 0.8% of the data training data is covered for the 'U2R' category. So, the model was not provided with a sufficient amount of 'U2R' attack category data to learn appropriately. Hence the low scores.

B. Extreme Gradient Boosting (XGBOOST)

For the Random Forest classifier, the highest accuracy of 78% was achieved when the Top 15 features were considered. The ROC AUC score is 94.8%. A review of the classification report is as below.

CLASS LABEL	PRECISION	RECALL
DOS	0.88	0.83
NORMAL	0.75	0.95
PROBE	0.59	0.62
R2L	0.88	0.22
U2R	0.21	0.13

CLASSIFIER	HIGHEST ACCURACY	ROC AUC SCORE
Random Forest	0.76	0.94
XGBOOST	0.78	0.948

C. Artificial Neural Networks

Accuracy: 99%
Loss: 0.034
Validation Accuracy: 80%

REFERENCES

[1] G. De Carvalho Bertoli et al., "An End-to-End Framework for Machine Learning-Based Network Intrusion Detection System," in IEEE Access, vol. 9, pp. 106790-106805, 2021, doi: 10.1109/ACCESS.2021.3101188.

[2] Devan, P., Khare, N. An efficient XGBoost–DNN-based classification model for network intrusion detection system. Neural Comput & Applic 32, 12499–12514 (2020). https://doi.org/10.1007/s00521-020-04708-x

[3] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez,
Anomaly-based network intrusion detection: Techniques, systems and challenges,
Computers & Security. Volume 28, Issues 1–2,2009,Pages 18-28,ISSN 0167-4048,
https://doi.org/10.1016/j.cose.2008.08.003,(https://www.sciencedirect.com/science/article/pii/S0167404808000692)

[4] Ahmad, Zeeshan, et al. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches." Transactions on Emerging Telecommunications Technologies 32.1 (2021): e4150.

[5] Lasitha Hiranya (2020, March 9). Most Important things of "NSL-KDD" data set
https://medium.datadriveninvestor.com/did-you-know-the-famous-data-set-called-nsl-kdd-293b39420c74

[6] Gerry Saporito (2019, Sep 17). A Deeper Dive into the NSL-KDD Data Set.
https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657

[7] DNSSTUFF (2020, Feb 18). 7 Best Intrusion Detection Software and Latest IDS System.

https://www.dnsstuff.com/network-intrusion-detection-software.

[8] Tek Tools (2020, Feb 14). Intrusion Detection System(IDS) - The Fundamentals.
https://www.tek-tools.com/security/what-is-an-intrusion-detection-system-ids

[9] Wikipedia contributors, "XGBoost," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=XGBoost&oldid=1095596992 (accessed July 18, 2022).

[10] https://www.coursera.org/learn/machine-learning-classification-algorithms, Instructor: Anna Koop.

[11] https://www.udemy.com/course/deeplearning/, instructor: Kirill Eremenko