Network Intrusion Detection using Machine Learning Algorithms

Dr.B.Sankara Babu
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana,India.
bsankarababu81@gmail.com

G.Akshay Reddy
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana,India.
akshayreddy2412@gmail.com

D.Kushal Goud
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana,India.
kushalnani143m@gmail.com

K.Naveen
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana,India.
Kota05473@gmail.com

K.Sai Tharun Reddy
Department of CSE
Gokaraju Rangaraju Institute of Engineering & Technology.
Telangana,India.
saitharunreddykatta@gmail.com

2023 3rd International Conference on Smart Data Intelligence (ICSMDI) | 978-1-6654-6487-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSMDI57622.2023.00071

ABSTRACT

The advancement in wireless communication technology has led to various security challenges in networks. To combat these issues, Network Intrusion Detection Systems (NIDS) are employed to identify attacks. To enhance their accuracy in detecting intruders, various machine learning techniques have been previously used with NIDS. This paper presents a new approach that utilizes machine learning techniques to identify intrusions. The findings of our model indicate that it outperforms other methods, such as Naive Bayes, in terms of accuracy. Our method resulted in a performance time of 1.26 minutes, an accuracy rate of 97.38%, and an error rate of 0.25%.

Keywords – Network Detection system, Intrusion, Network, Machine Learning, Random Forest, Decision Tree, Principal Component Analysis, Support Vector Machine.

INTRODUCTION

Nowadays, the internet is involved in human life in many aspects. Every Person is using internet in their daily life. It has a crucial role in human daily life. We are using internet daily. So, it plays a main role in everyone life for personal activities. We also need to be secure while using the internet. As there are many security concerns for using internet like accessing our personal information. It is necessary for everyone to keep their systems secure from the malicious activities from the intruders.

The use of intrusion detection systems enables organizations to safeguard their systems from potential threats that arise as a result of an increased connection to networks and dependence on information technology. To detect these types of attacks, previous research has employed various techniques. This study presents a new intrusion detection system that incorporates the utilization of both principal component analysis and support vector machine technique. Both methods serve distinct purposes: PCA provides granularity in the data, while SVM facilitates the classification of attacks. The proposed intrusion detection system aims to enhance the system security by detecting and addressing the presence of intruders. This system is designed to detect intrusions. It operates by importing a dataset and reducing its dimensions. To minimize the components of a dataset we required PCA which helps to reduce the dimensions. PCA compute the co-variance matrix which is used to find the co relations between the attributes of dataset. From the covariance matrix it computes eigen vectors and eigen values which are used to find the principal components. Now it creates a feature vector which helps to identify which principal components. We can select the principal components of our wish by passing the components number in the function as a parameter. This system will help the users to easily detect the intrusions in the network and classify them accordingly. The data set used here is NSL-KDD dataset which was downloaded from Kaggle website. The dataset contains the attributes which are used for classifying the intrusions.

978-1-6654-6487-1/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSMDI57622.2023.00071

1.1 DETECTING INTRUSION IN NETWORK (NIDS):

Intrusion refers to unauthorized access to a system and potentially damaging or corrupting the information
within it. Network Intrusion Detection Systems (NIDS) are utilized to observe network traffic and identify potential intrusions. An Intrusion Detection System's main purpose is to safeguard assets from potential security breaches by identifying and alerting on any suspicious activity. It analyses and predicts user behaviour, and based on that, it determines whether the behaviour is considered an attack or normal. The purpose of an intrusion detection system is to recognize and pinpoint security violations. However, it is crucial that the system detects attacks early on in order to minimize the extent of damage caused by them.

1.2 PRINCIPAL COMPONENT ANALYSIS(PCA):

Principal component analysis, or PCA, is a popular unsupervised learning method in machine learning that is used to minimise the number of features in a dataset. The PCA algorithm is selected to improve the accuracy. It helps to identify the important attributes required for classification. PCA reduces the attributes to a desired number. With the retrieval of attributes and the computation of the crucial attributes required for classification, PCA lowers the dimensionality of the data set. PCA compute the co-variance matrix which is used to find the co relations between the attributes of dataset. From the covariance matrix it computes eigen vectors and eigen values which are used to find the principal components. Now it creates a feature vector which helps to identify which principal components. Then sort in descending order of Eigen values. Now it obtains a set of components known as principal components. PCA is a technique that is used to analyse the structure of large, complex data sets by identifying patterns in the data. According to the obtained Eigen values and select n Eigen values. They are called principal components.

1.3 RANDOM FOREST

Random Forest is a popular machine learning algorithm that is utilized for data classification tasks due to its high accuracy and robustness. This algorithm is used because it provides high level of precision and can be implemented easily. It also operates well in large size databases and provides accurate predictions. It combines multiple decision trees generated from different subsets of the training dataset and uses the collective output to improve accuracy. Building a decision tree for each subset follows the initial selection of k random data points from the training set. The number of trees, N, is chosen by the user. During the testing phase, the algorithm makes a prediction for each tree and assigns the data point to the category that receives the most votes from the trees.

1.4 DECISION TREE

Decision trees are a widely used tool for classification and prediction tasks. The features are selected and split based on their statistical importance. In this algorithm, each internal node represents an attribute or feature, while each leaf node indicates a specific class label. They are structured like a flowchart, each internal node represents a test for a specific attribute, and the branches that extend from it represent the possible outcomes of that test. The leaf nodes at the end of these branches contain the final class label that the data point is assigned to. By analysing metrics like information gain and Gini index, the system determines which property at each node is the most informative. This process continues until a class identifier is reached at a leaf node.

1.5 SUPPORT VECTOR MACHINE

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm that can be used to solve both classification and regression problems. These algorithms build non-linear decision boundaries between data points, giving them a high degree of adaptability in managing classification and regression tasks of various levels of complexity. They can find complex relationships between your data without doing lot of transformations. One advantage of support vector machines (SVM) is their efficient use of memory. During the decision-making phase, SVMs employ only a subset of training points, resulting in lower memory usage. SVMs also perform well when there is a clear margin of separation and in high-dimensional spaces. SVMs might be sigmoid, radial basis function (RBF), polynomial, or even linear in shape.

II. LITERATURE REVIEW

The authors presented a method for creating a model for NIDS to classify intruders. They performed binary classification. That means he has two classes, with intrusion or without intrusion. The accuracy that is achieved by this model is good when compared to its predecessor. Here the authors provided a solution for NIDS by applying ML algorithms. They conducted experiments on the KDD dataset. They also provided results on accuracy and classification. In this paper, the authors applied a series of procedures. We have selected traits that are very important for classification. They analyzed all techniques used for intrusion detection. In this experiment, the authors used the KDD

dataset and found that their approach was effective. The paper provides an overview of network intrusion detection systems and the various machine learning algorithms used to classify intrusions. The authors proposed a method for detecting intruders and demonstrated that it provided high detection rates compared to previous techniques. By removing features from the dataset and using them more effectively as input for the intrusion detection system, they also intended to increase the system's effectiveness.

III. PROBLEM DOMAIN

Devices connected to the internet are often targeted by malicious activities, referred to as intrusions. A device known as a Network Intrusion Detection System (NIDS) is used to combat this. For NIDS to be effective, it must be both accurate and efficient in detecting intrusions. There are various machine learning algorithms that can be used for this purpose, such as SVM, Random Forest and Xg Boost. However, further improvements can be made to increase accuracy, detection rates and reduce false alarms. Alternative techniques may also be employed to replace existing methods.

IV. PROPOSED SOLUTION

We proposed a network intrusion detection system used to analyze network traffic and classify intrusions. The proposed system is an anomaly-based system that inspects and returns intrusions is detected when network traffic is not found normally. PCA is utilized in this system to decrease the dataset's dimensionality. After reducing the dimensions, we get the principal components. These attributes are used to train the algorithm and improve its accuracy. This system is designed for early detection of intrusion. It works by importing the dataset and minimizing the dimensions of the dataset.

PCA is used for dimensionality reduction and provides a set of attributes called principal components. A reliable machine learning technique that is frequently used for data classification is the Random Forest algorithm. This algorithm is used because it provides high level of precision and can be implemented easily. It also operates well in large size databases and provides accurate predictions. It combines multiple decision trees generated from different subsets of the training dataset and uses the collective output to improve accuracy.

Decision trees are a widely used tool for classification and prediction tasks. The features are selected and split based on their statistical importance. In this algorithm, each internal node represents an attribute or feature, while each leaf node indicates a specific class label. They are structured like a flowchart, each internal node represents a test for a specific attribute, and the branches that extend from it represent the possible outcomes of that test.

Support Vector Machines (SVM) is a powerful machine learning algorithm that aims to divide training data using a hyperplane such that the expected risk for the given application is minimized. To extend the algorithm to nonlinear data, SVM uses kernel functions, such as polynomial or radial basis functions, to enable the mapping of data into a higher-dimensional feature space, we employ the polynomial kernel in this study. By identifying the support vectors on the function's surface using this approach, SVM can classify new data points effectively.

Evaluation Metrics: In assessing the ability of machine learning methods to detect network intrusions, it is important to use a range of performance measures. These metrics are crucial in the evaluation of such techniques.

Detection Rate:
It typically refers to the ability of a model to correctly identify positive instances or cases.

Recall Rate:
Recall measures the accuracy of a classification model in identifying actual positive cases. Specifically, it determines the proportion of positive cases correctly detected by the model.

False Positive Rate:
The false positive rate (FPR) is a performance metric used in classification models to evaluate the model's effectiveness in identifying negative instances, measures the proportion of negative cases that the model incorrectly identifies as positive.

False Negative Rate:
The false negative rate (FNR) is a performance metric commonly employed in classification models to evaluate the model's ability to correctly identify positive instances, it measures the proportion of actual positive cases that the model incorrectly identifies as negative.

TYPES OF ATTACKS

1	Normal
2	Anomaly

F-Measure:
F-measure, also known as F1 score, is a widely used metric in machine learning to evaluate the effectiveness of binary classification models, measuring the model's precision and recall and providing a single score that represents its overall performance.

CONFUSION MATRIX-ACTUAL PREDICTIONS

	Predicted Class Positive (Normal)	Predicted Class Negative (Attack)
Actual Class Positive (Normal)	A(TN)	B(FP)
Actual Class Negative (Attack)	C(FN)	D(TP)

CONFUSION MATRIX-EXPECTED PREDICTION

	Predicted Class Positive	Predicted Class Negative
Actual Class Positive	E	F
Actual Class Negative	G	H

Root Mean Square Error:
Root Mean Square Error (RMSE) is a commonly used metric for evaluating the performance of machine learning models, commonly used in regression problems to quantify the disparity between predicted and actual values.

V. DATASET USED

NSL-KDD dataset: This dataset was downloaded from Kaggle website. This includes 42 attributes which describes about a network. The attributes in the dataset are used for the classification of the intrusion. The dataset is preprocessed such that it will not contain any noisy or missing data. Later these attributes are reducing to desired number by using PCA, so the accuracy gets improved.

VI. RESULTS:

The various machine learning algorithms are trained on the KDD dataset by using cross validation and after the training they were tested. The result obtained from the tested dataset shows good classification rate and less error rate. The System is detecting the intrusions based on the parameters obtained by PCA after applying PCA on the dataset. The results state that the model is classifying the given data correctly. When compared to earlier models, the model's accuracy is good. The error rate of the model is less which leads to good classification.

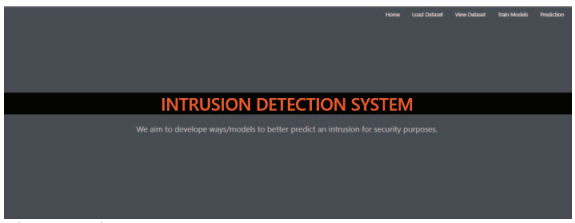

Fig 1:Main Page

This is the main page; it is obtained by compiling the code by which it generates a hyperlink which redirects us to this page.

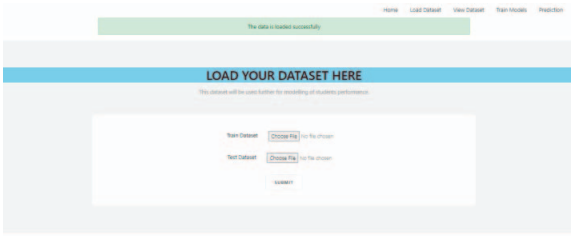

Fig 2:Data Upload

This is the place we can load our both training and testing data. We need to click on submit in order to load the dataset.

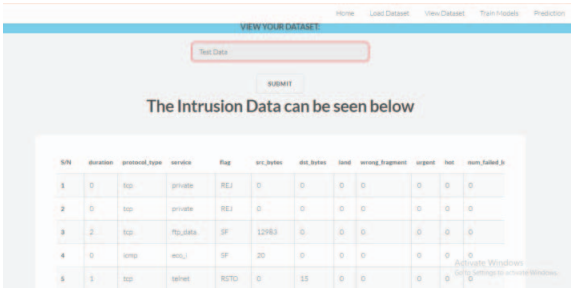

Fig 3:Dataset Information

This shows the dataset information. It includes attributes information and values. We can view both training and testing dataset.

Fig 4:Training Phase

Here we will train our models based on the uploaded datasets. We will be having several models like Random Forest, XGBoost, Decision Tree, SVM. We need to select the model and click on submit to get the model trained by the uploaded dataset.

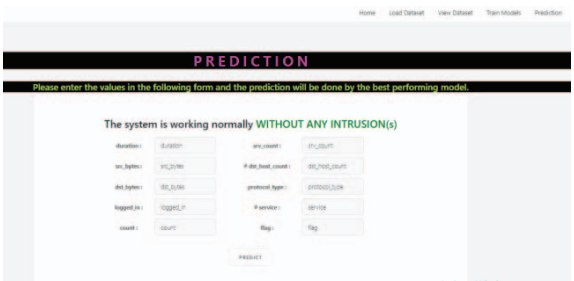

Fig 5:Results based on PCA Attributes

This page will provide the results based on the attributes generated by PCA. It will classify whether the intrusion is detected or not by taking the attribute values as input.

Fig 6:Final Result Display

If the intrusion is detected it displays intrusion detected else, it shows no Intrusion.

VII. CONCLUSION:

The proposed method is capable of effectively detecting and categorizing online intrusions based on their behaviour using network traffic metrics. Our approach achieves a low false positive rate, which is beneficial for accurately classifying attacks and normal network traffic. It is worth emphasising that, despite the use of various algorithms, no system can ever be completely secure, and computer security remains a constantly evolving and challenging field of study. By using this method, we can quickly detect an intrusion and take necessary measures to protect the system. The proposed approach has a runtime of 1.26 minutes, an accuracy rate of 97.38%, and an error rate of 0.25%.

VIII. REFERENCES

[1] C Chang and C J Lin, LIBSVM, Using LIBSVM, 2009. "Libraries Used for Support Vector Machines in Classification."

[2] Rung-Ching Chen, Kai-Fan Cheng, Chia-Fen Hsieh, International Journal of Network Security & Its Applications (IJNSA), Band 1, Nr. 1, 2009, "Using Rough Sets and Support Vector Machines for Network Intrusion Detection Systems."

[3] Liberios Vokorokos, Alzebeta Kleniova, "Intrusion Detection System Level Network Security", IEEE Netzwerk, 2004.

[4] Thomas Heyman, Bart De Win, IEEE Transactions 2004, "Improving Intrusion Detection with Alert Verification."

[5] D. E. Denning and P. G. Neumann, the technical report project 5910 by SRI International in 1985 covers the topics of collecting and processing usage data and analysing audit trails.

[6] D. E. Denning, The IEEE Transactions on Software Engineering published a paper in 1987 titled "An intrusion detection model", which presents a model for detecting intrusions in software systems. The paper is found in volume 13, issue 2, and spans pages 118-131.

[7] D. Barbara, J. Couto, S. Jajodia, L. Popyack, and N. Wu, In June 2001, the Proceedings of the 2nd Annual IEEE Workshop on Information Assurance and Security featured a paper titled "ADAM: Detecting intrusions by data mining." The paper, which spans pages 11-16, presents a method for detecting intrusions using data mining techniques.

[8] G. Wang, J. Hao, J. Ma, and L. Huang, Artificial neural networks and fuzzy clustering in intrusion detection, Expert Systems with Applications, vol. 37, no. 8, pp. 6225–6232, 2010.

[9] H. Debar and B. Dorizzi, "A neural network component for an intrusion detection system," in Oakland, CA, May 1992, pp. 240–250, Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy.

[10] H. A. Nguyen and D. Choi, Network intrusion detection using data mining: Y. Ma, D. Choi, and S. Ata, editors, classifier selection model, Proceedings of Challenges for Next Generation Network Operations and Service Management (APNOMS 2008), Lecture Notes in Computer Science, vol. 5297, Springer, pp. 399–408.