iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.61047

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: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

IDSUML: Intrusion Detection System Using Machine Learning

Miss. Roshni R. Makode[1], Prof. Anita Mahajan[2]
[1]*Student, Department of Computer Engineering Dr. D. Y. Patil School Of Engineering, Pune - 412 105,Savitribai Phule Pune University Pune Maharashtra, India*

Abstract: ***Network and system security is essential to modern digital communication. There will be multiple successful attempts by hackers and other intruders to obtain illegal access to networks and internet services. The development of secure systems is continuously accompanied by new dangers and the corresponding responses. Using intrusion detection systems (IDS) is one choice. An intrusion detection system. Main objective is to safeguard its resources from possible dangers. These acts are categorized as either potential or conventional attacks since it is capable of analysing and forecasting user behaviour. To find network intrusions, we typically employ Support Vector Machines (SVM) and Rough Pure Mathematics (RST). RST is used to preprocess and decrease the amount of data when packets are retrieved from the network. The RST-selected options provided to the SVM model enable it to provide many views and an explanation. A good application of the approach is to reduce the density of knowledge in a certain field. By comparing the data with Principal element analysis, the experiments show how RST and SVM schema may lower the false positive rate and increase accuracy (PCA)***

Keywords: Intrusion detection system, machine learning, nature inspired algorithms, deep learning, feature selection

I. INTRODUCTION

Since 1987, intrusion detection systems have been the focus of ongoing research and development. This is especially true now as the threats associated with computers and networks are increasing, making automated monitoring a crucial component of information technology security. The process of keeping an eye on system or network events and looking for indications of invasions is known as intrusion detection. Attempts to get around security measures or compromise a computer or network & availability, confidentiality, or integrity are known as intrusions. They are brought about by hackers breaking into systems via the Internet, unauthorized users trying to obtain additional unauthorized access, and authorized users abusing their power.

Abuse and anomaly detection are the two main techniques used to find invasions. The pre- recording representation of certain patterns for incursions serves as the foundation for misuse identification and permits any matches to be communicated with them during the current activity. Known attack vector patterns are referred to as signatures. Next, network traffic statistics are used to train a classifier that can distinguish between the two sets. The discriminative features, which are made up of different network traffic statistics, were extracted using a number of data mining techniques, including frequent episodes and association criteria. On the other hand, getting accurately labeled data examples can be challenging and time-consuming, particularly when working with novel attack kinds and patterns. But after training models for usual traffic patterns, anomaly detection classifies any network activity that significantly deviates from regular patterns as an invasion. While many concur, that neural networks are a helpful method for adaptively identifying patterns, their application is severely limited by their high processing overhead and lengthy training cycles, particularly in intrusion detection scenarios where a significant amount of pertinent data is needed

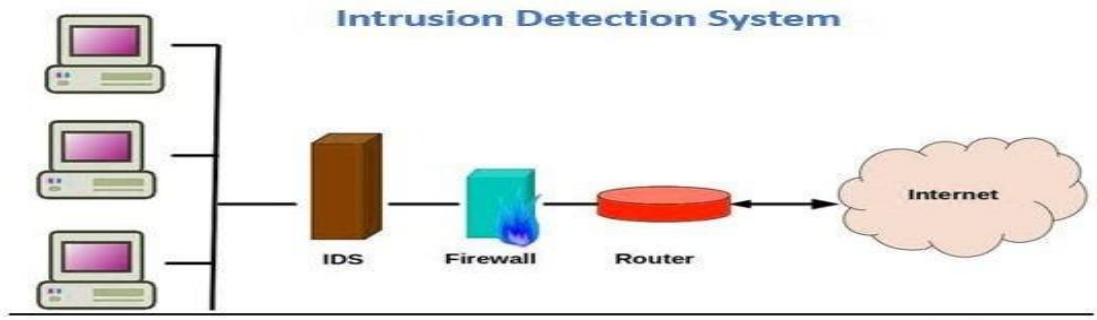


Fig 1. Intrusion detection systems

©IJRASET: All Rights are Reserved

II. LITERATURE REVIEW

A. *Using Apache Spark Platform to Apply a Deep Learning Model for Intrusion Detection*

This article proposes an IDS system based on deep learning algorithms for the attacks included in the NSL-KDD dataset. The training process will be implemented on Apache Spark. For short, we refer to this model as DLS-IDS (Deep Learning Spark Intrusion Detection System). The DLS-IDS solves the NSL-KDD dataset related problems, defines the best model arrangement and model elements to produce a high intrusion detection accuracy, as well as determines the best Apache Spark cluster configurations to reduce the implementation process time. Computer networks have proliferated over the years, adding to social and economic growth. The Internet Security Threat Report (ISTR) states that 1 in 13 Web requests is malware. The spam rate in e- mails had increased to 55%, ransomware had risen to 46 %, and other Internet threats [1]. Cybercrime and threat actions have grown and have become a critical threat. This growth promoted an increase in network security importance. By analyzing packets captured from the network, IDS helps to detect threats [2].

B. *Hybrid Deep Learning and Scalable K-Means C Random Forest IntrusionDetection System*

Our preliminary investigation reveals that both TIDCS and TIDCS-A show good results in terms of detection accuracy. TIDCS applies periodic trust updates based on the node's past information. TIDCS-A proposes a dynamic algorithm to compute the exact time for nodes cleansing states and restricts the exposure window of the nodes. TIDCS and TIDCS-A are used according to the applied security policy. It has been proven that TIDCS-A is faster in the detection of malicious nodes and more complex compared to TIDCS. By using the NSL-KDD and UNSW datasets, TICDS performs better than previous work (NB, AODF, CADF, and TANN) in terms of average accuracy, the detection rate, false alarm. The security of the networks has become an essential issue in any distributed system. Intrusion detection systems came to aid in adding a layer of protection over these networks by detecting unauthorized intrusion scenarios. In this paper, we propose a novel model for network intrusion detection, namely TICDS and TICDS-A. In particular, the proposed system combines machine learning techniques and past information to create a trusted cloud environment. TICDS and TICDS-A apply cleansing activities for the participants' nodes, regardless of the presence/absence of attack alarms. TIDCS has a fixed periodic cleansing window and TIDCS-A has a dynamic window for network cleansing and anomaly detection. Simulation results show the good performance of the two proposed models.

C. *A Novel Approach to Deep Learning for Intelligent Intrusion Detection Systems*

"A Novel Approach to Deep Learning for Intelligent Intrusion Detection Systems" by R Vijayakumar, the findings of the research are likely summarized, highlighting the effectiveness of the proposed deep learning approach compared to existing methods. The performance of the intrusion detection system would be evaluated based on metrics such as accuracy, false positive rate, and computational efficiency. The conclusion may also discuss implications for the field of cyber security and propose avenues for future research, such as exploring different deep learning architectures or addressing real-world deployment challenges. Additionally, there might be mention of the broader impact of the research beyond intrusion detection, underscoring its potential applications in related area

D. *A Deep Learning Technique for Wireless Intrusion Using Filter-Based Feature Engineering*

This paper presented the design, implementation and testing of a DL based intrusion detection system using FFDNNs. A literature review of ML and DL methods was conducted and it was found that the most efficient approach to intrusion detection has yet to be found. The FFDNN models used in this research were coupled to a FEU using IG in a bid to reduce the input dimension while increasing the accuracy of the classifier. The dataset used in this work is the NSL-KDD. For the binary and the multiclass classifications problems, the FFDNNs models both with a full and a FEU- reduced feature space achieved a performance that is superior to SVM, RF, NB, DT and KNN. In future work, we aim at finding a strategy to increase the detection rates of R2L and U2R attacks in the NSL-KDD dataset. Moreover, we will apply the FEU and the FFDDNs to the AWID dataset in order to investigate further the superiority of DL based methods for IDS over other ML approaches. "A Deep Learning Technique for Wireless Intrusion Using Filter-Based Feature Engineering" by Sydney Mambwe Kasongo (IEEE), one might expect to find a concise summary of the key findings and contributions of the research. This would likely include an evaluation of the effectiveness of the proposed deep learning technique for wireless intrusion detection, potentially comparing its performance with existing methods. The conclusion may also discuss any limitations encountered during the study and propose directions for future research to address these limitations or further improve the proposed approach.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Additionally, there might be reflections on the broader implications of the research for the field of wireless security and suggestions for practical applications or deployment strategies. Finally, the conclusion may end with some closing remarks, emphasizing the significance of the work and encouraging further exploration in the field of wireless intrusion detection using deep learning technique

E. TIDCS: A Feature Selection System Based on Dynamic Intrusion Detection and Classification

The security of the networks has become an essential issue in any distributed system. Intrusion detection systems came to aid in adding a layer of protection over these networks by detecting unauthorized intrusion scenarios. In this paper, we propose a novel model for network intrusion detection, namely TICDS and TICDS-A. In particular, the proposed system combines machine learning techniques and past information to create a trusted cloud environment. TICDS and TICDS-A apply cleansing activities for the participants' nodes, regardless of the presence/absence of attack alarms. TIDCS has a fixed periodic cleansing window and TIDCS-A has a dynamic window for network cleansing and anomaly detection. Simulation results show the good performance of the two proposed models.

III. IMPLEMENTATION

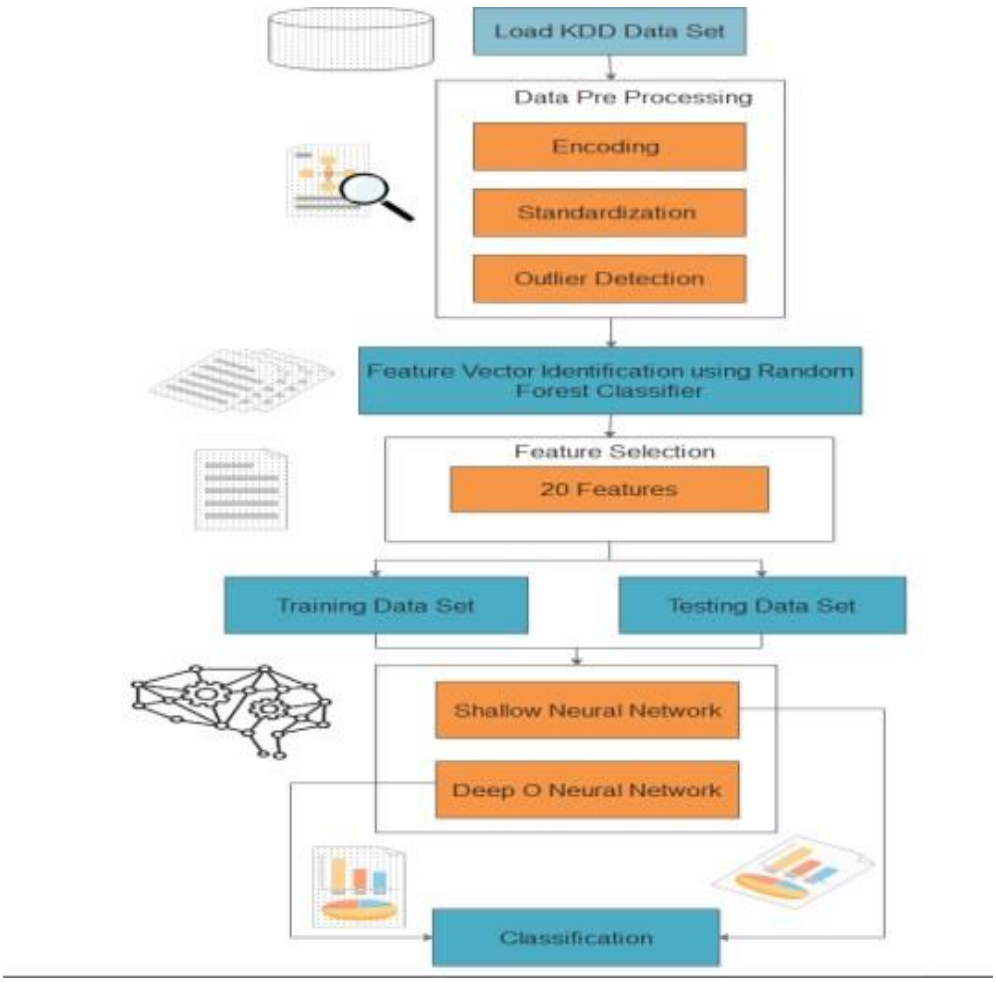


Fig 2 The system architecture for an AI-driven Intrusion Detection System (IDS) employing Generative Adversarial Network (GAN) models involves

A. Data Collection Layer

Network Sensors/Probes: Collect network traffic data in real-time from various network points. Preprocessing Module: Clean, filter, and preprocess the raw data before feeding itinto the analysis layer

B. Analysis and Anomaly Detection Layer

GAN-based Anomaly Detection Module: Incorporate GAN models designed specifically for anomaly detection within network traffic. Real-time Analysis Engine: Analyze incoming traffic for anomalies using GAN-generated models. Threshold and Classification Module: Set thresholds for anomaly classificationand classify detected anomalies.

C. Adaptive Learning and Improvement Layer

Adaptive Machine Learning Module: Incorporate mechanisms for continuous learning and adaptation to evolving threats based on GAN's outputs. Re-training Strategies: Define strategies for periodic re-training of GAN models to adapt to new attack patterns and minimize false positives/negatives.

D. Response and Alerting Layer

Response Protocols: Activate response protocols upon the detection of identified threats for immediate mitigation measures. Alerting Mechanism: Generate alerts or notifications for security personnel to take appropriate action based on threat severity.

E. Monitoring and Reporting Layer

Continuous Monitoring Framework: Establish a framework for ongoing monitoring of network activities. and anomalies. Threat Intelligence Reporting: Generate comprehensive reports providing actionable insights for proactive response strategies.

F. User Interface and Integration Layer

User-Friendly Interface: Develop intuitive interfaces for easy deployment, configuration, and management of the IDS.Integration into Network Infrastructure: Ensure seamless integration into existing network infrastructures with minimal disruption.

G. Security and Access Control Layer

Authentication and Authorization: Implement strict access control mechanisms to safeguard the IDS against unauthorized access. Data Encryption and Privacy: Ensure data encryption protocols to maintain the privacy and integrity of collected and analyzed data.

H. Scalability and Performance Layer

Scalable Architecture: Design the system architecture to accommodate varying network loads and scale resources accordingly.

I. Performance Optimization

Optimize the system for minimal impact on network performance and latency. This architecture enables the integration and orchestration of components, facilitating accurate anomaly detection, adaptive learning, real-time responsiveness, and proactive defense strategies against cyber threats with in complex network Environments.

J. Implementation Module

The proposed system consists of different modules, the main modules are listed below:

K. Module Description

1) *Module 1*: Login and registration module
2) *Module 2*: Read Dataset This code will read the data based on nsl and kdd datasets. which has 14 attributes.
3) *Module 3*: Preprocessing of data Preprocessing of data is done in this module with pandas reader. The label processing is done with the concatenated dataset and original testing and training dataset. Visualization of class distribution of the dataset both before and after the dataset. The binary classification is done with the count of each class and its percentage.
4) *Module 4:* Machine Learning Algorithms. The learning algorithm like k means, random forest, and are applied here on the test and Train data. The training of random fores, KNN and SVM Classifier for each attack type separately both all features and selected features.
5) *Module 5:* Deep Learning Algorithms. In deep learning algorithm we have used lstm algorithm. in this algorithm logarithmic scaling and min max scaling to the features to normalize their ranges. In this lstm model uses the kerns sequential Api. The model architecture uses the lstm layer with 80 units and dense layer with softmax activation function.
6) *Module 6*: Model evaluation is done with the pie chart and bar graph. the model evaluation is done with matplotlib and seaborn library.

IV. ALGORITHM

A. Machine Learning Algorithms

1) *K-means Clustering:* K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
2) *What is K-Means Algorithm?:* K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre- defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main motivation of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The k-means clustering, This algorithm mainly performs two tasks Determines the best value for K center points or centroids by an iterative process. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center create a cluster.

B. How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.
Step-2: Select random K points or centroids. (It can be other from the input dataset).
Step-3: Assign each data point to their closest centroid, which will form the predefined K- clusters.
Step-4: Calculate the variance and place a new centroid of each cluster.
Step-5: Repeat the third step, which means reassign each data point to the new closest centroid of each cluster.
Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.
Step-7: The model is ready.

Support Vector Machines (SVM) are a class of supervised learning algorithms used for classification, regression, and outlier detection tasks. Given a set of training data points (xi, yi), where xi is the feature vector and yi is the class label (yi = ±1), the goal is to find a hype rplane represented by:

$w \cdot x + b = 0$

Where:

- w is the weight vector perpendicular to the hyperplane,
- b is the bias term,
- x is the input feature vector.

The distance between a data point and the hyperplane is given by:

Distance $=|w \cdot x + b|/\| w \|$

C. Support Vector Machine or SVM

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyper plane. SVM chooses the extreme points/vectors that help in creating the hyper plane.

D. Applications of SVM

Support Vector Machines (SVM) are widely used in the field of Intrusion Detection Systems (IDS).

1) *Binary Classification:* SVMs are often used for binary classification tasks in IDS. Given a dataset containing both normal and attack instances, SVM can learn a decision boundary to separate the two classes. Any incoming data point can then be classified as either normal or an attack based on which side of the boundary it falls on.
2) *Multi-class Classification:* SVMs can also be extended to handle multi-class classification problems in IDS, where there are more than two classes of network traffic (e.g., normal, DoS, probing, malware).Methods like one-vs-all or one-vs

E. Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions and it predicts the final output The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting. Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase The Working process can be explained in the below steps.

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

F. Deep Learning Algorithms

A neural network is an oriented graph. It consists of nodes which in the biological analogy represent neurons, connected by arcs. It corresponds to Artificial Neural networks (ANN) or neural networks are computational algorithms. It intended to simulate the behavior of biological systems composed of “neurons”. ANNs are computational models inspired by an animal’s central nervous systems. It is capable of machine learning as well as pattern recognition. These presented as systems of interconnected “neurons” which can compute values from inputs. Dendrites and synapses. Each arc associated with a weight while at each node. Apply the values received as input by the node and define Activation function along the incoming arcs, adjusted by the weights of the arcs. A neural network is a machine learning algorithm based on the model of a human neuron. The human brain consists of millions of neurons. It sends and process signals in the form of electrical and chemical signals. These neurons are connected with a special structure known as synapses. Synapses allow neurons to pass signals. From large numbers of simulated neurons neural networks forms. Artificial Neural Network is an information processing technique. It works like the way human brain processes information. ANN includes a large number of connected processing units that work together to process information. They also generate meaningful results from it. We can apply neural network not only for classification. It can also apply for regression of continuous target attributes. Neural networks find great application in data mining used in sectors. For example economics, forensics, etc. and for pattern recognition. It can be also used for data classification in a large amount of data after careful training.

A Long Short-Term Memory (LSTM) network is a type of recurrent neural network
(RNN) architecture designed to overcome the vanishing gradient problem and capture
Long-term dependencies in sequential data. Let denote the input at time step.
Let denote the hidden state at time step .Let denote the cell state at time step.
The LSTM cell has four main components: input gate, forget gate, cell state update, and output gate. Each of these components involves several mathematical operations.

1 Input Gate (it) : Controls the flow of information into the cell state.

• Calculate the input gate activation:

$it = \sigma(Wi \cdot [ht-1, xt] + bi)$

2 Forget Gate (ft) : Controls the flow of information out of the cell state.

• Calculate the forget gate activation:

$ft = \sigma(Wf \cdot [ht-1, xt] + bf)$

3 Cell State Update ($\tilde{c}t$) : Calculating the new candidate cell state.

• Calculate the candidate cell state:

$\tilde{c}t = \tanh(Wc \cdot [ht-1, xt] + bc)$

4 Cell State (ct) : Update the cell state.

• Update the cell state using the input and forget gates:

$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$

Where $\odot$ denotes element-wise multiplication.

5. Output Gate (o_t) : Controls the flow of information from the cell state to the hidden state.

• Calculate the output gate activation:

$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

V. EXPERIMENTAL RESULTS

```
DOS Accuracy: 0.99814 (+/- 0.00171)
DOS Precision: 0.99852 (+/- 0.00223)
DOS Recall: 0.99678 (+/- 0.00343)
DOS F-measure: 0.99765 (+/- 0.00234)
Probe Accuracy: 0.99588 (+/- 0.00369)
Probe Precision: 0.99654 (+/- 0.00373)
Probe Recall: 0.99252 (+/- 0.00704)
Probe F-measure: 0.99457 (+/- 0.00397)
U2R Accuracy: 0.99755 (+/- 0.00228)
U2R Precision: 0.97188 (+/- 0.08437)
U2R Recall: 0.85942 (+/- 0.13142)
U2R F-measure: 0.90132 (+/- 0.13149)
R2L Accuracy: 0.97904 (+/- 0.00415)
R2L Precision: 0.97445 (+/- 0.01158)
R2L Recall: 0.96918 (+/- 0.01063)
R2L F-measure: 0.97084 (+/- 0.01078)
Accuracy: 0.99715 (+/- 0.00278)
Precision: 0.99678 (+/- 0.00383)
Recall: 0.99665 (+/- 0.00344)
F-measure: 0.99672 (+/- 0.00320)
Accuracy: 0.99077 (+/- 0.00403)
Precision: 0.98606 (+/- 0.00675)
Recall: 0.98508 (+/- 0.01137)
F-measure: 0.98553 (+/- 0.00645)
```

Fig.3 Accuracy Parameter

```
Train:
Dimensions of DoS: (113270, 123)
Dimensions of Probe: (78999, 123)
Dimensions of R2L: (68338, 123)
Dimensions of U2R: (67395, 123)

Test:
Dimensions of DoS: (17171, 123)
Dimensions of Probe: (12132, 123)
Dimensions of R2L: (12596, 123)
Dimensions of U2R: (9778, 123)
```

Fig 4. Testing Dos File



Fig.5 Dashboard



Fig.6 Selection of test file

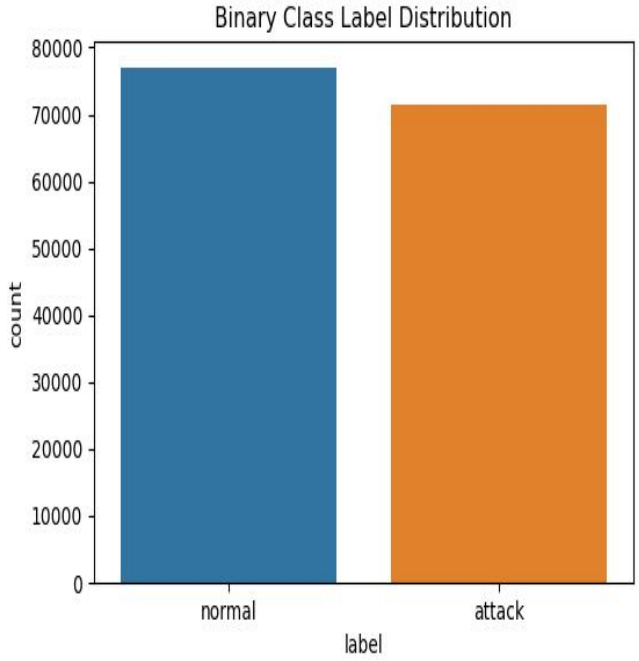


Fig. 7 Binary Class Distribution

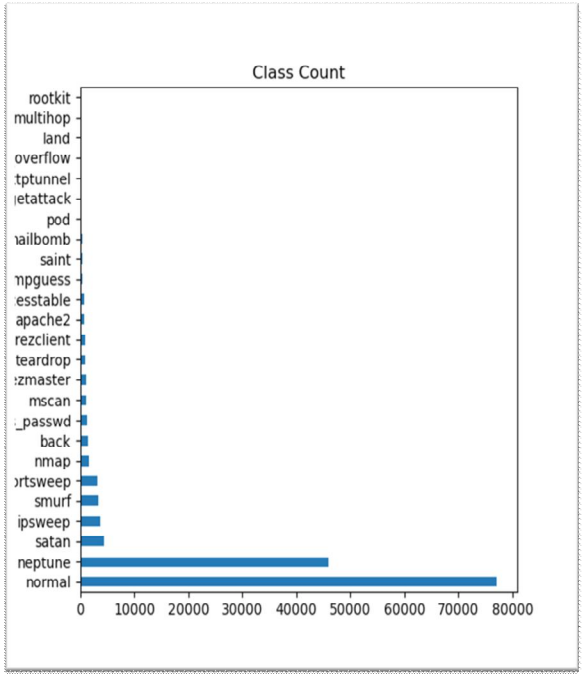


Fig 8 class count graph

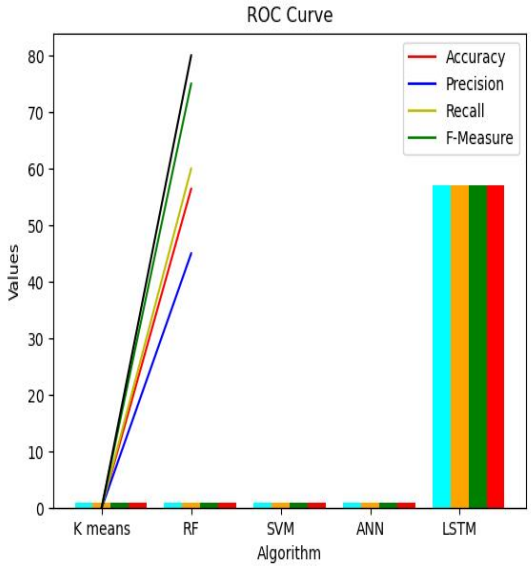


Fig 9. ROC curve

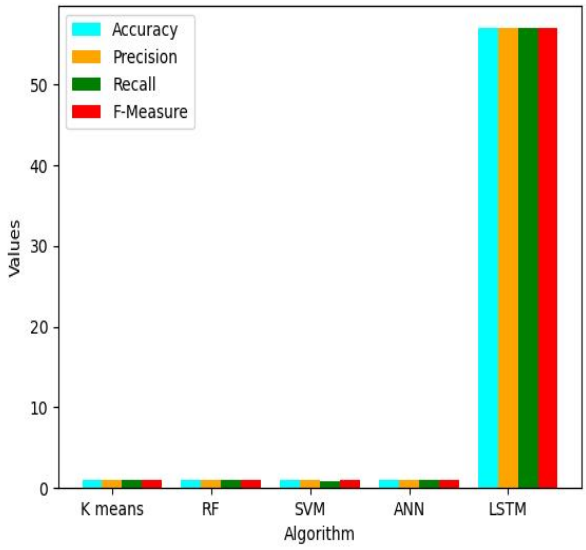


Fig 10. bar graph

VI. CONCLUSION

An intrusion detection system can be used to find both known and undiscovered incursions before the attacker damages any networked equipment. Using the NSL KDD dataset and the recurrent and deep neural network techniques, a suggested network intrusion detection alert system is developed in this study in order to build a dependable and adaptable network intrusion detection system. With a comprehensive taxonomy covering deep learning, machine learning, and nature-inspired approaches utilized in intrusion detection, this study provides readers with a fundamental understanding of intrusion detection systems. By critically analyzing each strategy, the benefits and drawbacks have become evident. The paper summarizes current research concerns in the field of network intrusion detection systems, provides a wide overview, and projects future advances.

©IJRASET: All Rights are Reserved

By showcasing the potential of machine learning and synthesizing the current state of the art, the work serves as an important appraisal. This will open up new avenues for advancement in the continuous hunt for reliable and flexible intrusion detection methods. The paper delves into the specifics of the datasets employed in these inquiries, presenting a plethora of sources that bolster the advancement of IDS methodologies using MLP algorithm; it recommends users by their interest.

REFERENCES

[1] CHAO LIU 1,2, ZHAOJUN GU2,3, AND JIALIANG WANG3., ",A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning" Received April 29, 2021, accepted May 16, 2021, date of publication May 20, 2021, date of current version May 27, 2021. Digital Object Identifier 10.1109/ACCESS.2021.3082147

[2] R. VINAYAKUMAR 1, MAMOUN ALAZAB2, (Senior Member, IEEE), K. P. SOMAN1, PRABAHARAN POORNACHANDRAN3 , AMEER AL-NEMRAT4 , AND SITALAKSHMI VENKATRAMAN5, A. Radford and V. Sivaraman, "Deep Learning Approach for Intelligent Intrusion Detection System," Received December 27, 2018, accepted January 3, 2019, date of current version April 11, 2019. Digital Object Identifier 10.1109/ACCESS.2019.2895334

[3] MOHAMED HAGGAG 1,3, (Member, IEEE), MOHSEN M. TANTAWY2 , AND MAGDY M. S. EL-SOUDANI3 , (Senior Member, IEEE) "Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform," Received July 15, 2020, accepted August 10, 2020, date of publication August 27, 2020, date of current version September 18, 2020. Digital Object Identifier 10.1109/ACCESS.2020.301993

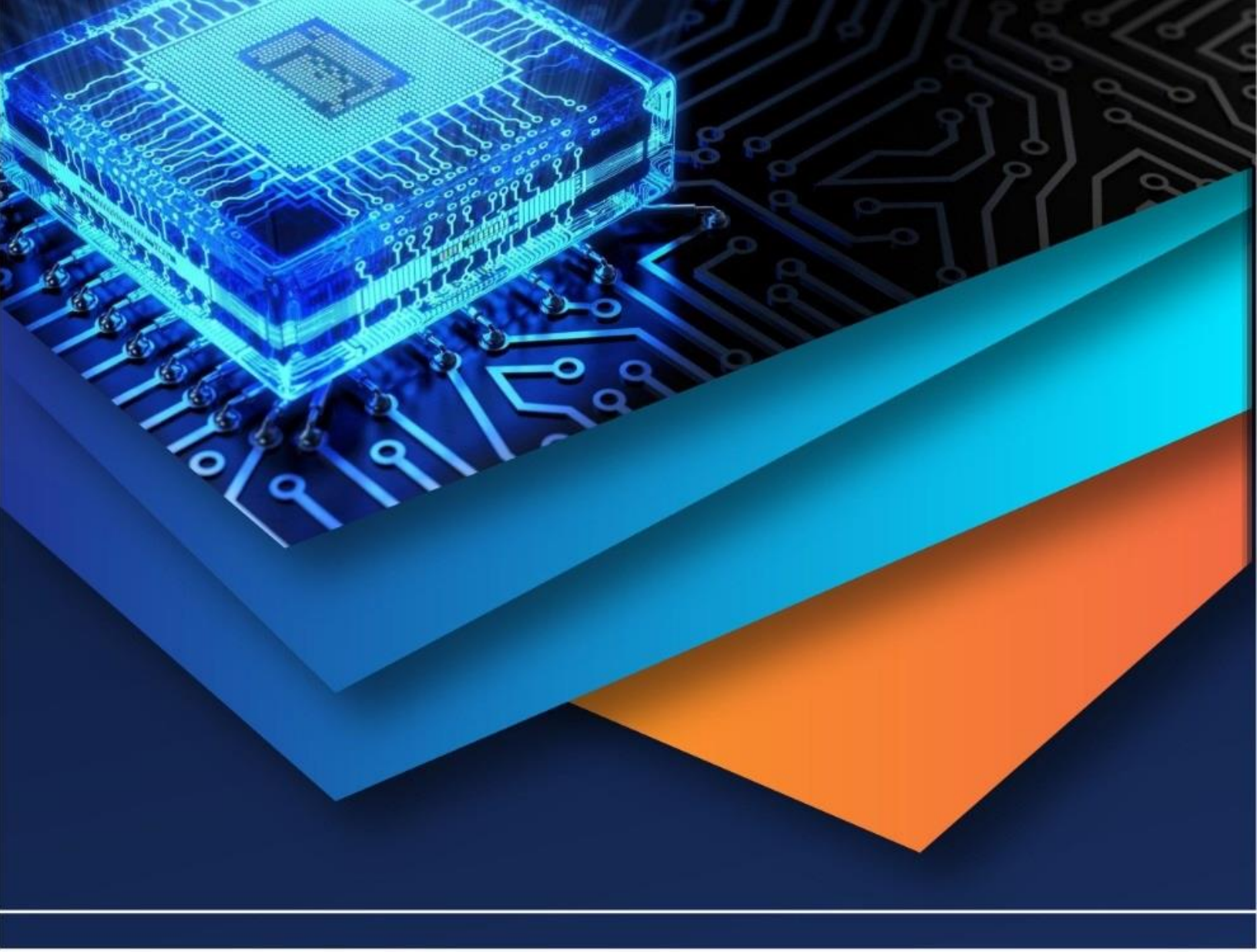
[4] ZINA CHKIRBENE 1, AIMAN ERBAD 1 , RIDHA HAMILA 1 , AMR MOHAMED 1 , MOHSEN GUIZANI 1 , AND MOUNIR HAMDI 2 "TIDCS: A Dynamic Intrusion Detection and Classification System Based Feature Selection." Received April 14, 2020, accepted April 28, 2020, date of publication May 15, 2020, date of current version June 3, 2020. Digital Object Identifier 10.1109/ACCESS.2020.2994931.

[5] Emad E. Abdallah"InformationAge. "Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey", April 2022

[6] Sim Hoong Kok., "A review of intrusion detection system using machine learning approach Mobile Compute., January 2019

[7] Ali, S. Sabir, and Z. Ullah, "Internet of Things security, device authentication and access control: A review," 2019. [Online]. Available: http://arxiv.org/abs/1901.07309.Dini, P.; Elashi, A.; Begni, A.; Saponara, S.; Zheng, Q.; Gasmi, K. Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity. Appl. Sci. 2023,13, 7507. https://doi.org/10.3390/app13137507

[8] M. Chalé and N. D. Bastian, "Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems," Expert Systems with Applications, vol.207, Article, ID117936,2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)