IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IDS FOR MEDIUM SIZE NETWORK USING DECISION TREE ALGORITHM

[1]ASHUTOSH KUMAR, [2]AVINASH KUMAR, [3]ABHISHEK KUMAR,[4] DR.R.SUDHAKAR, [5]DR. J. JAYAPRAKASH

[6]DR. G. VICTO SUDHA GEORGE

[1,2,3]CSE Students, [4,5,6]Faculty, Dept of Computer science and Engineering,

DR. M.G.R EDUCATIONAL AND RESEARCH INSTITUTE, Maduravoyal, Chennai-95, Tamil Nadu, India

***Abstract*:** Intrusion detection systems (IDS) are secure for protecting computer networks, and this study suggests an efficient IDS designed specifically for small to medium-sized networks. The system combines the decision tree algorithm with a machine learning framework by blending traditional anomaly detection with signature-based techniques. This combination provides a comprehensive system for detecting and responding to suspicious activities.The system strikes a compromise between accuracy and resource utilization by using lightweight algorithms and rule-based procedures, with a focus on the decision tree algorithm. The integration improves intrusion detection accuracy and overall system performance. The system's simplicity allows for quick deployment and maintenance, making it an effective solution for network security reinforcement. Testing findings confirm its ability to identify frequent network intrusions, making it a practical and cost-effective security solution for businesses with small to medium-sized networks.

Keyword: IDS, LAN network, Machine learning, Decision tree Algorithm, Signature-based detection.

I. INTRODUCTION

The internet's role in everyday life has grown significantly in recent years. Everyone's life has become increasingly dependent on the internet. Everyone now relies heavily on the internet. Protecting the system against harmful behavior is extremely important now that more people are using the internet for personal reasons. Various attacks have been discovered on the network or system. Wormholes, black holes, and grey holes are examples of network attacks. The aim of these attacks is to either damage or steal data from any system. Attackers utilize different methods, such as DoS, probe, snort, and r2l, to manipulate the data. An intrusion detection system was created to shield the system from these types of attacks. IDS monitors and prevents system attacks. Many studies have explored different strategies to defend against cyber threats. They have introduced an advanced intrusion detection system that works on a machine learning approach and incorporates the decision tree algorithm and principal component analysis (PCA) into the powerful random forest framework. This integration makes the system more efficient in identifying and preventing potential attacks. The decision tree method, nested within the random forest, improves classification accuracy, while PCA enhances data precision. This integration demonstrates an adaptive intrusion detection system that utilizes machine learning concepts and decision tree algorithms to provide accurate threat identification in a dynamic cybersecurity [3].

II. LITERATURE SURVEY

Jafar Abo Nada and Mohammad Rasmi Al-Mosa's paper "A Proposed Wireless Intrusion Detection Prevention and Attack System" addresses the drawbacks of standard network security solutions for wireless networks. The authors demonstrated a Wireless Intrusion Detection, Prevention, and Attack System (WIDPAS) with monitoring, analysis, and defense functionalities. The system is intended to detect and mitigate security threats in wireless networks by constantly monitoring for anomalies, analyzing attacks, and implementing protective measures. Overall, the study contributes to network security by offering a specialized answer to wireless technology challenges. [1]

In this paper, we unveil the outcomes of our examinations evaluating the efficiency in detecting various attack types (comprising IDS, malware, and shell code). We evaluate recognition effectiveness utilizing the Random Forest methodology across several datasets extracted from the Kyoto 2006+ dataset, encompassing the latest network packet data amassed for intrusion detection system development. We conclude with deliberations on prospective research endeavors [2].

The research paper titled "On the Selection of Decision Trees in Random Forests," authored by S. Bernard, L. Heutte, and S. Adam, delves into the limitations of conventional random forest (RF) induction techniques. It brings attention to issues stemming from the fixed number of randomly added decision trees, which adversely impact interpretability and ensemble performance. The authors frame this as a classifier selection dilemma and demonstrate that, even with subpar methods, it's possible to obtain superior subsets of decision trees. They argue that the traditional RF induction approach might not be the most optimal and advocate for a more strategic addition of trees to enhance RF classification accuracy [3].

In contemporary computer and network security, intrusion detection systems (IDS) are essential elements. The study utilizes the NSL-KDD intrusion detection dataset, an upgraded version of KDDCUP'99. However, inherent class imbalances in the NSL-KDD dataset impede effective machine learning applications for intrusion detection. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) is applied to the training dataset. Furthermore, a feature selection method based on information gain is employed to create a reduced feature subset of the NSL-KDD dataset. The proposed intrusion detection framework adopts random forests as its classifier. Empirical findings indicate that combining the Random Forests classifier with SMOTE and information gain-based feature selection leads to superior performance in developing efficient and effective IDS for network intrusion detection[4].

An intrusion detection system (IDS) scans networks or systems for harmful activities. Traditional IDS may face challenges in identifying sophisticated cyber attacks like low-rate DoS and unknown threats. Machine learning has gained popularity as a solution to these limitations. This study proposes using the PCA-Scale technique to enhance the Gated Recurrent Unit (GRU)'s intrusion detection accuracy. This method offers two options, PCA-Standardized and PCA-MinMax, which are integrated into the GRU layer. Experimental results on real-world datasets (KDD Cup 99 and NSL-KDD) demonstrate that the GRU model trained with the PCA-Scaled approach significantly improves performance [5].

III. OBJECTIVE

In this study, we develop an Intrusion Detection System (IDS) utilizing machine learning, specifically the Decision Tree algorithm, to identify and address network intrusions. By analyzing network data, the system categorizes behavior as normal or anomalous, enhancing cybersecurity. Evaluation metrics ensure the system's accuracy and reliability in identifying security risks. Focused to showcase the viability of machine learning for building robust intrusion detection systems capable of safeguarding networks against various cyber threats.

IV. EXISTING SYSTEM

Iftikhar Ahmad and his team investigated various machine learning techniques for detecting intrusions. They assessed SVMs, Extreme Learning Machines, and Random Forests. Their research revealed that the Extreme Learning approach outperformed the other algorithms [1]. B. Riyaz and his team aimed to enhance the dataset for the intrusion detection system. They elevated the dataset's quality through a fuzzy rule-based feature selection method. Utilizing the KDD dataset, they noted a significant enhancement in the intrusion detection system's outcomes [16].

Existing internet-based systems can be vulnerable to intrusions, which can compromise security. They often struggle with accuracy and detecting threats, and sometimes give false alarms. Techniques like SVM and Naïve Bayes might help improve their performance. By enhancing datasets, we can improve the quality of input and make the existing systems work better. These challenges remind us of the ongoing need to improve intrusion detection systems to better defend against cyber threats.

V. PROPOSED SYSTEM

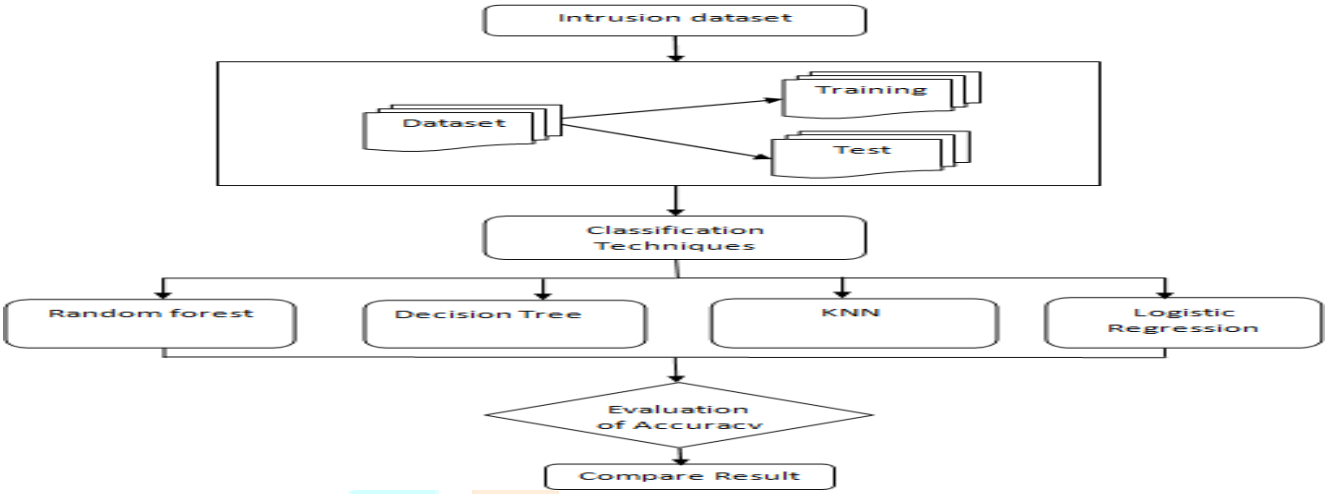
The Intrusion Detection System (IDS) is really important for keeping our systems safe from intruders.. The system is focused on recognizing and mitigating possible hazards, using innovative techniques to improve overall performance. The suggested IDS aims to address and correct flaws identified in prior generations, resulting in a more robust and efficient solution. This system relies on two primary methods: principal component analysis (PCA) and the Decision Tree algorithm. Principal Component Analysis (PCA): It is an important technique for reducing dataset dimensions. PCA greatly improves dataset quality by removing extraneous features and increasing overall accuracy. This reduction not only streamlines computational processes but also assures that the dataset contains just the most relevant information. In intrusion detection, PCA helps make analyses more efficient and accurate.

Decision Tree Algorithm: The proposed IDS is built around the decision tree algorithm. This machine-learning approach is useful for assessing and classifying data based on learned decision rules. Decision trees are especially well-suited for intrusion detection because they can successfully identify patterns suggestive of hostile activity. The algorithm's capacity to generate a tree-like model of decisions improves the accuracy of recognizing possible threats while reducing false positives.

This IDS project is distinguished by the inclusion of the Decision Tree method, which takes advantage of machine learning capabilities to provide more nuanced and adaptable intrusion detection. The decision tree, being a transparent model, allows for the interpretation of decision rules, which aids in understanding the reasoning behind classifications.

Our proposed system has several benefits. Firstly, it has a very low error rate, only 0.21%. Secondly, it's more accurate than previous algorithms. Lastly, it takes less time to perform compared to other algorithms. Decision trees have low computational complexity, making them both efficient and scalable. Decision trees can manage missing values in a dataset. In this work we used INTEL I5 Processor with 512 GB SSD, MYSQL, To run the project well, use Windows 7 and Python. Pick the right database, like SQLite, MySQL, or PostgreSQL, based on project needs. This ensures smooth operation and efficient data management.

VI. SYSTEM ARCHITECTURE



VII. MODULE

Data Collection: Collecting data stands as the inaugural and fundamental step in the tangible development of a machine learning mode. This step is really important because it can make a big difference in how well the model works overall. The quality and quantity of data acquired directly correlate with the model's effectiveness, with more and better data contributing to improved performance. There are various techniques available for data collection, such as web scraping, manual interventions, and others. The dataset used in this Intrusion Detection System dataset taken from kdd as shown in figure 2 which consist of four dos,

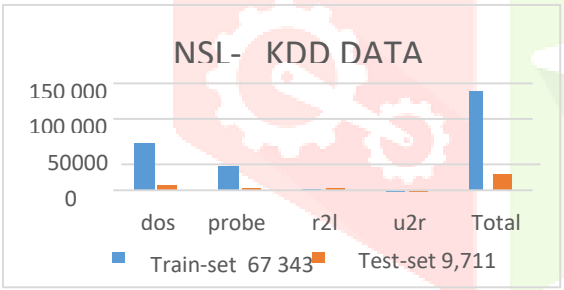


Fig. 2.kdd data set

Dataset: The dataset contains information about 125,974 individual data points. It has 42 different categories, each described in Table 2. The table includes details about the classes, both in the training and testing sets.

Table 1. NSL-KDD Datasets.

KDD Dataset	Classes	Train-sets	Test-sets
NSL-KDD Data set	normal	67343	9,711
	dos	45927	7 458
	probe	11656	2 421
	r2l	995	2 754
	u2r	52	200
	Total	125 973	22 544

Data Preparation: We will initiate the data transformation process by addressing missing data and removing specific columns. Firstly, we'll create a list of column names to retain. Following this, we will eliminate or drop all columns, preserving only those specified in the list. Lastly, we'll discard rows with missing values

from the dataset. The dataset will be split into training and evaluation sets, while keeping the internal processes of the testing phase concealed.During testing, inputs are generated, and outputs are evaluated without detailed insight into the software's internal mechanisms

TYPES OF TESTS

Integration testing: Integration tests assess whether integrated software components operate cohesively as a unified entity. Event-driven testing emphasizes the crucial outcomes of the software system's interfaces or screens. Whereas successful unit testing confirms individual component functionality, integration tests validate the accuracy and consistency of component integration. Integration testing seeks to identify potential problems that may develop when different system components are combined or integrated.

Functional test: Functional tests provide systematic validation that the functionalities under evaluation are accessible as per the bus's specifications. They methodically demonstrate that the tested functions meet the criteria outlined in the system documentation, user manuals, business and technical requirements, without directly borrowing words or phrases from the original text.

Functional testing focuses on the following elements:

Valid Input: Ensuring acceptance of valid input types during the examination process.

Input: It is essential to reject identified classes of invalid input.

Functions: Ensure that the designated functionalities are utilized to their full extent.

Output: Utilize acknowledged application output categories to verify their functionality. It's crucial to engage interacting systems or processes when dealing with systems or procedures.

System Test: System testing is a critical phase that validates whether the entire integrated software system aligns with the specified requirements. This comprehensive examination assesses a setup to guarantee anticipated and consistent outcomes. An instance of a system test is the configuration-oriented system integration test, which relies on process flows and descriptions, emphasizing integration points and predetermined process connections.

White box testing: White Box Testing involves a method where the tester can access the internal components, architecture, and source code of the program under test. This approach allows for a comprehensive assessment of areas that might be difficult to reach with black box testing alone. Figure 3 visually represents this concept.

Fig: 3 white box testing

Black Box Testing: Black box testing involves evaluating software without prior knowledge of its internal architecture, programming language, or workings. This technique relies on official source documents such as requirements or specifications. It treats the program as a "black box," focusing solely on its external behavior, inputs, and outputs, while disregarding internal mechanisms.

VIII. IMPLEMENTATION:

The ID S home page, made using HTML, CSS, and JavaScript, has a secure "Login Page" for user access. A simple navigation bar directs users to key sections: "Home" is the main page, "Abstract" provides a quick overview, and "Upload Data" lets users add information. "Prediction" offers useful analysis tools, "Future" outlines upcoming plans, and "Chart" visually shows data. "Performance Analysis" evaluates system efficiency. The user-friendly login page enhances security, ensuring a straightforward experience for intrusion detection, data management, prediction, and performance evaluation in a concise interface. Refer to fig 4 for visual representation.

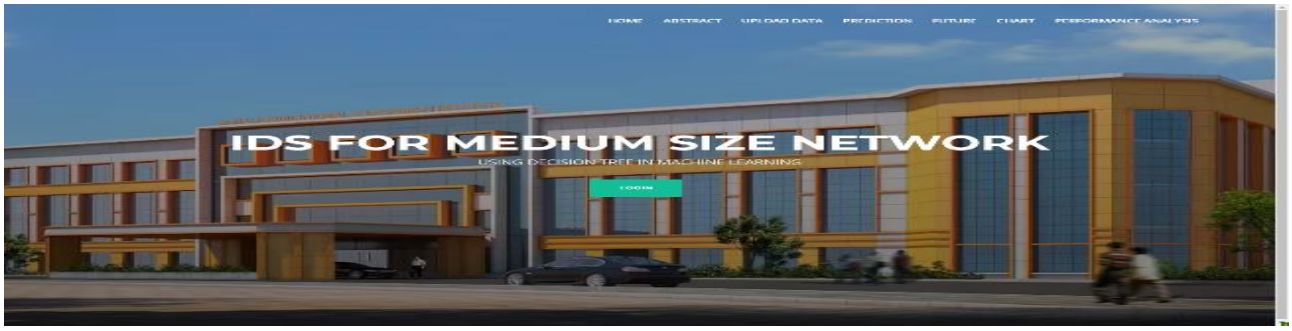


Fig: 4 home page

UPLOAD DATASET: We utilized dataset upload to input the KDD dataset, enabling training for intrusion detection systems, enhancing our ability to identify and mitigate potential security breaches effectively. Refer to fig 5 for visual representation

Fig : 5 upload dataset

KDD DATASET: The KDD dataset, sourced from kdd cup99 **,** encompasses 125974 data types. Its comprehensive nature facilitates thorough analysis crucial for intrusion detection systems (IDS) implementation and refinement. Refer to fig 6 for visual representation

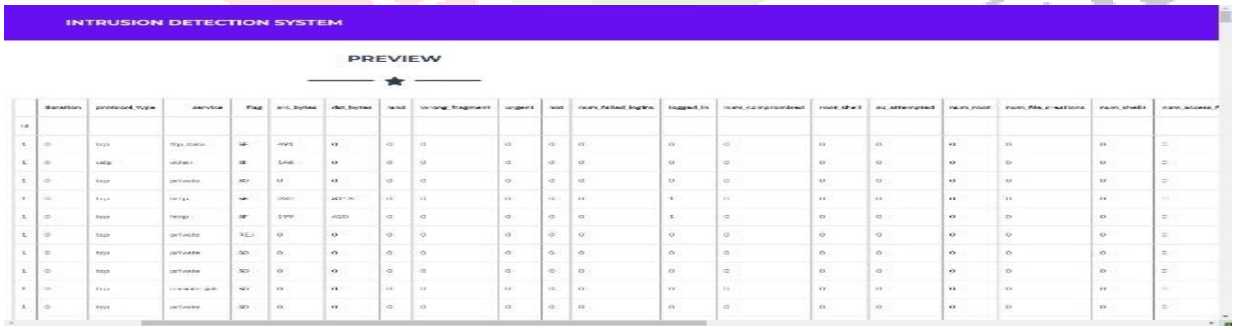


Fig:6 kdd dataset

INTRUSION DETECTOR: I've developed a form for intrusion detection. Users select the test type they wish to run, then check the status. The output categorizes as normal, probe, or R2L, aiding IDS operations. Refer to fig 7 for visual representation

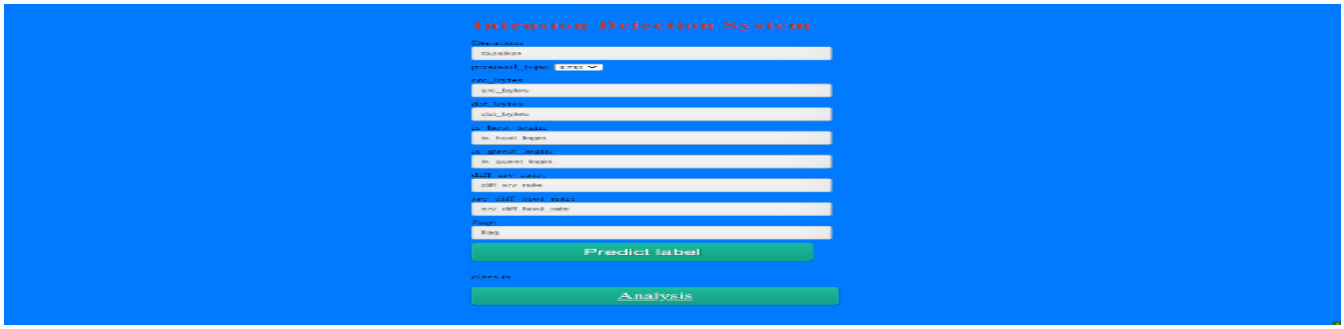


Fig: 7 intrusion detector

PIE CHART ANALYSIS: Following the status check, a pie chart displays attack analysis, indicating the prevalence of different attack types. This visual representation assists in identifying the most and least common attacks for further analysis and mitigation strategies. Refer to fig 8 for visual representation.

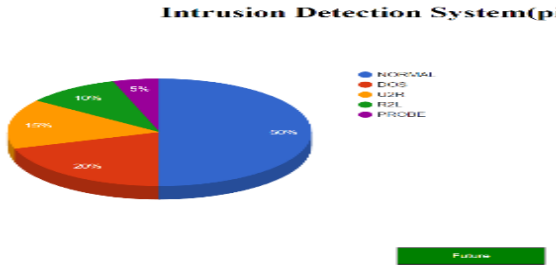


Fig:8 pie chart analysis

PERFORMANCE ANALYSIS: In this process, after the pie chart, performance analysis involves assessing recall and precision metrics. The confusion matrix is then generated to evaluate the detection of normal and attacker statuses, enhancing IDS effectiveness .Refer to fig 9 for visual representation

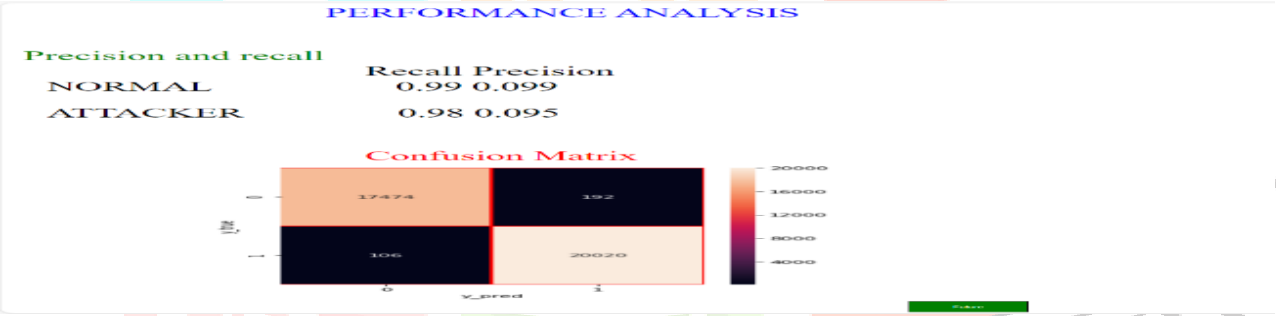


Fig: 9 performance analysis

IX. RESULT AND DISCUSSION

The project aimed to develop an Intrusion Detection System (IDS) using the KDD dataset. Initially, data was uploaded from kdd cup, containing 125,974 data types. A user interface was created, allowing users to select the type of intrusion test and check its status. Upon completion, the system generated outputs categorized as normal, probe, or R2L attacks. Following this, a pie chart displayed attack analysis, depicting the distribution of different attack types for further scrutiny. Performance analysis ensued, examining recall and precision metrics to assess the system's effectiveness. A confusion matrix was then generated, providing an overview of detection accuracy, false positives, false negatives, and other relevant metrics. Results showed promising performance, with high precision and recall rates across various attack types. The confusion matrix revealed robust detection capabilities, with minimal false positives and negatives. These findings suggest the IDS is effective in identifying and mitigating potential security breaches. However, further refinement and optimization are recommended to enhance system efficiency and accuracy. Overall, the project demonstrates the feasibility and efficacy of utilizing the KDD dataset for developing IDS solutions, contributing to enhanced cybersecurity measures.

X. CONCLUSION

As internet-connected systems become more prevalent, concerns about security also rise. Our innovative technique outperforms established algorithms such as SVM, Naïve Bayes, and Decision Tree in detecting online intrusions. It substantially enhances detection rates while minimizing false positives. We assessed our method using the Knowledge Discovery dataset, yielding remarkable results: 96.78% accuracy with an error rate as low as 0.21% achieved in just 3.24 minutes.

REFERENCE

[1] JafarAbo Nada; Mohammad Rasmi Al-Mosa, 2018 International Arab Conference on Information Technology (ACIT), A Proposed Wireless Intrusion Detection Prevention and Attack System

[2] Kinam Park; Youngrok Song; Yun-Gyung Cheong, 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (Big Data Service), Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning AlgorithmS. Bernard, L. Heutte and S. Adam "On the Selection of Decision Trees in Random Forests" Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia, USA, June 14-19, 2009, 978Alfonso P.: Blockchain and IoT Integration: A Systematic Survey. Sensors, 1424-8220 (2018).

[3] Le, T.-T.-H., Kang, H., & Kim, H. (2019). The Impact of PCA-Scale Improving GRU Performance for Intrusion Detection. 2019 International Conference on Platform Technology and Service (PlatCon). Doi:10.1109/platcon.2019.8668960

[4] Anish Halimaa A, Dr K.Sundarakantham: Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) 978-1-5386-9439-8/19/$31.00 ©2019 IEEE "MACHINE LEARNING BASED INTRUSION DETECTION SYSTEMJyoti D. and Amarsinh V.: Security Attacksin IoT: A Survey (2017).

[5] Rohit Kumar Singh Gautam, Er. Amit Doegar; 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence) "An Ensemble Approach for Intrusion Detection System Using Machine LearningAlgorithms."

[6] Kazi Abu Taher, Billal Mohammed Yasin Jisan, Md. Mahbubur Rahma, 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)"Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection."

[7] L. Haripriya, M.A. Jabbar, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)" Role of Machine Learning in Intrusion Detection System: Review"

[8] Mohammed Ishaque, Ladislav Hudec, 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) "Feature extraction using Deep Learning for Intrusion Detection System."

[9] Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar, Rashmi Bhattad, 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)"A Review of Machine Learning Methodologies for Network Intrusion Detection."

[10] Iftikhar Ahmad, Mohammad Basheri, Muhammad Javed Iqbal, Aneel Rahim, IEEE Access (Volume: 6) Page(s): 33789 – 33795 "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection."