IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 5, May 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

Network Intrusion Detection for IoT Security based on Stacking Ensemble: Survey

Abdullahi Hassan Adam[1]

M Tech, CSE, Sharda University, Greater Noida UP, India.[1]

ABSTRACT: An intrusion detection system (IDS) is one of the most common ways in a network architecture for ensuring the integrity and availability of key assets in protected systems. Despite the use of numerous supervised and unsupervised machine learning approaches to increase the efficacy of IDSs, current intrusion detection algorithms still struggle to achieve high performance. 1. In high-dimensional datasets, the classification process of an IDS is hampered by a considerable volume of redundant and irrelevant data. 2. it's possible that a single classifier won't be able to recognize all forms of assaults. 3. Many models are built to deal with old data, making them less susceptible to new threats.As a result, we provide a new intrusion detection framework based on the stacking ensemble technique in this study. We employed five different classifiers as a basis in the first stage, including k-Nearest Neighbor (kNN), Gaussian Naive Bayes (GNB), Random Forest (RF), Logistic Regression (LR), and Multi-Layer Perceptron Classifier (MLPC), and then used Decision Tree as a Meta classifier (DT). Finally, using the UNSW-NB15 dataset as a test, we arrive at a final conclusion for the stacking ensemble approach.

KEYWORDS: Intrusion Detection System, Internet of Things, Stacking Ensemble, Machine Learning, UNSW-NB15 dataset, Performance Evaluation, Security

I.INTRODUCTION

Previously, the Internet was only accessible via computers, mobile phones, and tablets. Many types of gadgets and appliances (e.g., televisions, air conditioners, and washing machines) may now be connected to the Internet thanks to the Internet of Things (IoT). Healthcare, agriculture, traffic monitoring, energy conservation, water supply, and unmanned air vehicles are just a few of the industries that might benefit from unmanned air vehicles, and automobiles are all examples of where the Internet of Things is being used today [1].

As (IoT) technological advancements, new threats arise every day, raising a slew of security problems. These hazards must first be identified before they can be prevented. As a consequence, intrusion detection is a top priority. Intrusion detection, on the other hand, heavily relies on datasets. The data collection that was used to test machine learning algorithms [2].

The use of classification algorithms to a network intrusion detection system (NIDs) is a key decision-making challenge. In the domain of machine learning, several methods have been used, including fuzzy logic, neural networks, support vector machines, Naive Bayes, K closest neighbour, and decision trees (NIDs) [3].

To limit the risk of security breaches, security experts implement a variety of preventative and detection approaches. Application of complicated configurations and the establishment of a robust security policy are examples of prevention strategies that try to make it more difficult to carry out such attacks. All security policies should adhere to the Central Intelligence Agency's trinity of principles: Confidentiality, honesty, and accessibility are all important considerations. [4].

This study provides a way for constructing an IDS that uses Machine Learning (ML) techniques to identify data threats in order to protect against cyber-attacks in the IoT. A dataset comprising normal and attack cases is required to design an intrusion detection system (IDS) utilising machine learning.

The study is divided into two sections: the first addresses the introduction, and the second explores related work in this field. The design of the IoT test bed, as well as the setup of adversarial systems to create attacks and collect data, are

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

discussed in Section 3. The performance of machine learning algorithms in categorising data is discussed in Section 4. The results of the investigation are discussed in Section 5.

II.BACKGROUND

2.1 Intrusion Detection Systems

An IDS is a harmful activity that attempts to disrupt a network's security policy by compromising the confidentiality, integrity, and availability of network components [4].

2.2 Classification of Intrusion Detection System:

2.2.1 Network Based (Network IDS): Network based intrusion detection looks for illegal, criminal, or unexpected behavior based only on network traffic. A network intrusion detection system (IDS) collects packets that pass over a network via a network tap, bridge port, or hub. Based on the collected data, the IDS system evaluates and flags any suspicious traffic. An intrusion detection system differs from an intrusion prevention system in that an intrusion detection system does not actively limit network traffic. The role of a network IDS is passive, consisting solely of data gathering, identification, logging, and alerting.

2.2.2 Host Based (HIDS):HIDS, or host-based intrusion detection, is a technique for detecting illegal, criminal, or unexpected behaviour on a computer or other device. In most cases, HIDS necessitates the installation of an agent on each machine that monitors and alerts on local OS and application activities. To detect illegal conduct, the installed agent employs a combination of signatures, rules, and heuristics. A host IDS's job is largely passive, consisting solely of data collection, identification, logging, and alerting.

2.3 Detection Types

2.3.1 Signature-based Intrusion Detection System: This system is based on the matching concept. The data is evaluated and compared to known attack signatures. If there is a match, an alert is sent out. This approach has the benefit of being more accurate and having standard alerts that the user can understand.

2.3.2 Anomaly-based Intrusion Detection System: It is made up of a statistical model of typical network traffic, which includes the bandwidth used, the protocols established for the traffic, the ports, and the network devices. It monitors network traffic on a regular basis and compares it to the statistical model. The administrator is notified if there is any abnormality or inconsistency. This method has the benefit of being able to identify new and unique threats.

2.4 Functioning Types

2.4.1 Passive Intrusion Detection System: It simply identifies the type of malware activity and notifies the system or network administrator. (This is what we've seen so far!) The administrator then takes the necessary steps.

2.4.2 Reactive Intrusion Detection System: Not only does it identify the risk, but it also takes precise action, such as resetting the suspect connection or blocking network traffic from the suspicious source. Intrusion Prevention System is another name for it.

2.5 Classification Methods

The base classifiers selected for the ensemble stacking are k-Nearest-Neighbor(kNN), Naïve-Bayes(NB), Random-Forest(RF), Logistic-Regression(LR), Multi-Layer-Perceptron-Classifier (MLPC) and Decision-Tree(DT) using as Meta classifier. These base classifiers are briefly discussed below.

2.5.1K-Nearest Neighbor (kNN)

The k-nearest neighbour algorithm, often known as KNN or k-NN, is a supervised learning classifier that uses proximity to make classifications or predictions about how an individual data point should be categorised. It may be used to solve regression or classification problems, but it is most typically used as a classification approach since it is based on the assumption that similar points can be found near together.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

2.5.2 Naive Bayes Classifier (NBC)

The Nave-Bayes approach, which is based on Bayes' theorem, is a supervised learning strategy for dealing with classification problems. It's mostly used for text classification using a big training set. The Nave Bayes Classifier is a basic and effective classification approach that assists in the creation of fast machine learning models that can make correct predictions.

P(A|B) = \frac{P(B|A) P(A)}{P(B)}

2.5.3Random Forest (RF)

The supervised learning technique Random Forest is well-known. Both classification and regression problems may be solved using (ML). It's based on ensemble learning, a technique for merging several classifiers to solve a complicated issue and increase the model's performance.

A Random-Forest, as the name implies, is "a classifier that consists of a number of decision trees on distinct subsets of a given dataset that are averaged to increase the dataset's predicted accuracy." Rather of depending on a single decision tree, the random forest gathers predictions from several trees and predicts the ultimate outcome based on the majority of votes.

2.5.4Logistic Regression

The term "logistic regression" refers to a supervised learning classification method for estimating the probability of a target variable. Only two valid classes exist due to the dichotomous character of the target or dependent variable.

2.5.5Multi-Layer Perceptron Classifier (MLPC)

Artificial-neural-networks(ANN) with a multi-layer perceptron design are the most complicated. Multiple layers of the perceptron make up the majority of it. TensorFlow is a well-known deep learning framework, and this notebook will show you how to use it to construct a neural network.

2.5.6 Stacking Ensemble of Classifiers

Stacking is a method for combining multiple classification models using a meta-classifier. Individual classification models are trained using the entire training set, and the ensemble's outputs are then used to fit the meta-classifier (meta-features). The meta-classifier may be trained using either the expected class labels or the ensemble probabilities.

Algorithm 19.7 Stacking

Input: Training data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{m}$ ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathcal{Y}$)
Output: An ensemble classifier H

1: Step 1: Learn first-level classifiers
2: **for** $t \leftarrow 1$ to T **do**
3: Learn a base classifier h_t based on $\mathcal{D}$
4: **end for**
5: Step 2: Construct new data sets from $\mathcal{D}$
6: **for** $i \leftarrow 1$ to m **do**
7: Construct a new data set that contains $\{\mathbf{x}'_i, y_i\}$, where $\mathbf{x}'_i = \{h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \ldots, h_T(\mathbf{x}_i)\}$
8: **end for**
9: Step 3: Learn a second-level classifier
10: Learn a new classifier h' based on the newly constructed data set
11: **return** $H(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_T(\mathbf{x}))$

Fig1.Stacking Ensemble Algorithm

2.4 Performance Evaluation Measures

A number of performance assessment approaches are used to evaluate the accuracy and efficiency of the Stacking Classifier, classifiers, and comparison. The performance evaluation approaches are as follows: True-Positive(TP), False-Positive(FP), True-Negative(TN), False-Negative(FN), Area-under-Curve(AUC), Root-Mean-Square-Error(RMSE), and Precision (PR).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

$$\text{True Positive(TP)} = \frac{TP}{\text{TP} + \text{FN}}$$

$$\text{True Negative(TN)} = \frac{TN}{\text{TN} + \text{FP}}$$

$$\text{False Positive(FP)} = \frac{FP}{\text{FP} + \text{TN}}$$

$$\text{False Negative(FN)} = \frac{FN}{\text{FN} + \text{TP}}$$

$$\text{Precision} = \frac{TP}{\text{TP} + \text{FP}}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(P_i - T_i)^2}$$

$$\text{Accuracy} = \frac{TP + TN}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

III. RELATED WORKS

According to the findings, the underlying dataset has to be updated in order to detect new IDS attacks with improved performance. Because bad gays employ a variety of procedures and technologies to carry out their attacks, this is the case. Furthermore, the pattern of launching several assaults replicates the necessity for datasets containing actual network circumstances. The properties of these datasets are reviewed in this work, as well as some of their flaws. In the future, we want to investigate the performance of these datasets using various ML and DM approaches, as well as feature engineering and data sampling, to overcome their flaws [5].

To deal with the complexity of the new smart system, smart procedures must be developed. A deep-neural-network(DNN) for (IDS)in IoT networks was reported in our paper. Intruded patterns were classified to detect intruders. We trained our network with three datasets and utilized DNN to test it. It was demonstrated that it was capable of successfully detecting attacking behavior. As a consequence, we achieved at least 90% accuracy and greater with each dataset. [6].

IoT networks are extremely sensitive to hacking, hence solutions to safeguard these devices and networks must be developed and tested. An IoT-based platform was constructed for this project, and it acted as a test bed for understanding and executing IoT network assaults. Data was collected from the established network in order to use a machine learning method to detect network threats. The data was classified as regular and modified malicious attack data using four machine learning techniques [7].

The algorithms were able to accurately categorize the data. According to the findings, machine learning techniques may be utilized to create IDS for IoT networks. The most difficult aspect of constructing an IDS based on machine learning principles is generating a realistic and high-quality training dataset; data flow in the network should be of good quality during the attack phase since interception is only achievable with a continuous flow of data. Because the network will be used by a variety of heterogeneous devices, the ML model will take into account a wide range of data. A viable IDS for the IoT context may be constructed by addressing these problems. To guarantee that IoT devices are safe from cyber-threats, security elements must be included early in the development process[8].

Based on this perspective, we may identify a number of challenges that limit the applicability of current techniques for distinct IDS types. This enabled us to identify the IDS techniques that look to be promising for IoT. Furthermore, we highlighted two research approaches that have the potential to address the shortcomings of IDSs when applied to IoT

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

networks. Overall, we received the idea that the bulk of existing IDSs aren't quite up to the task of dealing with the IoT's resource constraints, but that progress is being made. We see a lot of promise for sufficient solutions that will properly secure the IoT and its users after putting in some effort into research and development[9].

We offered a comparison of several models based on Long-Short Term Memory through the study of intrusion detection systems and neural networks (LSTM). As dimensionality reduction techniques, we employed Principal Component Analysis and Mutual Information. We then examined the performance and time processing of these approaches. During the training stage, the LSTM model may effectively learn the features collected from the dataset. This capacity allows the models to successfully discriminate between normal traffic and network threats[10].

IV. PROPOSED SYSTEM

The goal of the suggested strategy is to use an ensemble technique termed stacking to get trustworthy forecasts.

4.1 Dataset

UNSW-NB15 is a brand-new dataset that was released in 2015. It covers recent attacks (nine vs. 14 in the KDD'99 dataset). It contains 49 characteristics and a wide range of normal and attacked activities, as well as 25, 40,044 records with class labels. The total number of records consists of 2, 21,876 normal and 3, 21,283 attacked records. Basic-Features, Flow-Features, Time-Features, Content-Features, Additional-Generated-Features, and Labelled-Features are the six categories of features included in the UNSW-NB15 data collection. General Purpose Features are features with a number between 36 and 40. Connection features are those that start with 41 and end with 47.Analysis, Fuzzers, Backdoors, DoS Exploits, Reconnaissance, Generic, Shellcode, and Worms are among the nine types of attacks in the UNSW-NB15 dataset.

4.2 Proposed Flow Diagram

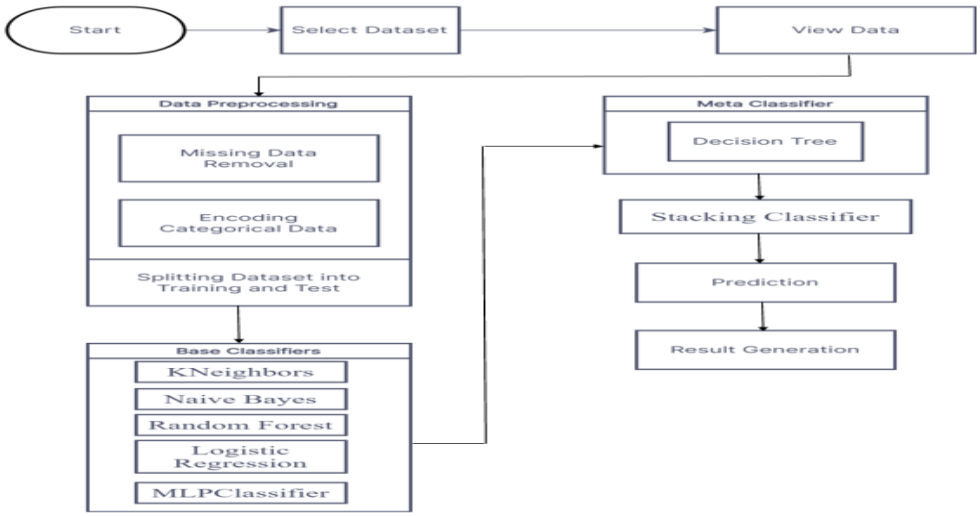

Fig2.Flow Diagram

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

4.3 Performance of Features

Algorithm	Accuracy	Execution Time(s)
KNeighbors	89.9009%	0.0250
Naive Bayes	84.9301%	0.1032
Random Forest	92.2118%	4.7459
Logistic Regression	88.1626%	1.3344
MLPClassifier	89.3945%	67.5907
Ensemble Stacking	96.5439%	314.2427

Table 1. Performance of Features

4.4 Evaluation Metrics for Stacking

Parameter	ensemble stacking
True Positive (%)	92.5130
False Positive (%)	7.4869
True Negative (%)	98.4195
False Negative (%)	1.5804
Precision (%)	0.9659
Recall (%)	0.9839
F1 score (%)	0.9748
Accuracy (%)	96.5439
RMSE (%)	0.1859
Roc_Curve (%)	0.955

Table 2. Evaluation Metrics

V. CONCLUSION

Based on the notion of stacking, this research provides an ensemble approach for effective network intrusion detection. The UNSW NB-15 was employed as a test subject. A mixture of algorithms, including K-nearest-neighbor, Nave-Bayes, Random-Forest, Logistic-Regression, and Multi-layer-perceptron-classifier, produced higher predictions on a real-time dataset. Experimentation on multiple datasets, including current attack categories, may be done using the implementation technique. Advanced computing engines, like as Apache Spark, and may be used in the future to increase processing speed and scalability for massive volumes of network data. Based on the results of a series of tests conducted throughout the course of this research, the suggested approach may be considered a competitive viewpoint for real-time network intrusion detection.To establish traffic periodicity and long-term evolution of network traffic, only classic packet-based intrusion detection datasets may be employed.

REFERENCES

1. Tomer, V.; Sharma, S.Detecting IoT Attacks Using an Ensemble Machine Learning Model. Future Internet 2022, 14, 102. https:// doi.org/10.3390/fi14040102
2. Hassan Adegbola Afolabi1, Abdurazzag Aburas2., Comparison of Single and Ensemble Intrusion Detection Techniques using Multiple Datasets. 2021, https://doi.org/10.30534/ijatcse/2021/161042021

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005305 |

3. Rajagopal, S., Kundapur, P. P., & Hareesha, K. S. (2020). A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets. Security and Communication Networks, 2020, 1–9. doi:10.1155/2020/4586875
4. Mahfouz, A., Abuhussein, A., Venugopal, D., & Shiva, S. (2020). Ensemble Classifiers for Network Intrusion Detection Using a Novel Network Attack Dataset. Future Internet, 2020 doi:10.3390/fi12110180.
5. Thakkar, A., & Lohiya, R. (2020). A Review of the Advancement in Intrusion Detection Datasets. Procedia Computer Science, 167, 636–645. doi:10.1016/j.procs.2020.03.330
6. Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. Procedia Computer Science, 167, 1561–1573. doi:10.1016/j.procs.2020.03.367
7. Sai Kiran, K. V. V. N. L., Devisetty, R. N. K., Kalyan, N. P., Mukundini, K., & Karthi, R. (2020). Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques. Procedia Computer Science, 171, 2372–2379. doi:10.1016/j.procs.2020.04.257
8. Khan, Z. A., & Herrmann, P. (2019). Recent Advancements in Intrusion Detection Systems for the Internet of Things. Security and Communication Networks, 2019, 1–19. doi:10.1155/2019/4301409
9. Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 8(1). doi:10.1186/s40537-021-00448-4
10. Ryan Mills, Angelos K. Marnerides, Matthew Broadbent, Nicholas Race. (2021). Practical Intrusion Detection of Emerging Threats.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details