A Comprehensive Approach for Detection of Intrusion in Computer Network

Sonam Jayaswal[1], Gholam Mursalin Ansari[2], Upendra Kumar[3]
[1]*Scholar, YBN University, Ranchi-834002*
[2]*Assoc. Professor, YBN University, Ranchi-834002*
[3]*Asist. Professor, BIT Mesra Patna*

Abstract- Now a day's Network Intrusion is the universal issue for users. One of the strategies view to prevent intrusion is to develop an Intrusion Detection System (IDS). Moreover, attackers frequently modify their tools and methodologies to change their attacking patterns, therefore putting a recognized IDS system in one place is generally difficult. Recently network environments getting more complex and having more hosts are becoming vulnerable to attack. The goal of this research is to analyze existing models which is the best suitable model to detect intrusion in minimum time. Although it's crucial to talk about methodical, effective, and automated intrusion detection models. In this research work, we explore machine learning strategies for Intrusion Detection on the UNSW-NB15 dataset. Although we are analyzing and implementing several model to find better output. The goal is to increase the intrusion detection rate by focusing on false positive and false negative performance indicators. The results of this research indicates that the Random Forest classifier will the best average accuracy rate Although the XG Boost classifier will the lowest value for false negatives. Further we also analyze time complexity of the various machine learning models.

Keywords: IDS, SVM. Random Forest, KNN, Bagging, Boosting, Network Intrusion

1.INTRODUCTION

As we know that world is based on computer network system without that no any communication is possible. To know this importance of communication hackers are also in middle. Therefore intrusion in network system is a very big challenge. The various reaction teams estimate that 2020-2024, computers were attacked into more frequently than once every two seconds. To improve businesses ability detection and tracking intrusions are required. The means of detecting and tracking intrusions is called "intrusion detection". As businesses migrate more of their crucial business transactions to the Internet, Intrusion Detection Systems (IDS), which have been the focus of theoretical research and development, are growing popularity. By identifying an intruder's actions, an intrusion detection system can alert users in advance of attacks or intrusion attempts. Intrusion Detection Systems are effective tools in the enterprise's struggle to protect its computing resources in this area.

One of the best sources of information for the modern world is the Internet, which has evolved into an invaluable tool for commercial and educational purposes. Data transmission over the Internet must therefore be safe. One of the primary issues in the modern day is Internet security. It is crucial to have a system that safeguards both the users and the data because the Internet is constantly under assault. As a result, the Intrusion Detection System (IDS) is a creation that perfectly satisfies this need. To stop malicious attempts, network administrators adjust the Intrusion Detection System. In light of this, intrusion detection systems have emerged as crucial components of security management. All infiltration attempts and network abuse are found and reported by an intrusion detection system. IDS can perform a thorough security analysis, detect and stop malicious assaults on the network, maintain normal functioning throughout a malicious outbreak.

Considering how quickly information technology has advanced over the last two decades. Computer networks are widely employed in commerce, industry, and many other facets of daily life. As a result, creating dependable networks is a crucial duty for IT administrators. On the other hand, the quick advancement of information technology has made it exceedingly challenging to create dependable networks. The availability, integrity, and

confidentiality of computer networks are threatened by a variety of intimidation. Denial of Service (DOS) attack is considered as one of the most common harmful attacks. Machine learning techniques have advanced quickly during the past ten years, allowing for new levels of automation and prediction. This is inspiring scientists and engineers to create fresh uses for these amazing methods. Machine learning techniques were quickly applied to bolster network security systems. Intrusion, including brute force attacks, denial-of-service attacks, or even infiltration from within the network, is the most frequent threat to network security. Given the shifting patterns of network activities, a dynamic method is required to identify and stop such intrusions. The widespread consensus in this field is that static data sets do not adequately reflect traffic composition and incursions. To detect and resist sophisticated attackers that may easily evade simple intrusion detection systems; we need a changeable, replicable, and expandable dataset for (IDS). Machine learning is useful in this situation. We will analyze many machine learning methods that can be applied to create reliable IDS.

2. BACKGROUND

All harmful network traffic and computer activity that a conventional firewall is unable to identify is detected by an intrusion detection system. These comprise malware, data-driven assaults on apps, host-based attacks including privilege escalation and unauthorized logins and access to sensitive information, as well as network attacks on susceptible services (viruses, Trojans and worms). [3.4,5] The purpose of intrusion detection systems (IDS) is to shield the business from the effects of the aforementioned situation. They keep an eye on network traffic for questionable activities and sound an alarm if anything goes wrong[9,10]. It is a piece of software that searches a network or system for harmful activities or rules that have been broken. Any unlawful behavior or malicious intent is often either recorded centrally using a security information and event management system or reported to an administrator (SIEM) system. A SIEM system integrates output from multiple sources and uses alert filtering techniques to distinguish malicious activity from false alerts [7,8]. While monitoring networks for potentially harmful behavior, intrusion detection systems are also susceptible to false alarms. Therefore, at initial installation, companies must fine-tune their IDS products. This entails configuring the intrusion detection systems to distinguish between legitimate network traffic and malicious activities [18,19].

Network packets entering the system are also watched by intrusion prevention systems to look for harmful behavior and promptly transmit notifications. Both IDS and firewalls are concerned with network security, however a firewall varies from an IDS in that it searches the outside for attackers in order to stop them. In order to prevent infiltration, firewalls impose access restrictions between networks

If an attack originates from within the network, it is not disclosed. A IDS describes a suspected intrusion once it has occurred, and then reports an alarm.

In this wonderful world, only those who know and trust will be able to access your network. We want our network to be accessible to suppliers and clients. Additionally, we want remote users to have access to the network during odd hours. But what is about the guy who is bored and alone in his bedroom, pumped full of sugar and coffee, and has no better use for his time than to experiment with our network is required. An intrusion detection system can help in this situation. You can find and stop hackers in conjunction with an intrusion prevention system before they get close to crucial data on your network. This is how they operate. In fact, networks have been a blessing for a very long time, bringing individuals and the world closer together. The risk of intrusion also materialized with networks. The idea of intrusion detection was created in response to intrusion. All incoming and outgoing network activity is monitored by an intrusion detection system (IDS)[14,15], which looks for any indications of intrusion that can jeopardize your systems. It is also known as a passive monitoring system because its primary purpose is to sound an alarm when it notices such activity. The capacity to both detect anomalies and stop them from happening on a network of a company makes an intrusion prevention system (IPS) a step beyond an IDS.

Organizations can profit from intrusion detection systems in a number of ways, starting with their capacity to identify security issues. You can analyse the volume and variety of attacks using IDS.

Organizations can make changes to their security systems or put in place more efficient controls using this information. Organizations can use an intrusion detection system to find faults or issues with the configuration of their network devices. Then, future risks can be evaluated using these metrics[10-12]. Additionally, intrusion detection systems can assist an organization in complying with regulations. By giving businesses improved network visibility, an IDS makes it simpler to abide by security rules. Additionally, businesses can utilize their IDS logs as part of their documentation to demonstrate that they comply with legal obligations.

Security measures can also be enhanced by intrusion detection systems. IDS sensors can find network hosts and devices, therefore they can also be used to look at data in network packets and figure out what operating systems the services being utilized are using. It may be considerably more effective to use an IDS to acquire this data than to manually count linked systems.

The IDS can either:

- be strategically positioned in the network as an NIDS (Network-based Intrusion Detection), which makes use of hardware sensors located at key points in the organization's network, or be installed in the network as a software-based IDS.
- or deployed as Host-based Intrusion Detection HIDS on each system or placed on network-connected system to analyses incoming and outgoing data.

IDS can play a crucial role in an organization's security, but they are just one component of a well-rounded, safe system. Understanding how to use IDS and the alternatives most effectively requires comparison. An IDS does not block or prevent attacks, it only helps detect theme but must therefore be a part of a comprehensive strategy which are also includes other security measures and personnel who are trained to react correctly.

Table 1: Intrusion detection in computer network

References	Year	Authours	Gist of research work
1	2021	G.S. Senthilvelan, P. Thangamani, and V. S. M. P. Senthil Kumar	Comprehensive study of NIDS datasets.
2	2022	Abbas, Khan, Latif, Ajaz, Shah & Ahmad	Ensemble model for IoT IDS using CICIDS2017 dataset with voting classifier; having max accuracy of Ensemble hard voting 88.92%.
6	2023	Andrew, Deepika & Chandran	Projected a CNN-based model for intrusion detection over MQTT protocol; high performance across flow types having Max F1 Score 0.99.
10	2020	Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar	Foreword of TON_IoT dataset; ML model assessment and having max RF accuracy 97%.
11	2015	B. Ingre, A. Yadav	Recital analysis of NSL-KDD dataset using Artificial Neural Networks (ANN).
12	2024	Nandhini, Rajeswari, Shanker	Cyber attack detection in IoT WSN devices using threat intelligence and has authenticated.
13	2024	Lagraa et al.	Graph-based methods for network security and botnet detection.
14	2023	Vipin Kumar, Vivek Kumar, N. Singh, Raj Kumar	Improving intrusion detection on edge computing systems has seen.
15	2024	Kumar, P.; Kumar, A.A.; Sahayakingsly, C.	Improving intrusion detection on edge computing systems.
16	2021	Mishra, B.; Smirnova, I.	Optimal configuration of intrusion detection systems for better performance.
17	2021	Md Al Imran, Shamim H. Ripon	detection using deep learning and ML models.
18	2020	Chauhan, M.; Joon, A.; Agrawal, A.; Kaushal, S.; Kumari, R.	Literature review of machine learning-based intrusion detection systems.

19	2020	Zuo, X., Chen, Z., Dong, L., Chang, J., Hou, B.	IDS in power information networks using data mining algorithms.

3. METHODOLOGY

There are numerous methods to classify intrusion detection systems due to the various ways in which IDS companies implement their products. The first is based on the breadth of monitoring offered by IDS, or whether it is a network-based product that monitors traffic across the network and also analyses data from individual computers or whether it is installed on a single host computer and uses data from that computer. The way the vendor offers the system—as a software product or as an integrated hardware device—is another distinction in implementation (appliance).

In order to find intrusions, a network-based IDS analyses data from host computers as well as data from network traffic. An "immoral" network adapter is often used by a network-based IDS to analyses data packets sent over the network (capable of reading all packets sent over the network, not just those addressed to it). The host-based IDS typically overlooks the packet headers, which are examined by the network-based IDS. This makes it possible to identify assaults like Denial of Service (DoS) and others that a host-based IDS would miss. Different techniques can be used by IDS systems to find alleged intruders. The two most popular categories are statistical anomaly detection and pattern matching. Known assaults are identified through pattern matching based on their "signatures," or the particular actions they employ. This is sometimes referred to as abuse detection or signature-based IDS. IDS searches for activity and actions that coincide with known attack patterns. The signature database, which needs to be updated, determines effectiveness. Pattern matching is comparable to identifying a culprit by his fingerprint being found at the scene of a certain crime. Pattern matching techniques are used in fingerprint analysis. Pattern matching's major drawback is its inability to identify new threats for which the software has a specified signature in its database.

Anomaly-based detection keeps an eye out for changes in typical usage patterns. To do this, a baseline profile must be created in order to ascertain the norm, and activities that deviate from those norms must then be watched for. This enables you to identify fresh incursions or attacks for which a signature is not yet known.

A police officer patrolling a particular beat every day and being familiar with what is "typical" there is comparable to anomaly detection. Even if he is unsure of the specific crime being perpetrated or the perpetrators, he has a reasonable suspicion that it is taking place if he notices something that deviates from the norm.

There are several different methods for detecting anomalies, including:

- metric model
- neural network
- machine learning classification

One problem with IDS, which is based on anomalies, is the higher number The increased rate of false positives with IDS, which is based on anomalies, is an issue since unexpected behavior is taken into account as a potential attack, even if it is not.

An intrusion detection system's objective is to give a warning of a prospective or real attack. A vulnerability is a risk that raises the possibility of an attack or intrusion, whereas an attack or incursion is a transient event. The distinction between an attack and a vulnerability is that an attack takes place at a certain time, whereas a vulnerability always exists. Another way to think of an attack is as an attempt to take advantage of a weakness (or in some cases, a perceived vulnerability). This leads us to categories different types of intrusion detection systems.

Network traffic is monitored by intrusion detection systems to identify unauthorized attackers conducting attacks. IDS offers security experts some or all of the following features:

- providing a method for administrators to reconcile, organize, and comprehend pertinent OS audit trails and other logs that are otherwise challenging to track or analyses.

- monitoring the operation of routers, firewalls, key management servers, and files required by other security controls to detect, prevent, or remediate cyberattacks.
- having a comprehensive database of attack signatures against which data from the system can be cross-referenced; offering a user-friendly interface so that non-expert staff can assist in managing system security.
- the identification and notification of data file modification detection by IDS.
- create a notification and alert that security has been compromised.
- blocking intruders or securing the server in response to them.

Benefits of intrusion detection systems

A IDS is quite valuable for network monitoring, but how useful it is totally depending on what you do with the data you receive. Without the proper staff and policies in place to manage them and respond to threats, detection technologies are useless as an additional layer of security since they cannot block or fix possible issues. Intruders can employ encrypted packets to enter the network because an IDS cannot see them. Your systems are exposed until the attacker is discovered since an IDS does not register these intruders until they have advanced farther into the network. This is a significant issue as encryption usage increases to protect personal data. A IDS can read the contents of an IP packet, but the network address can still be spoof. It is more challenging to identify and evaluate the threat when an attacker utilizes a faked address.

You being alerted to false positives on a frequent basis is a significant issue with IDS. False positives are frequently more common than genuine dangers in many situations. Despite the fact that IDS can be configured to produce fewer false positives, your technicians will still need to spend time responding to them.

You are susceptible to assaults based on logs. The same log-based attacks that target network hosts are also targeted against NIDS since they analyses logs as they are being gathered. Both incorrect data and log analyzer problems have the potential to crash an NIDS.

3.1 DATASET

- The IXIA Perfect Storm programmed was used in the Cyber Range Lab at UNSW Canberra to construct a mixture of real modern normal activities and synthetic contemporary attack behaviors for the raw network packets of the UNSW-NB 15 dataset. We used the tcp dump programmed to record 100 GB of unprocessed traffic (e.g., Pcap files). These nine attack types include fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms, and they are all included in this dataset. To create a total of 49 features with the class label, 12 algorithms are created and the Argus and Bro-IDS tools are utilized. The UNSW-NB15 features.csv file contains a description of these features.
  - The four CSV files UNSW-NB15 1.csv, UNSW-NB15 2.csv, UNSW-NB15 3.csv, and UNSW-NB15 4.csv contain a total of two million and 540,044 records.
  - The list of events file is UNSW-NB15_ LIST _EVENTS.csv, and the ground truth table is UNSW-NB15_ GT.csv.
  - This dataset's one partition, UNSW NB15 training-set.csv and UNSW NB15 testing-set.csv, has been set up as a training set and a testing set, respectively. The testing set has 82,332 records of various sorts, including Attack and Normal, while the training set has 175,341 records total.

Table2: Attack names and number of Instances for UNSW NB15 data set

Attack Name	Number of Instances
Generic	18460
Normal	9625
Exploits	5293
DOS	717
Fuzzers	535
Reconnaissance	504
Worms	34
Backdoor	11

- Original Dataset: UNSW NB15.csv

- The file UNSW NB15 features.csv contains 49 features with a class label. In the UNSW-NB15 freatures.csv file, these properties are described.

Over the past three decades, Network Intrusion Detection Systems (NIDS), particularly Anomaly Detection Systems (ADS), have taken a greater role in identifying new threats than Signature Detection Systems (SDS). Due to three key issues, the evaluation of NIDS using the KDD99 and NSLKDD benchmark datasets does not produce satisfactory findings. The absence of network packet inspection to differentiate between typical and aberrant findings In this study, we present three examples of the UNSW-NB15 dataset's complexity. First, an explanation of observation and attribute statistical analysis is provided. The investigation into feature correlations is given second. Third, the accuracy and false alarm rates of five current classifiers are utilized to assess the complexity (FARs). Following that, the outcomes are contrasted with the KDD99 dataset. The experimental findings demonstrate that UNSW-NB15 is a new benchmark dataset for NIDS evaluation and is more sophisticated than KDD99 contemporary low-impact assault types, the absence of contemporary examples of normal traffic a different split between the test and training sets.

The UNSW-NB15 dataset was recently developed in order to overcome these problems. This dataset includes nine different contemporary assault types as well as fresh traffic patterns. It has 49 properties, including ones related to data transfer between hosts.

3.2 MACHINE LEARNING ALGORITHMS OVERVIEW:

A quick summary of the various machine learning techniques demonstrates the necessity to apply them in various contexts, such as intrusion detection. Machine learning algorithms are needed more and more often as a result of the ongoing development of technologies in order to analyses a huge number of datasets and derive knowledge from them.

Decision Tree Classifier:

The main goal of this classifier is to build a lookup table that aids in identifying the output's expected output class. There are a number of lookup methods that can be employed to boost the decision table's effectiveness, including first-width search, evolutionary algorithms, and cross-validation. The predicted actions are connected to the predefined conditions in the lookup table, which contains a collection of conditions. In other words, a set of significant rules that aid in the prediction of fresh incoming inputs is the result of the decision table classifier. The decision table lookup can also be utilized in other contexts, such as when a fuzzy system is complicated and requires specialized knowledge, to display the key rules.

CART Algorithm:

CART (Classification and Regression Trees) can be used to solve both classification and regression issues, as the name suggests. The target variable is different in classification because we are attempting to predict a class label. In other words, classification is employed for issues where the output (target variable), such as whether or not it will rain tomorrow, can only take one of a limited number of values. Regression is used to forecast a numerical label, on the other hand. This implies that your output can have an endless range of values, such as the price of a house. The supervised subset of machine learning techniques applies to both situations. Decision trees can be used for both classification and regression applications. We'll try to comprehend the decision tree algorithm's fundamentals. The CART algorithm is then used to create the decision tree from the training dataset. A non-parametric supervised learning method is the decision tree. It is a tree containing numerous decision rules that were all derived from the features of the data. The Gini index serves as a purity or impurity indicator for creating a decision tree using the CART (Classification and Regression Tree) technique. An attribute with a low Gini index is preferable to one with a high Gini index. The CART approach only produces binary splits, and it does so by employing the Gini index.

The Gini index can be calculated using the formula shown below:

$$\text{Gini Index} = 1 - \sum_j P_j^2$$

KNN Classifier:

The K nearest neighbors (KNN) method is an easy-to-use approach that uses the entire dataset for training.

The entire training dataset is searched for the k-most comparable examples whenever a prediction is required for an unknown data instance, and the data with the closest instance is finally returned as the prediction.KNN is widely used in search applications, such as discover items similar to this one, where similar items are desired.

Random Forest Classifier:

With this method, forest-based tree classifiers are improved. It is one of the methods for classification trees. The cited research produced random forest classifiers with a recognized accuracy rate that can be utilized to manage dataset noise levels. There are no modifications made during the categorization stage. Since each tree in a forest forecasts the anticipated outcome, the number of trees in the forest must be calculated in order to apply this method. Then, using a voting mechanism, the estimated outcome with the largest number of votes is determined.

Overview of Ensemble Learning

Ensemble learning combines conclusions from various models to enhance performance as a whole.

- Stacking in Advanced Ensemble Techniques
- Techniques employed in the ensemble include bagging and boosting.

A technique for ensemble learning called stacking-stacking combines predictions from many models, such as decision trees, KNN, and svms, to produce a new model. On the test set, predictions are made using this model. Here is a step-by-step breakdown of a straightforward stacked ensemble: There are ten sections in the training set as deficted in fig2.

Fig. 2 for training and testing

Predictions are produced for the 10th part after fitting a base model (such as a decision tree) to the first nine parts. Each piece of the train set receives this treatment.

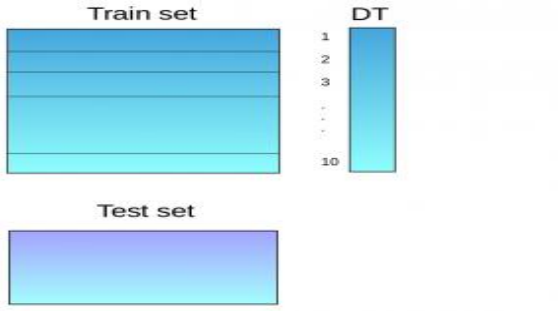

Fig.3 for training and testing for decision tree

The complete training data set is then run through the underlying model (in this example, the decision tree). Predictions for the test data set are based on this model.

Fig.4 for training and testing for decision tree

For a different base model (such as knn), steps 2 through 4 are repeated, producing a new set of predictions for the training and test data set.

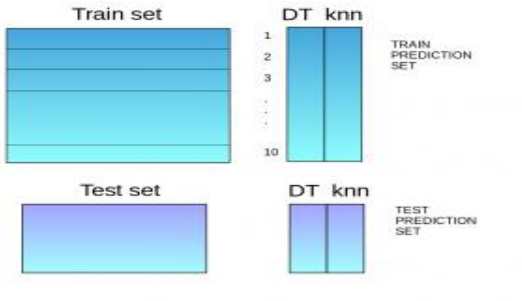

Fig.5 for training and testing for decision tree and KNN

In order to create a new model, the predictions from the training set are used as features.

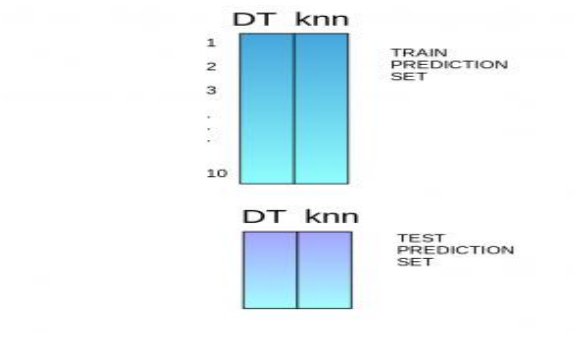

Fig.6 for training and testing for decision tree and KNN

Bagging-

The goal of bagging is to integrate the output from various models (such as all decision trees) to provide a more comprehensive result. Does it make sense to mix all the models that were created using the same data set? Given the same input, there is a high likelihood that these models will produce the same outcome. How therefore can we resolve this issue? The method of bootstrapping is one among them. Using the sampling approach known as "bootstrapping," we extract subsets of observations from the original dataset. The subsets' size is the same as the original dataset's size. These subsets (bags) are utilized in the bagging approach (or bootstrap aggregation) to acquire a clear picture of the distribution. (complete set). The size of the subsets created for bagging can be smaller than the original set.

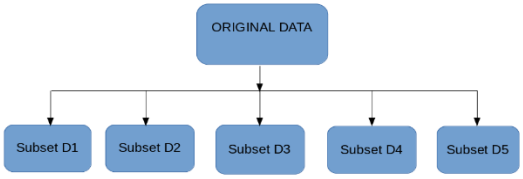

Fig.7 for training and testing for Bagging

- From the original dataset, several subsets are made, and observations are chosen through substitution.
- For each of these subgroups, a basic model (weak model) is built.
- The models operate independently of one another in parallel.
- The combined projections from all models yield the final predictions.

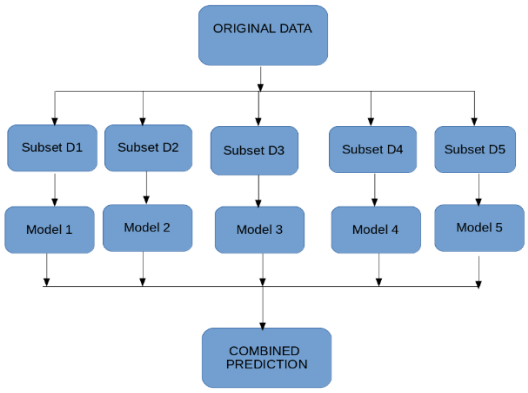

Fig.8 for training and testing for Bagging

Boosting-

Before continuing, we have one more query for you: can merging the predictions produce better outcomes when a data point is forecasted erroneously by the first model, the next, and likely by all models? Boosting is used to resolve such circumstances.

During the sequential process of "boosting," each new model tries to fix the flaws in the prior model. The preceding model is necessary for subsequent models. In the steps that follow, let's examine how boosting functions.

1. The original data set is divided into a subset.
2. All data points are initially weighted equally.
3. Based on this subset, a foundation model is developed.
4. Using this model, predictions are made for the complete dataset.
5. Using the actual values and the anticipated values, the errors are determined.
6. A greater weight is placed on the observations that were mistakenly predicted. (In this case, the three blue-plus points that were incorrectly assigned larger weights.)
7. Predictions are produced for the data set using a different model. (This model makes an effort to fix the flaws in the prior model)

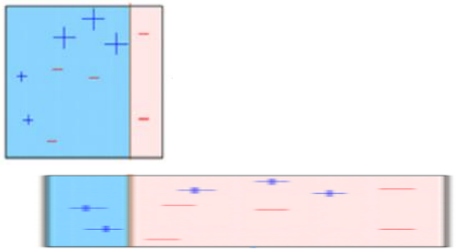

Fig.9 for training and testing for Bagging

In a similar vein, numerous models are developed, each one fixing the flaws of the prior model. The weighted average of all the models is used to create the final model (strong learner) (weak learner).

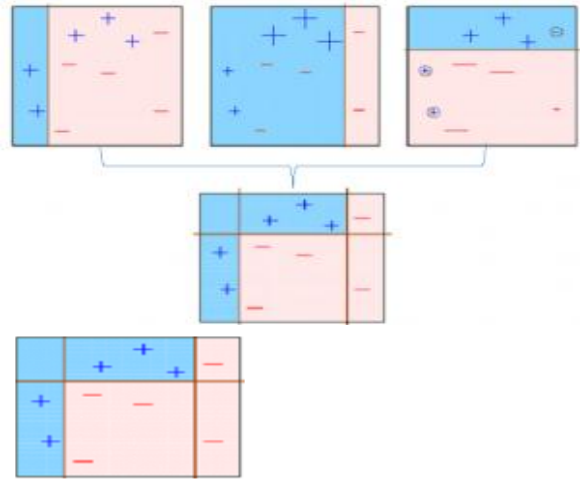

Fig.10 for training and testing for Boosting

As a result Fig.1, Fig2, Fig.3 Fig.4, Fig.5. Fig.6 Fig.7 Fig.8, Fig.9 and Fig.10 are depicted for correspond classification of different models. The boosting algorithm creates one strong learning model from several weak learning models. Although they oper Fig.1ate well for a chunk of the dataset, the individual models do not perform well on the complete dataset. Each model improves the performance of the ensemble as a result.

Ada Boost (Adaptive Boosting):
One of the simplest boosting algorithms is adaptive boosting, often known as AdaBoost. Decision trees are frequently employed in modelling. Each successive model is built to fix the flaws in the previous model. AdaBoost gives weights to the poorly predicted observations, and the ensuing model tries to forecast these values accurately.

XG Boosting:
A sophisticated use of the gradient boosting algorithm is called XG Boost (extreme Gradient Boosting). XG Boost has established itself as a very powerful machine learning algorithm and is frequently utilized in hackathons and contests for this field. In comparison to other gradient boosting methods, XG Boost is nearly ten times faster and has a strong predictive potential. Additionally, it contains a range of regularizations that lessen overfitting and enhance performance in general. It is often referred to as the "regularized boosting" technique.

PCA (Principal Component Analysis):
Consider that you have a large basket of fruits and that you want to arrange and organise them based on their commonalities. Although it would take a lot of time and work, you might accomplish this by carefully examining each fruit individually. Instead, you might utilise a programme called PCA (Principal Component Analysis) to assist you in quickly and effectively grouping the fruits.PCA works like a magic tool to help you discover trends and connections among a lot of data. The various fruit varieties in the basket serve as the data in this instance. PCA examines all the many characteristics of the fruits, including their size, shape, colour, sweetness, and texture, and seeks to identify the key characteristics that distinguish them from one another. After identifying these critical features, PCA generates a new collection of more compact, simplified features that retain the majority of the fruit-related data. Principal components are the name for these novel features. Consider them to be the fruits' essential qualities condensed into a handful of distinguishing characteristics. You may now classify the fruits based on their similarity using these key elements. You might discover, for instance, that some fruits are comparable because they are small, round, and red, while others are comparable because they are big, sweet, and green. Based on these similarities, you can then categorise these fruits into distinct categories. Because it can be used to identify patterns and similarities in any type of data, not just fruit-related data, PCA is a potent machine learning technique. It can be used to classify people according to their traits, look for patterns in financial data, or even analyse pictures and videos. When analysing massive data sets, PCA can help you save a tonne of time and effort and provide accurate results.

Table3: Comparative result

S.No.	Algorithm	Accuracy	Time(ms)
1	CART	97.79	0.0019
2	Decision Tree	97.83	0.003
3	KNN	98.05	6.42
4	Random Forest	98.49	0.131
5	Ada Boost	98.30	0.48
6	XG Boost	97.21	0.016

We have successfully reached the highest accuracy by Random Forest that is 98.49%, Accuracy after selecting 5 features by applying feature extraction using PCA as shown in table 3.

In Table 4 CART algorithm has taken minimum time to implement the process but max accuracy of Random forest.

Table4: Comparative results with Accuracy with time:

S.No.	Algorithm	Accuracy	Time(ms)
1	CART	97.89	0.002
2	Decision Tree	97.87	0.008
3	KNN	97.89	0.012
4	Random Forest	98.41	0.343
5	Ada Boost	98.07	0.39
6	XG Boost	97.47	0.01

CONCLUSION

Several experiments were performed and tested to evaluate the efficiency and the performance of the following machine learning classifiers: Decision Tree, CART Algorithm, Random Forest, KNN Classifier, AdaBoost and XG Boost. All the tests were based on the UNSW-NB15 intrusion detection dataset. After selection of more features than 5 even doesn't increases the accuracy and also decreasing the features less than 5 doesn't increases the accuracy From all the above mention models, we find the highest accuracy with Random Forest. We have compared CART algorithm with other research works done and we achieved the highest accuracy among all. We also find the time complexity of each algorithm and the least time to predict output is done by CART algorithm.

REFERENCE

[1] G.S. Senthilvelan, P. Thangamani, and V. S. M. P. Senthil Kumar A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System**"** published in the *Asian Journal of Research in Computer Science* (Volume 7, Issue 4, pages 4–33**,** 202.

[2] Adeel Abbas, Muazzam A. Khan, Shahid Latif, Maria Ajaz, Awais Aziz Shah, and Jawad Ahma., *A New Ensemble-Based Intrusion Detection System for Internet of Things Arabian Journal for Science and Engineering*, volume 47, pages 1805–1819, 2022.

[3] Tamara Zhukabayeva, Zulfiqar Ahmad, Aigul Adamova, Nurdaulet Karabayev, Assel Abdildayeva., An Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection to Enhance Cyber–Physical System Security in Industrial IoT**.**, International Journal of Sensors.,2025.

[4] Yasir Hamid, M. Sugumaran & V. Balasaraswathi., *IDS Using Machine Learning – Current State of Art and Future Directions., British Journal of Applied Science & Technology*, Volume 15, No. 3, Pages 122-135, 2016.

[5] Sharanya Chandran & K. Senthil Kumar *A Survey of Intrusion Detection Techniques*, *International Journal of Engineering & Technology*, Vol. 7-2.4, pp. 187–189., 2018.

[6] G. Andrew**,** M. P. Deepika**, and** S. Chandran., An MQTT IoT Intrusion Detection System Using Deep-Learning**,** published in the *Proceedings of the Third International Conference on Computing and Communication Networks (ICCCN 2023).*

[7] A. Nasr, M. Ezz, M. Abdulmageed*., Use of Decision Trees and Attributional Rules in Incremental Learning of an Intrusion Detection Model., International Journal of Computer Networks and Communications Security.,* Vol. 2, No. 7, pp. 216–224.,2014.

[8] S. Vijayarani, Maria Sylviaa S., *Intrusion Detection System – A Study., International Journal of Security Privacy and Trust Management.,* Vol. 4, No. 1, pp. 31–44, 2015.

[9] Anush Ananthakumar; Tanmay Ganediwal; Dr. Ashwini Kunte., *Intrusion Detection System in Wireless Sensor Networks: A Review International Journal of Advanced Computer Science and Applications (IJACSA).,* Vol. 6, Issue 12,2015.

[10] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar., TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems." IEEE Access**,** Volume 8, pages 165130–165150, 2020.

[11] Bhupendra Ingre, Anamika Yadav *Performance analysis of NSL-KDD dataset using ANN.,* International Conference on Signal Processing

and Communication Engineering Systems (SPACES), pages, 92–96, 2015

[12] S. Nandhini, A. Rajeswari, N. R. Shanker., *Cyber attack detection in IOT-WSN devices with threat intelligence using hidden and connected layer based architectures*., Journal of Cloud Computing: Advances, Systems and Applications,. Volume 13, Article 159,2024.

[13] Sofiane Lagraa, Martin Husák, Hamida Seba, Satyanarayana Vuppala, Radu State, Moussa Ouedraogo: *A review on graph-based approaches for network security monitoring and botnet detection*. Int. J. Inf. Sec. 23(1):pp119–140, 2024.

[14] Vipin Kumar, Vivek Kumar, N. Singh, and Raj Kumar: *Enhancing Intrusion Detection System Performance to Detect Attacks on Edge of Things,* SN Computer Science (SN COMPUT. SCI.), Volume 4, Article 802, 2023.

[15] Dr. P. Karunakar Reddy, T. Nalini Devi, Shaik Asiya, R. Swathi, Edge computing security: advanced feature selection adopted supervised learning models for real-time intrusion detection, International Journal of Communication Networks and Information Security (IJCNIS) *Year*:2024.

[16] Birendra Mishra & Inna Smirnova., *Optimal configuration of intrusion detection systems.,* Information Technology and Management., Volume 22, Issue 4, pages 231–244., 2021.

[17] Md Al-Imran & Shamim H. Ripon., *Network Intrusion Detection: An Analytical Assessment Using Deep Learning and State-of-the-art Machine Learning Models, International Journal of Computational Intelligence Systems*, Volume 14, Article Number 200.,2021.

[18] Mayank Chauhan, Ankush Joon, Akshat Agrawal, Shivangi Kaushal, Rajani Kumari., *Intrusion Detection System for Securing Computer Networks Using Machine Learning: A Literature Review., Advances in Intelligent Systems and Computing* volume 1334, part of the *Congress on Intelligent Systems (CIS 2020)* series.pp. 177–189,2020.

[19] Xiaojun Zuo, Ze Chen, Limian Dong, Jie Chang, and Botao Hou., *Power information network intrusion detection based on data mining algorithm., The Journal of Supercomputing.,* Volume 76, Issue 7, pages 5521–5539., 2020.