Intrusion Detection using Machine Learning

Sireesha Moturi[1], Divya Chintalapudi[2], Keerthana Tammuluri[3], Lahari Mattapalli[4]
[1] Professor, [2, 3 & 4] Students
[1]sireeshamoturi@gmail.com, [2] divyach090406@gmail.com, [3] tammulurikeerthana@gmail.com, [4] mattapallilahari@gmail.com
Department of Computer Science and Engineering,
Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

ABSTRACT - In the rapidly evolving digital landscape, maintaining the security of computer networks and systems has become essential. Conventional methods often struggle to remain relevant in light of the increasing complexity and diversity of cyber threats. The frameworks of cybersecurity and network analysis are necessary to detect and respond to malicious activity, unauthorized access, and potential threats within a network or system. Its ability to handle complex, unbalanced datasets and the exceptional performance of the few algorithms we used across several domains make it a serious contender for improving intrusion detection accuracy. Many in-depth experiments are conducted with the NSLKDD dataset to evaluate the performance of different methods. In our investigation, three algorithms XgBoost, CatBoost, and KNN achieved the accuracy score of 99%.

KEYWORDS - Intrusion Detection, Min Max & Standard Scalar Normalization, RandomForestClassifier, Support Vector Machine, Cat Boost, KNN, XGBoost, QDA, Naïve Bayes, Accuracy Prediction.

I. INTRODUCTION

The application of networked digital items, like CPUs and sensors, is known as the Internet of Things (IoT) [1]. Organizations can operate reliably and efficiently by leveraging cutting-edge digital technology like machine effectiveness and security. Because installed IoT endpoints are frequently outfitted with artificial intelligence (AI) capabilities to conduct and automate operational operations efficiently, installed IoT endpoints gather real-time data to assist organizations in making decisions. While deploying IoT can empower enterprises, putting strong preventive safeguards into effect is an important study area [2]. IoT ecosystems provide new challenges because they combine the digital and physical worlds. Mistakes can have severe repercussions, such as lost profits and business interruption. The majority of information system's security architecture depends heavily on intrusion detection systems (IDS). Finding harmful activity or policy infractions within a computer network or system is their main duty. Traditional intrusion detection techniques, which sometimes rely on preset criteria or signatures, find it challenging to stay up to date with the rapidly evolving world of threats from the internet. This is where machine learning comes into play, offering a dynamic and adaptable way to enhance intrusion detection abilities.

Botnet attacks can be a threat to Internet of Things (IoT) networks and devices with low resources [3]. An attack progressively increases congestion in networks, battery life, and CPU and memory utilization of IoT devices as it spreads throughout the victim network. It is therefore critical to detect hacked Internet of Things (IoT) gadgets and locate malevolent network behaviour when a botnet attack is underway. While identifying compromised IoT devices opens the door for preventive actions against the spread of malware and Botnet attacks, detecting illegal traffic permits proactive actions to lessen the impact of the attack and stop that.

But because the majority of IoT connections roughly 52% of them are to low-cost, low-maintenance devices that are a part of expansive IoT networks, it can be challenging to develop and implement complex security measures [4 In the early phases of research, a number of small artificial intelligence (ML)-based systems for detection (IDS) for Web of Things, or IoT, networks were developed in order to achieve this [5]. This study showed how machine learning-based system for intrusion detection (IDS) anomaly detection holds great potential in detecting zero-day assaults, which usually target vulnerable networks and devices and are based on unexpected invasions..

Because anomaly-based intrusion detection systems (IDS) primarily rely on the features of typical traffic used for parameter optimising it is more challenging to draw valid conclusions when network traffic's typical conduct shifts over time due to both internal and external factors. Like, the overall volume of traffic on the Internet of Things network may change dramatically if one or more new devices are added. Therefore, anomaly-based intrusion detection systems (IDS) would greatly benefit from real-time adaptation to time-varying network traffic characteristics, which is best accomplished via sequence learning via the internet [6, 7]. However, the difficulty of online learning to gather and classify adequate data, in addition to the requirement to choose appropriate times for parameter updates, can occasionally restrict its efficacy.

Predicting assaults using the dataset's information is the main goal of this work. the XG Boost, which is Cat Boost, KNN, SVM, naive Bayes, and the QDA are some of the methods we use to extract various predictions from the dataset. These techniques are useful for handling complex datasets like NSL-KDD. The Web of Things, also known as IoT, has been a disruptive force in the last few years, revolutionizing several industries with its seamless automation and connectivity. Every aspect of contemporary society has been influenced by IoT, which provides unparalleled simplicity and efficiency, from industrial sensors and autonomous cars to gadgets and intelligent residences to industrial sensors and driverless cars. But with all of these linked devices, a new wave of cybersecurity problems has also emerged. IoT networks are

particularly vulnerable because to their intrinsic heterogeneity and distributed nature, which makes them attractive to hackers and other bad actors looking to take advantage of security flaws for illicit gain. IoT ecosystems are vulnerable to many different threats, for which traditional safeguards like firewalls and detection systems for intrusions are usually insufficient to fully mitigate the harm. Therefore, innovative approaches to strengthen IoT device security posture and defend against cyber threats are desperately needed.

In today's connected world, protecting digital systems and networks is vital because they are the foundation of vital infrastructure. An important part of security measures is intrusion detection, which aims to identify and thwart hostile activity or unauthorized access to a network. Traditionally, intrusion detection systems (IDS) have used rule-based or signature-based detection techniques to identify known threats; however, as cyber threats evolve, so does the need for more flexible and dependable intrusion detection systems.

The application of Machine Learning (ML) has shown promise in improving intrusion detection capabilities. Through the utilization of Machine Learning algorithms, which possess the ability to examine large datasets and identify patterns that conventional rule-based systems would miss, more potent and productive malware detection could be created. Comparing ML-based intrusion detection technologies to conventional techniques, they may be able to detect attacks that were not previously known, detect evolving threats, and minimize false positives.

High-performing Machine Learning models designed especially for IoT security applications were chosen through this iterative procedure. The application of Machine Learning (ML) has shown promise in improving malware detection capabilities. Through the utilization of Machine Learning algorithms, which possess the ability to examine large datasets and identify patterns that conventional rule-based systems would miss, more potent and productive systems for detecting malware can be created. Comparing ML-based intrusion detection systems to conventional techniques, they may be able to detect attacks that were not previously known, detect evolving threats, and minimize false positives.

By having the thorough analysis of various algorithms to determine which models work best for IoT intrusion detection. we optimised model parameters, ensemble configurations, and feature selection strategies based on empirical research and experimentation to increase detection accuracy and reduce false positives. High-performing Machine Learning models designed especially for IoT security applications were chosen through this iterative procedure.

II. RELATED WORK

IDS (Intrusion Detection System) is crucial for detecting unauthorized access or anomalies in network traffic and system activities.There have been several studies conducted on this topicusing various statistical and Machine Learning techniques. Here are some of the notable literature surveys for IDS for finding different type of attacks:

A. Javaid et al. [8] demonstrated the efficacy of deep neural networks (DNNs) in detecting malicious activity inside network traffic, this study addresses the requirements of Deep Learning methods in the detection of network intrusions. In conclusion, model architectures, training approaches, and deployment techniques are all fast advancing in the field of DL-based NIDS research. An important contribution to this field is the work of Javaid and Niyaz [8], who show how Deep Learning techniques can be used for reliable and efficient network intrusion detection.

KT. Ahmad et al. [9] compares and assesses the efficacy of different Deep Learning models for intrusion detection, offering a comparative analysis that may be useful in choosing the right model for a certain kind of network environment. The objective is to illustrate the advancements made in this sector, evaluate different strategies, and pinpoint the gaps that the present research seeks to fill. Protecting information systems from hostile activity requires intrusion detection, and the development of Machine Learning(ML) has opened up new possibilities for more effectively identifying and countering such threats.

R. Geetha et al. [10] This comprehensive overview study covers a wide range of Machine Learning and Deep Learning techniques for cybersthreats, including how they relate to intrusion detection systems. Research on creating strong the emergence of adversarial Machine Learning, where attackers utilize sophisticated strategies to fool ML models, has led to an increase in DL models that are resistant to these kinds of attacks and safeguard cybersecurity systems.

Kim et al. [11] show how useful features can be extracted automatically from raw data on network traffic by deep learning models. for effective intrusion detection. K. Kim and M.E. Aminanto contributed to this topic; their focus was on using DL algorithms for ids. They most likely focused on building or fine-tuning Deep Learning (DL) models to increase detection rates, decrease false positives, and efficiently evaluate network data in real-time or almost real-time. They might have looked into architectures that were tailored to the nuances of network traffic and attack patterns in order to improve model performance. Additionally, they might have introduced state of the art methods for preparing data or training.

Prior to Deep Learning becoming widely used, signature-driven and asymmetry detection techniques have a hard time detecting zero-day attacks. Zero-day assaults are difficult for signature-based techniques to detect, although they can identify established patterns of malicious behavior. By detecting changes from typical network

behavior, anomaly-based techniques have the ability to detect novel threats; nevertheless, they frequently have significant false positive rates. The foundation for comprehending these methods was established by the research of scientists such as Denning (1987).

These studies demonstrate the wide range of techniques and approaches used to predict the attacks They also highlight the importance of features and performance. The integration of Machine Learning with IDS has significant improvement of accuracy.

A. MOTIVATION

We are entering a new era of connectedness and ease with the introduction of Internet of Things (IOT) devices. accross a range of industries, including healthcare, urban planning, and automation in factories. However, this has also made IoT gadgets vulnerable to hackers seeking to exploit security flaws in networks. The state of detection systems for intrusions (IDS) for Internet of Things systems is currently limited and inefficient. Traditional signature-based solutions often fail to adapt to the dynamic and heterogeneous nature of IoT settings, leaving them vulnerable to sophisticated and ever-evolving cyber threats. Thus, there is an urgent need for innovative solutions that address these fundamental issues and strengthen the security posture of IoT networks.

A creative approach that makes use of large datasets tailored for Internet of Things scenarios and machine learning algorithms is need to develop effective IDS that can consistently detect and neutralize a variety of cyberthreats. By solving these pressing issues, we hope to enhance IoT security and build a more robust and safe digital future. Federated learning is a paradigm shift in machine learning models training that takes place across several decentralized servers or devices that store local data samples and don't exchange them. This strategy is especially beneficial in Internet of Things settings where bandwidth, security, and privacy of data are critical considerations.

Deep learning models have demonstrated considerable potential in detecting abnormalities in huge datasets, especially those based on neural networks such as autoencoders. These models can identify patterns of behaviour by being trained on typical operational data from IoT devices and networks. When these models are implemented, they can efficiently detect departures from the standard, indicating possible security lapses or malevolent actions. By guaranteeing data integrity and traceability, integrating blockchain technology with Internet of Things security procedures can offer another degree of protection. A decentralized ledger allows for the verification and recording of every transaction—in this case, a data packet or command exchanged between devices—making it practically impossible for fraud to occur. These settings enable the creation of synthetic datasets that are realistic in nature and cover a wider range of potential dangers, such as zero-day assaults. IDS models can be trained in these simulated settings, which can significantly increase their capacity to generalize and adjust to threats that are not observed in real-world applications.

III. PROPOSED MODEL

This section discusses the recommended intrusion detection system. Smartphones and sensors produce data, which is then uploaded to the cloud. Because it is susceptible to attack, data stored on the cloud is not secure. The attack can be broadcasted directly onto the internet or fired. This study made use of the NSL-KDD dataset. The most recent forms of attack are included in this recently released dataset. Feature scaling was done in the first stage of this study methodology on the nsl-kdd dataset using the Standard Scalar technique and the min–max idea of normalization in order to prevent data releasing on the testing data. After that, training data from the XG Boost, Cat Boost technique, KNN, SVM, the QDA, and NB classifiers were used to reduce dimensionality using PCA. The Random Forest Classifier (RFC) was utilized to choose the features for the selection process.

Preparation of the data was the initial analysis carried out after the dataset was loaded and obtained. Preparing data is crucial since it lessens the likelihood of outliers and unnecessary characteristics. The data was prepared using the Standard Scalar and regularization process using the Min-max approach. The method of selecting features using principle component analysis (PCA) uses the results of the min-max and Standard Scalar algorithms. Ten significant components out of the dataset's forty-nine attributes were found by the PCA. The XGBoost, CatBoost method, KNN, SVM, QDA, and NB classifiers are then trained on a smaller dataset. The architecture of our suggested model is displayed in Fig. 1.

The current techniques for classifying attacks on NSL-KDD are shown in TABLE.I. These four publications show several approaches to increasing accuracy using the same NSL-KDD dataset. Using our dataset, ensemble learning and neural networks were essential in outperforming algorithms in terms of accuracy. A brief explanation was covered here, along with the methodology, findings, and limitations. We would hypothetically examine each approach that was presented, stressing its approach, results, and limitations based on standard techniques for attack classification using the NSL-KDD dataset. The distinct benefits and challenges that each technique presents progress the discipline of intrusion detection.

Using the NSL-KDD dataset, we would hypothetically analyze each strategy that was given, emphasizing its methodology, outcomes, and constraints based on accepted methods for attack classification. The field of intrusion detection is advanced by the unique advantages and difficulties that each technique offers. Deep learning models have demonstrated considerable potential in detecting abnormalities in huge datasets, especially those based on neural networks such as autoencoders.

TABLE I. Existing Methods.

Existing methods for attacks classification on NSL-KDD

Research Paper	Year	Methodology	Results	Limitations
M.Babichev a.,[12]	2023	Application of ML methods	Accuracy=99%	The algorithm may need to be further refined due to potential issues arising from the intricate and ever-changing nature of network intrusions. as well as possible false
Shashank [13]	2023	Ensemble Learning	99.4 % accuracy	Data scarcity and imbalance
Srinivas [14]	2023	Deep autoencoder (AE)	CNN,RNN,LSTM=99.55,94. 33,99.08%.	Extensible computational resources for training deep DL models.
Kalpna et al., [15]	2023	ML based predictive model	98.65%	Scalability to large networks.

A. Data Preprocessing

The NSL-KDD [30] dataset is a standard dataset that is widely used for expanding intrusion detection systems (IDS), particularly in the context of network safety. This improved dataset solves some of the biases and shortcomings of the kdd cup 1999 original dataset. The 1999 dataset was widely used, but it had a number of issues, such as duplicate records, incorrect background traffic, and an uneven distribution of attack and normal instances. The NSL-KDD dataset was created in order to solve these issues and provide a more equitable and realistic dataset for evaluating intrusion detection systems. Its creation involved processing the initial Knowledge discovery and extraction Cup 1999 dataset to remove redundant and duplicate records, add new attack types, and even out the distribution of attack types.

Nearly 41 columns in the dataset provide important information regarding various network-based assaults on Internet of Things devices. We have now categorized the attack types as protocol, service, and flag using group by functions. Figures 2, 3, and 4 illustrate this, accordingly. Nearly 41 columns in the dataset provide important information regarding various network-based assaults on Internet of Things devices. We have now categorized the attack types as protocol, service, and flag using group by functions. Figures 2, 3, and 4 illustrate this, accordingly. Moreover, Fig. 5 illustrates the classification of the attacks into Dos, R21, Probe, U2r, and Normal.

1. Treating Null and Missing values

Taking care of duplicates and missing values (nulls) is the first stage in data preprocessing. Machine Learning models may perform differently when null values are present in the dataset because they can induce biases.

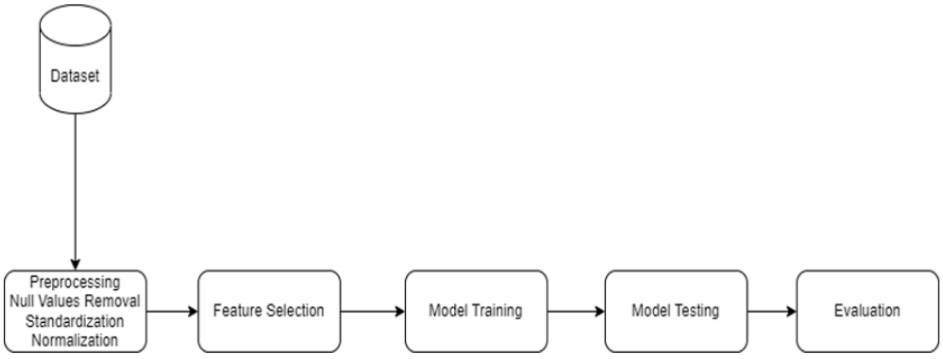

Fig.1. Data preprocessing

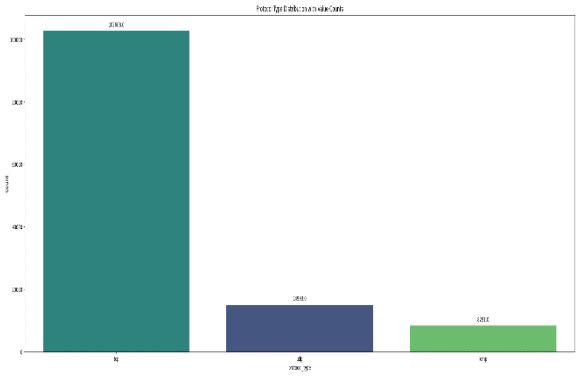

Fig.2. Protocol type distribution attacks.

In order to ensure that the data is presented in a way that highlights the most to least prevalent protocol kinds, we sorted the bars in descending order based on their value counts in Fig. 2. This sorting provides quick insights into how common or popular the different protocol types are in the dataset.

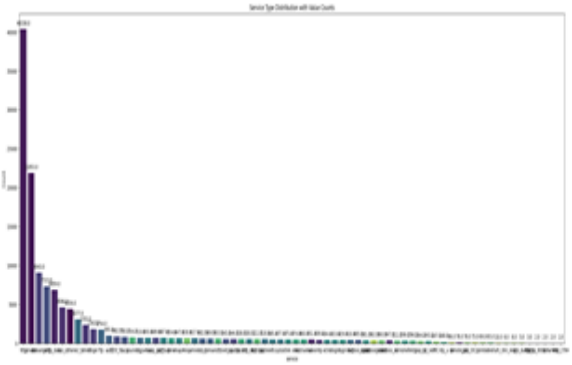

Fig.3. Service type distribution attacks

The services in Fig. 3 were organized in descending order according to their count. Because it is immediately apparent which services are more common in the dataset, this design facilitates more intuitive data interpretation on the part of viewers. C.

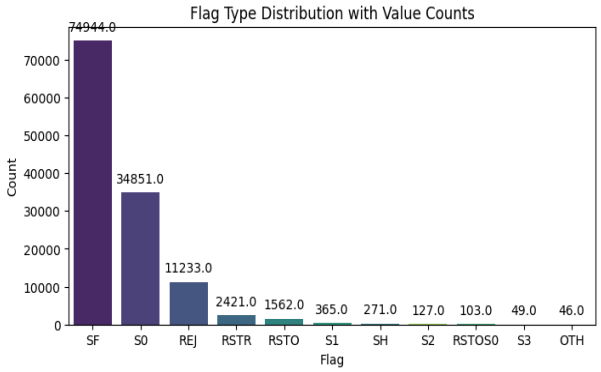

Fig.4. Flag type distribution attacks

Fig. 4 provides a quick, intuitive understanding of the distribution of the "flag" characteristic across our sample. It highlights the frequency or lack thereof of each category, offering crucial background information for comprehending the dataset's composition.

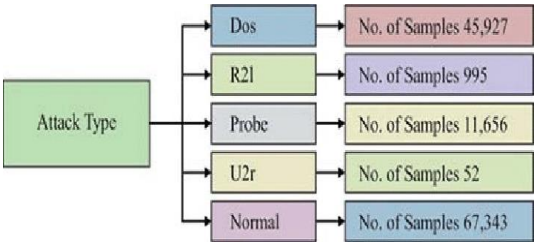

Fig.5. classification of Attack types

The distribution of these attack types in the dataset isn't even, as seen in Fig. 5, where there are noticeably more instances of some attack types—like DoS and Probe, in particular—than of U2R and R2L. Finding and removing null values ensures the data's integrity and completeness. In a similar vein, eliminating duplicate entries is essential to preserving the quality of the data since they may skew the analysis and lead to overfitting.

A set of network connections that are identified as either normal or attacked based on a number of criteria are included in the nsl-kdd dataset. The attacks are additionally separated into 4 categories:
Attacks known as denial of service (DoS) aim to render a computer or network resource unusable for the intended users by flooding it with so many requests. As instances, consider the words "pod," "Smurf," "back," "land," "neptune," and "teardrop."Probe: An attacker falls into this category if they attempt to get more network information through surveillance or other probing measures.

Using the NSL-KDD dataset, we would hypothetically analyze each strategy that was given, emphasizing its methodology, outcomes, and constraints based on accepted methods for attack classification. The field of intrusion detection is advanced by the unique advantages and difficulties that each technique offers. Federated learning can greatly improve the flexibility and efficiency of intrusion detection systems (IDS) in a variety of settings, allowing for the detection of new threats and lowering the risks to user privacy. Because raw data is not shared over the network; only model updates are, hence it also lowers bandwidth requirements. We would hypothetically assess each strategy offered using the NSL-KDD dataset, highlighting its methodology, results, and limitations based on recognized techniques for attack classification. The distinct benefits and challenges that each technique presents progress the discipline of intrusion detection. In a number of scenarios, federated learning can significantly increase the adaptability and effectiveness of intrusion detection systems (IDS), enabling the discovery of novel threats and reducing the dangers to user privacy. It also reduces bandwidth

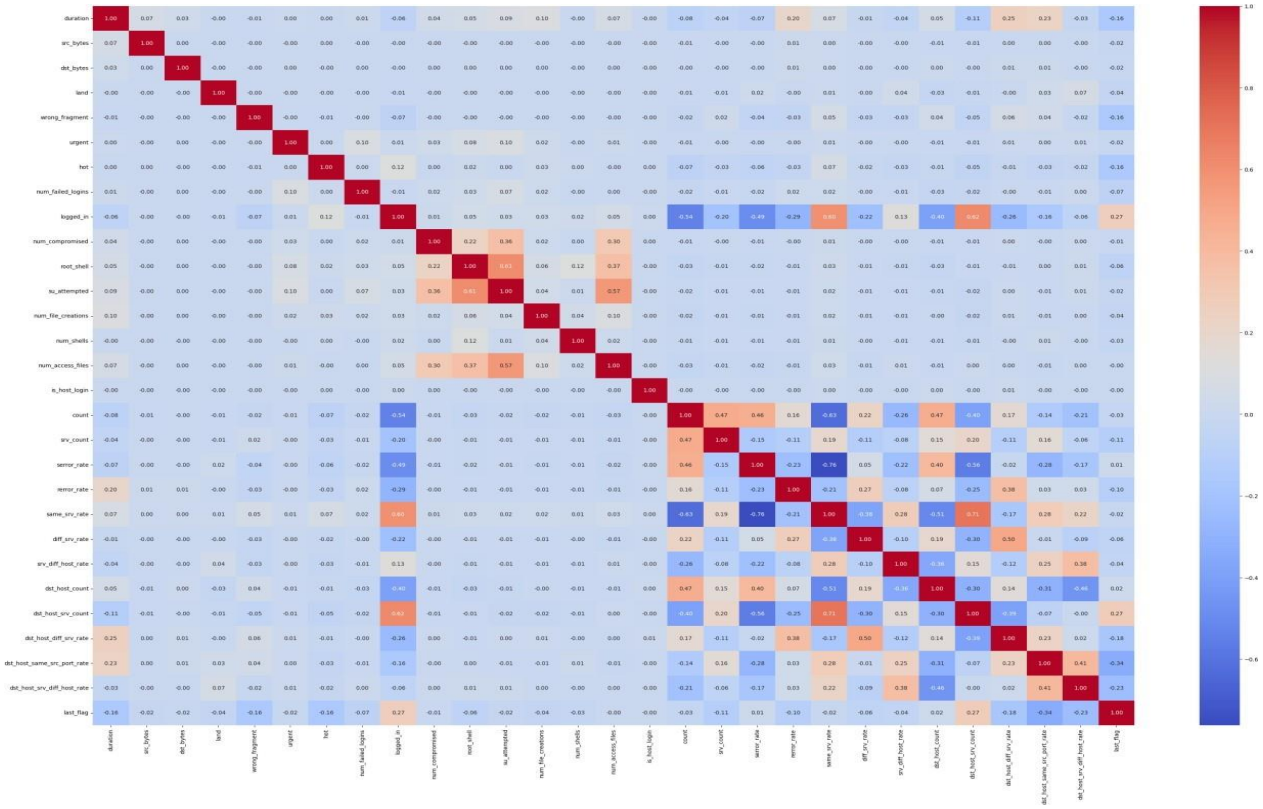

Fig. 6. Correlation Matrix

2. Building Correlation Matrix

Building a correlation matrix is an essential step in figuring out the links between the various features in the dataset, once null values and duplicates have been addressed. The correlation matrix shows how different variables are related to one another and can be used to spot possible problems with multicollinearity or redundancy. The coefficients of correlation can be employed to identify characteristic combinations that show significant connection, which denotes redundant information. We can see the co relation matrix in **Fig 6.**

3. Dropping Null Values

Constant or nearly constant values in a column provide little to no information for training the model, and they may even cause the model to perform less well. It is crucial to locate and eliminate these kinds of columns from the dataset. These columns add no variability to the dataset and could cause the model to become biassed or overfit. we can simplify the dataset and concentrate on pertinent features that provide valuable information for intrusion detection by removing constant value columns in the above co relation matrix we have seen the num_out_bounds columns have constant values so we have to drop the columnn.

4. Dropping Highly Correlated Features

Because highly correlated features essentially convey redundant information, they can also provide issues for Machine Learning algorithms. Removing highly correlated columns from preprocessing stage reduces multicollinearity. Problems and keeps duplicated information from unduly influencing the model. Retaining only one representative feature from pairs of highly correlated data allows you to increase interpretability and minimise model complexity without compromising predictive performance so, we have dropped the highly co related features.

B. Normalization

The goal of the feature scaling technique known as normalisation is to get all of the attribute values on the same scale. Many normalisation techniques exist, such as min-max normalisation, resilient scaling, Max Abs scaling, and Standard Scalar In this research, we employed standard scalar and the min–max normalisation approaches.

1. Standard Scalar

Standard Scalar does not confine the data to a certain range and is more resistant to outliers. Because of this, it can be used with algorithms that presume that the data has a normal distribution and is centered around zero. By using Standard Scalar, The data is rescaled to have a standard deviation of 1 and a mean of 0. This is how you calculate:

$$x_{std} = x - \mu/\sigma \qquad (1)$$

where the standardized value is denoted by x std.

x stands for initial value.

μ represents the feature mean.

σ represents the feature standard deviation.

2. Min-Max

By scaling the features to a given range, often [0, 1], Min-Max Normalization is applied. The Min-Max Normalization formula is as follows:

$$x_{norm} = x - x_{min} / x_{max} - x_{min} \quad (2)$$

Where the normalized value is denoted by x norm.

The initial value is x.

The equation's minimum and maximum values are represented by the variables x min and x max.

3. Dimensionality Reduction using PCA

Principal component analysis, or PCA, is a crucial technique we employ in our work on detection of intrusions for Web of Things (also known as IoT) networks. It assists in lowering the dataset's dimensionality while preserving its essential facts. We can model and analyze network traffic data more effectively by use PCA to transform the initial complex space of features into a lower-dimensional subspace.

By finding the principle components that account for the greatest amount of variance, principal component analysis (PCA) enables us to capture the inherent structure and variability of the dataset. Each of these major components captures a distinct combination of features, and together they create orthogonal axes in the feature space. We are able to minimise information while drastically reducing the dimensionality of the dataset by keeping only the highest principal components that account for most of the variance.

Additionally, PCA makes it easier to find and remove characteristics that are superfluous or unnecessary and may not have a major impact on the classification of typical and abnormal network behaviour. We can reduce the computational complexity of the analysis while simultaneously improving the discriminative capability of our intrusion detection models by concentrating on the principal components that reflect the most important variability in the data.

D. Feature Selection using RFC

In order to determine and choose the most significant characteristics from our dataset, we used the Random Forest Classifier in this investigation. Until the required number of features is obtained, the RFE method iteratively fits Random Forest by ranking and removing the least significant features. Improving the model's effectiveness, interpretability, and maybe lowering overfitting are the main objectives. Following the Random Forest Classifier's RFE process, the names of the top 10 features that it determined to be most important are listed in the selected_features list.

This subset of features can potentially enhance the model's predictive performance by focusing on the most informative aspects of the dataset. We used PCA also for Dimensionality Reduction**.** It is frequently used to reduce the computationally challenging nature of systems for machine learning. To recognize foremost imperative designs within the information. It is particularly useful when dealing with high-dimensional dataset. Be beyond any doubt that when using PCA, you will lose a few interpretabilities as the central components might not specifically compare to the first highlights

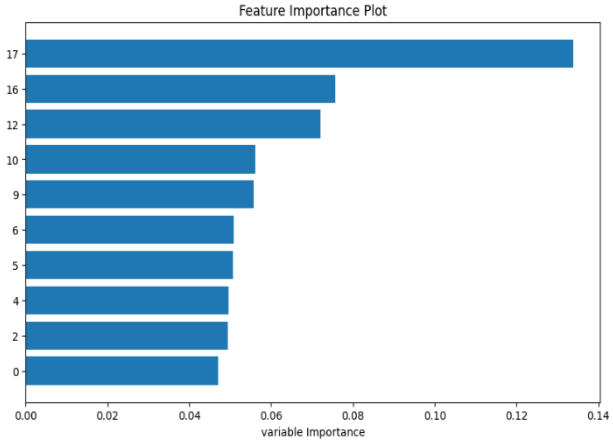

Fig.7. Feature importance plot

E. Model Creation & Evaluation

1. XG Boost

Another of the latest group learning algorithms is Xg-boost, a very efficient tree boosting algorithm. It has yielded state-of-the-art results in several applications. Xg-boost selects features and evaluates their importance by using the idea of forest bands. Designed for speed and performance, this approach is a slope enhanced tree of choices solution. It has been used in many real-world applications and machine-learning contests, frequently producing state-of-the-art outcomes. The success of XGBoost can be ascribed to its versatility in supporting a wide range of objective functions and evaluation criteria as well as its scalability, which enables it to handle large-scale data efficiently.

```
Accuracy: 0.9959515241320914
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     20082
         1.0       1.00      1.00      1.00     13837
         2.0       0.98      0.99      0.99      3547
         3.0       0.97      0.95      0.96       304
         4.0       0.50      0.18      0.27        22

    accuracy                           1.00     37792
   macro avg       0.89      0.82      0.84     37792
weighted avg       1.00      1.00      1.00     37792
```

2. CAT BOOST

Strong Machine Learning techniques like the Cat boost algorithm have produced excellent outcomes in a range of applications. However, the purpose of Cat Boost is to manage category traits. It can still manage numerical or continuous properties, albeit. The gradient-boosting decision tree technique is enhanced with the cat boost model, a unique feature. Among the things that distinguish CatBoost from other gradient boosting methods is its inbuilt support for categorical variables. Before being used as input to the

model, categorical features in traditional gradient boosting implementations must first be preprocessed or converted into numerical representations. This laborious preparation stage may cause information loss or add biases into the final product.

Accuracy: 0.9992591024555462
Classification Report:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	20082
1.0	1.00	1.00	1.00	13837
2.0	1.00	1.00	1.00	3547
3.0	0.99	0.99	0.99	304
4.0	1.00	0.68	0.81	22
accuracy			1.00	37792
macro avg	1.00	0.93	0.96	37792
weighted avg	1.00	1.00	1.00	37792

3. KNN

One of the most basic categories in machine learning is the KNN. The k-NN technique uses each labeled training instance to build a model of the function of interest. The totally non-parametric the K-NN algorithm method to object classification, which clusters items according to training examples that are closest to them in the feature space, makes use of instance-based learning. The the K-NN algorithm approach offers the benefit of being a manageable classifier conceptually in an IDS.

Accuracy: 0.9980948348856901
Classification Report:

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	20082
1.0	1.00	1.00	1.00	13837
2.0	0.99	1.00	0.99	3547
3.0	0.98	0.94	0.96	304
4.0	0.80	0.55	0.65	22
accuracy			1.00	37792
macro avg	0.95	0.90	0.92	37792
weighted avg	1.00	1.00	1.00	37792

4. SVM

One popular, flexible, and useful classification technique that can deal with issues with binary classification is a support raster network. Based on the structural danger reduction values, an SVM method of classification divides the beneficial and negatives class variables using a hyper-plane. SVM requires few parameters, is very resilient to local minima, and performs well in generalization. One of the most widely used and adaptable classification methods, the support vector machine (SVM), is well known for its efficiency in handling binary classification issues. It searches the feature space for the plane that best divides positive class occurrences from negative class instances, depending on the architectural risk reduction concept.

Accuracy: 0.9491162150719729
Classification Report:

	precision	recall	f1-score	support
0.0	0.96	0.96	0.96	20082
1.0	0.96	0.95	0.96	13837
2.0	0.84	0.89	0.86	3547
3.0	0.89	0.83	0.86	304
4.0	0.00	0.00	0.00	22
accuracy			0.95	37792
macro avg	0.73	0.73	0.73	37792
weighted avg	0.95	0.95	0.95	37792

5. QDA

Within the diagnostic assessment family, the following batch of classifiers is called quadratic discriminant modeling, or QDA. Compared to LDA, QDA produces more accurate results from analyses. It divides observations utilizing the quadratic formula concept. Using the QDA, the ten component qualities selected by PCA were categorized in this study. The QDA calibration map utilized for the analysis is shown in Fig. 9.

Accuracy: 0.9321284928027096
Classification Report:

	precision	recall	f1-score	support
0.0	1.00	0.98	0.99	20082
1.0	0.98	0.98	0.98	13837
2.0	0.91	0.56	0.69	3547
3.0	0.31	0.16	0.22	304
4.0	0.01	0.82	0.02	22
accuracy			0.93	37792
macro avg	0.64	0.70	0.58	37792
weighted avg	0.98	0.93	0.95	37792

6. NAÏVE BAYES

NB is a Bayesian Hypothesis-based classifier that is straightforward and incredibly scalable. The likelihood that a class will be in the attack or regular classes is predicted using NB. During the training and classification stages, it functions without a hitch. The underlying assumption of NB is that each vector's properties are distinct and equally significant. utilizing the Bayes Theorem, Naïve bayes is a straightforward classifier that is very scalable.

Accuracy: 0.8952159187129551
Classification Report:

	precision	recall	f1-score	support
0.0	0.97	0.92	0.95	20082
1.0	0.94	0.90	0.92	13837
2.0	0.66	0.73	0.69	3547
3.0	0.30	0.88	0.45	304
4.0	0.02	0.68	0.04	22
accuracy			0.90	37792
macro avg	0.58	0.82	0.61	37792
weighted avg	0.93	0.90	0.91	37792

TABLE.2. Comparison with past studies

Table 2 Comparison with past studies.

Authors	Security Threat	Validation Dataset Strategy	Accuracy	Precision	F1 Score	Recall
[26]	Network	NSL-KDD	86.53	-	-	-
[27]	Network	NSL-KDD	94.27	92.18	92.29	84.44
[28]	Network	NSL-KDD	91.39	-	-	-
[29]	Network	NSL-KDD	98	97	97	-
Proposed Xg Boost	Network	NSL-KDD	99.88	1.00	1.00	1.00
Proposed Cat Boost	Network	NSL-KDD	99.88	1.00	1.00	1.00
Proposed KNN	Network	NSL-KDD	99.98	1.00	1.00	1.00
Proposed SVM	Network	NSL-KDD	97.66	0.96	96.54	96.96
Proposed QDA	Network	NSL-KDD	95.44	1.00	99.98	99.94
Proposed NB	Network	NSL-KDD	97.64	0.97	95.94	95.41

Table.2 represents the comparison of past studies with different methods with same dataset NSL-KDD dataset. Past studies differentiate the past studies like neural networks and traditional methods. We examine historical research and make a distinction between using neural networks and conventional techniques on the NSL-KDD dataset. This dataset, renowned for its impartial portrayal of network threats, is used as a standard to assess the efficacy and efficiency of various cybersecurity techniques.

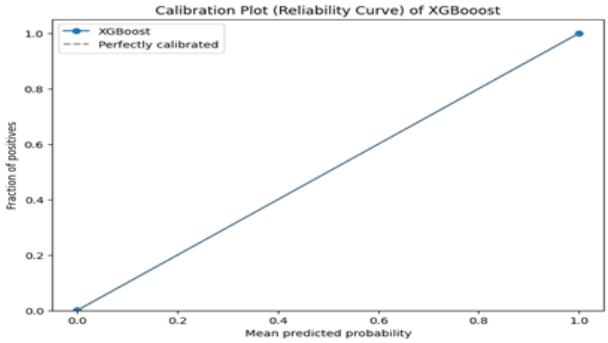

Fig.8. Calibration of Xg Boost

Fig.8 was plotted to illustrate the model's calibration process. The x-axis displays the mean predicted probability in each bin, while the y-axis represents the fraction of positives, or the actual percentage of positive outcomes in that bins is the degree of agreement between observed and projected probabilities of outcomes.

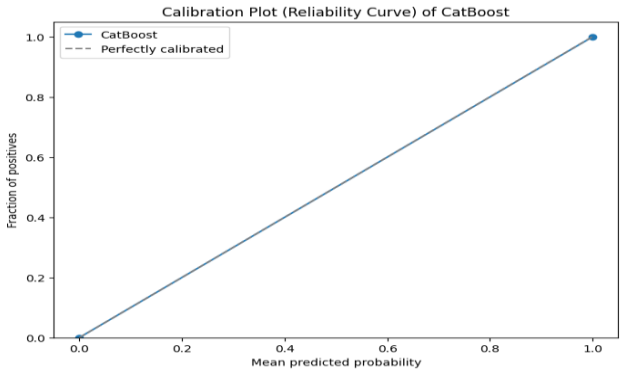

Fig.9. Calibration of Cat Boost

The accuracy of the CatBoost classifier's binary result prediction is the main focus of Fig. 10's calibration evaluation. The model's performance is measured by how well projected probability and observed outcomes agree, which is known as calibration.

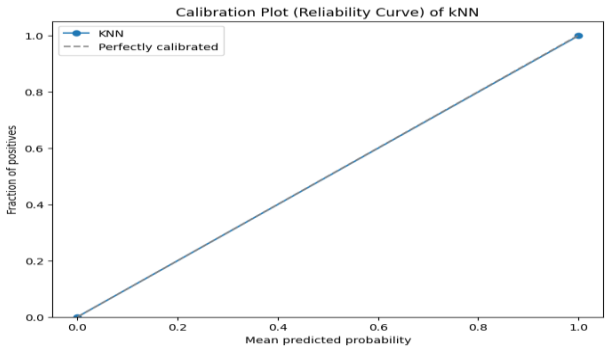

Fig.10. Calibration of KNN

In Fig. 10, the calibration of the K-Nearest Neighbours (KNN) classifier is evaluated, with special focus on the model's reliability in probabilistic predictions for binary classification tasks. Calibration is the degree of agreement between observed and projected probability outcomes.

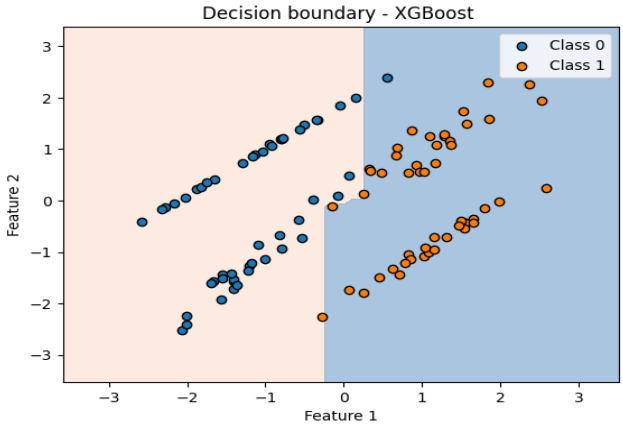

Fig.11. Decision Boundary of Xg Boost

In addition to demonstrating that the model successfully divides the two groups, Fig. 11 sheds light on the decision boundary's complexity and non-linearity situations is greatly aided by these insights, especially where interpretability and model decision-making procedures are crucial. In this paper, we tackle a binary classification problem using the scalable Machine Learning system for tree boosting, known as the XG Boost method. For a variety of classification jobs, XG Boost is the recommended option due to its exceptional efficiency.

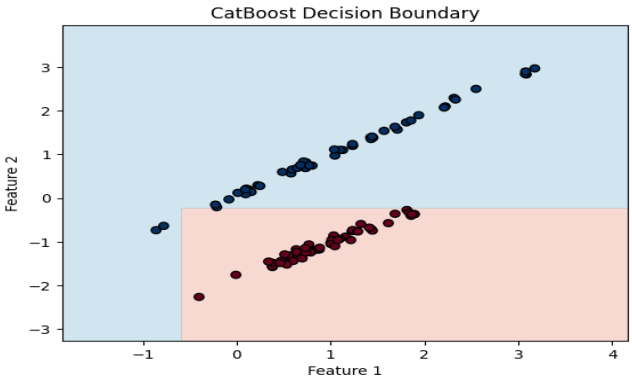

Fig.12. Decision Boundary of CatBoost

In this research, we tackle a binary classification problem using the scalable Machine Learning system for tree boosting, known as this Cat Boost method. For a variety of classification jobs, Cat Boost is the recommended option due to its exceptional efficiency.

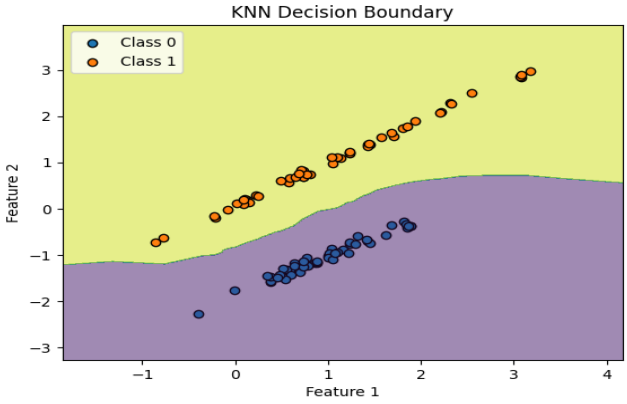

Fig. 13. Decision Boundary of KNN

The accuracy of the KNN's classifier's binary result prediction is the main focus of Fig. 13's calibration evaluation. The model's performance is measured by how well projected probability and observed outcomes agree, which is known as calibration. The above KNN understanding how Machine Learning models behave in real-world.

Table.3. Before and After applying K-Fold

Classifiers	Actual accuracy	K-Fold accuracy
XgBoost	99.5	99.8
CatBoost	99.9	99.8
KNN	99.8	99.9
SVM	94.9	97.6
QDA	93.2	95.4
Naïve Bayes	89.5	97.6

Table.3 describes about the accuarcies that were compares the accuracies before and after applying K-Fold.

VI. RESULTS AND ANALYSIS

The results of the experimental analysis are presented and analyzed in this section. This paper adopted the NSL-KDD dataset. This NSL-KDD is a perfect tool to extract a mixture of contemporary attacks and normal activities of network traffic. The attacks of NSL were grouped into four different attack types. They are Probe, DOS, R2L, U2R. The dataset was split into two, where 75% of the dataset was used for training the model, and 25% was used for testing the model. From the below figure we can observe the accuracy scores of our classifiers.

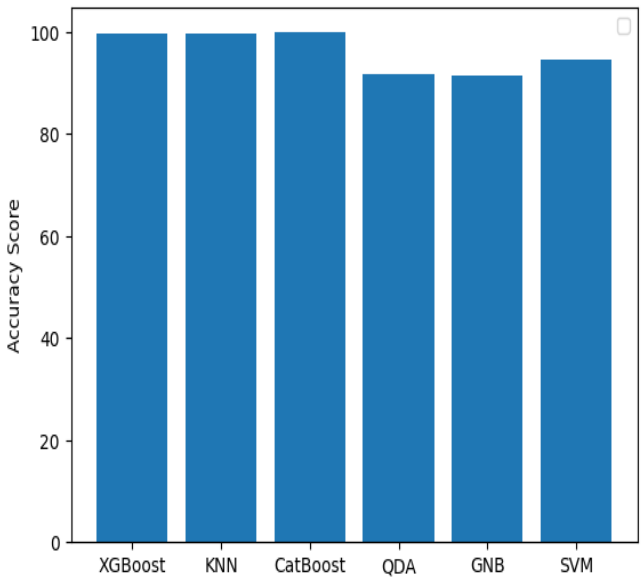

Fig. 14. Accuracy scores of different classifiers

V.PERFORMACE ANALYSIS

The model's performance was assessed using the following metrics: support, F1-Score, accuracy, recall, and precision. Table 3 displays the experimental outcomes of our conclusions regarding the suggested models. Each model's decision border was also developed to bolster the findings of our investigation. Each model's decision border was also developed to bolster the findings of our investigation. During the investigation, the decision boundary offers valuable insights about the methods used by each model to approach the problem. Xg Boost's decision boundary reveals how it takes on the role of categorization, as can be seen in Fig. 11.

The border of decision-making in machine learning refers to the line, surface, or manifold that separates multiple classes inside the feature space. Stated otherwise, it is the boundary that a model uses to identify the class to which a certain data item belongs, depending on features. Determining decision limits can reveal a model's benefits as well as any potential flaws or biases and offer important insights into the categorization process.

The decision border of KNN, as seen in Fig. 13, offers insights into the way KNN examines the classification task. The line, surface, or manifold dividing many classes inside the feature space is referred to as the border of decision-making in machine learning. Put another way, it's the limit that a model employs based on features to determine which class a given data item belongs to. Determining decision boundaries can provide valuable insights into the classification process and highlight a model's advantages as well as any potential biases or shortcomings.

VI. CONCLUSION

We looked at the practicality of using intrusion detection based on Machine Learning. In order to do this, we skillfully combined feature dimensionality reduction and Machine Learning techniques to create an intelligent intrusion detection system (IDS) that can identify unusual activity on vulnerable networks. To find the best strategy for Machine Learning-based intrusion detection systems, we assessed our scheme's performance using the NSL-KDD dataset. Machine Learning-based intrusion detection systems could handle security detection duties. PCA is utilized to reduce dimensionality, while the Random Forest technique is used to choose ten components based on features.

The model was tested on the most complex dataset available, NSL-KDD, which is ideal for intrusion detection and supports modern threats. The suggested approach yields higher F1 scores, indicating a stronger overall detection performance, according to our results. We may conclude that applying Machine Learning approaches for successful anomaly detection is both practical and practicable based on the experimental results from network simulations. The suggested Xg Boost, Cat Boost, CNN, SVM, QDA, and Gaussian NB algorithms exhibit exceptional accuracy when compared to previous research and can resolve the dataset's labelled data problem. Our work's experimental results, which achieved 99.99%, outperformed the state-of-the-art in terms of accuracy precision and support of the three models we offered.

Table.4. PERFORMANCE MEASURE OF PROPOSED MODEL

Classifiers	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*Support*
XGB	99.8	1.00	1.00	1.00	20082
CAT	99.8	1.00	1.00	1.00	20082
KNN	99.9	1.00	1.00	1.00	20082
SVM	97.6	0.96	0.96	0.96	20082
QDA	95.4	1.00	0.98	0.99	20082
NAÏVE BAYES	97.6	0.97	0.92	0.95	20082

REFERENCES

[1] Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. *Computers in industry*, *101*, 1-12.

[2] Yu, X., & Guo, H. (2019, August). A survey on IIoT security. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)* (pp. 1-5). IEEE

.

[3] Tushir, B., Sehgal, H., Nair, R., Dezfouli, B., & Liu, Y. (2021). The impact of dos attacks onresource-constrained iot devices: A study on the mirai attack. *arXiv preprint arXiv:2104.09041*.

[4] Fadlelmula, W. B., Mohamed, S. H., El-Gorashi, T. E., & Elmirghani, J. M. (2023, July). Energy Efficient Resource Allocation for Demand Intensive Applications in a VLC Based Fog Architecture. In *2023 23rd International Conference on Transparent Optical Networks (ICTON)* (pp. 1-6). IEEE.

[5] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 4396.

[6] Nakip, M., & Gelenbe, E. (2021, October). Botnet attack detection with incremental online learning. In *International ISCIS Security Workshop* (pp. 51-60). Cham: Springer International Publishing.

[7] Gelenbe, E., & Nakıp, M. (2022). Traffic based sequential learning during botnet attacks to identify compromised iot devices. *IEEE Access*, *10*, 126536-126549.

[8] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)* (pp. 21-26).

[9] Ahmad, T., Anwar, M. A., & Haque, M. (2020). Machine learning techniques for intrusion detection. In *Handbook of Research on Intrusion Detection Systems* (pp. 47-65). IGI Global.

[10] Geetha, R., & Thilagam, T. (2021). A review on the effectiveness of machine learning and deep learning algorithms for cyber security. *Archives of Computational Methods in Engineering*, *28*(4), 2861-2879.

[11] Kim, K., Aminanto, M. E., & Tanuwidjaja, H. C. (2018). *Network intrusion detection using deep learning: a feature learning approach*. Springer.

[12] Babicheva, M. V., & Tretyakov, I. A. (2023). Application of machine learning methods for automated detection of network intrusions. *Herald of Dagestan State Technical University Technical Sciences*, *50*(1), 53-61.

[13] Shashank, Rakesh, Sharma. (2023). Advancements in Machine Learning for Intrusion Detection in Cloud Environments. Indian Scientific Journal Of Research In Engineering And Management, doi: 10.55041/ijsrem24430

[14] Bingu, R., Jothilakshmi, S., & Srinivasu, N. (2023). An intelligent multiclass deep classifier-based intrusion detection system for cloud environment. *Concurrency and Computation: Practice and Experience*, *35*(26), e7840.

[15] Srivastav, S., Guleria, K., & Sharma, S. (2023, May). Machine Learning Based Predictive Model for Intrusion Detection. In *2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)* (pp. 1-5). IEEE.

[16] Gautam, R. K. S., & Doegar, E. A. (2018, January). An ensemble approach for intrusion detection system using machine learning algorithms. In *2018 8th International conference on cloud computing, data science & engineering (confluence)* (pp. 14-15). IEEE.

[17] Ogundokun, R. O., Basil, U., Babatunde, A. N., Abdulahi, A. T., Adenike, A. R., & Adebiyi, A. A. (2023, April). Intrusion Detection Systems Based on Machine Learning Approaches: A Systematic Review. In *2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)* (Vol. 1, pp. 01-04). IEEE.

[18] Vinod, D., & Prasad, M. (2023, April). A novel hybrid automatic intrusion detection system using machine learning technique for anomalous detection based on traffic prediction. In *2023 International Conference on Networking and Communications (ICNWC)* (pp. 1-7). IEEE.

[19] N., C., Thoutam., Mayur, Sonawane., Ghanshyam, R., Chaudhari., Om, Kathe., Prajwal, Sontakke. (2023). Machine Learning for the Identification of Network Anomalies. Indian Scientific Journal Of Research In Engineering And Management, doi: 10.55041/ijsrem18082

[20]Musa, U. S., Chhabra, M., Ali, A., & Kaur, M. (2020, September). Intrusion detection system using machine learning techniques: A review. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 149-155). IEEE.

[21] Liu, R. (2023, February). Multivariate Network Intrusion Detection Methods Based on Machine Learning. In *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)* (pp. 148-155). IEEE.

[22] D, Harshavardhan, Reddy., A., Elumalai. (2022). A Review on Intrusion Detection Using Machine Learning Techniques. International Journal of Engineering Research in Computer Science and Engineering, doi: 10.36647/ijercse/09.12.art013

[23] Sirisha, A., & Premamayudu, B. (2023, March). A brief analysis on efficient machine learning techniques for intrusion detection model to provide network security. In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 105-112). IEEE.

[24] Yilmaz, A. A. (2022, December). Intrusion detection in computer networks using optimized machine learning algorithms. In *2022 3rd International Informatics and Software Engineering Conference (IISEC)* (pp. 1-5). IEEE.

[25] Yilmaz, A. A. (2022, December). Intrusion detection in computer networks using optimized machine learning algorithms. In *2022 3rd International Informatics and Software Engineering Conference (IISEC)* (pp. 1-5). IEEE.

[26] Rathore, S., & Park, J. H. (2018). Semi-supervised learning based distributed attack detection framework for IoT. *Applied Soft Computing*, *72*, 79-89.

[27] Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., & Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory*, *101*, 102031.

[28] Pourreza, M., Mohammadi, B., Khaki, M., Bouindour, S., Snoussi, H., & Sabokrou, M. (2021). G2d: Generate to detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 2003-2012).

[29] Liang, C., Shanmugam, B., Azam, S., Jonkman, M., De Boer, F., & Narayansamy, G. (2019, March). Intrusion detection system for Internet of Things based on a machine learning approach. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)* (pp. 1-6). IEEE.

[30]https://www.kaggle.com/datasets/hassan06/nslkdd