A Survey on Intrusion Detection Systems (IDSs) Using Machine Learning and Deep Learning Techniques

Praveen Agrawal
Deptt. of Electronics & Communication Engineering
Shriram College of Engineering & Management (SRCEM)
Banmore, Gwalior(M.P)
praveen_agrawal90@yahoo.co.in

Prof. Ashish Duvey
Deptt. of Electronics & Communication Engineering
Shriram College of Engineering & Management (SRCEM)
Banmore, Gwalior(M.P)
adsrcem@gmail.com

Abstract*— Cybersecurity is becoming an increasingly important field of study because of the growing importance of networks in modern life. The most common cyber security measures include anti-virus software, firewalls, and intrusion detection systems (IDSs). Both internal and external threats can be protected by these methods. An IDS is a detection system that keeps tabs on the health of a network's software and hardware in order to keep that network's data safe. Analyzing the software or hardware of a network is the primary function of an IDS, a critical cyber security method. Current intrusion detection systems continue to face difficulties in increasing detection accuracy, reducing false alarm rates, and identifying unexpected threats, even after decades of research. Many academics have focused on developing IDSs that use machine learning approaches to address the issues raised above. Automatic and accurate detection of normal and aberrant data can be achieved through machine learning approaches. In addition, because machine learning (ML) techniques are so generalizable, they may uncover previously unknown attacks. Deep learning (DL) is a branch of machine learning (ML) that has grown in popularity as a result of its superior performance. This study offers an IDS taxonomy based on statistical objects as the primary dimension for classifying and summarizing, ML or DL-based IDS approaches. This form of classification structure, we feel, is appropriate for cyber cybersecurity experts. The survey defines the notion of IDSs and their classification. Furthermore, the MLand DL methods that are often employed in intrusion detection systems, measurements, including benchmark datasets are presented.

Keywords*— Intrusion Detection System, NIDS, AIDS, SIDS, IDS attcks IDS dataset, Machine Learning, Deep Learning.

I. INTRODUCTION

With the quick growth as well as widespread use of 5G, IoT, Cloud Computing, as well as other innovations, network size with real-time traffic have gotten more sophisticated and vaster, as have cyber-attacks, posing numerous problems to cyberspace security. IDS detects intrusions in the network that take the form of anomalies and alerts the user. There has been a lot of study in this topic in current history. IDSs are classified into two types: anomaly detection-based systems as well as signature-based systems. The Signature-based IDS examines network packets & compared them to recognized signatures that have been pre-configured as well as pre-identified based on previously known attack behaviour. The AIDS, on either hand, monitors regular network traffic including such bandwidth range, protocol types, ports or systems used this to connect as well as provides an alarm to the system administrator when abnormal behaviour is detected. NIDS must reliably detect hostile network assaults, offer real-time monitoring as well as dynamic security measures, including innovation strategy as the second line of defence the behind firewall[1].

The vast majority of traffic data in real-world cyberspace is routine activity, with just a few hostile cyber-attacks making up the majority. Because of the significant imbalances and duplicated nature of the network traffic data, intrusion detection is under a great deal of strain. Attacks on a network may masquerade as regular traffic if there is enough of it. Because of this, the algorithms for ML cannot properly understand the distribution of a small number of categories, and it is simple for it to get things wrong. [2].

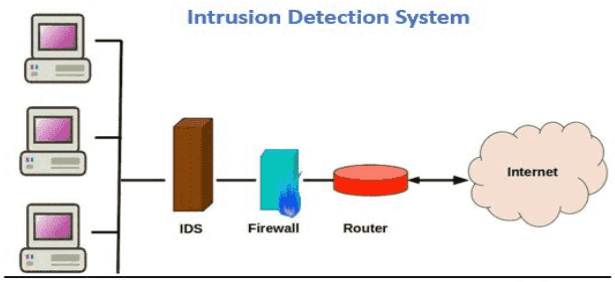

Figure 1: Intrusion Detection System

Typically, an IDS does not protect the system from being attacked by an intrusion; however, it only produces an alert after recognizing an attack in the implementation in real or before the assault arrives on the target. This may be done either before or after the attack occurs. Whereas an IDS keeps as well as updates an intrusion profiles in the log, it is just as important to cause notice of an attempt after it has occurred in

the system. This is because of the importance of preventing further damage to the system. The operating system should be able to support a variety of tasks, such as the analysis of logs, which need more space on the disc as well as resources from the CPU. Controlling the logs formats as well as contrasting those formats with the attack patterns that have been detected in accordance with the security violations that have been identified is another significant problem in the IDS [3] [4].

II. CLASSIFICATION OF IDS

According to V. Jyothsna [5] there are primarily three categories of intrusion detection systems, which are as follows: signature-based (SIDS), anomaly-based (AIDS), as well as network intrusion detection system (NIDS). SIDS systems like as Snort make advantage of pattern identification techniques by storing a catalog of signatures of previously identified attacks and comparing these signatures with freshly processed data. These approaches let the system to identify new threats more quickly. When it is determined that two things are similar, an alert is triggered. On the other hand, while systems including such PAYL construct a quantitative model to characterize the regular network traffic, ABS systems develop a framework to explain the usual network traffic and thereafter identify any aberrant behaviour that deviates from the model. Anomaly-based security solutions, on the other hand, offer the distinct benefit of being able to identify zero-day threats [6].

a) Signature based Detection (SIDS)

This type of detection is particularly successful against known attacks, but it is dependent on getting frequent updates of patterns in order to function properly. SIDSare often referred to as misuse-based detection systems. [7]. In the event that the user makes use of cutting-edge tools like a NOP generator or payload transponders with encrypted data routes, signature-based identification will fall short. Because each change necessitates a new signature, signature-based systems are significantly slower than those that don't. The efficiency of the system engine decreases as the number of signatures grows. Machines with multiple CPUs and Gigabit network cards have numerous intrusion detection algorithms installed on them. To prevent new attacks on the system, IDS engineers generate the based on developing before the attacker does. As designers and attackers work at different speeds, it has an impact on the algorithm's performance[6].

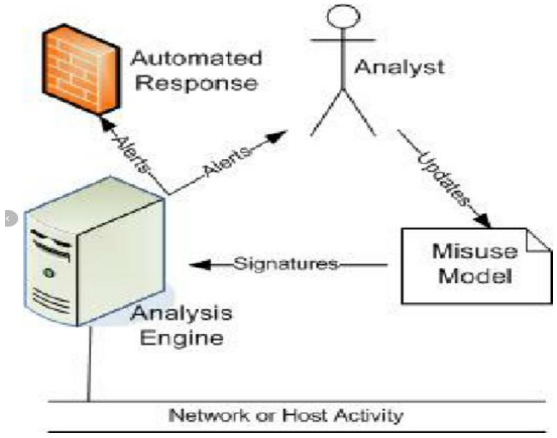

Figure 2: Signature Based Detection System

b) Anomaly based Detection (AIDS)

An anomaly-based intrusion detection system is a kind of intrusion detection system that monitors operation of the system as well as categorises it as either normal or abnormal. This allows the system to identify information as well as computer intrusions as well as abuse. Instead of looking for patterns or signs, the categorization, which is based on heuristics or rules, makes an effort to identify any kind of improper usage that deviates from the typical functioning of the system. SIDS methods, on the other hand, are limited in their ability to detect assaults since they can only identify those for which a signature has already been developed. [8] The definition of the network's behaviour is the foundation of the anomaly-based detection system. If the behaviour of the network matches the behaviour that has been specified, then it is allowed; otherwise, it will cause the event that is associated with the anomaly detection. The requirements of the network management are used to either prepare or learn the behaviour that is acceptable on the network.

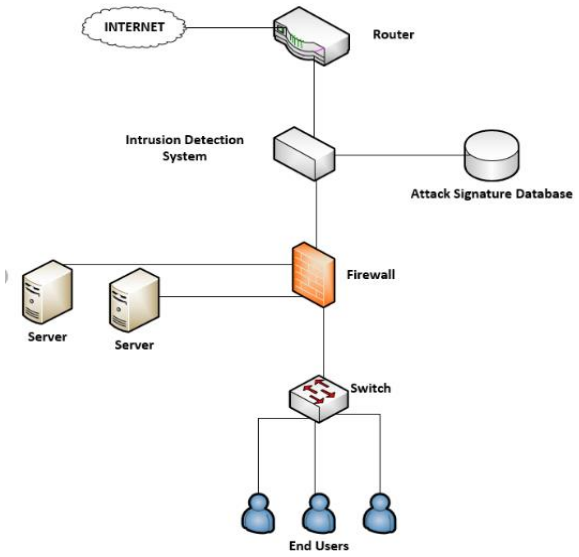

Figure 3: Anomaly-Based Intrusion Detection System

c) Network Intrusion Detection System

NIDS are installed at key nodes in the network architecture at important locations. The Network Intrusion Detection System (NIDS) is able to collect and examine data in order to discover previously unknown assaults by analysing patterns or signatures in the database. Additionally, the NIDS may detect illicit actions by scanning traffic for unusual behaviour. NIDS are also known as "packet-sniffers" due to the fact that they are able to intercept data packets as they travel across various communication media. The sensor as well as the administration station are the two logical components that are often included in a network IDS. The sensor is installed on a subnetwork and is responsible for monitoring that segment for any unusual traffic. The alerts that have been generated by the sensor(s) are sent to the centralized server, which then displays them to an administrator.

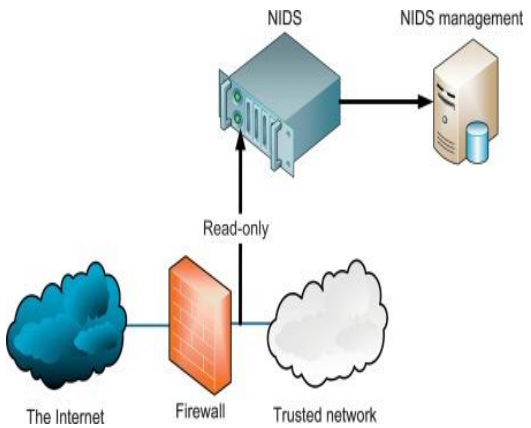

Figure 4: Network Intrusion Detection System

In most cases, the sensors are specialized systems that have no purpose other than to monitor the network. They have a network interface that is set to the promiscuous mode, that indicates that they receive overall network traffic instead of just the traffic that is meant for their IP address. Additionally, they collect passing network data for the purpose of doing analysis on it. When they come across anything that seems out of the ordinary, they report it to the station that does the study. The alerts may be shown at the analysis station, as well as it can also do extra analysis. NIDS, which are responsible for the passive monitoring of a network connection, face a basic challenge in the form of the ability of an experienced attacker to avoid detection by taking advantage of ambiguities in the traffic stream as observed by the NIDS [9].

III. TYPES OF ATTACKS

There are 22 different kinds of assaults in the Intrusion Detection data set, which is used to develop the IDS and evaluate the techniques provided in the book. These attacks are classified as [10]:

a) Denial of Service (DoS)

The attackers attempted to intercept the original users' efforts to utilise any service in this assault. Flooding or crashing services are two of the most common ways of DoS attacks. Whenever the server is overloaded with requests, a flood assault occurs. This slows down and finally stops the system from responding.

b) Remote to Local (R2L)

This assault is aimed at getting into the victim's computer without their having an account. As a result, an attacker who does not have an account on the victim system, but who is nevertheless interested in exploiting a vulnerability on that machine, sends packets across the network to the target machine in an attempt to get local access.

c) User to Root (U2R)

Attackers attempt to take control of the victim's computer by logging in as the user. Using remote to local exploits, fraudsters may use security flaws to launch malicious code, which can have disastrous effects for businesses. Such an attack may be used to steal data, disrupt corporate operations, and spy on users. Manual response methods have a long dwell time when trying to identify these kinds of assaults.

d) Probe

Attackers should be able to get access to critical information about the target host. Denial of Service (DoS)

assaults account for the majority of all attacks (DoS). There are many forms of assaults to be found [10] including spying, listening, intercepting, & Distributed Denial of Service (DDoS) assaults, to mention a few.

e) Distributed Denial of Service (DDoS)

Attacks against a server or network in an effort to interrupt regular flow. Internet traffic is encroaching on the target server and its immediate environs. Normal users are unable to reach the target and its surroundings as a consequence.

IV. BENCHMARK DATASETS IN IDS

Since the goal of machine learning is to extract useful information from data, its effectiveness is directly proportional to the quality of the data that is fed into it. The approach of machine learning is built on a foundation of data comprehension. The data that is used by IDSs must to be straightforward to get and ought to accurately represent the actions taken by hosts or networks. IDSs often get their information from packets, flows, sessions, and logs as their primary sources of data. Putting together a dataset is a difficult and time-consuming process. When a benchmark dataset is finished being compiled, it may be used again and again by a large number of researchers. The use of benchmark datasets comes with not one but two additional advantages in addition to its obvious ease. (1) The benchmark datasets have a high level of credibility, which helps to bolster the validity of the experimental findings. (2) Numerous published research have been carried out using the use of standard datasets, which enables the findings of recent studies to be compared with the findings of earlier studies.[4].

a) DARPA1998 [1]

A benchmark dataset that is often used in IDS research is called the DARPA1998 dataset. This set of data was developed either by Lincoln laboratory of MIT. The involved in such activities it by collecting data on Internet traffic over the course of nine weeks. The first seven weeks were used to create the training set, while the final two weeks were used to create the test set. The collection includes both unprocessed packets as well as labels. Normal, DOS, Probe, U2R and R2L labels are all available. Although raw packets cannot be used directly to conventional machine learning methods, the KDD99 dataset was created to circumvent this limitation.

b) KDD99 [2]

The KDD99 dataset is now the IDS benchmark dataset that is used the most often. Data from DARPA1998 were analysed by its programmers, and the results yielded 41-dimensional characteristics. The labels that appear in KDD99 are identical to those that appear in DARPA1998. For the purposes of KDD99, there are four distinct categories of feature: the most basic, the content-based, the host-based, as well as the time-based. Regrettably, there are a great deal of errors in the KDD99 dataset. First, there is a significant imbalance in the data, which leads to the different classifiers favouring the classes that have the majority of members. In addition, there are a large number of records that are identical to one another as well as records that are redundant. So that they can make use of it, a large number of academics need to rigorously filter the dataset. As a direct consequence of this, the study results of several research cannot always be compared to one another. Last but not least, the KDD statistics are too outdated to accurately depict the current state of the network.

c) NSL-KDD [3]

The NSL-KDD was conceived as a solution to address the deficiencies that were present in the KDD99 dataset. The recordings that are included in the NSL-KDD were chosen with great consideration with reference to the KDD99. The issue of categorization bias is circumvented because to the fact that the NSL-KDD evenly distributes records from various classes. In addition, the NSL-KDD eliminated entries that were duplicates or redundant; as a result, it only includes a reasonable amount of records. As a consequence, the tests can be carried out on the whole dataset, and the findings from the many studies are consistent and comparable to one another. The NSL-KDD helps to address some of the issues associated with biassed data and redundant data to some extent. However, the NSL-KDD does not include any new data; hence, there are still not enough samples from minority classes, and its sampling are still not up to date.

d) UNSW-NB15

The UNSW-NB15 dataset [11] was assembled by the University of South Wales, whereby academics set up three virtual servers to monitor network traffic and used a programme called Bro to extract 49-dimensional characteristics from the captured data. In comparison to the KDD99 dataset, this one has a greater variety of different kinds of assaults and a greater number of different attributes. The data categories consist of regular data as well as nine different kinds of assaults. In addition to flow features, basic features, content features, time features, extra features, and labelled features are included in the features. Subsequent

[1] http://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

[2] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3] https://www.unb.ca/cic/datasets/nsl.html

research has made use of the UNSW-NB15, which is a dataset that is typical of more recent additions to the IDS. It is important to generate fresh datasets in order to develop new IDS that are based on ML, despite the fact that the impact of UNSW-NB15 is now inferior to that of KDD99.

V. MACHINE LEARNING TECHNIQUES FOR IDS

In today's technological world[12], ML is without a doubt one of the most powerful & leading technologies. Artificial intelligence includes the field of machine learning. Samuel, an American computer game and AI inventor, introduced ML in 1959 and said that it gives machines "the capacity to learn without being taught.

Understanding the data is the first stage in the process of machine learning, which is a sort of approach that is driven by the data. As a result, we use the nature of the data source as the primary thread for categorization, as can be seen in Figure 5. In this part, we will discuss the many different ways that machine learning may be used to IDS design for the various kinds of data. The various forms of data provide light on the various attack behaviours, which may be broken down into two categories: host behaviours and network behaviours. When there is sufficient training data accessible as well as ML models have adequate generalisation to identify attack variations and new assaults, intrusion detection systems that are based on ML have the potential to reach excellent detection levels. In addition to this, machine learning-based IDS do not depend extensively on prior domain knowledge; as a result, it is simple to design and build these systems.

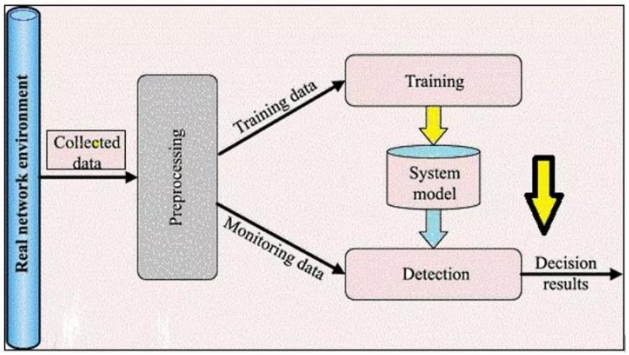

Figure 5: Real Time Network Intrusion Detection System Using Machine Learning Model

In addition to detection based on specifications, the training phase of the IDS is reliant on a ML technique of some kind, as was mentioned in the previous section. This section presents an overview of the many different machine learning methods that are used by IDSs. There are a number of positive and negative aspects associated with ML-based techniques, and references to works that are pertinent are supplied. The machine learning algorithms that are utilised most often for the creation of IoT networks [13].

1) Machine Learning Approach

A branch of artificial intelligence known as "machine learning" employs pre-trained models to learn new information. Machine learning is a concept created by Arthur Samuel in 1959 to describe study into how computers may learn on their own, without being explicitly programmed. Machine learning relies on prediction to do its work. Reinforcement learning is another example of a machine learning technique. [14][15].

a) Supervised Learning

An input-output mapping function is often learned using examples of input-output pairs in machine learning. It uses a database of training samples and tagged training data to infer a function. In this task-driven approach to supervised learning, a list of goals and a list of inputs are utilised to help students meet their goals. The term "regression" refers to fitting the data rather than classifying it. These are the most common duties that are overseen by a supervisor. Supervised learning may be shown in text categorization. For instance, supervised learning may be used to predict the class label or emotion of text, such as a tweet or a product review. Guided learning is also known as classification. The training process of supervised learning data necessitates labelling each occurrence. These kinds of long-term learning algorithms exist. “Hidden Markov Model, K-Nearest Neighbor (KNN), Bayesian Networks (C4.5, ID3, CART, and Random Forrest) and Gaussian Process Regression (GPR) are all examples of Bayesian networks. Boosting, Ensemble Classifiers, Linear Classifiers (Logistic regression, Fisher Linear discriminant, Naive Bayes classifier), Quadratic Classifiers” are some of the most often used supervised learning approaches.

b) Unsupervised Learning

There is no labelling in unsupervised learning. Clustering is a common method of implementing this strategy. Rather of relying on a human observer to categorise the data, unsupervised learning analyses the data on its own. For both practical and experimental reasons, many individuals use this extraction method. These include clustering, dimensionality reduction, feature learning, identifying associations, and anomaly detection, among others. Self-organizing maps, the Apriori and Eclat algorithms, as well as the K-means and fuzzy clustering algorithms, are all examples of unsupervised learning approaches.

c) Reinforcement Learning

"Reinforcement learning" is an example of this kind of learning. In a reinforcement strategy, a user (such as a subject matter expert) may be asked to identify an instance

from a collection of unlabeled examples. Machine learning approach Reinforcement learning allows software agents and computers to choose the best course of action for themselves in a particular circumstance or environment. Learners are encouraged to use environmental activists' ideas in order to increase the reward or reduce the threat. In high-tech systems like robots or autonomous driving or manufacturing or supply chain logistics, this strategy may help boost automation or maximise operational efficiency by developing AI models.

a) Decision Tree (DT)

Making a classifier for an unseen test case that predicts the value of a target class based on previously known examples is the job of DT. DT is used to classify an unknown test case by making a series of judgments. Its simplicity and ease of implementation make it popular as a single classifier. There are two forms of decision trees: classification trees and regression trees, the latter of which has class labels with numerical values.

b) Naive Bayes (NB)

Naive Bayes attempts to estimate the class-conditional probability based on the characteristics' conditional independence as assumed by the class label. Using naive Bayes as a classification algorithm frequently yields excellent results because of the simplicity of the categorization relations. NB uses just one scan of the training data, making categorization a lot easier.

c) Artificial Neural Network (ANN)

Human brains served as inspiration for the design of the Artificial Neural Network (ANN). NNs are often structured into layers, each of which has a no. of nodes that perform a certain task. Using a system of weighted connections, hidden layers interact with the network's input layer to do the actual processing. The detection result is subsequently sent to an output layer through the hidden layers.

d) Random forest (RF)

RF classifiers are widely utilised in ML and data science because they are well known ensemble classification approaches. Ensembling is the technique of fitting several decision tree classifiers to various subsamples of unique data sets in simultaneously. These findings are produced from this approach. Prediction accuracy and quality control are enhanced as a consequence of reducing over-fitting.

e) K-nearest neighbors (KNN)

KNN is a popular "lazy learning" method that uses "instance-based learning" or non-generalization. Rather than creating a general internal model, the n-dimensional space is used to store all instances that match to the training data. KNN classifies new data points built on similarity measures. If k of its closest neighbors votes in favor of classifying a point, it will be classified by a simple majority vote. The model's accuracy depends heavily on the quality of the training data.

f) Logistic regression (LR)

LR is a typical statistical model for classifying problems in machine learning that is based on probability. The mathematically defined sigmoid function is used in logistic regression to estimate the probabilities, which is also known as the logistic function. High-dimensional datasets may be overfitted using this method, but it works well when the data can be divided into linear chunks. To prevent over-fitting, regularisation (L1 and L2) approaches might be applied.

Table I: The pros and cons of various ML models.

Algo	pros	cons	Improvement Measures
ANN	Capable of dealing with nonlinear data; excellent fitting abilities	Overfitting is possible; becoming caught in a "local optimum" is easy to do; it takes a long time to train models.	Optimizers, activation functions, and loss functions were all upgraded and adopted.
SVM	Gain essential knowledge from a compact train set; excellent generating capabilities	Perform poorly when dealing with large amounts of data or several categorization jobs; sensitive to the parameters of the kernel function	PSO was used to fine-tune the parameters
KNN	Adapt to large amounts of data; Appropriate to nonlinear data; Put in some fast work; Resilient to noise	Low accuracy on the subset of the population being tested; Prolonged testing periods; Sensitivity to the parameter K	Time savings achieved by the use of trigonometric discrepancy; PSO used to get the parameters in the right place; datasets that are balanced via the use of the SMOTE
NB	tolerant to noisy environment;	Don't really function adequately when	Latent variables that were imported to

	capable of progressive learning	dealing with data relating to attributes	help loosen the assumption of independence
LR	Simple, and capable of being taught in a short amount of time; automatically scales features	Operate poorly when dealing with nonlinear data; prone to excessive fitting	Importing regularisation to prevent overfitting
DT	Choose options autonomously; a robust perception	The model is used to classify tends to fall into the majority class; disregard the correlation between the data.	SMOTE was used to provide balance to the datasets, and latent variables were included.
K-means	Simple, it can be taught in a short amount of time, and it has strong extensibility. Can fit to huge data	Do not operate well with data that is not convex; Sensitive to activation; Depending on the value of the parameter K	Enhanced approach to the beginning of the process

VI. DEEP LEARNING MODELS

Feature engineering is dependent on domain expertise, and feature quality is often a bottleneck of detection effects. Deep learning-based detection techniques automatically identify feature. These methodologies operate from beginning to finish and are increasingly becoming the standard strategy in IDS research. Deep learning algorithms may analyse raw data directly, enabling them to learn features while still doing categorization.

Deep learning is a subset of machine learning that may generate exceptional results. Deep learning technologies outperform typical machine learning techniques when dealing with large amounts of data. Furthermore, deep learning algorithms may learn representations from raw data and afterwards output outcomes; they are end-to-end as well as applicable. The structured programming, that has numerous hidden layers, is one distinguishing feature of deep learning.

Deep learning models are made up of several deep networks. Throughout 2015 to the current, the number of researches on deep learning-based IDSs has expanded fast. Deep learning methods learn classification model information from the source data, including such photos as well as texts, eliminating the need for human feature engineering. Deep learning approaches may therefore be used from start to finish. Deep learning approaches provide a considerable advantage over shallow models for huge datasets. The major focuses of deep learning research are network design, hyperparameter selection, as well as optimization approach. [4].

a) Autoencoder

Autoencoders consist of two symmetrical components: an encoder as well as a decoder, as illustrated in Figure 6. To decode a file, you must first decode it, and then reassemble the data by reusing the features that were decoded. There is a progressive narrowing of a gap between the inputs of the encoder and the outputs of the decoder during the training process.

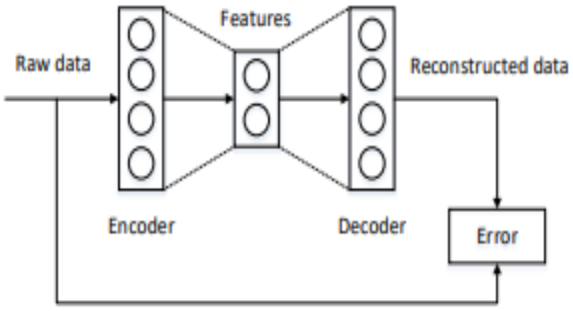

Figure 6: The structure of an autoencoder

b) Restricted Boltzmann Machine (RBM)

An RBM is a randomised neural network whereby the units follow the Boltzmann distribution. There are two parts to an RBM: one that can be seen and one that can't be seen. No connections exist between units in the same layer; nevertheless, the connections between units in other levels are complete, as seen in Figure 7. Visible and hidden layers are shown in this diagram.

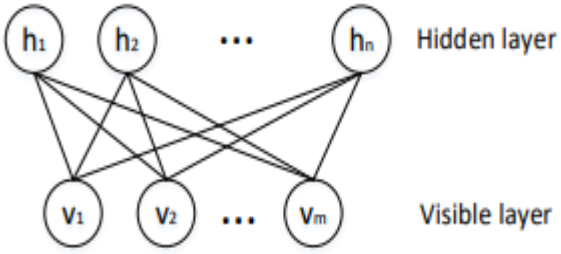

Figure 7: The structure of the RBM

c) Deep Brief Network (DBN)

As can be seen in Figure 8, the structure of a DBN, which includes numerous RBM layers and even a softmax classification layer. Training a DBN consists of two stages: pre-training as well as fine-tuning, which are both unsupervised. To begin, greedy layer-wise pretraining is used to train each RBM. Labeled data is then used to determine the weight of the softmax layer. Features extracted and classified using DBNs are employed in attack detection.

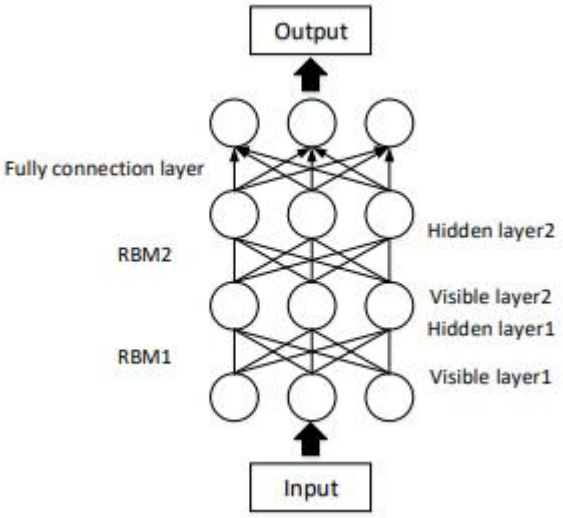

Figure 8: The structure of the DBN

d) Deep Neural Network (DNN)

DNNs with several layers may be built using a layer-wise pretraining as well as fine-tuning technique, as illustrated in Figure 9. During the unsupervised feature learning stage of DNN training, the parameters are initially learnt using unsupervised learning; the network is then tweaked using labelled data. The unsupervised feature learning step of DNNs is largely responsible for their incredible success.

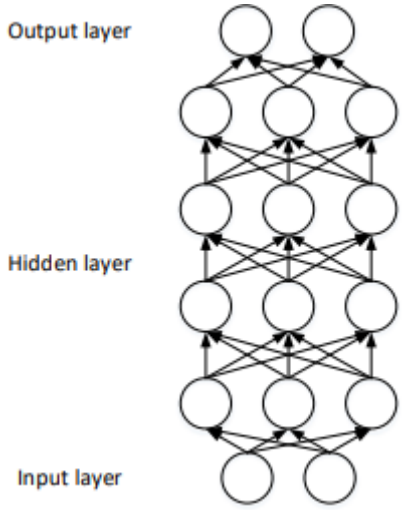

Figure 9: The structure of the DNN

e) Convolutional Neural Network (CNN)

When it comes to computer vision, CNNs have made considerable strides thanks to their ability to replicate the human visual system (HVS). As seen in Figure 10, a CNN is constructed by stacking convolutional and pooling layers in that order. For example, features may be extracted from convolutional layers as well as generalised using pooling layers. The input data for CNNs must be converted into matrices for attack detection since CNNs operate with 2D data.

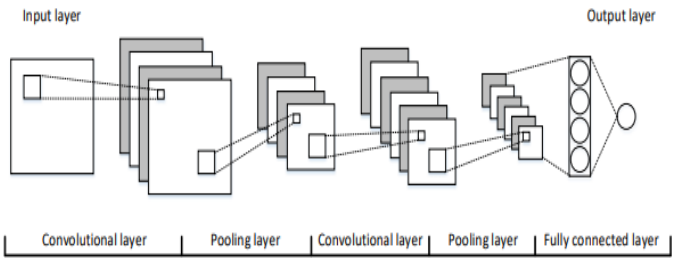

Figure 10: The structure of a CNN

f) Recurrent Neural Network (RNN)

A common use for RNNs in natural language processing is data sequencing (NLP). Evaluating a single piece of sequential data is illogical since it doesn't fit into the perspective of the whole. Every unit in an RNN gets not just the current state, but it also prior states. Figure 8 depicts the RNN's construction. In Figure 11, all of the W elements are the same. As a result of this property, RNNs often exhibit gradient vanishing or explosion. In actuality, RNNs can only process short sequences. Numerous RNN versions, including as LSTM, GRU, as well as bi-RNN, have been suggested to address the issue of long-term interdependence.

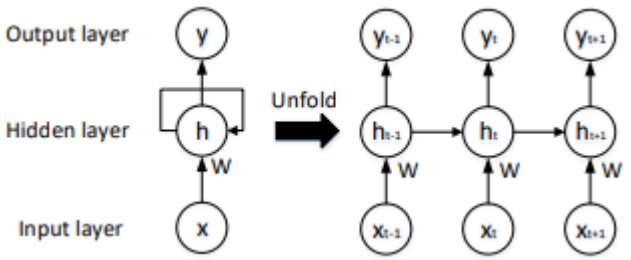

Figure 11: The structure of an RNN

g) Generative Adversarial Network (GAN)

A GAN model consists of two subnetworks, namely a generator and a discriminator. Discriminator is designed to discern between real versus synthetic data, while the generator is designed to provide synthetic data that is as close as possible to the actual data. As a result, both the generator as well as the discriminator become better as time goes on. The use of GANs to enhance attack detection data is now a prominent research area, helping to alleviate the lack of IDS datasets. Nevertheless, adversarial learning algorithms like GANs may improve model detection accuracy by including adversarial examples into the training set.

VII. LITERATURE REVIEW

In the recent decade, there have been a number of studies on intrusion detection. When it comes to dealing with large amounts of data,[16], offers a novel technique that utilises service-aware dataset separation, that either provides high adaptability to handle massive as well as rapidly expanding data on the network adaptively, as well as aids classifiers to improve classification accuracy and performance but instead speed. With the Kyoto2016 dataset, that is a well dataset for severely unbalanced data, we used multiple classification methods and settings to get the highest performance as well as compared it to current state-of the art methodologies. The results of the experiments show that our method is able to identify network traffics efficiently and simply, even when the datasets are large and unbalanced. According to our findings, current machine learning-based NIDS systems will no longer be plagued by problems related to unbalanced datasets.

In this paper[17], to NIDS, researchers employ MTL (multi-task learning) as well as oversampling algorithms. It is important to approach every terminal as a distinct job before using necessary information from other endpoints to learn each task. Oversampling is used to solve the issue of assaults being minorities. To investigate the usefulness of MTL with oversampling approaches for NIDS with minimal network attack data, the authors used the most recent UNSW-NB15 as well as CICIDS2018 datasets. They found that in various testing scenarios, they were able to obtain detection rates of over 90%.

In this article[18], SE-DAS (SMOTE and Edited Nearest Neighbors with Dual Attention SRU, SEDAS) is a network intrusion detection model that employs the SE algorithm to balance minority samples. It has been shown that a D model's ability to detect as well as identify minorities outperforms that of the classic SMOTE method on the UNSW-NB15 dataset by 0.037 percent, and the recall rate is 98.65 percent, which is greater than that of other DL techniques.

In this paper[19], with the use of WEKA, compare the performance of a number of ML algorithms, such as RF, NB, BN, Bagging, AdaBoost, and SVM, using network log data (KDD99, UNSW-NB15, and CIC-IDS2017). Researchers looked studied how modifying the number of output categories in publicly accessible network intrusion datasets affected sensitivity, TPR, FPR, AUC, as well as erroneously detected percent. Classifiers have become more efficient, thanks to the addition of highly correlated characteristics to the target classes. This is an interesting development. ML classifiers performed better with fewer target classes, as shown by the experiments. Effectiveness of classifiers may be improved by adding strongly correlated features to the output class.

This paper [20], effective intrusion detection in networks with uneven traffic, ML as well as DL are being studied. In order to address the issue of class imbalance, a new Difficult Set Sampling Technique (DSSTE) is proposed. The first step is to utilise the Edited Nearest Neighbor (ENN) method to partition the unbalanced training set into a tough and an easy set of data. Researchers undertake tests on both the old intrusion dataset NSL-KDD as well as the newer и far more thorough intrusion dataset CSE-CIC-IDS2018 in order to validate the suggested technique. RF, SVM, XGBoost, LSTM, AlexNet as well as Mini-VGGNet are some of the more traditional classification models designers employ. DSSTE beats the other 24 approaches in comparison, as shown by the experimental findings.

This paper [21], present a CNN -based intrusion detection model. Before CNN development, the network traffic is balanced using the SMOTE-ENN method. To test the model, designers utilise the NSL-KDD dataset. According to our calculations, the suggested SMOTE-ENN CNN IDS model is 83.31% accurate. Additionally, the detection rates of U2R as well as R2L attacks have been greatly enhanced. According to the findings, the new CNN IDS based on SMOTE-ENN is superior to the old IDS paradigm.

This study [22] deals with the issue of imbalance with a mixed method. Tomek connection is used in conjunction with SMOTE plus undersampling to minimise noise in this hybrid technique. A more effective intrusion detection system is achieved by combining two deep neural networks, the LSTM and the CNN. NSL-KDD and CICIDS2017 datasets are used to demonstrate the advantages of our proposed approach. Experimentation findings demonstrate that in the multiclass classification using NSLKDD dataset, the suggested framework achieved an overall accuracy and Fscore of 99.57 percent for LSTM and 99.70 percent as well as 98.27 percent for CNN. CICICD2017 has an LSTM Fscore of 98.65 percent, while its overall accuracy and Fscore on CNN are 99.85 percent and 99.98 percent, respectively.

VIII. CONCLUSION

Many current strategies were discovered, indicating that the Intrusion Detection System still need significant enhancements. With the attacker's strategy of penetrating a system as well as seeking new methods to infiltrate, the present IDS must improve in terms of detection accuracy as well as error rate in detecting assaults. Furthermore, the IDS should really be able to identify both known and unexpected assaults by increasing its intrusion detection approach. Network intrusion is now the most serious risk in network communications. The rising frequency of network assaults is a disaster for network services. Several studies have previously been undertaken in order to identify an efficient and appropriate method to prevent network intrusion as well as protect network privacy as well as security. Machine learning is a powerful analytical method for detecting unusual occurrences in network data flow. This study evaluates as well as discusses the research field for IDSs based on ML as well as DL approaches into a cohesive taxonomy, as well as highlights a gap in this critical research field.

IX. REFERENCES

[1] C. Zouhair, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A Review of Intrusion Detection Systems," *Cloud Secur.*, pp. 54–83, 2019, doi: 10.4018/978-1-5225-8176-5.ch003.

[2] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," *Proc. 2000 Int. Conf. Artif. Intell.*, 2000.

[3] S. Kumar, S. Gupta, and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," *IEEE Access*. 2021, doi: 10.1109/ACCESS.2021.3129775.

[4] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences (Switzerland)*. 2019, doi: 10.3390/app9204396.

[5] H. M. Shirazi, "Anomaly intrusion detection system using information theory, K-NN and KMC algorithms," *Aust. J. Basic Appl. Sci.*, 2009.

[6] V. Jyothsna, V. V. Rama Prasad, and K. Munivara Prasad, "A Review of Anomaly based Intrusion Detection Systems," *Int. J. Comput. Appl.*, 2011, doi: 10.5120/3399-4730.

[7] A. S. Ashoor and S. Gore, "Importance of Intrusion Detection System (IDS)," *Int. J. Sci. Eng. Res. 2011*, 2011.

[8] V. Jyothsna and K. Munivara Prasad, "Anomaly-Based Intrusion Detection System," in *Computer and Network Security*, 2020.

[9] M. Handley, V. Paxson, and C. Kreibich, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics," 2001.

[10] G. M. Gandhi, K. Appavoo, and S. K. Srivatsa, "Effective Network Intrusion Detection using Classifiers Decision Trees and Decision rules," *Int. J. Adv. Netw. Appl.*, 2010.

[11] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015, doi: 10.1109/MilCIS.2015.7348942.

[12] I. E. Faculty and M. Engineering, "ANALYSIS OF DIFFERENT MACHINE LEARNING Master thesis," pp. 1–59, 2020.

[13] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electron.*, vol. 9, no. 7, 2020, doi: 10.3390/electronics9071177.

[14] N. Farah, M. Avishek, F. Muhammad, A. Rahman, M. Rafni, and D. Md., "Application of Machine Learning Approaches in Intrusion Detection System: A Survey," *Int. J. Adv. Res. Artif. Intell.*, 2015, doi: 10.14569/ijarai.2015.040302.

[15] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, 2021, doi: 10.1007/s42979-021-00592-x.

[16] Y. Uhm and W. Pak, "Service-Aware Two-Level Partitioning for Machine Learning-Based Network Intrusion Detection with High Performance and High Scalability," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2020.3048900.

[17] L. Sun, Y. Zhou, Y. Wang, C. Zhu, and W. Zhang, "The effective methods for intrusion detection with limited network attack data: Multi-task learning and oversampling," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.3029100.

[18] X. Jiao and J. Li, "An Effective Intrusion Detection Model for Class-imbalanced Learning Based on SMOTE and Attention Mechanism," 2021, doi: 10.1109/PST52912.2021.9647756.

[19] T. Acharya, I. Khatri, A. Annamalai, and M. F. Chouikha, "Efficacy of Machine Learning-Based Classifiers for Binary and Multi-Class Network Intrusion Detection," 2021, doi: 10.1109/I2CACIS52118.2021.9495877.

[20] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2020.3048198.

[21] X. Zhang, J. Ran, and J. Mi, "An Intrusion Detection System Based on Convolutional Neural Network for Imbalanced Network Traffic," 2019, doi: 10.1109/ICCSNT47585.2019.8962490.

[22] M. Mbow, H. Koide, and K. Sakurai, "An Intrusion Detection System for Imbalanced Dataset Based on Deep Learning," 2021, doi: 10.1109/CANDAR53791.2021.00013.