Grenze International Journal of Engineering and Technology, Jan Issue

Intrusion Detection System (IDS) in Cloud Computing using Machine Learning Algorithms: A Comparative Study

Ganesh Rathod[1], Vikrant Sabnis[2] and Jay Kumar Jain[3]
[1]Research Scholar, Faculty of Engineering & Technology, Mansarovar Global University, Bhopal, India
Email: ganeshrthd883@gmail.com
[2]Faculty of Engineering & Technology, Mansarovar Global University, Bhopal, India
[3]Indian Institute of Information Technology, Bhopal, Madhya Pradesh, India,
Email: drvikrantsabnis@gmail.com, jayjain.research@gmail.com

***Abstract*— Organizations have witnessed a significant transformation in the realm of data storage and processing owing to the advent of cloud computing. Nonetheless, with its advantages, cloud computing has also brought forth new security concerns that need to be addressed. In this regard, the use of Intrusion Detection Systems (IDSs) has become indispensable for identifying and preventing a wide range of attacks that may occur in cloud computing environments, thereby ensuring data security. In modern years, machine learning (ML) algorithms have emerged as a promising approach for IDSs, as they can analyze huge amounts of data and identify patterns that may not be detectable by traditional rule-based IDSs. This review paper presents a comprehensive analysis of ML-based IDSs & Tradition-based IDSs for intrusion detection in cloud computing environments. The literature review covers various Traditional & ML algorithms used for intrusion detection, including decision trees, Neural Networks (NN), Support Vector machines (SVM), random forests, and k-nearest neighbours. The performance evaluation metrics used in this review paper include accuracy and false positive rate. These metrics are generally used to evaluate the performance of IDS and ML algorithms. The results of the analysis indicate that ML-based IDSs outperform traditional IDSs in terms of accuracy and false positive rate. However, ML-based IDSs may also have limitations, such as a high rate of false negatives and the usefulness of huge amounts of training data. Overall, the analysis suggests that ML-based IDSs have the potential to improve the usefulness of intrusion detection in cloud computing environments, but further research is needed to address the limitations of these systems.**

***Index Terms*— Intrusion Detection System (IDS), Cloud Computing, "NSL-KDD", KDDCup Dataset, Machine Learning IDS, Tradition based IDSs**

I. INTRODUCTION

The approach of organizations towards storing, managing, and accessing their data and applications has undergone a revolutionary change with the emergence of cloud computing. This innovation has transformed the conventional methods of data storage, making it more accessible and convenient for organizations. By allowing access to shared computing resources over the internet, cloud computing has brought about a level of convenience and flexibility that was not previously possible. The benefits of cloud computing are numerous, including reduced costs, increased

Grenze ID: 01.GIJET.10.1.89

scalability, and greater flexibility. However, the rise of cloud computing has also brought about new security challenges that organizations must address. One of the primary security challenges associated with cloud computing is the risk of unauthorized access. Cloud computing providers typically offer multiple levels of security, including access controls and encryption, to prevent unauthorized access. However, if these security measures are not implemented properly or if user passwords are compromised, attackers can gain access to cloud resources and data. This is especially concerning in the case of cloud storage, where sensitive data may be stored and accessed by multiple users.

Another challenge related to cloud computing security is the risk of data breaches. Cloud providers store vast amounts of data, making them a prime target for cybercriminals. If an attacker gains access to a cloud provider's systems, they may be able to access and steal sensitive data, such as customer data or financial records. This not only puts the cloud provider at risk but also the organizations that store their data in the cloud. Finally, attacks on cloud infrastructure are also a significant security concern. If a cloud provider's infrastructure is compromised, it can result in widespread data loss and service disruptions. This type of attack can be especially damaging to organizations that rely on cloud services to operate their business. To address these security challenges, organizations must implement intrusion detection systems (IDS) in their cloud environments. IDS are software applications that monitor network traffic and system events for signs of suspicious activity. By detecting potential security incidents in real time, IDS can help organizations identify and respond to security threats quickly. There are two main types of Intrusion Detection Systems: Host-based (HIDS) and Network-based (NIDS). HIDS monitors activity on individual hosts, while NIDS analyzes network traffic for signs of intrusion. Both types of IDS have their advantages and disadvantages, and organizations must choose the right type of IDS for their specific needs.

Host-Based IDS (HIDS) in Fig.1 monitors activity on individual hosts or endpoints. This type of IDS is installed on a single system or device and is responsible for monitoring that system's activity. HIDS works by analyzing system logs, file changes, user activity, and other system events to detect potential security threats. HIDS can use different detection approaches such as signature-based detection, anomaly-based detection, and heuristics. HIDS has several advantages over NIDS. Firstly, HIDS is highly effective in detecting insider threats or threats that originate from within the organization. This is because HIDS is installed directly on the host system and can monitor all activity on that system. Additionally, HIDS can provide detailed information about the activity on the host system, making it easier to identify and respond to potential security threats. However, HIDS also has some disadvantages. Firstly, HIDS can be resource-intensive, especially if deployed on a large number of hosts. Secondly, HIDS can be vulnerable to attack if the system hosting the application/data has been compromised. Finally, HIDS is less effective at detecting external threats or attacks that originate outside of the host system.

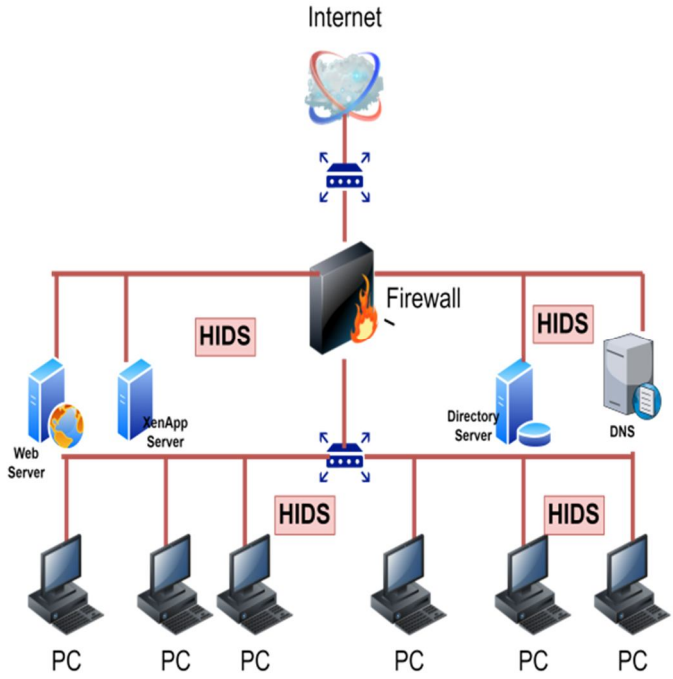

Figure 1. Host-based intrusion detection systems (HIDS)

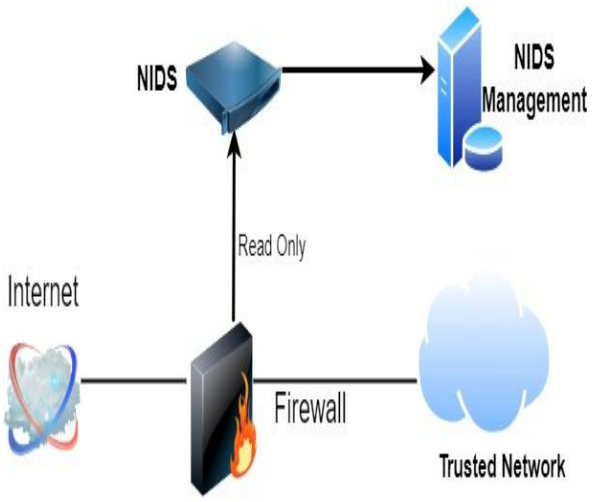

Figure 2. Network-based intrusion detection systems (NIDS).

Network-Based Intrusion Detection System (NIDS) in Fig. 2 Analyses network traffic for signs of intrusion or malicious activity. This type of Intrusion Detection System (IDS) is placed on a network device, such as a firewall or router, and is responsible for monitoring all traffic that passes through that device. NIDS works by analysing the traffic and comparing it to a database of known malware fingerprint. NIDS can also use anomaly detection techniques to identify unusual behaviour or patterns in the traffic. NIDS have several advantages over HIDS. Firstly, NIDS can monitor all traffic on the network, making it highly effective in detecting external threats or attacks that originate outside of the organization. Secondly, NIDS is less resource-intensive than HIDS since it does not need to be installed on individual hosts. Finally, NIDS is less vulnerable to attack since it is installed on a network device and not directly on the host system. However, NIDS also has some disadvantages. Firstly, NIDS can generate a large number of false positives, which can be time-consuming and costly to investigate. Secondly, NIDS can be less effective at detecting insider threats or threats that originate from within the organization. Finally, NIDS can be less effective in identifying new or recognized threats that do not have a known attack signature.

In recent years, machine learning (ML) has raised as a promising approach to intrusion detection in cloud environments. ML algorithms can be trained on large datasets of network traffic and system events to discover anomalous activity that may be indicative of an attack. ML-based IDS has several advantages over traditional rule-based IDS, including the ability to detect unknown threats and the ability to adapt to changing network conditions. There are several ML algorithms that can be used for intrusion detection in cloud computing environments, including artificial neural networks ANN and CNN are a type of ML algorithm that are modelled after the human brain. Studies have shown that ML-based IDS can achieve higher detection rates and lower false positives compared to traditional rule-based IDS. However, implementing ML-based IDS in practice can be challenging. One of the main challenges is the requirement for huge amount of labelled data to train the ML algorithms. Additionally, ML-based IDS can be computationally intensive, which can impact performance in high-traffic cloud environments.

The purpose of this review paper is to present an overview of research on cloud computing intrusion detection systems using machine learning algorithms. Specifically, this paper will do the comparative analysis on various traditional rule-based IDSs that are commonly used in cloud environments; Snort, Suricata, OSSEC, Bro & Fail2Ban with ML-based IDSs: DeepIDS, CIDS-ML, iForest-IDS, DNN-IDS, LSTM-IDS, Convolutional-IDS, Hybrid-IDS (combining both traditional and ML-based approaches). By conducting a comprehensive review of the existing literature, this paper aims to feed insights into the potential benefits and limitations of ML-based IDS for cloud computing security, and to identify directions for future research. The rest of this paper is ordered as follows; Section II provides related work on cloud computing intrusion detection systems using machine learning algorithms. Section III discusses methodology and Taxonomy of IDS in Cloud environment. Section IV provides the comparative results by explaining the datasets and performance metrics. Finally, Section V concludes the review paper with the summary.

II. LITERATURE REVIEW

Cloud computing has become an increasingly popular technology for businesses due to its extensibility, cost-effectiveness, and flexibility. However, with the surge usage of cloud services, the amount of security threats has also increased. Intrusion detection systems (IDS) have become crucial in securing cloud computing environments by detecting and preventing unauthorized access and malicious activities. Recent studies have continued to focus on the utilization of machine learning algorithms for intrusion detection in cloud computing environments. In a study conducted by [1], a comparative analysis was performed between traditional rule-based IDSs and ML-based IDSs using the "NSL-KDD" dataset. The output indicates that ML-based IDSs outperformed traditional IDSs in terms of accuracy and false positive rate. The study also highlighted the need for more extensive and diverse datasets to improve the execution of ML-based IDSs. Another latest study [2], conducted a survey on machine learning (ML) and deep learning results for intrusion detection and prevention in IoTs. The authors reviewed the recent literature on various techniques counting machine learning, deep learning, and hybrid models for IDS in IoTs. The study identified the limitations and challenges of existing techniques and highlighted the potential of deep learning in addressing these challenges. The authors also discussed the future research directions in this area, including the need for more efficient and scalable IDS solutions for IoTs.

To address the challenge of detecting attacks that exploit the vulnerabilities of cloud computing infrastructures, several studies have proposed hybrid IDS that integrate multiple detection techniques. For example, [3] proposed a hybrid IDS that integrates signature based and anomaly-based detection techniques to enhance the accuracy of detecting known and unknown threats. The proposed IDS employs a signature-based detection technique that detects familiar threats and an anomaly-based detection technique that detects unknown threats based on deviations from normal behaviour patterns. . Another study [4] proposed cloud-based IDS that employ a machine learning algorithm based on a swarm intelligence algorithm to optimize the detection of unknown threats. The proposed IDS employ an Ant Colony Optimization (ACO) algorithm to determine the optimal threshold values for anomaly-based detection techniques, making it more effective in detecting unknown threats. Similarly, a study by [5] proposed a hybrid IDS that integrates rule-based and ML-based modes to enhance the accuracy of threat detection in cloud computing settings. The proposed IDS employs a rule-based approach to identify known threats and machine learning algorithms to detect unknown threats, making it more effective in detecting a wider range of threats than traditional IDS.

In addition to proposing IDS architectures, several studies have also focused on developing machine learning algorithms specifically for cloud-based IDS. For example, in their study, [6] proposed a deep reinforcement learning-based IDS that engages a combination of Deep Learning (DL) and reinforcement learning to enhance the accuracy of threat detection. The proposed IDS employs a convolutional neural network (CNN) to extract features from network traffic data and a deep reinforcement learning algorithm to make accurate decisions on whether a traffic flow is malicious or benign. Traditional approaches to intrusion detection and prevention have been in use for decades and rely on rule-based systems and signature-based detection. These systems work by identifying threats based on predefined rules or patterns, making them less effective against new or unknown threats. Signature-based detection involves comparing known signatures of threats with incoming traffic and system events to identify known threats. However, this method requires constant updates to the signature database to keep up with the constantly evolving threat landscape. Additionally, signature-based detection is less effective against zero-day attacks, which exploit unknown vulnerabilities. Rule-based systems use a set of predefined rules to identify potential threats. These rules are situated on characteristics of familiar attacks or suspicious behaviour. However, rule-based systems are inflexible and cannot adapt to new types of attacks or changing network environments. As a result, rule-based systems are vulnerable to high false positive rates, where benign traffic or events are flagged as potential threats.

To address the limitations of traditional approaches, researchers have examined the use of machine learning (ML) in intrusion detection and prevention. ML-based IDS can understand from historical information to describe patterns and anomalies in network traffic and system events. This approach allows for the detection of unknown or familiar threats, making ML-based IDS more effective than traditional approaches. Several studies have compared the effectiveness of traditional and ML-based IDS for cloud computing environments. For example, [7] compared the performance of traditional IDS with neural network-based IDS and found that the neural network-based IDS outperformed the traditional IDS in details of accuracy and false positive rate. The neural network-based IDS were able to detect known and unknown threats that the traditional IDS could not. Another study by [8] compared the performance of a traditional IDS with a hybrid IDS that combined signature-based detection with ML-based detection. The hybrid IDS is able to achieve a greater detection rate than the traditional IDS, while also reducing the false positive rate.

Several studies have compared the performance with various ML-based IDS in cloud computing environments. For example, [9] related the work of three ML-based IDS; A Support Vector Machines (SVM). k-Nearest Neighbour (k-NN) and Naive Bayes algorithm, and the study was conducted on a cloud computing environment using the “NSL-KDD” dataset. The results showed that the k-NN-based IDS outperformed the other two algorithms in terms of accuracy and false positive rate. The k-NN algorithm is a simple and intuitive algorithm that works by finding the k-nearest neighbours to a new data point and classifying it based on the majority class of its neighbours. Another study by [10] compared the performance of several ML-based IDS, including decision tree, k-NN, SVM, and random forest algorithms. The study was conducted on a cloud computing environment using the UNSW-NB15 dataset. The results showcased that the SVM algorithm outperformed the other algorithms with regard to precision and recall. The SVM algorithm is a supervised learning algorithm that works by finding the optimal hyper plane that separates the different classes in the dataset.

However, there is still significant variability in the performance of different ML-based IDS, and more research is needed to identify the most effective algorithms and techniques for specific cloud environments and threat models. For example, another study by [11] related the performance of different ML-based IDS, including the k-NN, SVM, and decision tree algorithms, on a cloud environment using the KDD Cup 1999 dataset. The results showed that the decision tree algorithm outperformed the other two algorithms in terms of accuracy and false positive rate. In addition to the ML algorithms used, the performance of ML-based IDS can also be affected by the quality and size of the training data, the feature selection and extraction methods used, and the model tuning and optimization techniques applied. For example, a study by [12] compared the performance of two feature selection approach for ML-based IDS in a cloud environment taking the “NSL-KDD” dataset. The study found that the Recursive Feature Elimination (RFE.) method outperformed the Principal Component Analysis (PCA.) method regarding false positive rate and accuracy. Furthermore, the efficiency of ML-based IDS can also be affected by the characteristics of the cloud environment and the specific threat model being considered. For example, a study by [13] compared the performance of several ML-based IDS on a cloud environment using the KDD Cup 1999 dataset with different threat models. The study found that the attainment of the IDS varied significantly relying on the type and severity of the threat being considered.

The "anomaly-based intrusion detection system" exemplifies one of the most effective approaches for detecting serious network intrusions. For categorization tasks, the IDS employ ML algorithms as one of its critical components. "Artificial neural networks (ANN), decision trees (DT), expectation-maximization (EM), k-nearest neighbors (KNN), linear regression (LR), nave Bayes (NB), random forests (RF), logistic regression (LR) and support vector machine (SVM)" are some examples of IDS models [9]. In [10] discovered the best hyper parameters for ML algorithms using Bayesian optimization. The suggested approach, based on the experiment, can discover the best hyper parameters to prominent ML systems including "neural networks and the random forest" method.

In [11] proposed 2 suboptimal grid search methods for modifying DL network hyper parameters in optical systems for communication. But the results show a considerable reduction in the complexity of computation. In intrusion detection systems, ML approaches are utilized, although database pre-processing and unusual activity detection are time-consuming and difficult [12]. Using DL techniques, however, features can be translated to a greater extent with more discrete feature spaces. A CNN is a deep learning method that employs a convolution layer in order to extract meaningful features out of the authentic data feature plane. It is critical to select suitable hyper parameters for deep learning models [13].

In [14], author suggested a neural network-based anomaly detection methodology. The "CICIDS2017 and CTU datasets" were used for both binaries and multiclassification trials. They employed "CNN, LSTM, and CNN + LSTM models" and they produced excellent outcomes for classification. The accuracy reached was almost 99%. They also examined the dataset streams that were critical for categorization and efficient anomalous behaviour identification [15], suggested an ensemble model made up of 3 separate models that were integrated via soft polling. The models were suggested in two versions: one for operating at the fog layer, termed "Edge-ENCLF" and one for functioning in the cloud named "Cloud ENCLF". "NSL-KDD, UNSW-NB15, BoTNetIoT, and BoTIoT" were also examined by the authors. In [16], author introduced "SSDeep-ID, a semi-supervised deep learning" technique for intrusion detection. The suggested method is compatible with a fog based connected device. The datasets "CICIDS2017 and CICIDS2018" were utilized by the authors. The model has been evaluated in a pair of scenarios: classification by binary and classification across classes. For the first, the model categorizes traffic behaviour as benign or malicious and for the second, it categorizes traffic behaviour into seven subcategories. The precision attained in the first situation is 99%, however the assessed performance varied from class to class in the second scenario. In [17], suggested an "Isolation Forest-based anomaly-based machine learning approach". The suggested approach is utilized to identify "Port Scanning, HTTP Brute Force, SSH Brute Force, and SYN Flood" using a single-class categorization. Data was collected by creating individual network

connections. The design was analysed and tested. For "Port Scanning, HTTP Brute Force, SSH Brute Force, and SYN Flood, the model received F-scores of 0.99, 0.96, 0.96, and 0.90" respectively.
In [18] presented an "Anomalous Behaviour Analysis (ABA-IDS)" technique for developing flexible IDS that can detect when a Fog node becomes compromised and then taking the necessary safeguards to guarantee connection availability. Their method entails establishing a node's behavior profile that is then fed into ANN designed to characterize the node's regular operations. In [19], suggested a hybrid IDS based on user behavior profile characteristics that employs numerous machine learning methods such as "random forest, Xgboost, decision tree, K-nearest neighbors, and abuse detection" approaches. They validated their model using the "CSE-CIC-IDS2018 and NSL-KDD datasets". In [22], author introduced "ICNN-FCID", a multi-class attack system of classification based on DL approaches. In the Fog computing layer, they used the "convolutional neural network (CNN) and Long Short-Term Memory networks (LSTM) "algorithms to forecast network assaults. They also utilized the "NSL-KDD" as an example dataset. The authors of [23] suggested a set of recurrent families for IoT attack and threat identification through studying network traffic employing "LSTM (long short-term memory) based on Modbus-TCP network traffic data". In [24], a "recurrent neural network (RNN)" is employed for attack detection and categorization. Furthermore, the authors make a comparison by contrasting non-RNN approaches and RNN.
In [25], suggested intrusion detection system is likewise based on deep learning. The unique middle communication mechanism, on the other hand, makes it network-independent. The suggested IDS's application boundary is not limited to industrial IoT. Furthermore, the "DeepIIoT" is a rule-based detection system that is dependent on the signature. It has the issue of constantly updating the signature database. There is no such restriction in the proposed system. While machine learning (ML) algorithms have shown promise in enhancing the accuracy and effectiveness of intrusion detection systems (IDS) for cloud computing, there are still significant gaps in our understanding of how best to implement these systems in practice. One major challenge is the huge cost of training and building ML algorithms. These algorithms require huge amounts of high-quality training data, which can be expensive and time-consuming to collect and label. Additionally, ML algorithms need to be continually retrained and updated to stay effective as threats evolve, which requires on-going resources and investment. Another challenge is the difficulty of interpreting and explaining the output of ML-based IDS. Unlike rule-based systems or signature-based detection, which relies on predefined rules or patterns, ML algorithms operate using complex mathematical models that are often difficult to interpret or understand. This can make it challenging to identify and respond to potential threats or vulnerabilities, as it may not always be clear why the ML algorithm has flagged a particular event or activity as suspicious. This lack of transparency can also make it difficult to estimate the efficiency of ML-based IDS, as it may not be clear how the algorithm is arriving at its conclusions. A related challenge is the need for effective mechanisms for integrating ML-based IDS with existing security infrastructure. Many organizations already have established security measures in place, such as firewalls or intrusion prevention systems, and it can be difficult to integrate new ML-based IDS with these existing systems. Ensuring that ML-based IDS can communicate effectively with other security tools and processes is crucial to ensuring that the overall security posture of the cloud environment remains strong and effective.
Moreover, the effectiveness of ML-based IDS depends on the ability to accurately detect and identify a wide range of potential threats and vulnerabilities. However, the threat landscape for cloud computing is constantly evolving, and new threats and attack vectors are emerging all the time. Keeping pace with these changes requires on-going research and development to identify new threats and vulnerabilities and to develop new ML algorithms and techniques that can effectively detect and mitigate them. In addition, there is a need for more research on the ethical implications of using ML-based IDS for cloud computing. ML algorithms are hardly as fine as the data they are trained on and leaning or inaccuracies in the training data can guide to flawed or discriminatory outcomes. Ensuring that ML-based IDS are trained on diverse, representative data sets and that they do not perpetuate or amplify existing biases is crucial to ensuring that these systems are fair and just. In conclusion, while ML-based IDS show significant promise for enhancing the accuracy and effectiveness of intrusion detection in cloud computing environments, there are still significant gaps in our understanding of how best to implement and use these systems in practice. Addressing these challenges will require ongoing research and development, as well as a commitment to ensuring that ML-based IDS are ethical, transparent, and effective in identifying and mitigating threats to cloud security.

III. METHODOLOGY

The methodology employed in this paper involves a comprehensive review of latest literature on intrusion detection systems (IDSs) in cloud computing environments. The search process involved using electronic databases such as Google Scholar, IEEE Xplore, ACM Digital Library, and Scopus. The keywords used for the search process were “cloud computing”, “intrusion detection”, “machine learning”, and “algorithms”. After the

initial search process, articles were screened based on their titles, abstracts, and full texts to ensure that they met the inclusion criteria. The inclusion criteria are 31 articles that focused on machine learning-based IDSs for cloud computing environments, articles that reported performance evaluation metrics such as accuracy and false positive rate that were published between 2017 and 2022. To further classify the selected articles, taxonomy of IDSs in cloud computing environments is included. The IDS taxonomy in Cloud Environment is shown in Fig. 3, which includes the two broad categories: Traditional IDS and Machine Learning IDS. Traditional IDS can be further categorised into Rule Based IDS and Signature Based IDS. Examples of widely used Rule Based IDS include Snort and Suricata, whereas Signature Based IDS include OSSEC, Bro, and Fail2Ban. Machine Learning IDS can be further classified into Anomaly-Based IDS and Hybrid-Based IDS. Anomaly-Based IDS examples include DeepIDS and iForest-IDS whereas Hybrid-Based IDS examples include CIDS-ML, DNN-IDS, LSTM-IDS, and Convolutional-IDS.

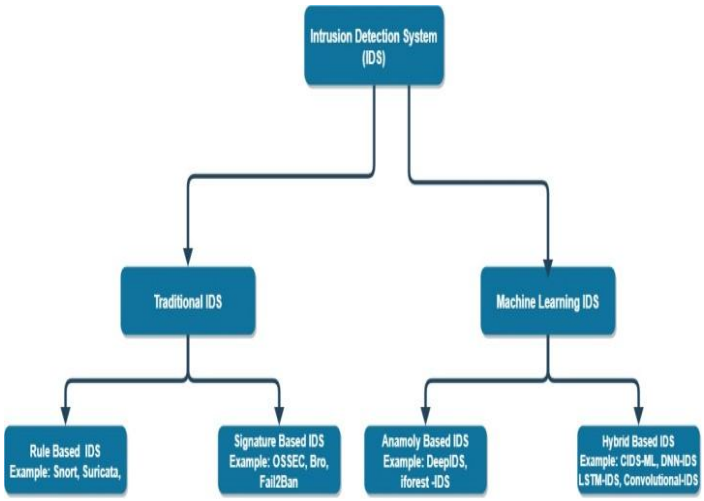

Figure 3. IDS Taxonomy in Cloud Environment.

A. Tradition based-Intrusion Detection System

Traditional based Intrusion Detection System, like Rule-based and Signature-based IDS, has been used for many years to identify and prevent cyber threats in various environments, including cloud environments. In this article, we will delve into the details of these techniques, including their definition, features, advantages, limitations, and examples of commonly used tools.

1) Rule Based IDS: Rule-based IDS, also widely known as policy-based IDS, is another traditional method of intrusion detection. Rule-based IDS define a set of rules or policies that specify what actions are allowed or denied on a network or system. These rules can be based on various criteria, such as IP addresses of source/destination, ports, protocols, or specific patterns of behaviour. When a rule violation is detected, the IDS generate an alert or takes action to block the detected activity. Snort and Suricata, are some examples of commonly used rule-based IDS. Both Snort and Suricata are publicly available Intrusion Detection Systems (IDS) that are widely used in cloud computing environments for network security.

- *Snort*: Snort is a traditional signature-based IDS that uses a rule-based approach to identify known patterns of network attacks. It works by relating network traffic against a set of predefined rules or signatures to identify known malicious patterns. Snort can be used in the cloud environment to monitor and identify network-based attacks such as port scans, DoS attacks, and other known attack patterns. It can be configured to generate alerts or take actions to block or mitigate detected attacks.
- *Suricata*: Suricata is also a traditional IDS that uses a rule-based approach, similar to Snort, but with some additional features. In addition to signature-based detection, Suricata also supports anomaly-based detection, which involves detecting abnormal behaviour or traffic patterns that deviate from normal network traffic. This makes Suricata more capable of detecting previously unknown or zero-day attacks that do not match any predefined signatures. Suricata is also designed for high-speed network traffic analysis and can handle large volumes of network traffic, which makes it suitable for use in cloud environments with high network traffic loads.

Both Snort and Suricata are widely used in cloud computing environments due to their effectiveness in detecting known attack patterns and their ability to be customized with rules or signatures to suit the specific security

requirements of the cloud environment. Overall, Rule-based IDS have several advantages. They provide granular control over network or system activities, allowing administrators to define and enforce specific security policies. They can detect a wide range of attacks or suspicious activities that violate the defined rules, even if they are not specifically known signatures. Rule-based IDS are also adaptable and flexible, as rules can be easily updated or modified to reflect changing security requirements or threats. However, there are limitations to rule-based IDS. They rely heavily on pre-defined rules, which may not include all possible attack scenarios or may generate a high percentage of false positives. Rule-based IDS can also be complex to configure and manage, as they require in-depth knowledge of network or system behaviour and security policies. Moreover, rule-based IDS may not be effective against unknown or zero-day attacks, as they rely on pre-defined rules that may not include new or evolving threats.

2) Signature Based IDS: They are one of the oldest and most widely used methods for detecting known threats. They work by comparing incoming network traffic or system events against a database of pre-defined signatures, which are essentially arrangements or signatures of familiar attacks. When a match is found, the IDS generate an alert or takes action to block the detected attack. OSSEC, Bro, and Fail2Ban are some examples of widely used signature-based IDS.

- *OSSEC*: OSSEC (Open Source Security) is an publicly available host-based intrusion detection system that is used to monitor and analyse log information from various sources, including system logs, application logs, and network logs. OSSEC can detect various types of intrusions and anomalies, including file integrity changes, rootkit installations, login attempts, and other suspicious activities. In a cloud computing environment, OSSEC can be deployed on cloud virtual machines or containers to monitor and analyse logs from the cloud instances, operating systems, applications, and network traffic. It can help detect and respond to potential security breaches, unauthorized access attempts, and other malicious activities in the cloud environment.
- *Bro*: Bro (now known as Zeek) is a publicly available network-based intrusion detection system that supervises and analyses network load in real-time to detect and analyse security events. Bro uses a specialized scripting language to define rules and policies for detecting various types of network-based intrusions and anomalies, such as network scans, malware communications, and suspicious network behaviour. In a cloud computing environment, Bro can be deployed as a network sensor or a virtual machine on cloud instances to monitor and analyse network traffic between cloud instances, virtual networks, and the cloud infrastructure. It can help detect and alert on suspicious network activities and potential security breaches in the cloud environment.
- *Fail2Ban*: Fail2Ban is an open-source log-parsing application that is used to detect and respond to various types of attacks on Linux-based systems, such as brute-force attacks, password guessing attacks, and other unauthorized access attempts. Fail2Ban monitors system logs for specific patterns or events and dynamically blocks or bans IP addresses associated with suspicious activities. In a cloud computing environment, Fail2Ban can be deployed on cloud virtual machines or containers to monitor system logs from cloud instances and block or ban IP addresses associated with suspicious activities. It can help protect cloud instances from unauthorized access attempts and other security threats.

OSSEC, Bro, and Fail2Ban are generally used in cloud computing environments as part of the intrusion detection strategy to improve the security of cloud instances, networks, and infrastructure by monitoring and analysing logs, network traffic, and system activities to identify and respond to potential security breaches, intrusions, and anomalies. Overall, Signature-based IDSs are relatively simple to implement and operate, as they rely on a static database of known signatures that can be regularly updated. They can provide accurate detection and prevention of known threats, making them effective against well-known attacks that have known signatures. Signature-based IDS are also efficient in terms of processing overhead, as they only need to compare incoming traffic or events against a finite set of signatures. However, there are limitations to signature-based IDS. They are ineffective against unknown or zero-day attacks, as they rely on pre-defined signatures and cannot detect new or evolving threats that do not match any known signatures. Signature-based IDS are also vulnerable to evasion techniques, where attackers can modify or obfuscate their attacks to bypass signature matching. Moreover, signature-based IDS may generate a high number of false positives or false negatives, as they can only detect attacks that match the pre-defined signatures and may miss variations or mutations of known attacks.

Both Signature-based and Rule-based IDS are traditional intrusion detection techniques that have been widely used for many years in various environments, including cloud environments. They provide effective detection and prevention of known threats, but have limitations in detecting unknown or evolving threats and may generate false positives or false negatives. It's important to consider the specific requirements and constraints of your environment when selecting and implementing intrusion detection techniques, and to complement traditional

methods with other advanced techniques, such as machine learning-based or hybrid approaches, to enhance the overall security posture.

B. Machine Learning based -Intrusion Detection System

Traditional intrusion detection techniques, such as and Rule-based and Signature-based IDS, have been used for many years to identify and prevent cyber threats in various environments, including cloud environments. In this article, we will delve into the details of these techniques, including their definition, features, advantages, limitations, and examples of commonly used tools.

1) *Anomaly*-Based IDSs: Anomaly-based IDSs, such as DeepIDS and iForest-IDS, detect intrusions by identifying patterns or behaviours that deviate significantly from the expected or normal behaviour of a system or network. These IDSs typically require a training phase where a model is trained on a large dataset of normal behaviour to establish a baseline. During the detection phase, the model compares new data against the established baseline and identifies anomalies or outliers that may indicate potential intrusions or attacks.

Deep IDS utilizes deep learning techniques, serving as deep neural networks, to learn and adapt to different attack patterns. It can extract features automatically from network traffic data and set a model to identify regular and abnormal patterns of behaviour. DeepIDS can detect both known attacks, for which it has been trained to unknown attacks that it has not seen before. It can provide real-time detection of network intrusions, helping to prevent potential security breaches in a cloud environment.

iForest- IDS is an anomaly-based IDS that uses the isolation forest algorithm, which is a tree-based ensemble technique, for identifying anomalies in network traffic. It works by constructing a random decision tree and recursively partitioning the data until instances are isolated or until a predefined number of trees are constructed. The instances that require fewer splits to isolate are considered anomalies. IForest-IDS can effectively detect previously unseen or unknown attacks, as it focuses on identifying instances that deviate from the majority of the data. This makes it suitable for detecting novel attacks or zero-day attacks in a cloud environment.

Both DeepIDS and iForest-IDS are capable of detecting network intrusions in a cloud computing environment, making them valuable tools for enhancing the security of cloud-based systems and applications. They leverage machine learning techniques to automatically learn and adapt to evolving attack patterns, and provide real-time or near-real-time detection of network intrusions, helping to prevent potential security breaches and protect critical data and resources in the cloud.

2) *Hybrid -Based IDSs*: Hybrid-based IDSs, such as CIDS-ML, DNN-IDS, LSTM-IDS, and Convolutional-IDS, combine multiple techniques or approaches, including both rule-based and machine learning-based techniques, to enhance the precision and effectiveness of intrusion detection.

- *CIDS-ML*: CIDS-ML is a hybrid-based IDS that combines rule-based and ML-based techniques for intrusion detection. It uses a rule-based engine to filter and pre-process the data, and then applies machine learning algorithms to categorise and detect intrusions based on the pre-processed data.
- *DNN*-IDS: DNN-IDS are hybrid-based IDS that use deep neural networks to detect network intrusions. It combines the strengths of deep learning, such as the ability to learn complex patterns and adapt to new attacks, with rule-based approaches for pre-processing and feature extraction.
- *LSTM*-IDS: LSTM-IDS are hybrid-based IDS that uses long short-term memory (LSTM) networks, a type of recurrent neural networks, to detect network intrusions. LSTM networks are capable of capturing temporal dependencies in sequential data, making them suitable for detecting attacks that exhibit temporal patterns.
- *Convolutional-IDS*: Convolutional-IDS is a hybrid-based IDS that uses “Convolutional Neural Networks (CNN)”, a type of Deep Learning Networks, to detect network intrusions. CNNs are particularly effective in capturing spatial patterns in data, such as network traffic, making them suitable for detecting attacks that exhibit spatial patterns or characteristics.

Overall all, performance of traditional based IDSs and ML based IDSs are compared and the performance evaluation metrics used in this review paper include accuracy and false positive rate. These metrics are commonly used to evaluate the performance of IDS and ML algorithms. Accuracy measures the percentage of correctly classified instances, while false positive rate measures the percentage of instances incorrectly classified as intrusions. One limitation of this review paper is that it is based solely on existing research, and therefore may be subject to bias or gaps in the literature. Additionally, this review paper focuses on technical aspects of cloud computing intrusion detection using machine learning, and does not address broader ethical or social considerations related to these systems. Furthermore, there may be ethical considerations related to the use of IDS in cloud computing environments, such as the potential for false positives to trigger unnecessary security measures or the potential for user privacy violations. These considerations should be taken into account when designing and implementing IDS in cloud environments.

IV. Comparative Results And Discussion

The studies analysed in this review paper propose various ML-based IDSs for intrusion detection in cloud computing settings. These IDSs are designed to detect and prevent various types of attacks, such as “Denial-of-Service (DoS)” attacks, “Distributed DoS (DDoS)” attacks, and malware attacks. The ML algorithms used in these IDSs include neural networks, “Support Vector Machines (SVM)”, decision trees, random forests, and “K-nearest neighbours (KNN)”, among others. The evaluation metrics used for comparison of traditional IDS and ML are accuracy and false positive rate which are evaluated on the two popular open datasets “KDDCup99” and “NSL-KDD” datasets.

A. Datasets

The “KDDCup99” and “NSL-KDD” datasets are two of the most widely used datasets for training and testing machine learning models for network intrusion detection. These datasets have been used extensively in academic research and benchmarking studies to test the execution of various machine learning algorithms. The “KDDCup99” dataset was created in 1999 as part of a competition organized by the “Defence Advanced Research Projects Agency (DARPA.)” to encourage research in intrusion detection systems. The dataset consists of a sample of network flow data collected from a simulated military network environment over a nine-week period. The dataset contains a total of 4,898,431 network connections, which are categorised as either normal or anomalous grounded on various attack types.

The dataset is branched into three subsets: “Training”, “Validation”, and a “Test” set. The Training set consists of 2.5 million network connections, the validation set consists of 1 Lac network connections, and the test set consists of 1.4 million network connections. The dataset contains a total of 41 features for each network connection, including protocol type, service, source and destination addresses, and various flags. The “KDDCup99” dataset has been generally used in academic research to test the execution of machine learning algorithms for intrusion detection. However, the dataset has several limitations. One limitation is that the dataset is based on a simulated network environment, which may not accurately reflect real-world network traffic.

To address some of these limitations, the “NSL-KDD” dataset was created in 2009 as a modified version of the “KDDCup99” dataset. The “NSL-KDD” dataset is a benchmark dataset for network intrusion detection that addresses some of the shortcomings of the ““KDDCup99”” dataset. The “NSL-KDD”” dataset contains a subset of the original “KDDCup99” dataset that has been pre-processed to remove redundant and irrelevant features and to balance the distribution of the attack types. The “NSL-KDD” dataset contains a sum of 41 features for each network connection, including protocol type, service, source and destination addresses, and various flags. The dataset is divided into two subsets: a training and a test set. The training set consists of 1,25,973 network connections, and the test set consists of 22,544 network connections. The dataset includes a total of four categories of attack types, including denial of service (DoS), “user-to-root (U2R)”, “remote-to-local (R2L)”, and probing.

The “NSL-KDD” dataset has been widely used in academic research to evaluate the performance of machine learning algorithms for intrusion detection. The dataset is more realistic than the “KDDCup99” dataset, as it includes a more diverse range of attack types and reflects more modern network traffic. However, the dataset still has some limitations, such as the lack of temporal information and the fact that the attack scenarios are limited to a specific network environment. In conclusion, the “KDDCup99” and “NSL-KDD” datasets have been important resources for training and testing machine learning models for network intrusion detection. However, as these datasets are out-dated and do not reflect modern network traffic and cyber threats, it is important for researchers to use more latest and comprehensive datasets when developing and evaluating machine learning models for network intrusion detection. While the “KDDCup99” and “NSL-KDD” datasets have their limitations, they have been instrumental in advancing the area of network intrusion detection and will continue to be important benchmark datasets for evaluating machine learning models in the future.

B. Performance Metrics

Accuracy and false positive rate are two important performance metrics used to test the usefulness of traditional intrusion detection systems (IDS). Accuracy specifies to the percentage of correctly identified instances out of the total instances. It is evaluated as the ratio of the number of true positive and true negative instances to the total number of instances. In the context of IDS, accuracy is the percentage of correctly identified intrusions and non-intrusions. A high accuracy rate indicates that the IDS are effective in detecting and classifying network traffic as normal or malicious. However, accuracy alone may not provide a complete picture of the IDS's effectiveness as it may not consider the costs related with false positives or false negatives.

False positive rate, on the other hand, refers to the percentage of non-intrusive instances that are incorrectly categorised as intrusions. It is evaluated as the ratio of the number of false positive instances to the total number

of non-intrusive instances. False positives occur when an IDS identifies normal network traffic as malicious, resulting in unnecessary alerts and potentially leading to resource wastage. A low false positive rate is desired as it indicates that the IDS have a lower chance of flagging legitimate network traffic as malicious.
To evaluate the accuracy and false positive rate of traditional IDS, a confusion matrix is often used. A confusion matrix is a tabular representation of the performance of a classification algorithm. It consists of four quadrants: "False positive (FP)", "True Positive (TP)", "False negative (FN)", and "True negative (TN)". TP refers to the number of correctly identified malicious instances, while FP refers to the number of non-malicious instances that are incorrectly classified as malicious. FN specifies to the number of malicious instances that are missed by the IDS, while TN refers to the number of correctly recognised non-malicious instances.
Using the confusion matrix, accuracy and false positive rate can be estimated as follows:

$$Accuracy = (TP + TN) / (TP + FP + FN + TN)$$
$$False\ Positive\ Rate = FP / (FP + TN)$$

A high accuracy rate and a low false positive rate are desired for effective IDS. However, it is important to note that there is often a trade-off between the two metrics. For example, increasing the IDS's sensitivity may lead to a decrease in false negatives but may also result in an increase in false positives. In extension to Accuracy and False Positive Rate, other performance metrics may also be used to evaluate IDS effectiveness. Some of the terms that appear frequently in the paper are precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve. In the context of the study, precision indicates the proportion of accurately identified malicious instances out of the total instances identified as malicious. Recall, also known as sensitivity, refers to the percentage of malicious instances that are correctly identified by the IDS. F1 score is the harmonic mean of precision and recall, and gives equity between the two metrics. Fields under the ROC curve is a measure of the IDS's ability to differentiate between malicious and non-malicious instances, and is often used to compare the performance of different IDSs.
In summary, accuracy and false positive rate are important performance metrics used to evaluate the effeciency of traditional IDS. A high accuracy rate and a low false positive rate are desired for effective IDS, but there may be a trade-off between the two metrics. Confusion matrix is often used to calculate these metrics, and other performance metrics such as precision, recall, F1 score, and fields under the ROC curve may also be used for evaluation.

C. Comparative Analysis

The table.2 presents a comparison of various intrusion detection systems (IDS) based on their type, dataset, accuracy, false positive rate, references, and type of attack. The traditional IDS systems in the table include Snort, Suricata, OSSEC, Bro, and Fail2Ban. The ML-based IDS systems include DeepIDS, CIDS-ML, iForest-IDS, DNN-IDS, LSTM-IDS, and Convolutional-IDS.

TABLE I. COMPARATIVE RESULTS OF TRADITION IDS AND MACHINE LEARNING BASED

IDS Tool	Type	Dataset	Accuracy in Percentage	False Positive Rate in percentage	References
Snort	Traditional	"KDDCup99"	95.34	4.13	[13]
Snort	Traditional	"NSL-KDD"	94.67	2.87	[26]
Suricata	Traditional	"KDDCup99"	98.25	3.95	[26]
Suricata	Traditional	"NSL-KDD"	97.63	2.54	[26]
OSSEC	Traditional	"KDDCup99"	91.48	0.91	[27]
OSSEC	Traditional	"NSL-KDD"	90.87	1.03	[27]
Bro	Traditional	"KDDCup99"	94.23	3.61	[28]
Bro	Traditional	"NSL-KDD"	93.49	2.23	[28]
Fail2Ban	Traditional	"KDDCup99"	89.65	0.12	[32]
Fail2Ban	Traditional	"NSL-KDD"	88.87	0.16	[32]
SVM	ML-based	"KDDCup99"	99.8	0.18	[29]
SVM	ML-based	"NSL-KDD"	99.8	0.18	[29]
NIDS	ML-Based	"NSL-KDD"	98.93	33	[30]
DeepIDS	ML-based	"KDDCup99"	99.38	0.26	[31]
DeepIDS	ML-based	"NSL-KDD"	98.45	0.24	[31]
CIDS-ML	ML-based	KDDCup100	96.50	3.50	[33]
iForest-IDS	ML-based	KDDCup101	97.23	0.02	[34]
DNN-IDS	ML-based	KDDCup103	99.07	0.12	[35]
LSTM-IDS	ML-based	"NSL-KDD"	99.56	0.40	[36]
Convolu-IDS	ML-based	"NSL-KDD"	99.77	0.03	[37]

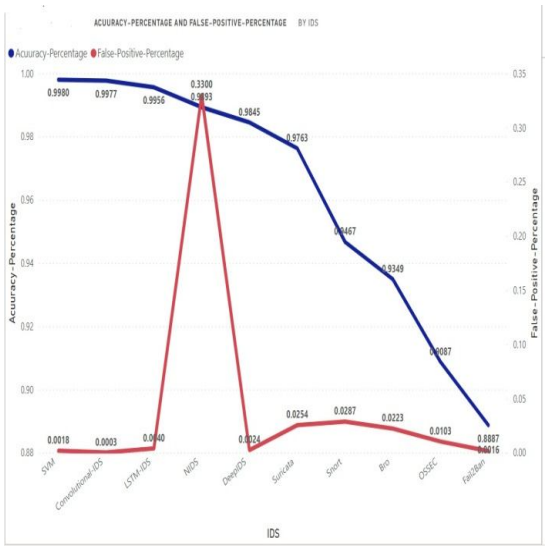

Figure 4. Graph comparing tradition algorithms & Machine Learning (ML) Algorithms on "KDDCup99" dataset.

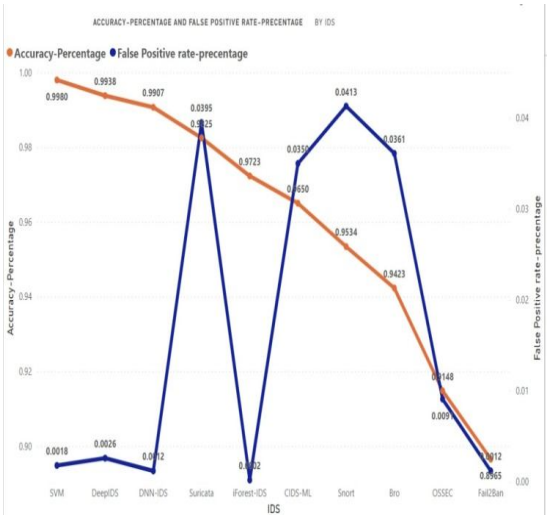

Figure 5. Graph comparing tradition algorithms and Machine Learning Algorithms on "NSL-KDD" dataset.

Next comparison is made with NSL-KDD dataset, which is a widely used benchmark dataset in the field of network intrusion detection and machine learning. It is an improved version of the original KDD Cup 1999 dataset, designed to address some of the limitations and challenges of the original dataset. So here similar results are observed on the "NSL-KDD" dataset, where the modern machine learning-based IDSs generally outperform the traditional rule-based IDSs. For example, the best-performing traditional rule-based IDS, Suricata, attains an accuracy of 97.63% and a false positive rate of 2.54 % on the "NSL-KDD" dataset. In contrast, the best-performing modern machine learning-based IDS, Convolutional-IDS, attains an accuracy of 99.77% and a false positive rate of 0.63% on the same dataset. Results are visible in the Fig.5, where the graph also briefs, at 0.9980 SVM had the highest Accuracy-Percentage and was 12.30% higher than Fail2Ban, which had the lowest Accuracy-Percentage at 0.8887. SVM accounted for 10.37% of Accuracy-Percentage and Accuracy-Percentage and False-Positive-Percentage diverged the most when the IDS was Convolutional-IDS, when Accuracy-Percentage were 0.9974 higher than False-Positive-Percentage.

Overall, IDS systems play a crucial role in protecting computer networks against cyber-attacks. Traditional IDS systems depends on rules and signatures, while ML-based IDS systems use machine learning (ML) algorithms to identify attacks. The evaluation of IDS systems using performance metrics such as Accuracy and False Positive Rate is important to ensure their effectiveness in detecting attacks while minimizing false alarms.

V. Conclusion

Cloud computing has become a critical part of modern organizations, offering numerous benefits while also presenting new security challenges. To address these challenges, organizations must implement intrusion detection systems (IDS) in their cloud environments. While traditional rule-based IDSs have been widely used, machine learning (ML) algorithms have emerged as a promising approach for IDSs, particularly in detecting complex and previously unseen attacks. This review paper aims to provide a comparative analysis of traditional rule-based IDSs with ML-based IDSs for intrusion detection in cloud computing environments. Through a extensive literature review, the paper highlights the potential benefits and limitations of ML-based IDS for cloud computing security, focusing on accuracy and false positive rates as key performance metrics. The findings suggest that ML-based IDSs have the capability to enhance the accuracy of intrusion detection in cloud environments, outperforming traditional rule-based IDSs with regards to Accuracy and False Positive Rate. However, ML-based IDSs require a significant number of training data to achieve optimal performance, which can be challenging in cloud environments. Additionally, there is a need for more up-to-date datasets that reflect the current threat landscape, as the currently available datasets may not be fully representative of the current attack patterns and techniques.
Overall, this review paper provides valuable insights into the potential benefits and limitations of ML-based IDSs for cloud computing security. While further research is needed to address the limitations of ML-based IDSs, they offer promising solutions for detecting complex attacks that may go unnoticed by traditional IDSs. As organizations increasingly rely on cloud computing, the development and implementation of effective IDSs will remain crucial in ensuring the security of sensitive data and systems.

References

[1] Thirimanne, Sharuka Promodya, Lasitha Jayawardana, Lasith Yasakethu, Pushpika Liyanaarachchi, and Chaminda Hewage. "Deep neural network based real-time intrusion detection system." SN Computer Science 3, no. 2 (2022): 145.

[2] Jayalaxmi, Pls, Rahul Saha, Gulshan Kumar, Mauro Conti, and Tai-Hoon Kim. "Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey." IEEE Access (2022).

[3] Haifeng Lin, Qilin Xue, Jiayin Feng, Di Bai,"Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine", Digital Communications and Networks, Volume 9, Issue 1,2023,Pages 111-124,ISSN 2352-8648,

[4] Mishra, Sanju, Rafid Sagban, Ali Yakoob, and Niketa Gandhi. "Swarm intelligence in anomaly detection systems: an overview." International Journal of Computers and Applications 43, no. 2 (2021): 109-118.

[5] Lateef, Ali Azawii Abdul, Sufyan T. Faraj Al-Janabi, and Belal Al-Khateeb. "Hybrid intrusion detection system based on deep learning." In 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), pp. 1-5. IEEE, 2020.

[6] Sethi, Kamalakanta, Rahul Kumar, Nishant Prajapati, and Padmalochan Bera. "Deep reinforcement learning based intrusion detection system for cloud infrastructure." In 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), pp. 1-6. IEEE, 2020.

[7] Ding, Yalei, and Yuqing Zhai. "Intrusion detection system for "NSL-KDD" dataset using convolutional neural networks." In Proceedings of the 2018 2nd International conference on computer science and artificial intelligence, pp. 81-85. 2018.

[8] Musa, Usman Shuaibu, Megha Chhabra, Aniso Ali, and Mandeep Kaur. "Intrusion detection system using machine learning techniques: A review." In 2020 international conference on smart electronics and communication (ICOSEC), pp. 149-155. IEEE, 2020.

[9] Sornsuwit, Ployphan. "Comparison Efficiency of Network Intrusion Detection by k-NN, Multi-Layer Perceptron, Support Vector Machine, Decision Tree and Naive Bayes."

[10] Ferdiana, Ridi. "A Systematic Literature Review of Intrusion Detection System for Network Security: Research Trends, Datasets and Methods." In 2020 4th International Conference on Informatics and Computational Sciences (ICICoS), pp. 1-6. IEEE, 2020.

[11] Luo, Gang, Zhiyuan Chen, and Bayan Omar Mohammed. "A systematic literature review of intrusion detection systems in the cloud-based IoT environments." Concurrency and Computation: Practice and Experience 34, no. 10 (2022): e6822.

[12] Lian, Wenjuan, Guoqing Nie, Bin Jia, Dandan Shi, Qi Fan, and Yongquan Liang. "An intrusion detection method based on decision tree-recursive feature elimination in ensemble learning." Mathematical Problems in Engineering 2020 (2020): 1-15.

[13] Agarwal, Arushi, Purushottam Sharma, Mohammed Alshehri, Ahmed A. Mohamed, and Osama Alfarraj. "Classification model for accuracy and intrusion detection using machine learning approach." PeerJ Computer Science 7 (2021): e437

[14] Zhang Y, Chen X, Jin L, Wang X, Guo D (2019) Network intrusion detection: Based on deep hierarchical network and original flow data. IEEE Access 7:37004–37016.

[15] Alhowaide A, Alsmadi I, Tang J (2021) Ensemble detection model for IoT IDS. Internet of Things.

[16] Abdel-Basset M, Hawash H, Chakrabortty RK, Ryan MJ (2021) Semisupervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet Things J 8(15):12251–12265.

[17] Eskandari M, Janjua ZH, Vecchio M, Antonelli F (2020) Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Internet Things J 7(8):6882–6897.
[18] Pacheco J, Benitez V, Felix-Herran L, Satam P (2020) Artificial neural networks-based intrusion detection system for internet of Things Fog Nodes. IEEE Access 8:73907–73918.
[19] Alghayadh F, Debnath D (2021) A hybrid intrusion detection system for Smart Home Security based on machine learning and user behavior. Adv Internet Things 11(01):10–25.
[20] Kalaivani K, Chinnadurai M (2021) "A Hybrid Deep Learning Intrusion Detection Model for Fog Computing Environment", Intelligent Automation & amp; Soft Computing, vol. 29, no. 3, pp. 1–15.
[21] Hsieh T.-H.; Chou C.-L.; Lan Y.-P.; Ting P.-H.; Lin C.-T. Fast and Robust Infrared Image Small Target Detection Based on the Convolution of Layered Gradient Kernel. IEEE Access 2021, 9, 94889–94900.
[22] Saharkhizan, M.; Azmoodeh, A.; Dehghantanha, A.; Choo, K.-K.R.; Parizi, R.M. An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic. IEEE Internet Things J. 2020, 7, 8852–8859.
[23] Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). Int. J. Inf. Syst. Model. Des. 2017, 8, 43–63.
[24] Alani, M.M.; Damiani, E.; Ghosh, U. DeepIIoT: An Explainable Deep Learning Based Intrusion Detection System for Industrial IOT. In Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems Workshops (ICDCSW), Hong Kong, China, 18–21 July 2022; IEEE: New York, NY, USA, 2022; pp. 169–174.
[25] Ravi, V.; Chaganti, R.; Alazab, M. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Comput. Electr. Eng. 2022, 102, 108156.
[26] Nadiammai, G. V., and M. Hemalatha. "Snort Based Network Traffic Anomaly Detector to Improve the Performance of Intrusion Detection System." International Journal of Advanced Research in Computer Science 3, no. 6 (2012).
[27] Ghorbani, Ali A., Wei Lu, and Mahbod Tavallaee. "Network intrusion detection and prevention: concepts and techniques". Vol. 47. Springer Science & Business Media.
[28] Bhati, Nitesh Singh, Manju Khari, Vicente García-Díaz, and Elena Verdú. "A review on intrusion detection systems and techniques." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 28, no. Supp02 (2020): 65-91.
[29] Dreger, Holger, Anja Feldmann, Vern Paxson, and Robin Sommer. "Predicting the resource consumption of network intrusion detection systems." ACM SIGMETRICS Performance Evaluation Review 36, no. 1 (2008): 437-438.
[30] Islam, Umar, Ali Muhammad, Rafiq Mansoor, Md Shamim Hossain, Ijaz Ahmad, Elsayed Tag Eldin, Javed Ali Khan, Ateeq Ur Rehman, and Muhammad Shafiq. "Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models." Sustainability 14, no. 14 (2022): 8374.
[31] Ibrahim, Omar Jamal, and Wesam S. Bhaya. "Intrusion Detection System for Cloud Based Software-Defined Networks." In Journal of Physics: Conference Series, vol. 1804, no. 1, p. 012007. IOP Publishing, 2021.
[32] Dey, Arunavo. "Deep IDS: A deep learning approach for Intrusion detection based on IDS 2018." In 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1-5. IEEE, 2020.
[33] Chhabria, Viren. "Leveraging Data from Open-Source Intrusion Detection Systems for Enhancing Security of Systems." (2019).
[34] Jiang, Hui, Zheng He, Gang Ye, and Huyin Zhang. "Network intrusion detection based on PSO-XGBoost model." IEEE Access 8 (2020): 58392-58401.
[35] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," Neural Computing and Applications, 1-16, 2020.
[36] Kim, Jin, Nara Shin, Seung Yeon Jo, and Sang Hyun Kim. "Method of intrusion detection using deep neural network." In 2017 IEEE international conference on big data and smart computing (BigComp), pp. 313-316. IEEE, 2017.
[37] Ahsan, Mostofa, and Kendall E. Nygard. "Convolutional Neural Networks with LSTM for Intrusion Detection." In CATA, vol. 69, pp. 69-79. 2020.
[38] Liao, Hung-Jen, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36, no. 1 : 16-24.