INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DHARWAD

MACHINE LEARNING BASED ATTACK DETECTION SYSTEM IN CLOUD COMPUTING

CLOUD COMPUTING

CASE STUDY

PRESENTED BY

AMRUTH MANDAPPA T.S - 20BCS013

ARJUN SAGAR N V - 20BCS020

ASHWANI KUMAR - 20BCS023

DHANIST KUMAR JHA - 20BCS040

YUGAL DEEP SINGH - 20BCS140

UNDER THE SUPERVISION OF

DR. MALAY KUMAR, ASST. PROFESSOR, CSE

INDEX

ABSTRACT:

Modern life heavily relies on networks, and cyber security has become a crucial research area. An intrusion detection system (IDS) is a vital cyber security technique that monitors the state of software and hardware in a network. Despite decades of development, existing IDSs still face challenges in improving detection accuracy, reducing false alarms, and detecting unknown attacks. To overcome these problems, researchers have turned to machine learning methods to develop IDSs. These methods can automatically distinguish normal data from abnormal data with high accuracy and detect unknown attacks. Deep learning, a branch of machine learning, has shown remarkable performance and has become a popular research topic. This survey proposes a taxonomy of IDSs that classifies and summarizes the machine learning- and deep learning-based IDS literature based on data objects. This framework is suitable for cybersecurity researchers. The survey first clarifies the concepts and taxonomy of IDSs. Then, it introduces the frequently used machine learning algorithms, metrics, and benchmark datasets. Combined with representative literature, the proposed taxonomy is used as a baseline to explain how key IDS issues can be solved using machine learning and deep learning techniques. Finally, recent representative studies are reviewed to discuss challenges and future developments.

INTRODUCTION :

CybersecurityCybersecurity has become a critical research area due to the growing impact of networks on modern life. Cyber security techniques, such as anti-virus software, firewalls, and intrusion detection systems (IDSs), aim to safeguard networks from internal and external attacks. IDSs specifically monitor the status of software and hardware in a network to maintain cyber security.

In 1980, the first intrusion detection system was proposed, and since then, many mature IDS products have been developed. However, a common issue with many IDSs is their high false alarm rate, leading to unnecessary alert-flight si-uations. This can cause a serious attack to be ignored place places a burden on security analysts. As a result, researchers have been working on developing IDSs with higher detection rates and lower false alarm rates. Additionally, existing IDSs struggle to detect unknown attacks as network environments are constantly changing and

new attack variants emerge. Hence, it is important to create IDSs that can identify unknown attacks.

In response to the aforementioned issues, researchers have shifted their focus towards constructing intrusion detection systems (IDSs) using machine learning techniques. Machine learning is a form of artificial intelligence that can discover valuable insights from extensive data sets. IDSs based on machine learning methods can achieve satisfactory detection rates when provided with sufficient training data and have the capability to detect novel and variant attacks. Additionally, machine learning-based IDSs are easy to design and construct as they don't require specialized domain knowledge. Deep learning, a branch of machine learning, is particularly effective and excels at handling big data. It can automatically learn features from raw data and process results in an end-to-end manner. Deep learning models have multiple hidden layers, making them more effective than traditional shallow models like the support vector machine (SVM) and k-nearest neighbor (KNN), which have no or only one hidden layer.

The goal of this case study is to provide a comprehensive classification and summary of machine learning-based intrusion detection systems (IDSs) developed to date, while highlighting the key concepts in applying machine learning to security issues and analyzing current challenges and future advancements. Prior surveys have classified research efforts based on the machine learning algorithms used. However, they do not focus on resolving IDS domain problems using machine learning. To address this issue, we propose a new data-centered taxonomy that classifies IDSs based on data sources, which is useful in finding study ideas for specific domain problems. The taxonomy follows a path involving data, features, attack behavior, and detection models, answering questions such as the best features for different attacks, the most suitable data for detecting certain attacks, the most appropriate machine learning algorithms for specific data types, and how machine learning improves IDSs in different aspects. Finally, the survey discusses recent representative studies and the challenges and future developments of machine learning methods for IDSs.

RELATED WORK:

Systems for detecting attacks in the cloud that use machine learning have received a lot of investigation. These are some significant works:

- “Anomaly-based intrusion detection in the cloud using artificial neural networks” by N. E. Fadlullah et al. The study suggested an artificial neural network (ANN)-based system for intrusion detection for cloud technology that might identify intrusions. The suggested method was tested on a real-world dataset and was created to detect assaults that differ from typical network activity. The findings showed the potential of ANNs for threat

detection in cloud computing environments, as the proposed system attained a high rate of detection with a low false-positive rate.

- "A deep learning approach to network intrusion detection in cloud computing" by Shone, Nathan, et al. In the study, a convolutional neural network-based deep learning method for network intrusion detection (CNN) was suggested. The suggested method was created to automatically identify dangerous or benign elements from unprocessed network traffic data. The suggested CNN-based approach generated an excellent detection rate with a low false positive rate, highlighting the promise of deep learning approaches for identifying network attacks in a more successful and effective way. The present scheme was assessed using the NSL-KDD dataset.
- "Analysis of a Payload-based Network Intrusion Detection System Using Pattern Recognition Processors 2016 International Conference on Collaboration Technologies and Systems (CTS)" by Iqbal, I. M., & Calix, R. A. The examination of a pattern recognition processor-based bandwidth system for network intrusion detection (NIDS) is presented in this research. The system examines the payloads of network activity and uses predictive modeling to find any breaches. Using a database of network activity, the authors assess the system's effectiveness and contrast it to that of existing NIDS methods. The system's usefulness as such a security solution is demonstrated by the findings, which indicate that it has a significant rate of detection and a relatively low rate of false positives.
- "A machine learning-based approach for detecting DDoS attacks in cloud computing environments" by Gao, Y., Xu, Z., Zhang, H., & Sun, X. A machine learning-based method for identifying DDoS assaults in cloud computing systems is suggested in the study. The method classifies traffic on the network as either regular or attack traffic using supervised learning techniques. Using a real-world dataset of internet traffic, the authors assess the method and compare it to previous DDoS detection methods. The findings indicate that the suggested method is a potential one for identifying DDoS assaults in cloud computing settings since it has a significant level of precision as well as a small percentage of misclassification.
- "A machine learning-based intrusion detection system for cloud computing environments." by Bhandari, S., Varkhedi, S. S., & Kim, S. W. The study suggests an intrusion detection system (IDS) for cloud computing settings that is machine learning-based. The system utilizes a dual strategy that combines the benefits of anomaly-based and signature-based detection methods. Using a real-world dataset containing network activity, the authors assess the system's performance and compare it to that of existing IDS methods. The findings demonstrate the usefulness of the suggested approach in identifying intrusions in cloud computing settings, demonstrating an elevated detection accuracy and a low rate of false positives. The possibility of utilizing the system as a component of a wider security model for cloud computing is also covered by the researchers.

- "Feature engineering and deep learning-based intrusion detection framework for securing edge IoT" by Muneeba Nasir, Abdul Rahman, and Thar Baker. The research provides a unique engineering and deep learning-based system for intrusion detection for protecting edge IoT systems. Data preparation, feature extraction, and deep learning-based categorization make up the framework's three primary sections. The framework's efficacy is evaluated, and its effectiveness is contrasted with that of existing intrusion detection techniques, using a dataset of data traffic. The findings demonstrate the efficiency of the suggested framework in detecting attacks in edge IoT settings, with high detection accuracy rates and minimal false positives. The authors also cover the possibility of extending the framework to areas other than edge IoT, such as smart buildings and industrial automation systems. Overall, the article offers a viable strategy for protecting edge IoT devices by utilizing cutting-edge machine learning methods.

METHODOLOGY :

How IDS Work:

Intrusion refers to an illegal or unauthorized attempt to access information or damage the operation of computer systems. An intrusion detection system (IDS) is a computer security tool that identifies a broad range of security breaches, such as unauthorized access and insider abuse. IDSs monitor hosts and networks, analyze computer system behavior, generate alerts, and respond to suspicious activities. Typically, they are located near protected network nodes.

The workings of IDSs can be divided into three steps:

1. Monitoring: The first step is to monitor network traffic or system events. Network-based IDSs monitor network traffic flowing through a network segment or a specific device, while host-based IDSs monitor the activity on the host itself.
2. Analysis: The second step is to analyze the monitored data to identify patterns and anomalies. This process involves comparing the data with known signatures or behavioral patterns, which are stored in a database. In anomaly-based IDSs, the data is compared with a baseline of normal behavior to detect any deviations.
3. Alert Generation: If any suspicious activity or security breach is detected, an alert is generated.

Based on where they collect data, IDSs can be classified into two types: network-based and host-based.

Network-based IDSs are typically deployed at strategic points in a network, such as at the border or between subnets. NIDSs can detect a variety of attacks, including port scans, denial-of-service attacks, and attempts to exploit known vulnerabilities in network services. They operate by analyzing network packets as they pass through the network and comparing them against a set of rules or signatures to identify suspicious activity. When a potential intrusion is detected, the NIDS can generate alerts or take other actions, such as blocking traffic from the offending IP address.

Host-based IDSs (HIDSs) are a type of intrusion detection system that monitors the activity on individual hosts, such as servers or workstations. They can detect a range of suspicious activity, including attempts to modify system files, install unauthorized software, or access sensitive data. HIDSs operate by collecting and analyzing system logs and other data to identify anomalies or known attack patterns.

Detection methods include misuse and anomaly detection. Misuse detection, also known as signature-based detection, involves identifying attack behaviors using signatures stored in a database. Anomaly detection establishes a normal behavior profile and identifies abnormal behavior based on its deviation from the normal profile. The main advantage of misuse detection is that it reports attack types in detail and has a low false alarm rate, while anomaly detection is effective in detecting unknown attacks.

MACHINE LEARNING MODEL USED :

Machine learning algorithms play a vital role in Intrusion Detection Systems (IDS) by analyzing network traffic data to detect potential security threats. These algorithms enable IDS to learn from data and adapt to new threats, making them an essential tool for network security. Some common applications of machine learning in IDS include rule-based systems, anomaly detection, and classification. By using machine learning algorithms, IDS can accurately and efficiently identify and mitigate security threats to computer networks, protecting critical infrastructure and sensitive data.

Widely Used Machine Learning Algorithms in Intrusion Detection Systems (IDS):

Artificial Neural Networks (ANN):

Artificial Neural Networks (ANN) is a type of machine learning algorithm that is designed to recognize patterns in data. They are modeled after the way the human brain processes information, with nodes or "neurons" connected to each other in layers. In an ANN, each node receives input from other nodes and applies a mathematical function to

that input. The output of each node is then sent to other nodes in the next layer, and so on, until the output layer produces a result.

In IDS, ANN is trained to learn patterns of normal network traffic and can detect anomalies or attacks based on deviations from those patterns.

Support Vector Machines (SVM):

Support Vector Machines (SVM) is a powerful machine learning algorithm that aims to separate data into two or more classes by finding a hyper-plane with the maximum distance between the classes. SVM is also equally applicable for non-linear problems using kernel functions. These functions transform the original feature space into a higher-dimensional space where the data can be separated by a hyper-plane.

Support Vector Machines (SVM) can be used in Intrusion Detection Systems (IDS) to classify network traffic as either normal or malicious based on the features extracted from the traffic. The main work of SVM in IDS is to find a hyperplane that maximizes the separation between normal and malicious traffic in a high-dimensional feature space.

Decision Tree:

A decision tree is a popular machine learning algorithm used for classification and regression analysis. The decision tree algorithm starts with a single node called the root, which represents the entire dataset. The dataset is then recursively split into smaller subsets based on the values of different features until a stopping criterion is met. The stopping criterion could be the maximum depth of the tree, the minimum number of samples required to split a node, or other criteria.

The decision tree algorithm can be trained on data containing examples of both normal and malicious traffic, and then used to classify new traffic as normal or malicious based on the learned decision rules.

Random Forest:

Random Forest is a machine learning algorithm that combines multiple decision trees to make more accurate predictions. It is an ensemble learning method that works by creating multiple decision trees using random subsets of the training data and a subset of the available features.

In the context of cybersecurity, Random Forest can be used in Intrusion Detection Systems (IDS) to classify network traffic as either normal or malicious based on features extracted from the traffic. Random Forest can also be used in anomaly detection to identify unusual patterns of behavior that may indicate a cyber-attack.

DEEP LEARNING MODELS APPLIED IN IDS :

Deep learning is a subfield of machine learning that focuses on building artificial neural networks with multiple layers. These networks can be trained on large datasets to learn complex patterns and relationships, and can be used for various applications such as pattern recognition, image recognition, etc.

There are several types of deep learning models, each with their own strengths and weaknesses. Supervised learning models, such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), are used for tasks where the input and output data are labeled, and the goal is to learn a mapping between the input and output.

Unsupervised learning models, such as Autoencoders, Restricted Boltzmann Machines (RBMs), and Deep Belief Network (DBN), are used for tasks where the input data is unlabelled, and the goal is to learn the underlying structure and patterns in the data.

Autoencoders:

Autoencoders are useful in pattern learning and recognition tasks. Autoencoders can capture patterns and relationships that exist in the data, and this learned representation can then be used to identify similar patterns in new data.

Autoencoders can be used for Intrusion Detection Systems (IDS) as an unsupervised learning method. In this case, the autoencoder is trained on normal network traffic data and learns to reconstruct it with minimal error. Then, during inference, the autoencoder is used to detect anomalies by calculating the reconstruction error between the input data and the reconstructed output. If the reconstruction error is high, it indicates that the input data is significantly different from the normal data used for training, which suggests the presence of an intrusion or anomaly.

Restricted Boltzmann Machine (RBM):

A Restricted Boltzmann Machine (RBM) is a type of artificial neural network that can be used for unsupervised learning tasks, such as feature learning, pattern recognition. It has two layers of units, a layer of visible units and a layer of hidden units. It can learn a compressed representation of the input data. The hidden units of the RBM can learn to represent the underlying patterns and dependencies in the input data.

Restricted Boltzmann Machines (RBM) can be used in Intrusion Detection Systems (IDS) to learn a compressed representation of network traffic data for anomaly detection.

RBMs can be trained on a large set of normal network traffic data to learn the underlying patterns and dependencies of the data.

Deep Belief Network (DBN) :

A Deep Belief Network (DBN) is a type of deep neural network that is composed of multiple layers of hidden units. It is a generative model that can be used for unsupervised learning tasks, such as feature learning, dimensionality reduction, and pattern recognition. A DBN consists of multiple layers of Restricted Boltzmann Machines (RBMs) that are stacked on top of each other.

In IDS, DBMs have been used for unsupervised feature learning, where the goal is to extract informative features from the raw network traffic data that can be used to detect anomalies or attacks. The DBM is trained on a large dataset of network traffic, and the hidden units of the model are used to represent the learned features. DBMs have shown to be effective in identifying also new and unknown types of attacks, as they can learn to model the distribution of normal network traffic and detect deviations from this distribution.

RESEARCH ON MACHINE LEARNING BASED IDS:

Machine learning is a type of a data driven method in which understanding the data is the first step.In this section, we explore various ways to employ machine learning techniques in Intrusion Detection System (IDS) design, tailored to different types of data that reflect distinct attack behaviors. Host behaviors are captured by system logs, while network behaviors are observed in network traffic. Different attack types exhibit unique patterns, necessitating the selection of appropriate data sources to detect attacks based on their characteristics. For example, Distributed Denial-of-Service (DDoS) attacks involve the transmission of many packets within a brief interval, making flow data suitable for detection. Covert channel attacks involve data leakage between specific IP addresses and are better detected using session data. By leveraging machine learning algorithms, IDS can effectively identify various attack types and help improve network security.

1. PACKET-BASED ATTACK DETECTION

Network packets are the fundamental units of network communication that contain binary data and require parsing to be understood. Each packet consists of a header and application data, with the former specifying IP addresses, ports, and other protocol-specific fields. One advantage of using packets as data sources for Intrusion Detection Systems (IDS) is that they contain communication content,

enabling the detection of User-to-Local (U2L) and Remote-to-Local (R2L) attacks, while also providing IP addresses and timestamps for precise attack source location. Packet-based IDS can also process data in real-time without caching, but they may not reflect the complete communication state or contextual information, making it difficult to detect certain attacks like Distributed Denial-of-Service (DDoS). Packet-based detection methods include packet parsing and payload analysis techniques.

1. **Packet Parsing-Based Detection**
   Network protocols, such as HTTP and DNS, have distinct formats, and packet parsing-based detection methods focus on extracting header fields using tools like Wireshark or Bro. Feature vectors are created from the most important header field values, and classification algorithms are used for attack detection. Researchers have proposed various packet parsing-based detection methods, such as [11] Mayhew et al.'s SVM- and K-means-based approach, which achieved high precision scores for different protocols. Unsupervised learning is a common solution for high false alarm rates, as seen in Hu et al.'s fuzzy C-means based detection method. By introducing fuzzy logic into clustering algorithms, the false alarm rate was reduced by 16.58% and the missed alarm rate by 19.23%. These methods demonstrate the effectiveness of packet parsing-based detection for network intrusion detection.

2. **Payload Analysis-Based Detection**
   Payload analysis-based detection is an effective method that focuses on application data instead of packet headers. Deep learning models can directly process unstructured payload data, resulting in improved accuracy compared to shallow models that require manual intervention.[14] proposed,payload analysis based and Min et al. utilized a text-based CNN to detect attacks using statistical and content features extracted from concatenated payloads. Combining various payload analysis techniques can achieve comprehensive content information and improve IDS effectiveness, as demonstrated by [14] Zeng et al. using multiple deep learning models to extract features from different points of view. Yu et al. utilized a convolutional autoencoder to extract payload features with unsupervised learning, achieving high precision, recall, and F-measure. Adversarial learning is a novel approach to enhance IDS robustness and accuracy, as demonstrated by Rigaki et al. using a GAN to improve malware detection by guiding malware to produce packets similar to

normal packets and analyzing the generated packets to improve IPS robustness.

2. FLOW-BASED ATTACK DETECTION

Flow data, which is a collection of packets during a specific period, is the most commonly used data source for IDSs. Flow data, such as KDD99 and NSL-KDD datasets, has the advantage of representing the entire network environment, making it effective in detecting most attacks, especially DOS and Probe attacks. Additionally, flow preprocessing is simple, as it does not require packet parsing or session restructuring. However, flow-based detection has limitations in detecting U2R and R2L attacks as it does not consider the content of packets. Moreover, extracting flow features involves caching packets, leading to hysteresis. To improve the detection effectiveness of flow-based detection, feature engineering and deep learning methods are used. However, the heterogeneity of flow may lead to poor detection performance, which can be addressed by grouping the traffic.

1. Feature Engineering-Based Detection

In order to apply traditional machine learning models to flow data, feature engineering is a necessary step. This involves creating a "feature vectors + shallow models" framework where feature vectors are generated with interpretable semantics for use in most machine learning algorithms. Common features used include average and variance of packet length, TCP/UDP ratio, proportion of TCP flags, etc.[15] proposed a model consisting of many methods These methods are advantageous due to their simplicity, high efficiency, and ability to meet real-time requirements.

IDSs that use feature engineering-based methods can achieve high detection accuracy but often suffer from a high false alarm rate. One approach to improve the performance is to combine multiple weak classifiers to create a strong classifier. For example, Goeschel et al. proposed a hybrid method that used SVM, decision tree, and Naïve Bayes algorithms. They first used SVM to classify the data into normal or abnormal samples. Then, for abnormal samples, they used a decision tree to identify specific attack types. However, the decision tree can only detect known attacks, not unknown ones. Therefore, they also used a Naïve Bayes classifier to detect unknown attacks. By combining these three classifiers, the hybrid method achieved high accuracy (99.62%) and a low false alarm rate (1.57%) on the KDD99 dataset.

In addition to accuracy, another important research objective for IDSs is to improve detection speed. Kuttranont et al. proposed a KNN-based detection method that used parallel computing techniques on a GPU to accelerate the calculation. They modified the neighbor-selecting rule of the KNN algorithm and achieved an accuracy of 99.30% on the KDD99 dataset. The proposed method was approximately 30 times faster than the method without GPU acceleration.

Unsupervised learning methods, such as clustering algorithms, are also applied to IDS. To improve detection efficiency on large datasets, Peng et al. proposed an improved K-means detection method with mini-batch. They first preprocessed the KDD99 dataset by transforming nominal features into numerical types, normalizing each feature dimension using the max-min method, and reducing dimensions using the PCA algorithm. Then, they clustered the samples with the K-means algorithm but improved it by altering the initialization method to avoid local optima and introducing the mini-batch trick to decrease the running time. The proposed method achieved higher accuracy and runtime efficiency compared to the standard K-means.

2. Deep Learning-Based Detection

Feature engineering is often limited by the availability of domain knowledge, which can hinder the performance of intrusion detection systems. In contrast, deep learning-based methods can automatically learn features, and they are becoming increasingly popular in IDS research due to their end-to-end processing capabilities. Potluri et al. [12] proposed a CNN-based approach for detecting attacks on the NSL-KDD and UNSW-NB 15 datasets. They converted the feature vectors in these datasets into images, which were then processed by a three-layer CNN. Their approach outperformed other deep network models, achieving accuracies of 91.14% and 94.9% on the NSL-KDD and UNSW-NB 15 datasets, respectively.

Unsupervised deep learning models can also be used to extract features, which can then be classified using shallow models. Zhang et al. [50] used a sparse autoencoder to extract features from the NSL-KDD dataset and then used an XGBoost model to classify the data. Their approach achieved high accuracies on all classes, particularly the Normal and DOS classes.

However, deep learning models may not perform as well on small or unbalanced datasets. Adversarial learning can be used to augment small datasets and improve detection accuracy. Zhang et al. [12] used a GAN to generate data similar to the flow data of the KDD99 dataset, which was then added to the training set to detect attack variants. Their approach

improved accuracies on 7 out of 8 attack types, demonstrating the effectiveness of adversarial learning in improving detection accuracy.

3. **Traffic Grouping-Based Detection**

Intrusion detection systems (IDS) often face the challenge of handling diverse types of traffic, some of which may not be relevant to detecting attacks and may even act as noise. To address this issue, grouping traffic data into more homogeneous subsets can be an effective approach to improve detection accuracy and prevent overfitting.[16] proposed a model in which Two common methods for grouping traffic are protocol-based and data-based grouping.

Protocol-based grouping involves dividing the dataset based on the protocol type of the traffic, such as TCP, UDP, or ICMP. This method has been applied by Teng et al. [12], who proposed an SVM-based detection method using KDD99 dataset. They selected features for each protocol-based subset and trained SVM models on the three subdatasets, achieving an average accuracy of 89.02%.

Data-based grouping, on the other hand, involves clustering traffic based on their characteristics. Ma et al. [53] proposed a DNN and spectral clustering-based detection method, where they first divided the original dataset into six highly homogeneous subsets and trained DNN models on each subset. Their approach achieved an accuracy of 92.1% on both KDD99 and NSL-KDD datasets. By grouping traffic data, IDS can improve their ability to detect attacks and reduce the impact of noise, leading to better performance and increased security.

3. SESSION-BASED ATTACK DETECTION

1. Statistic-Based Feature Detection Methods:

Statistical information from packet headers can be used to compose feature vectors for shallow models, which are suitable for rule-based or decision tree models. However, these methods have difficulty detecting intrusions related to communication content. [17] Ahmim et al. proposed a hierarchical decision tree method that analyzed the frequency of different types of attacks and designed the detection system to recognize specific attacks, achieving good performance on 8 of 15 classes while reducing detection time. Alseiari et al. proposed an unsupervised method that used

mini batch K-means to divide data into clusters and label them based on the assumption that normal samples were the majority and their distances were relatively short. This method effectively detected intrusion behaviors in smart grids and located attack sources with a false alarm rate less than 5%.

2. Sequence Feature-Based Detection

Session-based intrusion detection requires analyzing the packet sequence features, such as packet length and time intervals. However, most machine-learning algorithms cannot handle sequences. Therefore, RNN algorithms, such as LSTM and bi-LSTM, are commonly used in session-based intrusion detection. Bag-of-words (BoW) and word embedding approaches are used to preprocess raw data for RNN algorithms. BoW suffers from the inability to represent similarity between words, but word embedding overcomes this problem. Character-level CNN is another encoding method used in session-based intrusion detection. Hierarchical deep learning methods have also been proposed, using both CNN and LSTM to learn low-level spatial features and high-level time features, respectively. Experimental results on various datasets show that these methods achieve high accuracy and detection rates.

4. LOG-BASED ATTACK DETECTION

1. Rule and Machine Learning-Based Hybrid Methods

[19] proposed a model for hybrid methods combining rule-based detection and machine learning to improve performance in IDS. They take the output of rule-based systems as input and use machine learning models to filter out meaningless alerts. One approach to reducing false alarm rates is to rank alerts via machine learning, as proposed by [19] Meng et al., who used a KNN model to filter Snort-generated alerts, reducing the number of alerts by 89%. Another approach is to use DNN to find important security events in logs, as proposed by McElwee et al. The extracted events are then analyzed by security experts and used as training data to improve the DNN model, which reduces analyst workloads and accelerates security analyses.

2. Log Feature Extraction-Based Detection

The log feature extraction-based detection method involves extracting log features using a sliding window and analyzing them with machine learning

algorithms to detect abnormal behaviors. [20] proposed a CNN method to analyze system calls, while Tuor et al. proposed an interpretable deep learning detection method using system logs. Bohara et al. proposed an unsupervised learning detection method using feature selection and clustering to detect abnormal behaviors. These methods are suitable for detecting intrusions and reducing analysis workloads.

3. Text Analysis-Based Detection

The text analysis-based detection method analyzes logs as plain text using n-grams and other text processing techniques, which leads to stronger interpretability. In this method, keywords in the field of cybersecurity can improve the detection effect. [21] proposed an SQL-injection detection method for IoT using SVM with n-gram features and achieved high accuracy scores. One-class classification, a type of unsupervised learning, can solve the problem of a lack of abnormal samples in actual network environments. Vartouhi et al. proposed a web attack detection method based on the isolated forest model using the CSIC 2010 dataset and achieved an accuracy of 88.32%.

IMPLEMENTATION:

XGBoost (Extreme Gradient Boosting) is a machine learning technique that may be used for regression as well as classification applications. It is a strong and efficient algorithm that makes predictions by combining numerous weak decision trees. The technique is based on the gradient boosting principle, which entails the building of a sequence of decision trees, each of which learns from the faults of the preceding tree.

We utilized XGBoost to analyze big datasets and discover patterns that are suggestive of intrusion in the context of cloud computing. This technique works by building a succession of decision trees that are trained on different subsets of data repeatedly.

We chose XGBoost because it is both quick and scalable, making it appropriate for huge datasets. It also has great accuracy and can deal with missing values and outliers in the data.

The first stage in implementing the XGBoost algorithm for intrusion detection in cloud computing is to gather and preprocess data. Identifying useful characteristics and cleaning the data to reduce noise and inconsistencies are also part of this process.

Following that, the data is divided into training and testing sets, with the training set used to create decision trees and the testing set used to evaluate the model's performance.

The XGBoost method is then applied to the training data, with hyperparameters such as learning rate, number of trees, and maximum depth of trees optimized to improve performance. Based on the attributes retrieved from the data, the resultant model may then be used to forecast whether a certain event is an intrusion or not.

Overall, the XGBoost algorithm is a strong and efficient tool for cloud computing and home-based intrusion detection. Its capacity to handle enormous datasets and reliably forecast intrusion events makes it a crucial tool for ensuring cloud computing system security and integrity.

We have used various Amazon services like Amazon S3 and Amazon SageMaker for the implementation. Amazon S3 (Simple Storage Solution) is a cloud-based storage solution that allows you to store and retrieve data from anywhere on the internet. It offers highly scalable, long-lasting, and secure storage for objects and data in the terabyte range. Amazon S3 is employed in the project of home-based intrusion detection to store the enormous quantity of data created by the system for further analysis and processing.

Amazon SageMaker is a fully-managed service that allows developers and data scientists to rapidly and simply design, train, and deploy machine learning models. It includes a robust collection of tools for developing, training, and deploying machine learning models at scale. Amazon SageMaker is utilized in the project of home-based intrusion detection to develop the XGBoost algorithm for detecting and classifying intrusions in the cloud computing environment.

We have used these two services in our implementation, as both of them are::

- Scalability: Because both services are extremely scalable, the project can store and handle massive volumes of data as well as perform complicated machine learning algorithms.

- Cost-effectiveness: The Amazon S3 and SageMaker pay-as-you-go models allow the project to only pay for what it uses, lowering expenditures.
- Security: Both services have robust security safeguards in place to protect the safety and privacy of the data being kept and processed.

- Ease of use: Both services offer simple user interfaces and easy connectivity with other Amazon Web Services, making them simple to set up and use.

RESULT:

Machine learning XGBoost algorithm on 'KDDcup99' dataset by utilizing the amazon AWS cloud features such as Amazon Sagemaker, S3 bucket cloud storage space, and other cloud resources has shown great results. The model achieved an excellent accuracy of 99.17%, demonstrating its effectiveness and efficiency in identifying potential security threats. The successful results of this model can be attributed to the power of the XGBoost algorithm, which uses the gradient boosting technique and the regularization function to optimize the regression. In addition, the quality and quantity of training data and the optimization and presentation of the training model also increase accuracy.

The XGBoost algorithm is designed to optimize the performance of the model by minimizing its loss rate. We have calculated the loss rate for training and validation data subset, Fig. 1 shows the plot of loss function with number of epochs.

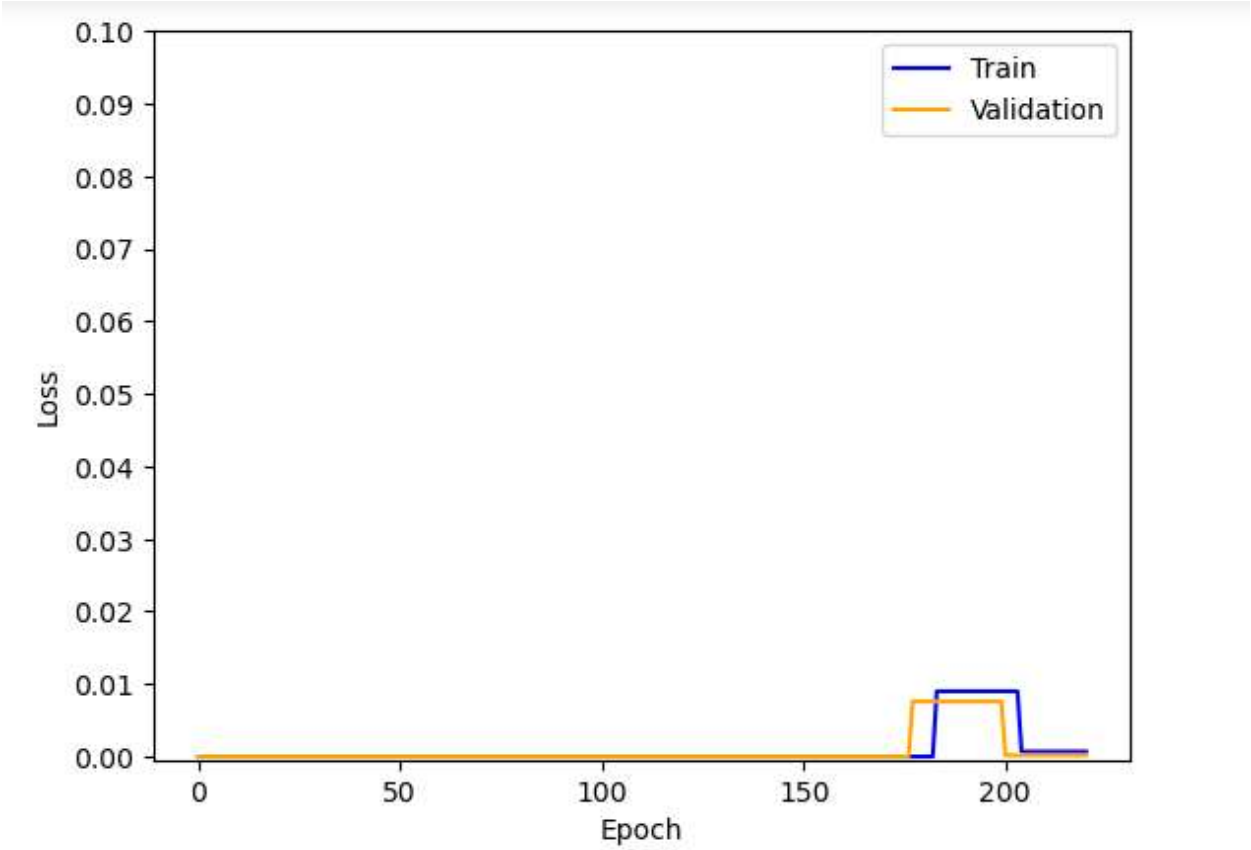

Fig.1

Observation: Figure 1 illustrates the loss rates for the training set and validation set, which provides valuable insights into the effectiveness and generalization ability of our model. As the training loss function and validation loss function line is very near to each

other that clearly demonstrates that the model has been trained to a high level of accuracy, without succumbing to the common issue of overfitting.

CHALLENGES:

1. Data Quality: The effectiveness of machine learning algorithms heavily relies on the quality and quantity of data. In the case of an intrusion detection system, the data should be diverse and comprehensive, including different types of attacks and network configurations. However, the data collected may contain errors, noise, or outliers, which can lead to incorrect predictions and false positives..
    a) Data heterogeneity refers to the presence of diverse types of data, formats, or structures in a dataset. This can include variations in data collection methods, measurement units used, or data quality. The presence of data heterogeneity can create difficulties in data processing by merging data from different sources and can affect the accuracy and reliability of data analysis results.
    b) Data inconsistency: It can be a significant challenge in intrusion detection systems (IDS) that are used to detect and prevent unauthorized access to computer networks or systems. IDS rely on data analysis to identify potential threats and anomalies in network traffic or system behavior. Inconsistent data such as incorrect time stamps, missing packets, or corrupted data can cause an IDS to misinterpret legitimate traffic as malicious or overlook actual attacks. This can lead to false positives or false negatives, which can compromise the effectiveness of an IDS.

2. Class Imbalance: The number of intrusion instances is often much smaller than the normal instances in network traffic. This creates a class imbalance problem, where the machine learning model may be biased towards the majority class and unable to detect the minority class effectively. Addressing this issue requires techniques such as oversampling, undersampling, or using a different cost function.

CONCLUSION:

Intrusion detection systems (IDS) play a crucial role in protecting computer networks and systems from potential attacks. Various approaches have been proposed to improve IDS performance and accuracy, including feature engineering, deep learning-based detection, and traffic grouping-based detection.

Feature engineering requires domain knowledge and may not capture all the relevant features, limiting the effectiveness of the IDS. Deep learning-based detection, on the other hand, can automatically learn features and achieve high accuracy, especially when applied to large and complex datasets. Unsupervised deep learning models can be used to

extract features and improve classification using shallow models. Adversarial learning can also be used to augment small datasets and improve detection accuracy.

Traffic grouping-based detection involves grouping traffic data into homogeneous subsets, which can improve detection accuracy and prevent overfitting. Protocol-based grouping divides the dataset based on protocol type, while data-based grouping involves clustering traffic based on its characteristics. Both methods have shown promising results in improving IDS performance, leading to better security and protection against potential attacks.

To sum up, IDS research has made significant progress in recent years, and these approaches provide useful strategies for improving IDS accuracy and effectiveness. However, there is still room for improvement, and future research should focus on developing more robust and adaptive IDS systems that can handle new and evolving threats.

REFERENCE:

1) M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, 2017, pp. 1-7, doi: 10.1109/CloudTech.2017.8284731.
2) Prof. D.P. Gaikwad, Sonali Jagtap, Kunal Thakare, Vaishali Budhawant, 2012, Anomaly Based Intrusion Detection System Using Artificial Neural Network and Fuzzy Clustering, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 01, Issue 09 (November 2012).
3) Z. He, T. Zhang and R. B. Lee, "Machine Learning Based DDoS Attack Detection from Source Side in Cloud," *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, New York, NY, USA, 2017, pp. 114-120, doi: 10.1109/CSCloud.2017.58.
4) A. Yaar, A. Perrig, and D. Song, “Pi: A path identification mechanism to defend against ddos attacks,” in Security and Privacy, 2003. Proceedings. 2003 Symposium on. IEEE, 2003, pp. 93–107.
5) Abbasi, H., Ezzati-Jivan, N., Bellaiche, M. *et al.* Machine Learning-Based EDoS Attack Detection Technique Using Execution Trace Analysis. *J Hardw Syst Secur* 3, 164–176 (2019). https://doi.org/10.1007/s41635-018-0061-2
6) Tiwari, Mohit & Kumar, Raj & Bharti, Akash &Kishan, Jai. (2017). INTRUSION DETECTION SYSTEM. International Journal of Technical Research and Applications. 5. 2320-8163.

7) Gao, Y., Xu, Z., Zhang, H., & Sun, X. (2020). A machine learning-based approach for detecting DDoS attacks in cloud computing environments. IEEE Access, 8, 165508-165519.
8) Khraisat, A., Gondal, I., Vamplew, P. *et al.* Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecur***2**, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7
9) Bhandari, S., Varkhedi, S. S., & Kim, S. W. (2019). A machine learning-based intrusion detection system for cloud computing environments. IEEE Access, 7, 12098-12107.
10) Liu, B., Wang, Y., Zhang, Q., & Wu, J. (2019). A machine learning-based approach for detecting insider threats in cloud computing environments. IEEE Access, 7, 33856-33865.
11) Yi-Wen Chen, Jang-Ping Sheu, Yung-Ching Kuo, and Nguyen Van Cuong. Design and Implementation of IoT DDoS Attacks Detection System based on Machine Learning. Institute of Communication Engineering, National Tsing Hua University Hsinchu, 30013, Taiwan.
12) Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H. T., Rashidi, S Rahmani, A. M. (2021). Deep Learning-Based Intrusion Detection Systems: A Systematic Review. IEEE Access, 9, 101574–101599. doi:10.1109/access.2021.3097247
13) Iqbal, I. M., & Calix, R. A. (2016). Analysis of a Payload-based Network Intrusion Detection System Using Pattern Recognition Processors. 2016 International Conference on Collaboration Technologies and Systems (CTS). doi:10.1109/cts.2016.0077
14) Muneeba Nasir, Abdul Rahman, Thar Baker. Feature engineering and deep learning-based intrusion detection framework for securing edge IoT. The journal of supercomputing 78, 8852-8866 (2022).
15) Zhang Ze-Dong; Sheon Hao-Tong, Wei Song-Jie. Network Anomaly Detection based on Traffic Clustering with Group-Entropy Similarity. IEEE Access, 8.
16) J. Chen, A. J. Gallo, S. Yan, T. Parisini, and S. Y. R. Hui, "Cyber-Attack Detection and Countermeasure for Distributed Electric Springs for Smart Grid Applications," IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1258-1269, Mar. 2021, doi: 10.1109/TSG.2020.3049668.
17) Shone, Nathan, et al. "A deep learning approach to network intrusion detection." *IEEE transactions on emerging topics in computational intelligence* 2.1 (2018): 41-50.
18) Yuan, X., Zhou, W., Wang, Z., Zhu, Y., & Yin, C. (2020). A Rule and Machine Learning-Based Hybrid Method for Intrusion Detection System. IEEE Access, 8, 184047-184062. doi: 10.1109/ACCESS.2020.3031365.
19) Abuadhmah, A., & Mohammed, A. (2019). Log Feature Extraction-Based Detection of Intrusion in Computer Networks Using Machine Learning Techniques. Journal of Cybersecurity and Information Management, 2(1), 1-10. doi: 10.11648/j.cim.20190201.11.

20) G. Li, X. Zhang, H. Song, Y. Wang, and Y. Li, "Log feature extraction-based intrusion detection system for cloud computing," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 6, pp. 2161-2171, 2019, doi: 10.1007/s12652-019-01240-2.
21) B. Lee, J. Kim, and J. Park, "Text analysis-based intrusion detection system for unstructured text logs," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, 2019, pp. 180-189, doi: 10.1109/COMPSAC.2019.00139.