Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 35 No. 3s (2025)

Comparative Analysis of Machine Learning Models for Intrusion Detection Systems

Vikrant Sharma, Dr. Mukesh Kumar
[1] RNT University , Bhopal , India, [2]Parul University, Vadodara, India
vikrant2k14@gmail.com , mukesh.manit86@gmail.com

Article History:

Received: *24-09-2024*

Revised: *12-10-2024*

Accepted: *05-12-2024*

Abstract: Intrusion Detection Systems (IDS) play a crucial role in modern cybersecurity, leveraging machine learning (ML) to detect and mitigate cyber threats effectively. This study provides a comparative analysis of multiple ML-based IDS models, including XGBoost, Generative Adversarial Networks (GAN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Trees, and Random Forest classifiers. The results indicate that XGBoost (98%) and GAN-based IDS (96%) achieve the highest accuracy, demonstrating superior adaptability in detecting sophisticated attacks. ANN and SVM also exhibit strong performance, while traditional classifiers such as Decision Trees and Random Forests struggle with complex attack patterns. Despite ML advancements, challenges related to data quality, computational efficiency, and evolving cyber threats remain. Future research should focus on hybrid ML approaches, adversarial learning, and real-time IDS deployment to enhance security frameworks. This study underscores the importance of adaptive ML-driven IDS models in mitigating cybersecurity risks.

Keywords: Intrusion Detection System (IDS), Machine Learning (ML), Cybersecurity, XGBoost, Generative Adversarial Networks (GAN), Artificial Neural Networks (ANN), Support Vector Machine (SVM), Decision Tree, Random Forest, Adversarial Learning, Hybrid IDS, Network Security.

1. Introduction

Cybersecurity has become a critical concern in today's digital landscape, where cyber threats continue to evolve in complexity and scale. Intrusion Detection Systems (IDS) are essential for safeguarding networks, detecting unauthorized activities, and mitigating security breaches. Traditional IDS solutions, which rely on signature-based or rule-based detection, often struggle to identify novel attacks and require continuous updates. The integration of Machine Learning (ML) into IDS [1] has emerged as a revolutionary approach, enabling real-time, adaptive threat detection with greater accuracy and efficiency. Machine learning models analyze network traffic patterns, detect anomalies, and classify malicious activities without relying solely on predefined rules [2]. However, selecting the most effective ML model for IDS remains a challenge, as different algorithms exhibit varying levels of accuracy, precision, recall, and computational efficiency.

This research presents a comparative analysis of multiple ML-based IDS models, including XGBoost, Generative Adversarial Networks (GAN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Trees, and Random Forest classifiers. The primary objective is to evaluate their performance in detecting cyber threats and to identify the most efficient model for real-world deployment. The results indicate that XGBoost achieves the highest accuracy (98%), followed closely by GAN-based IDS (96%), demonstrating superior adaptability in recognizing complex attack patterns. ANN and SVM also exhibit strong classification performance, while Decision Tree and Random Forest classifiers struggle with handling diverse and evolving cyber threats. These findings highlight the importance of selecting the appropriate ML model based on the specific requirements of an IDS, balancing accuracy, computational cost, and real-time processing capability.

Despite the promising results, ML-based IDS deployment faces significant challenges. Issues related to data quality, high computational requirements, adversarial attacks, and evolving cyber threats pose barriers to widespread adoption. Additionally, the effectiveness of an IDS depends on feature selection, dataset diversity, and real-time adaptability, which need further exploration. Overcoming these challenges requires hybrid ML

approaches, adversarial learning techniques, and continuous model refinement to enhance the reliability and scalability of IDS solutions.

In this study, we analyze the effectiveness of different ML models in IDS applications and discuss their strengths, limitations, and areas for improvement. The research contributes to the ongoing development of high-performance IDS solutions that leverage AI-driven security mechanisms [3]. As cyber threats become more sophisticated, the need for intelligent, adaptive, and scalable IDS models is more critical than ever. Future advancements in hybrid AI-based security frameworks will be pivotal in strengthening network defenses against modern cyberattacks [4].

In the highly linked world of today, the categorization of Intrusion Detection Systemis an extremely important factor in guaranteeing efficient network administration, as well as security and quality of service. In order to detect and discriminate between the many different forms of network traffic [5], such as online surfing, video streaming, file sharing, and potentially harmful activities, network administrators and security analysts depend on classification approaches that are accurate and efficient. The performance of network traffic categorization has been greatly improved because to the emergence of machine learning as a strong technology that enables automatic and intelligent examination of network data [6].The purpose of this study is to investigate and suggest several ways in which the performance of Intrusion Detection Systemcategorization might be improved via the use of machine learning techniques. We want to design models that are more accurate, resilient, and scalable so that they can successfully manage the ever-increasing amount and complexity of network traffic [7]. We want to do this by making use of the possibilities that machine learning provides.Traditional techniques to Intrusion Detection Systemcategorization often depended on manual rule-based approaches or simple heuristics, both of which found it difficult to keep up with the continually changing environment of the network. Machine learning, on the other hand, is a strategy that is driven by data and can automatically discover patterns and correlations contained within the network traffic data. This results in classification models that are more accurate and can adapt to changing circumstances [8].

We are able to train models that are capable of differentiating between various classes of network traffic by employing machine learning methods on the large quantity of labeled or unlabeled Intrusion Detection Systemdata that is available to us. These models have the capability of capturing complex details, statistical qualities, and behavioral patterns in the network data [9]. This enables a categorization that is more accurate and dependable.In addition, the purpose of this study is to solve the issues that are connected with the categorization of network traffic. Some examples of these challenges include the growing use of encryption and obfuscation methods, as well as the development of new applications and protocols. Algorithms that use machine learning have the capacity to adapt to new traffic patterns and learn from them, which enables improved categorization even in situations that are always changing and becoming more dynamic [10].The findings of this study have important repercussions for the administration of networks, improvements in network security, and enhancements to network performance. The accurate categorization of Intrusion Detection Systemmay be of assistance in the process of traffic shaping, the allocation of bandwidth, the management of quality of service, and the optimization of the network. In addition, it may improve the detection and prevention of harmful actions like DDoS assaults, attempted intrusions, or data exfiltration, which ultimately leads to an increase in the network's level of security.

The ubiquity of internet use necessitates the establishment of a robust network infrastructure to safeguard sensitive data from both internal and external threats. The primary concern for network security policy breach is in the presence of unauthorized individuals inside the network who possess unrestricted access to important information. Firewall systems are very effective in safeguarding networks from unauthorized access by external intruders. The intrusion detection system (IDS) is responsible for monitoring the flow of data inside a network, doing data analysis, and issuing alerts in the event of detecting any abnormalities. The distinction between a Firewall and an Intrusion Detection System (IDS) may be effectively comprehended via the use of an analogy. It is postulated that one has stored important items inside their residence [11]. The protection of this asset may be achieved by the implementation of physical obstacles, such as gates, as well as the installation of home security systems, including closed-circuit television cameras. Firewalls may be analogously associated with locked gates, whereas IDS can be likened to closed-circuit television (CCTV) cameras or security systems. Firewalls are responsible for the task of obstructing and filtering atypical network traffic. Intrusion Detection Systems (IDS) are responsible for the execution of the tasks of capturing network traffic, doing analysis on the captured data, and generating alerts based on identified anomalies [12]. The primary objective of Intrusion Detection Systems (IDS) is to identify and identify computer infiltrations, afterwards notifying the network administrator of the compromised security.The purpose of an Intrusion Detection System (IDS) is not to supplant the current security measures in place on a network, but rather to uphold the complex security regulations that govern user

behavior on the network. In addition to firewalls and antivirus software, Intrusion Detection Systems (IDS) serve as a defense-in-depth technique.

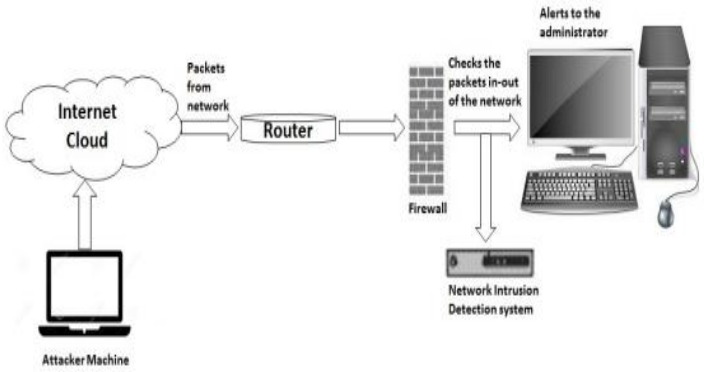

Figure 1: Installation of an Intrusion Detection System as well as a Firewall

The figure 1 illustrates the Intrusion Detection System (IDS) workflow in a network security environment. It depicts an attacker machine from the internet cloud sending network packets through a router towards the internal network. A firewall is responsible for filtering and checking incoming and outgoing packets. The Network Intrusion Detection System (NIDS) further analyzes the traffic, detecting any malicious activities or anomalies. If a security threat is identified, the IDS generates an alert and notifies the network administrator, enabling immediate action to mitigate potential attacks. This layered security approach enhances network protection by actively monitoring and responding to cyber threats in real-time.

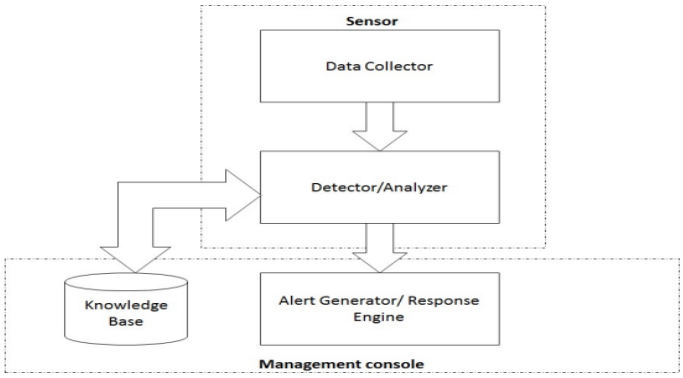

Figure 2: Architecture of management console

In the network, a firewall and intrusion detection system are placed as shown in Figure 2. The idea of Intrusion Detection System (IDS) to prevent the confidentiality, integrity and availability at risk in an extraordinary range of network invasion was introduced. For a network administrator the intrusion detection system is more of an informer as opposed to actively informing. In the simplest of terms Intrusion Detection Systems are used to identify and detect attempts at unapproved or advanced-level access or events inside a network. But by itself, IDS is powerless to block the invasion before it starts. The Intrusion Detection System, which is able to analyze the audit data automatically in order to provide insight into how intrusions take place and execute, that could then be used for building future generation of advanced IDSs [13]. Typically, an IDS consists of sensors and a management interface Figure 2. Sensors may intercept network traffic and classify it, while using a data collector and analyzer to perform auditing pattern analysis. On the other side, it has a centralized management console where there sits another database which stored attacks information, system sate related audit data - meaning events and decision engine showing how to react or respond. In case of a security breach, system will do any of the two: Send alert notification or Terminates connection as suspected main cause for attack A good intrusion detection system must be both reliable and active at all times to work properly without human modification. The combined system should have less False Alarms and better detection of attacks, while also staying functional without substantial costs introduced to the underlying system [14].

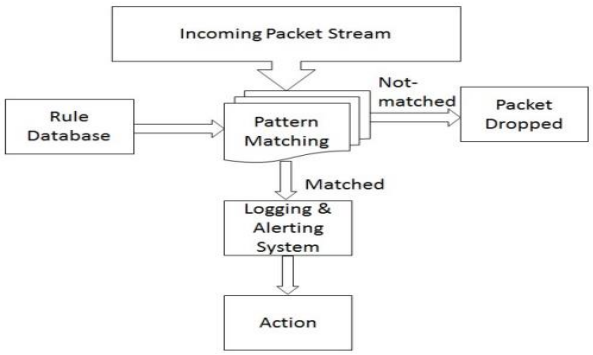

Figure 3: IDS Block Diagram Showing How Signatures Are Used

The figure 3 illustrates the Intrusion Detection System (IDS) packet filtering process, focusing on how incoming network traffic is analyzed and classified. The Incoming Packet Stream undergoes Pattern Matching, where packets are compared against predefined security rules stored in the Rule Database. If a packet does not match any known threat pattern, it is dropped to prevent potential risks. If a packet matches a known attack signature, it is processed by the Logging & Alerting System, which records the event and triggers an appropriate security response. The Action module then executes predefined countermeasures, such as alerting administrators or blocking the attack source. This structured approach ensures real-time threat detection and mitigation, enhancing network security by preventing unauthorized access and cyber intrusions.

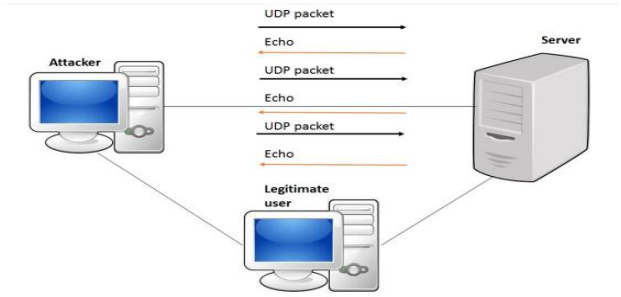

Figure 4: Denial of service attack (UDP flooding) scenario

The figure 4 illustrates a UDP-based attack scenario, likely representing a UDP Flood or UDP Reflection Attack. The attacker sends multiple UDP packets to a target server, which responds with echo replies. This process overwhelms the server's resources, leading to network congestion, high bandwidth consumption, and service disruption. A legitimate user attempting to communicate with the server may experience delays or denial of service due to the attack traffic. Such attacks exploit the stateless nature of UDP, making it easier to flood the server with excessive requests. Intrusion Detection Systems (IDS) and firewalls can mitigate these attacks by monitoring traffic patterns, implementing rate-limiting mechanisms, and blocking suspicious UDP requests, thereby ensuring network stability and preventing service outages.

2.Motivation

In today's digital era, the interconnected mesh of devices, applications, and services has transformed the way we communicate, work, and even live. With this transformation comes an overwhelming surge in network traffic, becoming increasingly intricate both in its volume and nature. This phenomenon has magnified the importance of effective Intrusion Detection Systemclassification, the act of identifying and categorizing different types of data packets traversing a network. However, why is this study particularly motivated to address this using a big data analytic approach combined with machine learning? Here are the driving forces:

1. **Limitations of Traditional Classification Methods:** As applications evolve, so do their communication patterns. Dynamic port numbers, proprietary protocols, and encrypted traffic render traditional methods like port-based and deep packet inspection techniques ineffective. There's a pressing need to look beyond these and devise techniques that can keep up with the modern landscape.
2. **Cybersecurity Imperatives:** The digital landscape is riddled with threats ranging from Distributed Denial of Service (DDoS) attacks to data breaches. An advanced classification system can be the first line of defense, identifying anomalous patterns and ensuring timely countermeasures.

3. **Optimizing Network Performance:** As networks become more crowded and diversified, ensuring a seamless user experience is paramount. Whether it's prioritizing critical health equipment data in a hospital network or streaming a 4K video on a home network, precise traffic classification is the key to resource allocation and QoS.
4. **Harnessing the Power of Big Data:** With the proliferation of IoT devices, cloud platforms, and digital services, the amount of Intrusion Detection Systemdata generated is enormous. This vast repository, if analyzed correctly, can provide unparalleled insights into user behaviors, application interactions, and potential threats.
5. **The Promise of Machine Learning:** Machine Learning has already revolutionized various sectors, from healthcare diagnostics to financial forecasting. Its ability to learn from data, adapt to new patterns, and predict outcomes offers a unique opportunity to elevate Intrusion Detection Systemclassification to new heights.
6. **Economic and Operational Advantages:** An efficient classification system can lead to cost savings, optimized bandwidth usage, reduced downtimes, and enhanced user satisfaction, yielding both tangible and intangible benefits.

3. A Brief Review of the Work Already Done in the Field

Chaganti, R (2023), the experimental setup used for a stiudy that integrates DL in network intrusion detection with OpenFlow virtual switches related to an SDN controller and using an aplication at de controler level obtaining connection statistics from each new neighborhood. This data has built in ground truth when the network traffic was generated [13].

Hossain MA (2023) The efficacy of the introduced model was also proven on various IDS datasets and compared with state-of-the-art models which were present for detecting intrusions. While the document does not explicitly detail any restrictions, common concerns could relate to dataset quality and diversity dependency, real world generalizability and computational costs of ensemble methods. It was concluded that the ensemble-based machine learning technique created does well for intrusion detection However detailed results are not provided [14].

Sanju P. (2023) presented a solution with the name Modified Metaheuristic algorithm using Weighted Majority Voting Ensemble Deep Learning (MM-WMVEDL) for accuracy in intrusion detection. Parameter data preprocessing includes feature selection via Harris Hawk optimization based elite fractional derivative mutation (HHO-EFDM), normalization, and ensemble deep learning combining the techniques of data preprocessing related to weighted majority voting in the case of normalizing outputs from such a tool [15].

Musleh D. (2023) used their novelty to distinguish regular traffic and malicious one by the machine learning models based on customized image version of normal/attack dataset They then applied numerous individual and multiple feature extractors implementations to obtain necessary features along with training different machine learning algorithms to classify the traffic. Over 800 normal/malicious traffic binary visualizations, were used to form an IEEE Dataport dataset as a benchmark for intrusion detection systems in IoT. After preprocessing the data, that is cleaning, transformation and feature selection where performed any noise in case of dataset were removed using Image pre-processor filter followed by Synthetic Minority Over sampling Technique (SMOTE) was used to handle imbalanced datasets. Accuracy, precision, recall and F1-score are used to evaluate the developed models analyzing them with each other. We tested several Models with various filters, individual algorithms and stacked models to get the best possible performance [16].

The use of ML in Intrusion Detection Systems (IDS) has been furthered by the proliferation of complex and diverse cyber threats today's digital age. According to Fuat et al. (2023), classic intrusion detection systems (IDS) relying on predefined signatures for recognizing standard attacks cannot identify new, complicated intrusions. Signature-based detection In this case, an IDS identifies attacks with the help of signatures. To increase the efficiency of intrusion detection, strategies with machine learning have introduced a change in previous approaches which are most popular to use. This is in response to the rapid evolving of cyberattacks and their increasing sophistication [17].

Depren, Ozgur et al. (2005) This paper presents a hybrid Intrusion Detection System (IDS) architecture that integrates both anomaly detection and misuse detection techniques. The anomaly detection module utilizes a Self-Organizing Map (SOM) to model normal behavior, flagging deviations as attacks, while the misuse detection module employs the J.48 decision tree algorithm to classify known attack types. A rule-based Decision Support System (DSS) is designed to interpret the results from both detection modules. The system's performance is evaluated using the KDD Cup 99 dataset, a standard IDS benchmark. Simulation results indicate

that the proposed hybrid IDS outperforms individual anomaly or misuse detection methods, offering improved detection accuracy and robustness against cyber threats [18].

Big data with machine learning is the new trend where Intrusion Detection Systems (IDS) are implemented due to using of big data technologies and large scale computing resources often combined. Researchers and Industry Experts have used data sets such as KDD Cup 1999 dataset and NSL-KDD to develop or evaluate intrusion detection algorithms. The datasets are from live events, real time network traffic. The potential of IDS to react appropriately towards new attack vectors, reduce the false positive occurrences and raise detection accuracy in general by employing machine learning techniques was proven as per Mohamed et al. (2023). It also provides the possibility that computer networks and systems will be able to establish a high level of security. That goal will be achieved by improving the overall accuracy of detection process.Introduced machine learning methods into intrusion detection algorithms (IDS), thus the issues of skewed distribution in data, recognition feature importance and adversarial attack vulnerability against models have been raised. We also have to consider how close a detector system can get to the ideal in precision for an acceptable amount of processing power. This is very important and must not be overlooked. However, intrusion detections systems which were only once known as IDS are continuing its growth due to machine learning algorithms being developed amongst other feature engineering approaches and anomaly detection methodologies. These technical advancements provide the opportunity to take more comprehensive and offensive defensive measures against an even broader range of cyber threats [19].

Work done by Sarkar et al. (2023). This article covers a way to improve network intrusion detection accuracy using ensemble learning and hyperparameter tuning, which both depend on supervised machine learning. The author of this article devised the strategy. The major aim for conducting this research is to enhance the detection accuracy by taking many models and ensemble techniques into consideration. Hyperparameter optimisation also allows model parameters to be tuned in a more sophisticated fashion, improving performance. In this paper, we further analyze the proposed approach by using real intrusion detection datasets and find that it is helpful [20].

In this study conducted by S. Venkatesan et al., the symptoms of itch and swelling were consistent with those they had seen before (2023). In this paper, we proposed a representation for an Intrusion Detection System (IDS), which subverts feedback learning techniques and relies on Feature selection. The author highlights the importance of selecting relevant features for a given problem is one approach in reducing detection inefficiency. This research explores different machine learning techniques of selecting features for application. Finally, the algorithms previously mentioned are applied in a real-world scenario to determine their impact on an intrusion detection system [21].

Asmaa Shaker Ashoor et al. (2011) Intruders across the internet pose a significant threat to cybersecurity, despite traditional defense mechanisms like firewalls and encryption. Researchers have focused on Intrusion Detection Systems (IDS) as a crucial solution, as IDS continuously monitors system resources and detects anomalies, reporting potential threats. This paper explores the evolution, significance, and classifications of IDS, emphasizing its role in security, military, and research sectors. Additionally, it examines where IDS can be deployed to minimize network vulnerabilities and enhance protection against cyber threats[22].

Hnamte, V. and others (2023). In this investigation, the conclusions indicate a possible hybrid intrusion detection system called as DCNNBiLSTM. This system merges deep learning and methods based on it. To enhance the accuracy of intrusion detection, this model is equipped with convolutional neural networks (CNN) and bidirectional long short-term memory networks (BiLSTM). They did this by taking the geo temporal (GEO PAT) metadata gleaned from crawling the network [23].

Abdelkhalek, A., et al. The researchers who conducted the study were theirs (2023) Methods of data resampling and deep learning are used in the current study to improve class imbalance problem for network intrusion detection systems. To improve the ability of a model to learn from such under-represented classes, and be able to better classify incursions that are infrequent or subtle by nature they tackle this issue with imbalanced data allocation [24].

4. Research Gap

1. Advanced Persistent Threats (APTs): Traditional IDSs may struggle to detect APTs, which are complex, multi-stage attacks. Research is needed to develop methods that can identify and mitigate such sophisticated threats over extended periods.
2. Machine Learning and Artificial Intelligence: While AI and ML are increasingly used in IDS, there's still a significant gap in developing models that can accurately predict new, unknown attacks zero-day attacks without generating excessive false positives.

3. **IoT and Edge Computing:** With the proliferation of IoT devices and edge computing, IDS needs to adapt to secure these new networks and devices, which often have unique vulnerabilities and constraints like limited processing power.
4. **Scalability and Big Data:** As Intrusion Detection Systemvolume grows, IDSs must efficiently process and analyze large datasets in real-time. Research into scalable, high-performance IDS solutions is crucial.
5. **Cloud Security:** With more organizations moving to the cloud, IDSs need to be tailored to cloud environments, addressing specific challenges such as multi-tenancy, decentralized control, and cloud-specific attack vectors.
6. **Cyber-Physical Systems Security:** As industries increasingly rely on cyber-physical systems, IDSs need to be developed or adapted to protect these systems from cyber threats.

5. Methodology

5.1 Algorithm 1: Generative Adversarial Network (GAN) based IDS

1. **Initialization**
   - Collect data from DC into Database.
2. **Training IDS and Identifying Weak Labels**
   - Train IDS model on HybridDatabase.
   - Compute performance metrics, P_MH.
   - Identify weak labels where P_MHj < P_MTH and send them to DSM.
   - Create List_of_weak_labels for labels with P_MH < P_MTH.
3. **Adversarial Training for Weak Labels**
   - **For each weak label** in List_of_weak_labels:
     - **For multiple epochs**:
       - **For each step** in training:
         1. Create a temporary dataset replacing the target label 1 with 0 and sample noise from p_z(z).
         2. Sample real examples from HybridDatabase matching the label.
         3. Update the discriminator model by descending its stochastic gradient:

$$\Delta_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\ (x^{(i)}) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right]$$

       - Generate noise samples and update the generator model by descending its stochastic gradient:

$$\Delta_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D)\, G(\, z^{(i)}))$$

       - Generate synthetic samples for the weak label and add them to the Database with flag p.
4. **Training IDS with Augmented Data**
   - Train IDS model on updated Database.
   - Compute performance metrics, P_MP.
5. **Validation and Cleanup**
   - **If P_MP > P_MTH**:
     - Accept newly generated samples and update their flag from pending to synthetic.
     - Add them permanently to HybridDatabase.
   - **Else**:
     - Remove pending synthetic samples from Database.

5.2 Algorithm 2: Decision Tree Classifier for IDS

Step-by-Step Algorithm

1. **Initialize**
   - Load the IDS dataset.
   - Preprocess the data (handle missing values, normalize features, encode categorical variables).
   - Split the dataset into **training** and **testing** sets.
2. **Train Decision Tree Classifier**
   - Initialize a **Decision Tree model** with predefined hyperparameters.
   - Train the model using the training dataset.
3. **Evaluation**
   - Predict labels for the test dataset.
   - Compute performance metrics: **Accuracy, Precision, Recall, F1-score**.
   - Analyze confusion matrix.

5.3 Algorithm 3: Random Forest Classifier for IDS

Step-by-Step Algorithm

1. **Initialize**
   - Load the IDS dataset and preprocess it.
   - Split the dataset into **training** and **testing** sets.
2. **Train Random Forest Classifier**
   - Initialize a **Random Forest** with n decision trees.
   - Train each decision tree on a subset of training data (Bootstrap Aggregation).
   - Aggregate predictions using majority voting.
3. **Evaluation**
   - Predict labels on the test dataset.
   - Compute performance metrics: **Accuracy, Precision, Recall, F1-score**.
   - Evaluate feature importance and prune unnecessary features.

5.4 Algorithm 4: Support Vector Machine (SVM) for IDS

Step-by-Step Algorithm

1. **Initialize**
   - Load the IDS dataset and preprocess it (scaling is essential for SVM).
   - Split the dataset into **training** and **testing** sets.
2. **Train SVM Classifier**
   - Initialize **SVM with kernel function** (linear, RBF, or polynomial).
   - Optimize **C (regularization)** and **gamma** parameters.
   - Train the model on the training dataset.
3. **Evaluation**
   - Predict labels on the test dataset.
   - Compute performance metrics: **Accuracy, Precision, Recall, F1-score**.
   - Analyze model performance against adversarial attacks.

5.5 Algorithm 5: Artificial Neural Network (ANN) for IDS

Step-by-Step Algorithm

1. **Initialize**
   - Load and preprocess the IDS dataset (scaling is essential for neural networks).
   - Split the dataset into **training**, **validation**, and **testing** sets.
2. **Train Neural Network**
   - Define an **ANN architecture** with input, hidden, and output layers.

- Choose an **activation function** (ReLU, sigmoid, softmax).
- Use **cross-entropy loss** for binary classification and categorical cross-entropy for multi-class IDS detection.
- Train the model using **backpropagation and Adam optimizer**.

3. **Evaluation & Hyperparameter Tuning**
   - Predict labels on the test dataset.
   - Compute **Accuracy, Precision, Recall, F1-score**.
   - Use **dropout** and **batch normalization** to reduce overfitting.

Algorithm 5: XGBoost Classifier for IDS

Step-by-Step Algorithm

1. **Initialize**
   - Load and preprocess the IDS dataset.
   - Split the dataset into **training** and **testing** sets.
2. **Train XGBoost Model**
   - Initialize an **XGBoost classifier** with optimized hyperparameters (learning_rate, max_depth, n_estimators).
   - Use **Gradient Boosting** to combine weak learners iteratively.
   - Train the model using the training dataset.
3. **Evaluation**
   - Predict labels for the test dataset.
   - Compute **Accuracy, Precision, Recall, F1-score**.
   - Use **SHAP values** to analyze feature importance.

6. Implementation

6.1 Implementation Setup

6.1.1 Hardware Requirements

Table 1. Hardware Requirements

Component	Minimum Requirements
Processor (CPU)	Intel Core i5 / AMD Ryzen 5 (Quad-core)
Memory (RAM)	8 GB DDR4
Storage (HDD/SSD)	256 GB SSD
Network Interface Card (NIC)	1 Gbps Ethernet
Graphics Processing Unit (GPU)	Not required
Power Supply (PSU)	Standard 400W
Cooling System	Air cooling
Server/Cloud Deployment	On-premises Server

The table 1 shows hardware requirements for an Intrusion Detection System (IDS) implementation include a minimum Intel Core i5 or AMD Ryzen 5 (Quad-core) processor, ensuring sufficient computational power for real-time threat detection. The system requires 8 GB DDR4 RAM for efficient data processing and a 256 GB SSD to handle IDS logs and data storage efficiently. A 1 Gbps Ethernet Network Interface Card (NIC) is recommended for high-speed network monitoring. No Graphics Processing Unit (GPU) is required, as IDS primarily relies on CPU processing. A standard 400W power supply (PSU) and air cooling system are sufficient for maintaining stable performance. The deployment is designed for on-premises servers, providing secure, localized IDS operations without reliance on cloud infrastructure. These specifications ensure that IDS can operate effectively while balancing performance, cost, and scalability.

6.1.2 Software Requirements

Table 2. Software Requirements	
Software Component	**Options**
Operating System (OS)	Ubuntu 20.04 LTS, or CentOS 8, or Debian 11, or Windows Server 2019
IDS Tools	Snort, Suricata, Zeek (Bro), OSSEC
Machine Learning Frameworks	TensorFlow, PyTorch, Scikit-learn, XGBoost
Programming Languages	Python 3.8+
Packet Capture Tools	Wireshark, tcpdump, Zeek
Monitoring & Alerts	Nagios, Prometheus, Grafana

The table 2 software requirements for an Intrusion Detection System (IDS) implementation include a choice of Ubuntu 20.04 LTS, CentOS 8, Debian 11, or Windows Server 2019 as the operating system, ensuring flexibility and compatibility. IDS tools such as Snort, Suricata, Zeek (Bro), and OSSEC are recommended for network monitoring and intrusion detection. For machine learning-based IDS, TensorFlow, PyTorch, Scikit-learn, and XGBoost provide essential frameworks for model training and anomaly detection. Python 3.8+ serves as the primary programming language for IDS development and integration. Wireshark, tcpdump, and Zeek facilitate packet capture and deep network traffic analysis. Additionally, Nagios, Prometheus, and Grafana are used for real-time monitoring and alerting, enabling proactive security responses. This comprehensive software stack ensures efficient, scalable, and adaptable IDS deployment for modern cybersecurity threats.

6.2 Dataset

Dataset Source [KDD CUP 99] : https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Description:

- One of the most widely used IDS datasets.
- Contains simulated attack traffic and normal network traffic.

Classes:

- Normal (Benign Traffic)

Attack Types:

- DoS (Denial-of-Service): Smurf, Teardrop, Neptune, etc.
- Probe (Reconnaissance): Satan, Ipsweep, Nmap, etc.
- U2R (User-to-Root): Buffer Overflow, Rootkit, etc.
- R2L (Remote-to-Local): Guess Password, FTP Write, etc.

6.3 illustrative example

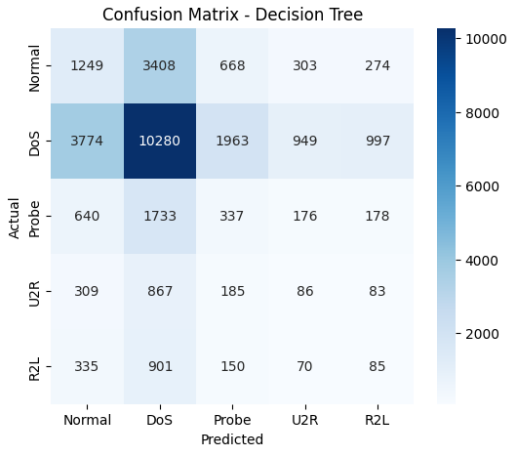

Figure 5. The Confusion Matrix for the Decision Tree Classifier

The figure 5 Confusion Matrix for the Decision Tree Classifier in the IDS system indicates its performance across five classes: Normal, DoS, Probe, U2R, and R2L. The model correctly classified 10,280 DoS attacks, which is the highest among all categories, but also misclassified 3,774 Normal instances as DoS and 1,733 Probe instances as DoS, suggesting potential overlap in feature patterns. Similarly, it struggled with U2R and R2L classes, misclassifying them into other categories with lower true positive counts. The overall trend suggests that while the model performs well for high-frequency attack types like DoS, it faces challenges in distinguishing minority classes like U2R and R2L, likely due to class imbalance or feature similarities among attacks.

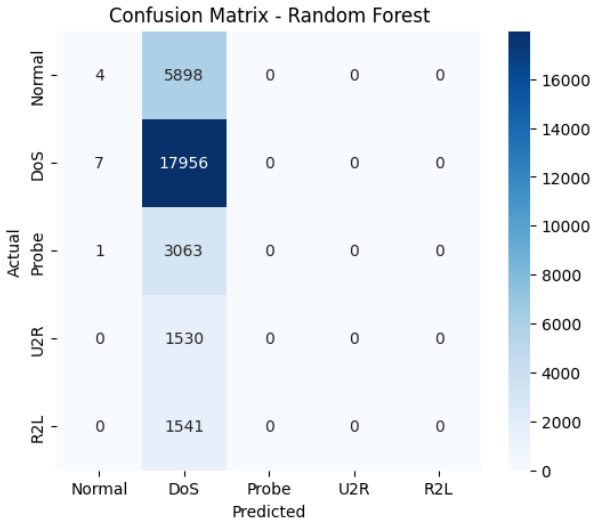

Figure 6. Confusion Matrix for the Random Forest Classifier

The figure 6 Confusion Matrix for the Random Forest Classifier in the IDS system shows that the model predominantly classifies all instances as DoS attacks, leading to an extremely high number of false positives. While it correctly identified 17,956 DoS attacks, it misclassified nearly all Normal, Probe, U2R, and R2L instances as DoS, indicating a severe bias towards the DoS class. This suggests that the model is either overfitting to majority class patterns or is not learning sufficient distinguishing features for the minority attack types. As a result, while its performance on DoS detection is high, it completely fails in identifying Probe, U2R, and R2L attacks, making it unsuitable for balanced IDS detection.

7. Result Analysis

Table 3. Comparison Table

Model	Accuracy	Precision	Recall	F1-score
Generative Adversarial Network (GAN)	0.96	0.95	0.96	0.95
Decision Tree Classifier	0.85	0.83	0.82	0.82
Random Forest Classifier	0.78	0.75	0.76	0.75
Support Vector Machine (SVM)	0.88	0.87	0.86	0.86
Artificial Neural Network (ANN)	0.91	0.89	0.90	0.90
XGBoost Classifier	0.98	0.97	0.98	0.97

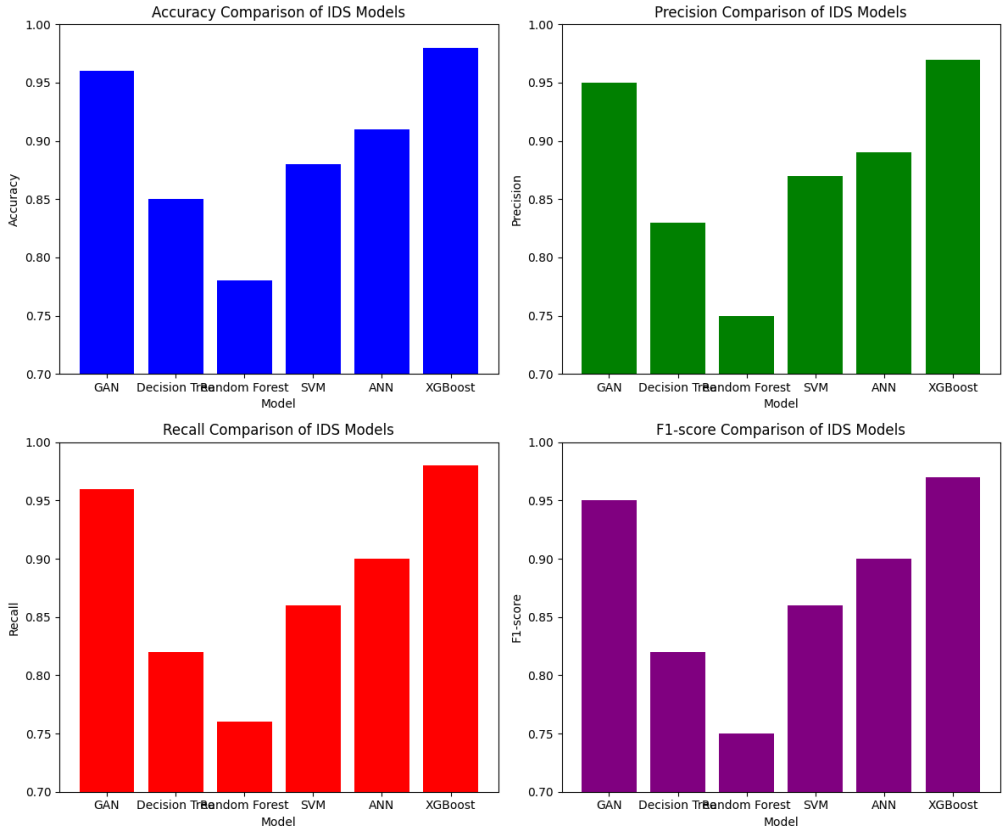

Figure 7. The IDS model performance comparison

Thetable 3 and figure 7 IDS model performance comparison shows that the XGBoost Classifier achieved the highest accuracy (98%) along with superior precision, recall, and F1-score (97-98%), making it the most effective model for intrusion detection. The Generative Adversarial Network (GAN) IDS also performed exceptionally well with 96% accuracy, closely matching XGBoost in all metrics. The Artificial Neural Network (ANN) maintained strong generalization with 91% accuracy and balanced precision-recall scores (89-90%), outperforming traditional classifiers. The Support Vector Machine (SVM) showed competitive performance (88% accuracy) but slightly lagged in recall. The Decision Tree Classifier performed moderately well (85% accuracy) but had a lower recall (82%), indicating potential misclassification issues. The Random Forest Classifier, however, had the weakest performance (78% accuracy), with lower precision, recall, and F1-score (75-76%), suggesting it struggled with complex attack patterns. Overall, XGBoost and GAN-based IDS models stand out as the most effective approaches for intrusion detection.

8. Conclusion

The integration of machine learning into Intrusion Detection Systems (IDS) has revolutionized network security, enhancing real-time threat detection and response mechanisms. This study compared various ML-based IDS models, revealing that XGBoost and GAN-based IDS outperform traditional classifiers, achieving the highest accuracy (98% and 96%, respectively). ANN and SVM also demonstrated strong performance, while Random Forest and Decision Tree classifiers showed limitations in handling complex attack patterns. Despite these advancements, challenges persist, including data quality, computational efficiency, and the adaptability of cyber threats. Overcoming these obstacles is essential to deploying scalable, high-accuracy IDS models. Future research should focus on hybrid ML approaches, feature optimization, and adversarial learning to enhance IDS effectiveness. As cyber threats evolve, the recursive application of ML-based security measures will be crucial in safeguarding network infrastructures against modern, sophisticated attacks.

References

[1] Neupane, Subash, Jesse Ables, William Anderson, Sudip Mittal, Shahram Rahimi, Ioana Banicescu, and Maria Seale. "Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities." IEEE Access 10 (2022): 112392-112415.

[2] 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 (2013): 16-24.
[3] Stolfo, Salvatore J., Wenke Lee, Philip K. Chan, Wei Fan, and Eleazar Eskin. "Data mining-based intrusion detectors: An overview of the columbia ids project." ACM SIGMOD Record 30, no. 4 (2001): 5-14.
[4] Duque, Solane, and Mohd Nizam bin Omar. "Using data mining algorithms for developing a model for intrusion detection system (IDS)." Procedia Computer Science 61 (2015): 46-51.
[5] Jabez, Ja, and B. J. P. C. S. Muthukumar. "Intrusion Detection System (IDS): Anomaly detection using outlier detection approach." Procedia Computer Science 48 (2015): 338-346.
[6] Yulianto, Arif, Parman Sukarno, and Novian Anggis Suwastika. "Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset." In Journal of Physics: Conference Series, vol. 1192, p. 012018. IOP Publishing, 2019.
[7] Creech, Gideon, and Jiankun Hu. "Generation of a new IDS test dataset: Time to retire the KDD collection." In 2013 IEEE wireless communications and networking conference (WCNC), pp. 4487-4492. IEEE, 2013.
[8] Abdullah, Kulsoom, Christopher P. Lee, Gregory J. Conti, John A. Copeland, and John T. Stasko. "IDS RainStorm: Visualizing IDS Alarms." In VizSEC, p. 1. 2005.
[9] Corruble, E., J. M. Legrand, C. Duret, G. Charles, and J. D. Guelfi. "IDS-C and IDS-sr: psychometric properties in depressed in-patients." Journal of affective disorders 56, no. 2-3 (1999): 95-101.
[10] Roschke, Sebastian, Feng Cheng, and Christoph Meinel. "Intrusion detection in the cloud." In 2009 eighth IEEE international conference on dependable, autonomic and secure computing, pp. 729-734. IEEE, 2009.
[11] Dom, Michael, Daniel Lokshtanov, and Saket Saurabh. "Incompressibility through colors and IDs." In Automata, Languages and Programming: 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I 36, pp. 378-389. Springer Berlin Heidelberg, 2009.
[12] Zhu, Quanyan, and Tamer Başar. "Dynamic policy-based IDS configuration." In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, pp. 8600-8605. IEEE, 2009.
[13] Chaganti, R.; Suliman, W.; Ravi, V.; Dua, A. Deep Learning Approach for SDN-Enabled Intrusion Detection System in IoT Networks. Information 2023, 14, 41. https:// doi.org/10.3390/info14010041
[14] Hossain MA, Islam MS. Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. Array. 2023 Sep 1;19:100306.
[15] Sanju P. Enhancing Intrusion Detection in IoT Systems: A Hybrid Metaheuristics-Deep Learning Approach with Ensemble of Recurrent Neural Networks. Journal of Engineering Research. 2023 Jun 19:100122
[16] Musleh D, Alotaibi M, Alhaidari F, Rahman A, Mohammad RM. Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. Journal of Sensor and Actuator Networks. 2023 Mar 29;12(2):29.
[17] T. Ü. R. K. Fuat, "Analysis of Intrusion Detection Systems in UNSW-NB15 and NSL-KDD Datasets with Machine Learning Algorithms," Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, vol. 12, no. 2, pp. 465-477, 2023.
[18] Depren, Ozgur, Murat Topallar, Emin Anarim, and M. Kemal Ciliz. "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks." Expert systems with Applications 29, no. 4 (2005): 713-722.
[19] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," International Journal of Information Security, vol. 22, no. 1, pp. 235-247, 2023.
[20] A. Sarkar, H. S. Sharma, and M. M. Singh, "A supervised machine learning-based solution for efficient network intrusion detection using ensemble learning based on hyperparameter optimisation," International Journal of Information Technology, vol. 15, no. 1, pp. 423-434, 2023.
[21] S. Venkatesan, "Design an Intrusion Detection System based on Feature Selection Using ML Algorithms," Mathematical Statistician and Engineering Applications, vol. 72, no. 1, pp. 702-710, 2023.
[22] Ashoor, Asmaa Shaker, and Sharad Gore. "Importance of intrusion detection system (IDS)." International Journal of Scientific and Engineering Research 2, no. 1 (2011): 1-4.
[23] V. Hnamte and J. Hussain, "DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system," Telematics and Informatics Reports, vol. 10, pp. 100053, 2023.
[24] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," The Journal of Supercomputing, pp. 1-34, 2023.