INTRUSION DETECTION SYSTEM

Prof. Megha Jain
Airuddha K., Shantanu w.
Atharva K. and Siddhant C.

B.E., Department of Artificial
Intelligence and Data Science
SIES Graduate School of
Technology, Nerul, Navi Mumbai.

Abstract— **In today's digital landscape, securing networks and systems against unauthorized access is paramount. This project focuses on the development of an Intrusion Detection System (IDS) using Python, aimed at identifying potential threats and unusual activities within a network. The IDS leverages various machine learning algorithms to analyze network traffic and detect anomalies that may signify security breaches. Key Python libraries, such as Pandas for data manipulation and Scikit-Learn for implementing machine learning models, are utilized to process large datasets and accurately classify normal and intrusive patterns. The system is tested on benchmark datasets, and the performance is evaluated based on accuracy, detection rate, and false-positive rate. This project not only demonstrates the effectiveness of Python for cybersecurity applications but also highlights the IDS's potential as a proactive tool for protecting organizational assets from cyber threats**.

Keywords—Intrusion detection system, cyber security, network traffic ,Data analysis

Introduction

As technology advances, cyber threats have become increasingly sophisticated, making network security a critical concern for organizations worldwide. Intrusion Detection Systems (IDS) play a pivotal role in safeguarding networks by monitoring and analyzing traffic for signs of malicious activities. These systems are designed to detect unauthorized access, policy violations, and other potentially harmful actions that could compromise network integrity.

Modern networks face a wide range of security threats that can compromise data integrity, confidentiality, and availability. Some of the most prevalent challenges include:

- **Intrusion Detection**: Unauthorized access attempts can lead to data breaches, loss of sensitive information, and disruption of services. Detecting such intrusions promptly is crucial for minimizing damage.
- **Anomaly Detection**: Malicious activities often generate unusual network behaviors. Identifying these anomalies — such as unusual traffic spikes or unexpected protocol usage — is key to early threat detection.
- **Malware Detection**: Networks are constantly targeted by various types of malware, including ransomware, spyware, and worms. Recognizing and blocking these threats before they execute is essential to maintaining network security.
- **Phishing Attacks**: Cybercriminals often use deceptive methods to steal user credentials or trick users into performing harmful actions. Identifying suspicious traffic patterns associated with phishing campaigns can help mitigate this threat.
- **Distributed Denial of Service (DDoS) Attacks**: Overwhelming a network with a flood of traffic can cause service outages and financial losses. Distinguishing between legitimate surges in traffic and DDoS attacks is vital for ensuring service availability.

Traditional IDS models often rely on predefined rules and signature-based detection, which can struggle to identify new or evolving threats. To address this limitation, the proposed system employs anomaly-based detection, which can recognize unfamiliar patterns and adapt to new threat landscapes.

Role of Machine Learning in Network Security

Machine learning (ML) offers a powerful, data-driven approach to network security. Here's why it's particularly suited for tackling these challenges:

- **Predictive Capabilities**: ML models can be trained on historical data to learn patterns of normal and malicious behavior. This enables the system to predict and identify potential intrusions based on past attack patterns.
- **Adaptability**: Unlike rule-based systems that require constant manual updates, ML algorithms can adapt to new and evolving threats. By continuously learning from new data, the IDS remains effective even as attack strategies change.
- **Automation**: ML-driven IDS can automate the detection and alerting processes, significantly reducing the need for human intervention. This allows cybersecurity teams to focus on strategic threat mitigation rather than manual monitoring.
- **Feature Engineering and Data Handling**: Python's robust ecosystem — including libraries like Pandas for data manipulation and Scikit-Learn for machine learning — empowers the IDS to process large network datasets efficiently. Features such as packet size, protocol type, and connection duration can be extracted, analyzed, and modeled to distinguish between benign and malicious traffic.

- **High Detection Accuracy**: By leveraging algorithms such as Decision Trees, Random Forest, and Support Vector Machines (SVM), ML-based IDS can achieve high accuracy rates in classifying traffic as normal or intrusive. Techniques like cross-validation and hyperparameter tuning further enhance performance.
- **Behavioral Analysis**: Anomaly detection algorithms, such as Isolation Forest or k-means clustering, allow the system to learn what "normal" behavior looks like. Any significant deviation from this baseline can trigger an alert, even if the specific attack type is unknown.

Utilizing Python’s robust data processing and machine learning libraries, such as Pandas and Scikit-Learn, the IDS is capable of handling large volumes of network data. By training the system on well-known intrusion datasets, it can classify normal and intrusive traffic patterns, enabling real-time threat detection. This project not only underscores the effectiveness of Python as a versatile tool for cybersecurity applications but also demonstrates the potential of machine learning in enhancing network defense mechanisms.

In this research paper, we detail the system architecture, the data preprocessing steps, the selection of machine learning algorithms, and the performance evaluation metrics. The results demonstrate the IDS’s capacity to accurately detect intrusions, highlighting its relevance in today’s cybersecurity landscape.

Literature Review

1. A Comprehensive Review of AI-Based Intrusion Detection Systems

This paper analyzes 72 research studies, focusing on the algorithms and performance metrics utilized in AI-based IDS. It highlights that while AI methods improve detection accuracy, there's a predominant focus on detecting attacks rather than classifying individual attack types.

2. Survey of Intrusion Detection Systems: Techniques, Datasets, and Challenges

This survey categorizes IDS methodologies into Signature-based and Anomaly-based systems, discussing their respective strengths and limitations. It emphasizes the challenges posed by sophisticated cyber-attacks and the necessity for advanced detection techniques.

3. Intrusion Detection Systems Using Supervised Machine Learning Techniques

The study investigates the application of supervised machine learning methods in IDS, providing a taxonomy linking intrusion detection systems with specific algorithms. It offers an in-depth discussion on various cyber-security attacks and the effectiveness of machine learning techniques in detecting them.

4. Meta-Analysis and Systematic Review for Anomaly Network Intrusion Detection Systems: Detection Methods, Dataset, Validation Methodology, and Challenges

This comprehensive study reviews anomaly-based NIDS models, discussing detection methods, datasets, validation methodologies, and associated challenges. It provides a thorough benchmarking assessment of current NIDS-based publications.

5. Intrusion Detection Systems Using Support Vector Machines on the KDDCUP'99 and NSL-KDD Datasets: A Comprehensive Survey

This survey reviews intrusion detection techniques that apply Support Vector Machines (SVMs) as classifiers, focusing on studies evaluated on the KDDCUP'99 and NSL-KDD datasets. It summarizes each method, highlighting performance measures, strengths, and limitations.

II. PROPOSED SYSTEM

The proposed goal for an Intrusion Detection System (IDS) is to enhance the security posture of the organization by identifying, monitoring, and mitigating unauthorized access or malicious activity within the network or system environment. This can be broken down into specific objectives:

1. Threat Detection:

- **Detecting Unauthorized Access:** Identify attempts to gain unauthorized access to the network or system resources, whether internally or externally.
- **Recognizing Malicious Behavior:** Detect patterns or behaviors indicative of cyber threats such as malware, denial-of-service (DoS) attacks, or brute force attempts.

2. Real-time Monitoring and Alerts:

- **Continuous Network Monitoring:** Provide real-time analysis of network traffic and system activity to identify suspicious or abnormal behavior as early as possible.
- **Real-time Alerts and Reporting:** Generate alerts for security personnel in real-time to enable immediate investigation and response to potential threats.

3. Reducing False Positives and Negatives:

- **Minimize False Positives:** Fine-tune detection algorithms to ensure legitimate activities are not mistakenly flagged as threats.
- **Minimize False Negatives:** Ensure that the system does not overlook actual threats, achieving a high detection accuracy.

4. Proactive Defense and Incident Response:

- **Automated Response Capabilities:** Enable automatic defensive actions (e.g., blocking malicious traffic, isolating compromised hosts) upon detection of critical threats.

System Architecture

System Overview

The IDS architecture comprises key components that work together to ensure comprehensive threat detection and mitigation. The primary components include:

- **Data Collection Module:** Captures data from network traffic, system logs, and user behavior.
- **Preprocessing Module:** Cleans and normalizes raw data to prepare it for analysis.
- **Feature Engineering:** Extracts key features such as packet size, source/destination IP addresses, connection duration, and protocol type.
- **Machine Learning Module:** Implements a supervised learning model (e.g., Random Forest or Support Vector Machine) to classify traffic as normal or malicious based on historical data.
- **Detection/Prediction Module:** Uses the trained model to detect anomalies or predict attacks in real time.
- **Alert and Response System:** Generates alerts and, if necessary, triggers automatic mitigation actions.

Data Flow

1. **Data Input:** Network traffic and logs are collected.
2. **Preprocessing:** Data is cleaned and structured.
3. **Feature Extraction:** Relevant features are derived from the raw data.
4. **Model Training:** The machine learning model is trained on labeled datasets.
5. **Prediction/Detection:** New data is analyzed, and the model classifies it as normal or suspicious.
6. **Alert/Response:** Alerts are generated, and mitigation steps are executed.

Key Modules

- **Data Collection Module:** Captures data from multiple sources (network traffic, logs, user activity).
- **Preprocessing Module:** Handles data cleaning, normalization, and transformation.
- **Feature Engineering:** Extracts critical attributes like packet size, source IP, destination IP, protocol type, and connection duration.
- **Machine Learning Module:** Implements supervised learning (e.g., Random Forest) for high accuracy. This model was chosen for its ability to handle large datasets and balance between precision and recall.
- **Detection/Prediction Module:** Analyzes data streams, comparing behavior against trained models to detect threats.
- **Automated Response Module:** Initiates immediate actions, such as blocking malicious IPs or isolating compromised hosts.

Machine Learning Algorithms and Models

- **Model Selection:** Implements algorithms like Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks. Deep learning approaches such as CNNs, RNNs, LSTM, and Autoencoders are also considered for complex pattern recognition.
- **Training Process:** Uses cross-validation, hyperparameter tuning, and data augmentation to enhance model performance.
- **Model Performance:** Evaluated using metrics like accuracy, precision, recall, F1 score, and ROC curves to ensure high reliability.

- **Optimization:** Techniques include gradient descent, learning rate adjustments, and ensemble methods like boosting and bagging to improve performance.

Dataset

- **Type of Data:** Labeled network traffic data.
- **Data Sources:** The system uses the KDD Cup 1999 dataset, a benchmark dataset for intrusion detection.
- **Data Size and Splits:** The data is split into 70% training, 15% validation, and 15% testing sets to ensure a balanced model evaluation.

III. SYSTEM IMPLEMENTATION

- **Technological Stack:** Python, TensorFlow, Keras, Scikit-learn, and Pandas for machine learning and data manipulation. Wireshark and Tshark for packet capture.
- **Hardware Requirements:** Minimum configuration includes a modern CPU, 16GB RAM, and GPU (NVIDIA RTX 3060 or equivalent) for deep learning model training.
- **Real-time vs. Offline:** The system operates in real-time for live network monitoring, with an option for offline analysis on historical data for performance benchmarking.

1)Gaussian Naïve Bayes

In an Intrusion Detection System (IDS), the Gaussian Naive Bayes algorithm is used to classify network traffic as normal or malicious based on the assumption that features are normally distributed. It works well with smaller datasets and is computationally efficient, making it suitable for real-time detection. The model calculates the likelihood of each feature belonging to a class and uses Bayes' theorem to predict the probability of an intrusion. Despite its simplicity, it may struggle with complex data where feature dependencies exist. It is often used in combination with other algorithms for improved accuracy.

2) Decision Tree

In an Intrusion Detection System (IDS), the Decision Tree algorithm is used to classify network traffic by splitting the data into branches based on feature values, ultimately leading to a decision (normal or malicious). It works by creating a tree structure where each internal node represents a feature test, and each leaf node represents a class label. Decision Trees are easy to interpret and handle both categorical and numerical data. They can capture complex decision boundaries but are prone to overfitting, so pruning techniques are often used to improve generalization. They perform well with large datasets and non-linear relationships.

3)Random Forest

In an Intrusion Detection System (IDS), the Random Forest algorithm is used for classification by creating an ensemble of multiple decision trees, each trained on random subsets of the data. This technique reduces overfitting compared to a single decision tree and improves prediction accuracy. Each tree votes for a class (normal or malicious), and the final decision is made based on majority voting (classification) or averaging (regression). Random Forest can handle large datasets and complex features while providing robustness to noise and missing data. It also offers feature importance insights, helping to identify the most relevant attributes for intrusion detection.

4)Support Vector Machine

In an Intrusion Detection System (IDS), Support Vector Machine (SVM) is used to classify network traffic by finding the optimal hyperplane that separates different classes (normal and malicious) in a feature space. SVM works well with both linear and non-linear data by using kernel functions (e.g., linear, polynomial, RBF) to map input features into higher-dimensional spaces for better separation. It is effective in cases where the data is not clearly separable and is robust to overfitting, especially in high-dimensional spaces. SVM excels in precision but can be computationally intensive with large datasets.

5)Logistic Regression

In an Intrusion Detection System (IDS), Logistic Regression is used to classify network traffic as either normal or malicious by estimating the probability that an input belongs to a particular class. It models the relationship between the input features and the binary outcome using a logistic function, outputting probabilities that are then thresholded for classification. Logistic Regression is computationally efficient and works well when the classes are linearly separable. It is interpretable and often used as a baseline model for IDS. However, it may struggle with complex non-linear patterns in the data, requiring feature engineering or combination with other algorithms.

6)Gradient Boosting

In an Intrusion Detection System (IDS), Gradient Boosting is used to build a strong classifier by sequentially training multiple weak learners (typically decision trees), each one correcting the errors of its predecessor. The model focuses on instances that were misclassified by previous trees, progressively improving detection accuracy for both normal and malicious traffic. Gradient Boosting is highly effective for handling complex patterns and non-linear relationships in IDS datasets. It can achieve high accuracy but may be slower to train compared to other models, and it requires careful tuning of hyperparameters (e.g., learning rate, number of estimators) to avoid overfitting.

7)Artificial Neural Networks

In an Intrusion Detection System (IDS), Artificial Neural Networks (ANN) are used to detect complex patterns and relationships in network traffic, classifying it as normal or malicious. ANNs consist of multiple interconnected layers (input, hidden, and output layers) where each neuron applies weights and activation functions (e.g., ReLU, sigmoid) to process the data. The network learns feature representations through backpropagation, adjusting weights to minimize error. ANN is powerful in capturing non-linear patterns and works well with large, high-dimensional datasets. However, it can be computationally intensive and requires careful tuning of parameters (e.g., number of layers, learning rate) to ensure convergence and avoid overfitting.

In conclusion, leveraging multiple machine learning algorithms, such as Gaussian Naive Bayes, Decision Trees, Random Forest, Support Vector Machines, Logistic Regression, Gradient Boosting, and Artificial Neural Networks, can significantly enhance the effectiveness of Intrusion Detection Systems (IDS). Each algorithm brings unique strengths:

- Gaussian Naive Bayes offers speed and simplicity for smaller datasets.

- Decision Trees provide interpretability and handle both numerical and categorical data.

- Random Forest enhances accuracy through ensemble learning, reducing the risk of overfitting.

- SVM excels in separating complex data with a robust approach.

- Logistic Regression serves as an efficient baseline for binary classification.

- Gradient Boosting improves prediction through iterative learning, focusing on errors.

- ANN captures intricate patterns and relationships in large datasets.

By integrating these models, an IDS can achieve a higher detection rate of intrusions while minimizing false positives, providing a comprehensive solution for safeguarding network security. Continuous evaluation and tuning of these models are essential to adapt to evolving threats and maintain robust performance in real-world scenarios.

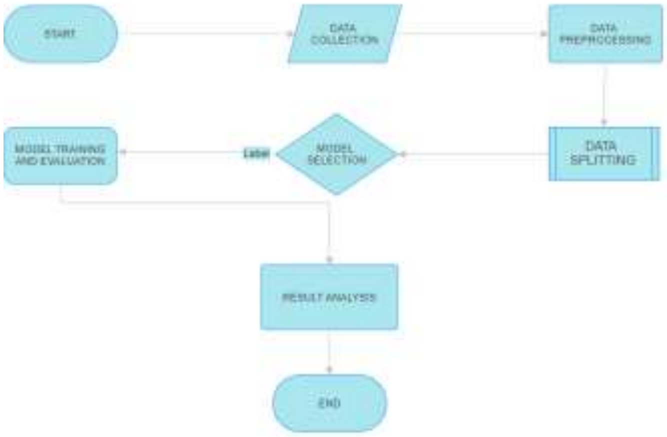


Fig. 1. Flow Chart

Security Considerations

1. Types of Threats Addressed

Our IDS is designed to detect and mitigate a variety of cyberattacks, including:

- Unauthorized Access: Detects suspicious login attempts, brute-force attacks, or privilege escalation attempts.
- Denial of Service (DoS/DDoS): Identifies traffic patterns indicative of DoS attacks, like an unusually high number of requests from a single IP.
- Malware Infection: Recognizes behavior patterns typical of malware (e.g., unusual outbound traffic or repeated failed access to system resources).
- Phishing Attacks: Detects domains or packet payloads containing suspicious or known phishing indicators.
- Port Scanning: Identifies reconnaissance attempts by recognizing frequent probing on multiple ports from the same source.
- Data Exfiltration: Detects large, unexpected data transfers, which might indicate an attacker trying to steal sensitive information.

Result: A wide spectrum of threat detection capabilities, covering both known and emerging attacks.

2. Adaptability

The system is built with adaptability in mind, ensuring it stays effective as attackers evolve their techniques:

- Continuous Learning: Using machine learning models (e.g., Random Forest or LSTM), we enable the system to retrain on new data periodically to recognize novel attack patterns.
- Feature Update: New features (e.g., emerging protocol behaviors, attack signatures) can be added to enhance detection without redesigning the entire system.
- Transfer Learning: For deep learning models, pre-trained weights can be fine-tuned with new data, speeding up adaptation to emerging threats.

Result: The IDS remains resilient against zero-day attacks and evolving threats.

3. False Positives/Negatives

Minimizing incorrect classifications is critical to avoid overwhelming analysts or missing real threats:

- Anomaly Threshold Tuning: The model’s decision thresholds are fine-tuned during the training phase using cross-validation to balance sensitivity (catching attacks) and specificity (reducing false alarms).
- Ensemble Learning: Combining models (e.g., Random Forest + SVM) increases robustness by reducing single-model bias.
- Behavioral Baselines: The system builds a baseline of normal behavior for different times of the day and types of traffic — helping flag deviations without false positives from routine network spikes.
- Confidence Scores: Each prediction includes a confidence score, helping prioritize high-confidence alerts.
- Feedback Loop: Misclassified traffic (false positives/negatives) is logged and added back into the training dataset for periodic retraining, improving accuracy over time.

Result: Reduced false alarms while ensuring malicious activities aren't overlooked.

Challenges and Limitations

1. Performance Trade-offs

High Detection Accuracy vs. Speed:

Achieving high accuracy often comes at the cost of increased processing time, especially for deep learning models. This may cause delays in real-time systems.

Resource Consumption:

Advanced models (e.g., LSTM, CNN) demand significant computational power, memory, and storage — challenging for lightweight or distributed environments.

2. Data Challenges

Imbalanced Datasets:

Cyberattack data is often scarce compared to normal traffic. This imbalance can lead to models biased toward predicting "normal" traffic, increasing false negatives.

Noisy and Incomplete Data:

Real-world network data may be incomplete or noisy (e.g., packet loss, encryption), reducing detection accuracy.

3. Evolving Threat Landscape

Zero-Day Attacks:

IDSs trained on historical data may fail to detect unknown or novel attack patterns — a critical limitation for signature-based detection methods.

Adversarial Evasion:

Attackers can deliberately craft malicious traffic that mimics normal behavior to bypass detection models, especially those reliant on static features.

4. False Positives and False Negatives

High False Positives:

Overly sensitive systems may flood analysts with alerts, causing "alert fatigue" and delayed responses to real threats.

False Negatives:

Missing an attack (false negative) is even more dangerous — attackers may exploit undetected vulnerabilities, leading to data breaches or system damage.

5. Model Limitations

Overfitting:

Machine learning models may overfit on training data, reducing their performance on unseen, real-world traffic.

Explainability:

Complex models like deep neural networks are often "black boxes," making it difficult to interpret why a particular decision was made — a problem for security audits and compliance.

6. Deployment Constraints

Real-time Constraints:

Processing network traffic in real-time requires balancing speed and accuracy. Delayed detection may render the system ineffective against fast-acting attacks like ransomware.

Integration with Legacy Systems:

Many organizations still rely on older infrastructure, which may not support modern IDS implementations without significant redesign or performance degradation.

7. Privacy and Legal Considerations

Deep Packet Inspection (DPI):

Advanced detection may require inspecting packet payloads, raising privacy concerns and compliance challenges (e.g., GDPR, HIPAA).

Encrypted Traffic:

Growing adoption of encryption (TLS 1.3, VPNs) makes traffic analysis harder without decrypting data — potentially violating privacy laws.

IV. Analysis and Results

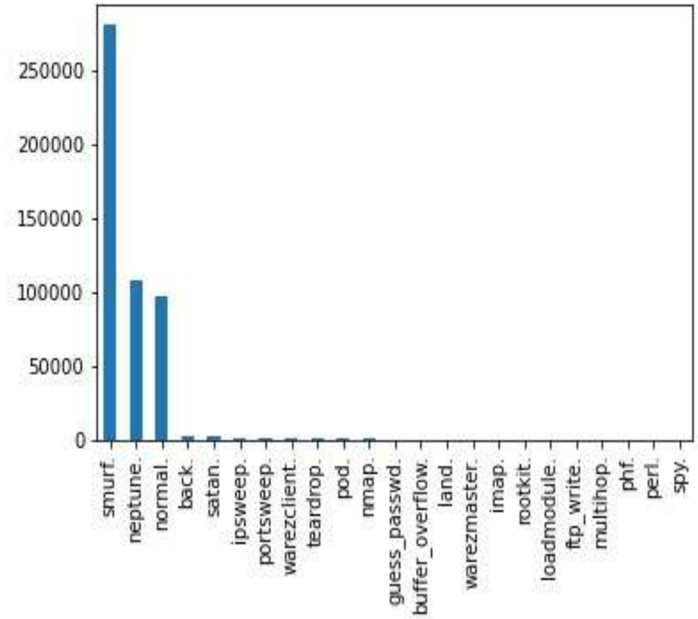


Fig. 2 Bar charts of attacks

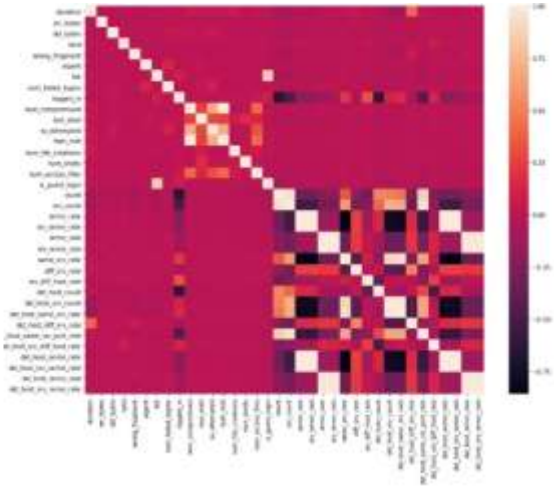
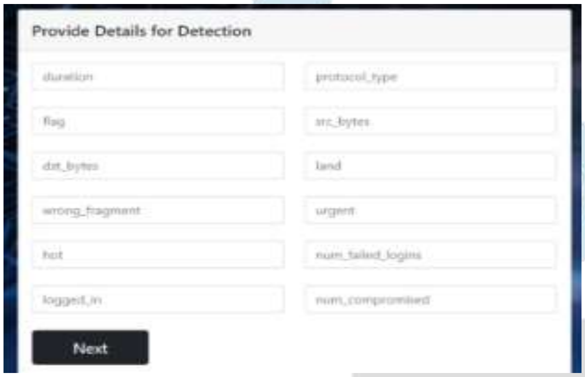


Fig. 3 Heatmap



Provide Details for Detection

duration

protocol_type

flag

src_bytes

dst_bytes

land

wrong_fragment

urgent

hot

num_failed_logins

logged_in

num_compromised

Fig. 4 Interface of Information

V. CONCLUSION AND FUTUREWORK

- The results of each model were compared and analyzed to determine the best-performing algorithm. The analysis included visual representations of model performance through graphs and tables, summarizing key metrics.
- Through this comprehensive methodology, the IDS was developed and evaluated, aiming to achieve high accuracy and reliability in identifying various types of network intrusions using the KDD Cup 1999 dataset. This structured approach ensures a robust assessment of the models' capabilities and provides insights into their applicability in real-world scenarios
- We explored a comprehensive and straightforward analysis for anyone who wants to compare various approaches used to design Network Intrusion Detection models.
- This review is established based on numerous research papers in different journals/publications between 2005 and 2020. In this article, we took citation as a quantitative measure to review the popularity of the intrusion detection system among various approaches .This paper presents various tables that offer a rapid analysis of different NIDS, research trends, and research scope.
- A review of diverse datasets with their characteristics, merits, demerits, and citation analysis has also been presented. The various approaches used in the network intrusion detection system are tabulated with their advantages and disadvantages also.
- A review concerning research trends regarding different techniques in IDS is presented.

REFERENCES

[1] https://ieeexplore.ieee.org/document/9623451

[2] Alazab, M., & Hossain, M. S. (2020). "Intrusion Detection Systems: A Comprehensive Survey." *IEEE Communications Surveys & Tutorials*

[3] glewski, M., & Janus, M. (2022). "Anomaly-Based Intrusion Detection Systems: A Review." *ACM Computing Surveys*

[4] Moustafa, N., & Slay, J. (2015). "The Significant Features of Network Traffic for Intrusion Detection: A Survey."

Proceedings of the 2015 International Conference on Cloud Computing and Big Data

[5] anaie, M. A., & Ullah, S. (2021). "Deep Learning Techniques for Intrusion Detection Systems: A Survey." *IEEE Access*

[6] Faraoun, M., & Djoudi, A. (2019). "A Survey of Intrusion Detection Systems in the Cloud Environment." *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*