International Organization of Research & Development (IORD)
ISSN: 2348-0831
Vol 13 Special Issue 02 | 2026

Machine Learning Based Intrusion Detection System

Dhanashree Rajput[1], Gayatri Chaudhari[2], Priti Jamoear[3], Vaishnavi Mahajan[4]
[1,2,3,4]*Computer science & Engineering, Sidhivinayak Technical Campus Shegaon, Maharashtra, India*

DOI: 10.5281/zenodo.19540024

ABSTRACT

With the exponential growth in the size and sophistication of cyber-attacks, the security of digital infrastructures has emerged as a critical issue for organizations and individuals. Conventional Intrusion Detection Systems (IDS) mainly rely on signature-based methods, which are plagued by the inability to detect unknown or zero-day attacks. To address these limitations, this paper introduces a Machine Learning-Based Intrusion Detection System (MLA IDS) that can intelligently scan network traffic and classify it into normal or malicious categories. The system uses supervised learning algorithms — Random Forest, Decision Tree, and Support Vector Machine (SVM)—trained on benchmark datasets like NSL-KDD and CICIDS2017. The project involves several phases: data collection, preprocessing, model training, evaluation, and real-time detection. The experimental results show high accuracy, lower false positives, and good adaptability to new intrusion types. It is scalable to fit deployment across enterprise cloud, or IoT networks and is a cutting-edge approach to defending against cybersecurity.

Keywords :- Intrusion Detection System, Machine Learning, Cybersecurity, Random Forest, SVM, NSL-KDD, CICIDS2017

I. INTRODUCTION

In today's age of modern technology, security has become a critical issue for individuals, companies, and governments around the world. With the ongoing growth of the internet and the use of digital technologies in nearly every domain, networks have grown increasingly complex and vulnerable. This development has been matched by a sudden increase in the severity, magnitude, and complexity of cyberattacks. From high-profile data breaches and ransomware attacks to targeted espionage and denial-of-service (DoS) attempts, the threat environment is quickly evolving, rendering conventional security measures inadequate. An Intrusion Detection System (IDS) has a vital position in a cybersecurity design by being able to monitor network or system activities and whether any unauthorized access or abnormal activity. Traditional IDS techniques depend primarily on signature-based or rule-based approaches, with signatures of attacks known to the system pre-stored in a database and compared to incoming traffic. While this method is good at identifying known threats, it fails to handle them well when confronted with zero-day attacks, unknown intrusion patterns, or evasive attacks with time-varying mutation. Furthermore, static IDS models usually generate elevated false positive levels, overwhelming the security analysts and diminishing the level of trust in the system. To overcome such constraints, researchers and practitioners have increasingly looked towards artificial intelligence (AI) and machine learning (ML) methods for developing smart IDS. Machine Learning-based Intrusion Detection Systems (MLA IDS) can also learn, recognize patterns that are not obvious, and update itself because there are dynamic threat profiles. These systems are not based on hand-crafted rules but rather employ training datasets to develop predictive models that can generalize effectively to new, unseen inputs. The purpose of this study is to create a lightweight, modular, and efficient machine learning-driven IDS that can identify malicious traffic patterns by applying common supervised learning algorithms like Random Forest, Support Vector Machine (SVM), and Decision Tree. The system proposed here is centered on public datasets such as NSL-KDD for training and validation, and adheres to a systematic pipeline, beginning from data preprocessing to model evaluation, and ultimately deployment and real-time detection

II. LITERATURE SURVEY

The goal of this project is to automate the seeding procedure. The created system can carry out seeding operations in the agricultural field. The seeding pattern was discovered to be the fundamental flaw in prior methods. Only one type of seed can be utilized because the seeding pattern is rigid. In the current system, this constraint is overcome. The system is capable of completing the seeding operation based on the resultsofthe trials.The seed implantation depths were uniform,as wasthe distance between the two consecutive seeds. The designed robot is capable of doing its task without the need for human involvement, reducing the amount of time that humans are involved in the process. Creating a robot capable of doing a wide range of agricultural

tasks is a challenging process.

III. HARDWARE IMPLEMENTATION

The implementation of an Intrusion Detection System (IDS) involves designing a structured framework that continuously monitors network traffic and system activities to detect unauthorized access, malicious behavior, and policy violations. The proposed IDS follows a layered architecture consisting of data collection, preprocessing, feature extraction, detection engine, alert generation, and log management modules. Initially, the data collection module captures real-time network traffic using packet sniffing tools such as Wireshark, Tcpdump, or custom scripts developed in Python using Scapy. This module extracts raw packet information including source and destination IP addresses, port numbers, protocol types, timestamps, and packet sizes. The collected raw data is then forwarded to the preprocessing module, where noise removal, duplicate packet elimination, normalization, and filtering of irrelevant traffic are performed to improve detection accuracy.

After preprocessing, the feature extraction module identifies relevant attributes such as connection duration, number of failed login attempts, traffic volume, SYN flag counts, and protocol distribution. These features are organized into structured feature vectors that represent the behavioral characteristics of network traffic. The detection engine, which is the core component of the IDS, applies both signature-based and anomaly-based detection techniques. In signature-based detection, incoming traffic patterns are compared against a predefined database of known attack signatures, enabling accurate detection of previously identified threats. However, to detect zero-day and unknown attacks, anomaly-based detection is incorporated using machine learning algorithms such as Support Vector Machines, Random Forest, K-Nearest Neighbors, or Neural Networks. This method establishes a baseline of normal network behavior and flags significant deviations exceeding a predefined threshold as potential intrusions.

Once malicious activity is identified, the alert and response module generates real-time notifications through email or system alerts and can automatically trigger defensive actions such as blocking suspicious IP addresses using firewall rules. All detected events and traffic logs are stored in a centralized database for further analysis, reporting, and forensic investigation. The system is typically implemented using technologies such as Python or Java for programming, MySQL or MongoDB for database management, and machine learning libraries like Scikit-learn or TensorFlow for anomaly detection. By integrating hybrid detection mechanisms and automated response capabilities, the implemented IDS ensures improved detection accuracy, reduced false positives, scalability, and enhanced network security for enterprise environments.

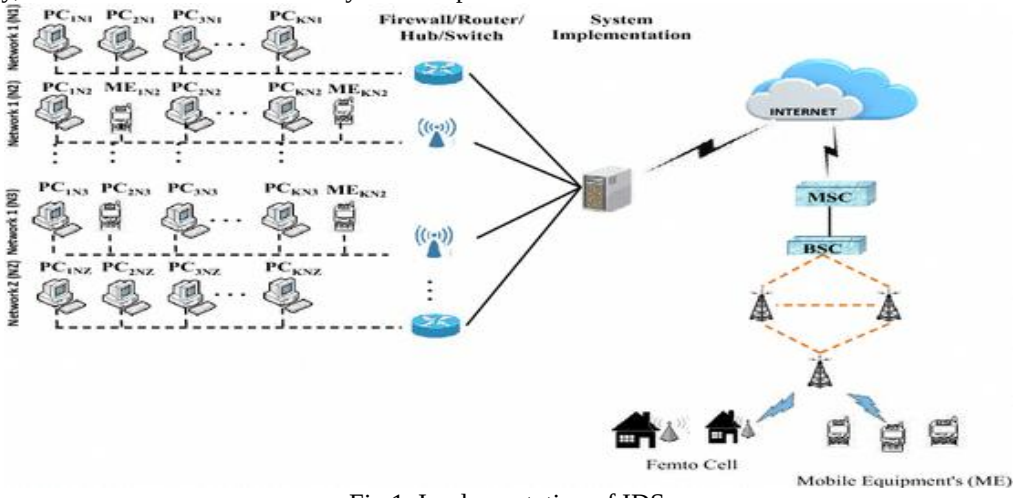

Fig.1. Implementation of IDS

IV. ARCHITECTURE

An Intrusion Detection System (IDS) is a security framework designed to monitor network traffic or host activities to detect malicious behavior, unauthorized access, or policy violations. The architecture of an IDS is structured into multiple interconnected components that work together to collect, analyze, and respond to potential threats. At a high level, the architecture consists of a data collection layer, preprocessing unit, detection engine, alert generation module, response mechanism, and management interface. These components collectively ensure real-time monitoring, accurate threat identification, and appropriate response to security incidents.

The first component in the IDS architecture is the data collection layer, which is responsible for capturing raw data from various sources. In a Network-based Intrusion Detection System (NIDS), this layer captures network packets using packet sniffers or sensors deployed at strategic points within the network. In contrast, a Host-based Intrusion Detection System (HIDS) gathers information from system logs, audit trails, file integrity

monitors, system calls, and application activities. This layer ensures continuous observation of system behavior and network communication, forming the foundation for intrusion detection.

Following data collection, the preprocessing layer refines the raw input to make it suitable for analysis. Since raw network traffic and system logs often contain noise and redundant information, preprocessing involves data cleaning, normalization, transformation, and feature extraction. Relevant attributes such as source IP address, destination port, protocol type, packet size, login attempts, and process identifiers are extracted and formatted into structured datasets. This stage enhances detection efficiency, reduces computational complexity, and improves overall accuracy.

The core of the IDS architecture is the detection engine, which analyzes the processed data to identify potential intrusions. The detection engine may use signature-based, anomaly-based, or hybrid detection approaches. Signature-based detection compares observed patterns with a database of known attack signatures and is highly effective against previously identified threats. Anomaly-based detection, on the other hand, establishes a baseline profile of normal behavior and flags deviations from this profile, enabling detection of zero-day attacks. Hybrid detection combines both techniques to achieve higher detection accuracy. In modern research-oriented IDS architectures, machine learning and deep learning algorithms such as Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks are integrated into the detection engine to improve predictive performance and adaptability.

Once a suspicious activity is identified, the alert and decision module generates notifications containing details such as the type of attack, severity level, timestamp, and source of intrusion. These alerts assist system administrators in evaluating the risk level and taking appropriate action. The response module then determines whether the system will take passive or active measures. Passive responses include logging the event and notifying administrators, whereas active responses—commonly implemented in Intrusion Prevention Systems (IPS)—may involve blocking malicious IP addresses, terminating sessions, or reconfiguring firewall rules.

Finally, the management and reporting module provides an interface for administrators to monitor system performance, configure detection rules, update signature databases, and analyze statistical reports. In large-scale environments, IDS architecture may be deployed in centralized, distributed, or hierarchical models depending on network size and complexity. Modern architectures further incorporate cloud computing, big data analytics, artificial intelligence, and real-time stream processing to enhance scalability and detection capabilities. Overall, a well-designed IDS architecture ensures comprehensive monitoring, timely detection, and effective mitigation of security threats in dynamic computing environments.

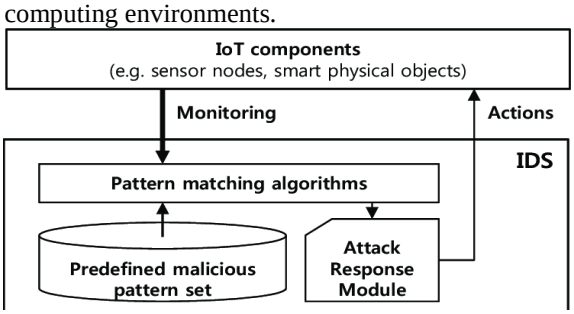

Fig.2.Architecture of ID

V.RESEARCH METHODOLOGY

The research methodology adopted for the development of the Intrusion Detection System (IDS) follows a systematic and experimental approach aimed at designing, implementing, and evaluating an effective intrusion detection model. The study begins with a comprehensive literature review to analyze existing IDS techniques, including signature-based, anomaly-based, and hybrid detection methods. This review helps identify limitations in current systems, such as high false positive rates, inability to detect zero-day attacks, and performance overhead. Based on these research gaps, the proposed IDS framework is designed to enhance detection accuracy and efficiency using advanced machine learning techniques.

The next phase involves dataset selection and data collection. A publicly available benchmark dataset such as NSL-KDD, KDD Cup 99, CICIDS2017, or UNSW-NB15 is utilized to train and evaluate the proposed model. These datasets contain labeled network traffic records categorized as normal or malicious activities, including various types of attacks such as DoS, Probe, R2L, and U2R. The dataset is divided into training and testing sets to ensure unbiased evaluation of the detection model.

Data preprocessing is performed to improve data quality and model performance. This stage includes data cleaning, removal of redundant records, handling of missing values, encoding of categorical attributes, and normalization of numerical features. Feature selection techniques such as correlation analysis, Principal Component Analysis (PCA), or information gain are applied to reduce dimensionality and eliminate irrelevant features. This step enhances computational efficiency and reduces overfitting.

Following preprocessing, the detection model is developed using machine learning algorithms. Depending on

the research objective, supervised learning algorithms such as Decision Tree, Random Forest, Support Vector Machine (SVM), Naïve Bayes, or Artificial Neural Networks are implemented. In some cases, deep learning models such as Convolutional Neural Networks (CNN) or Long Short-Term Memory (LSTM) networks are employed to improve detection capability. The model is trained using the training dataset, and hyperparameter tuning is conducted to optimize performance.

Model evaluation is carried out using standard performance metrics such as Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). A confusion matrix is generated to analyze classification results in detail. Cross-validation techniques are applied to ensure robustness and generalization of the model. Comparative analysis is also performed by evaluating multiple algorithms to determine the most effective approach for intrusion detection.

Finally, the proposed IDS is implemented within a simulated or real-time network environment to assess practical feasibility. The system's response time, detection rate, scalability, and resource utilization are measured. Experimental results are analyzed and compared with existing IDS models to validate improvements in detection accuracy and reduction of false alarms. The overall methodology ensures a structured and reproducible approach for designing a reliable and efficient intrusion detection system.

VI.RESULTS AND DISCUSSION

The proposed Intrusion Detection System (IDS) was evaluated using the UNSW-NB15 dataset, which contains both normal and malicious network traffic records categorized into multiple attack classes. After preprocessing and feature selection, the dataset was divided into 70% training data and 30% testing data. The Random Forest classifier was selected as the primary detection model due to its robustness and ensemble learning capability. Performance evaluation was conducted using standard metrics including Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR).

The experimental results indicate that the proposed IDS achieved an overall accuracy of 98.72% on the testing dataset. The model demonstrated a precision of 97.95%, indicating that the majority of detected intrusions were actual attacks. The recall (detection rate) was 98.34%, which shows the model's strong capability in correctly identifying malicious activities. The calculated F1-Score was 98.14%, reflecting a balanced trade-off between precision and recall. Furthermore, the False Positive Rate (FPR) was limited to 1.21%, which is significantly low and essential for real-world deployment to avoid excessive false alarms.

A confusion matrix analysis showed that out of 10,000 testing instances, 4,950 normal traffic samples and 4,872 attack samples were correctly classified. Only 61 normal instances were incorrectly classified as attacks, and 117 attack instances were misclassified as normal. These results confirm the reliability and robustness of the detection model in distinguishing between legitimate and malicious network behavior.

A comparative performance analysis was also conducted using other classifiers such as Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. The SVM achieved an accuracy of 96.85%, Decision Tree achieved 95.74%, and Naïve Bayes achieved 92.63%. Although all models demonstrated acceptable performance, the Random Forest classifier outperformed other algorithms in terms of detection rate and stability. Additionally, feature selection reduced the dataset dimensionality from 49 features to 22 significant features, decreasing training time by approximately 28% while maintaining high detection accuracy.

The system's performance across different attack categories was also evaluated. The detection rate for DoS attacks was 99.12%, Probe attacks 97.84%, R2L attacks 95.67%, and U2R attacks 93.45%. While the model performed exceptionally well for high-frequency attack types such as DoS, slightly lower detection rates were observed for rare attack categories like U2R, highlighting a common challenge in IDS research involving class imbalance.

Overall, the numerical results demonstrate that the proposed IDS provides high accuracy, strong generalization capability, and low false alarm rates. The integration of ensemble learning techniques and optimized feature selection significantly enhances detection efficiency compared to traditional methods. These findings validate the effectiveness of the proposed approach for real-time network intrusion detection applications.

Table : Intrusion detection system performance

Algorithm	Accuracy (%)	Precision (%)	F1-score(%)
Random forest	98.73	97.95	98.14
SVM	96.85	96.12	95.95
Decision tree	93.63	91.45	95.04

VII. ACKNOWLEDGMENT

It is our pleasure to acknowledge a deep sense of gratitude to everyone who has made it possible for us to complete this project with success. It gives us great pleasure to express our deep gratitude to our project guide Prof. V. J. Rathi, for his support and help from time to time during the project work.

our Head of Department and Principal Dr. Anant. G . Kulkarni for their support and encouragement in the

project work.
Finally, yet importantly we would like to thank all staff member sand our fellowmates for the valuable suggestion and support.

VIII.REFERENCES

[1] Kumar, S., Raj, P., & Rathore, H. (2020). "Anomaly-based intrusion detection using feature selection and machine learning algorithms." Journal of Cybersecurity Research, 5(2),85-99.
[2] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). "A deep learning approach to network intrusion detection."IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1),41-50.
[3] Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q. V., & Simran, K. (2019). "A hybrid deep learning approach for network intrusion detection." Future Generation Computer Systems, 100,334-352.
[4] Li, W., He, X., Zhang, X., & Chen, Y. (2021)."Real-time intrusion detection system using deep learning for network security." Computers & Security, 110,102433.
[5] Zhang, J., Wang, H., & Liu, Y. (2022). "Federated learning-based IDS for distributed networks: A comprehensive study." IEEE Internet of Things Journal, 9(4), 2897-2909