IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412151

INTRUSION DETECTION SYSTEM

Manoj S[1], Rajeshwari N[2]

Department of MCA, BIT, K.R. Road, V.V. Pura, Bangalore, India[1,2]

Abstract: An Intrusion Detection System (IDS) is a critical security mechanism designed to monitor network traffic and system activities to identify malicious actions or policy violations. With the rapid growth of interconnected systems and the increasing sophistication of cyberattacks, traditional security solutions such as firewalls are no longer sufficient on their own. This project focuses on the design and implementation of an effective intrusion detection system that enhances network security by identifying both known and unknown attacks in real time.

The proposed IDS analyzes incoming data packets and system behavior to detect abnormal patterns that may indicate unauthorized access, denial-of-service attacks, or data breaches. Machine learning techniques are employed to learn from historical data and classify activities as normal or malicious, thereby improving detection accuracy and reducing false alarms. The system continuously adapts to new attack patterns, making it suitable for dynamic network environments.

Keywords: Intrusion Detection System, Network Security, Cyber Attacks, Machine Learning, Anomaly Detection, Signature-Based Detection, Network Monitoring, Malicious Traffic, Threat Detection, Data Analysis

I. INTRODUCTION

With the rapid expansion of computer networks and internet-based services, securing digital information has become a major challenge. Organizations increasingly rely on networked systems for data storage, communication, and online transactions, making them attractive targets for cyberattacks. Traditional security mechanisms such as firewalls and access control systems provide basic protection, but they are often insufficient to detect sophisticated or insider attacks. As a result, there is a strong need for intelligent security solutions that can actively monitor and analyze system activities. An Intrusion Detection System (IDS) is designed to identify unauthorized access, misuse, or abnormal behavior within a network or host system. It works by continuously observing network traffic or system logs and comparing them against expected behavior patterns or known attack signatures. When suspicious activity is detected, the IDS generates alerts to inform administrators, enabling timely response and mitigation.

1.1 Project Description

The Intrusion Detection System (IDS) project is designed to enhance network security by continuously monitoring system activities and network traffic to detect unauthorized access and malicious behaviour. As cyber threats continue to evolve in complexity and frequency, conventional security mechanisms alone are not sufficient to ensure complete protection. This project aims to provide an intelligent and automated solution for identifying security breaches in real time.

1.2 Motivation

The rapid growth of digital networks and online services has significantly increased the risk of cyberattacks, making network security a critical concern for organizations and individuals. Attackers continuously develop new techniques to exploit system vulnerabilities, bypass traditional security measures, and gain unauthorized access to sensitive data. Conventional security tools such as firewalls and antivirus software are limited in their ability to detect complex and evolving threats, particularly zero-day and insider attacks.

II. RELATED WORK

Paper [1], This paper presents one of the earliest frameworks for detecting intrusions by analysing network traffic patterns and identifying suspicious activities in real time.
Paper [2], The study introduces the Bro system, which monitors live network traffic and applies policy-based analysis to detect intrusions with high flexibility and accuracy.
Paper [3], This work explains the nature, types, and impact of denial-of-service attacks while highlighting the challenges involved in detecting and mitigating them.
Paper [4], The paper proposes an intrusion detection approach that models normal system behaviour using sequences of system calls to identify anomalies.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412151

Paper [5], This research introduces the concept of self-behaviour modelling in Unix processes to detect intrusions by recognizing deviations from normal execution patterns.

III. METHODOLOGY

The methodology of the proposed Intrusion Detection System focuses on identifying malicious activities by systematically analyzing network traffic and system behavior. The process begins with data collection, where network packets or system logs are gathered from a monitored environment. This data represents both normal activities and potential attack patterns.

In the next stage, data preprocessing is performed to remove noise, handle missing values, and normalize the dataset to ensure consistency. Relevant features such as protocol type, connection duration, packet size, and traffic flow characteristics are then extracted, as these attributes play a crucial role in distinguishing between legitimate and malicious activities.

The processed data is used to train a detection model using machine learning techniques. The model learns behavioral patterns from historical data and classifies incoming activities as either normal or intrusive. During real-time operation, the trained model continuously analyzes live traffic and compares it with learned patterns to detect anomalies or known attack signatures.

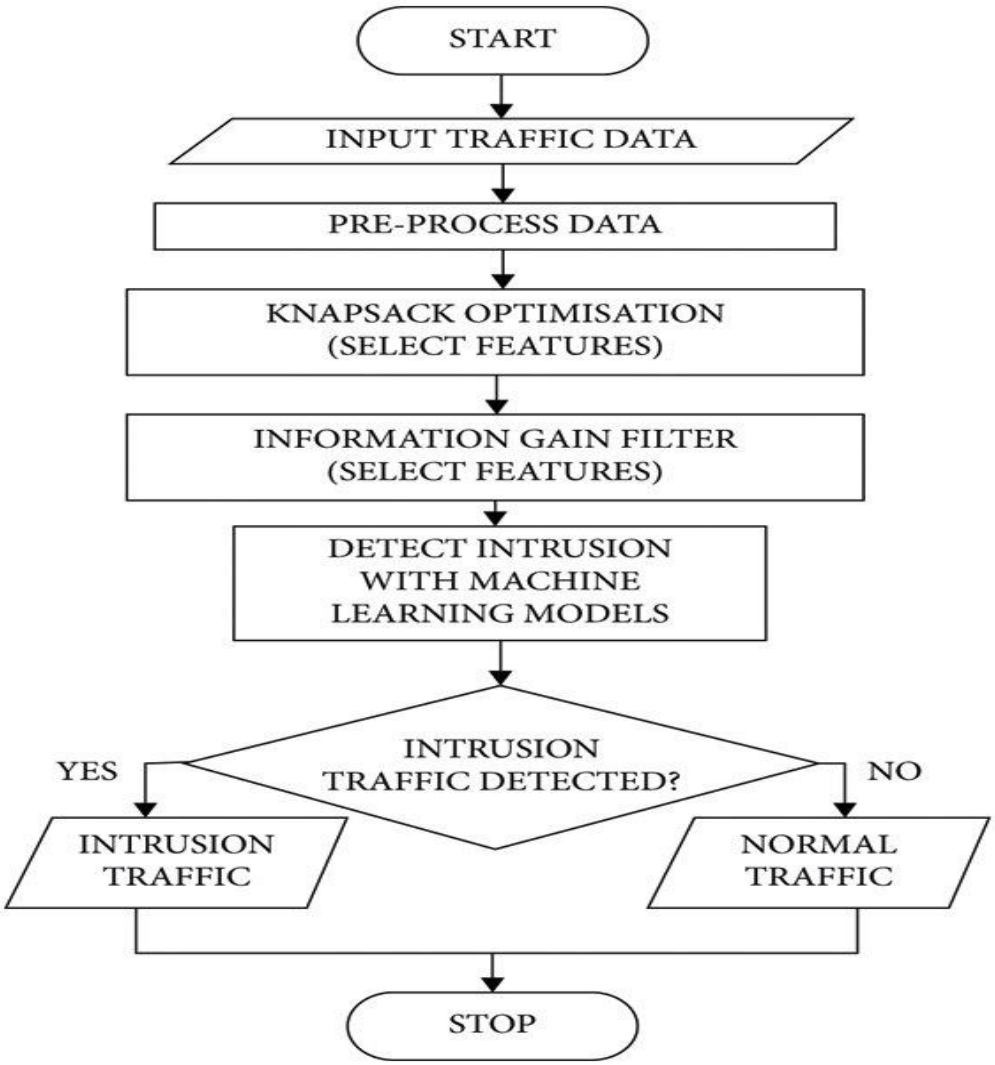

Fig.1.Flowchart

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412151

Implementation Flow

The implementation of the Intrusion Detection System begins with the acquisition of network traffic data from the monitored environment. This traffic data consists of various attributes related to network connections and packet behavior, which serve as the input to the system.

In the next stage, the collected data is pre-processed to improve quality and reliability. Pre-processing includes removing duplicate records, handling missing values, converting categorical data into numerical form, and normalizing feature values. This step ensures that the data is suitable for further analysis and model training.

After pre-processing, feature selection is carried out using Knapsack Optimization. This technique selects an optimal subset of features by maximizing their contribution to intrusion detection while minimizing redundancy and computational cost. Selecting relevant features helps improve detection accuracy and reduces processing time.

Following this, an Information Gain Filter is applied to further refine the selected features. Information Gain measures the importance of each feature based on how well it distinguishes between normal and malicious traffic. Only the most informative features are retained for classification.

Hardware and Software Requirements

- A computer or laptop with at least an Intel i5 processor, 8 GB RAM, sufficient storage, and internet connectivity is required.
- The system runs on Windows or Linux using Python, along with tools like Jupyter Notebook or Google Colab.
- Machine learning libraries such as NumPy, Pandas, and Scikit-learn are used with datasets like NSL-KDD or CICIDS.

IV. IMPLEMENTATION DETAILS

The Intrusion Detection System is implemented using a machine learning–based framework to analyze and classify network traffic. The system begins by importing a standard intrusion detection dataset, which contains records of both normal and attack traffic. The implementation is carried out using Python in a development environment such as Jupyter Notebook or Google Colab.

Initially, the dataset undergoes preprocessing to improve data quality. This includes removing duplicate entries, handling missing values, encoding categorical attributes, and normalizing numerical features. After preprocessing, feature selection techniques are applied to identify the most relevant attributes, which helps in reducing computational complexity and improving detection accuracy.

The processed data is then divided into training and testing sets. Machine learning models are trained using the training dataset to learn patterns associated with normal behavior and intrusion activities. Once trained, the model is evaluated using the testing dataset to validate its performance.

During execution, the trained model analyzes incoming network data and classifies it as either normal traffic or intrusion traffic. The classification results are displayed as output, and intrusion alerts are generated whenever malicious activity is detected. This implementation ensures efficient, accurate, and reliable intrusion detection suitable for experimental and academic purposes.

V. RESULTS AND DISCUSSION

The results obtained from the simulation demonstrate that the proposed Intrusion Detection System is effective in identifying malicious activities within network traffic. After training the model using the selected features, the system was tested with unseen data to evaluate its performance. The IDS was able to accurately classify traffic as normal or intrusive, showing a high detection rate for common attack types.

The evaluation metrics such as accuracy, precision, recall, and detection rate indicate that the system performs reliably with a low false alarm rate. Feature selection contributed to improved performance by reducing redundancy and enhancing the model's ability to focus on significant traffic characteristics. The discussion of results highlights that the machine learning–based approach provides better adaptability compared to traditional rule-based systems. Overall, the outcomes confirm that the proposed IDS is suitable for detecting intrusions in simulated network environments and can serve as a strong foundation for further real-time security implementations.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412151

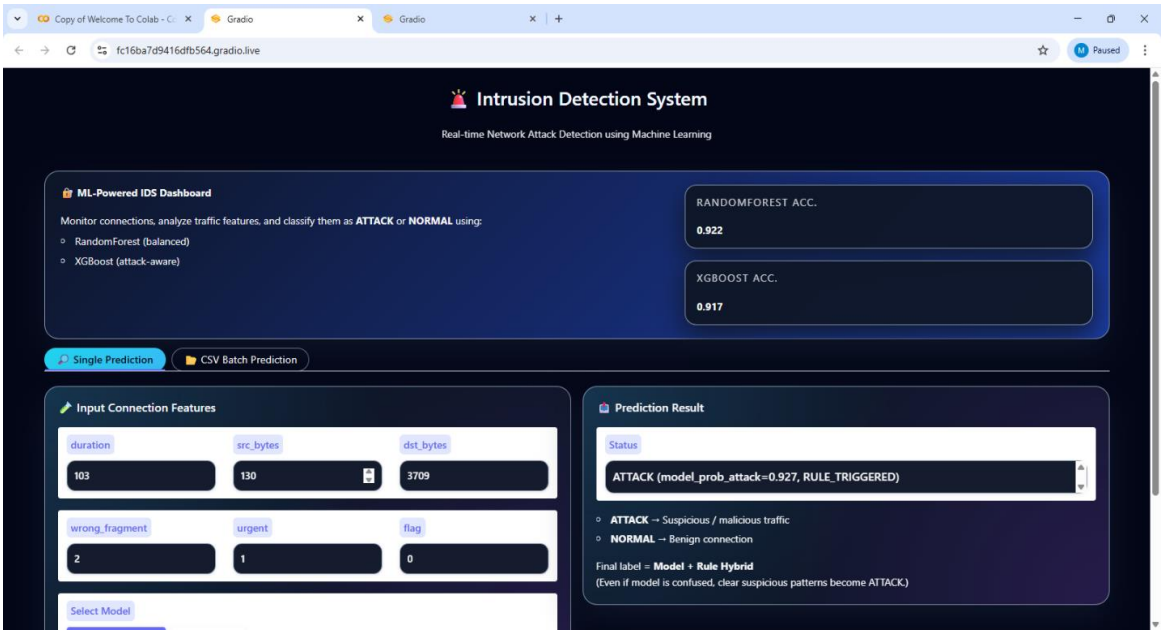

Fig.2. Single Prediction Input Interface

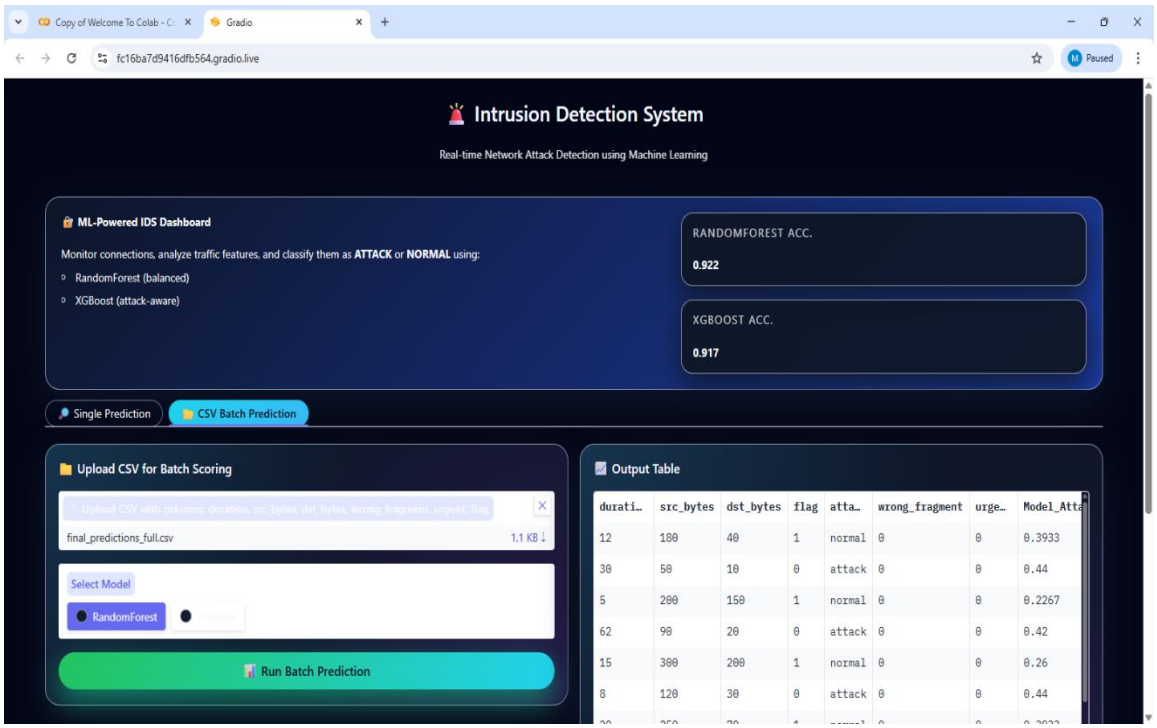

Fig.3. CSV Batch Prediction Module

This image illustrates the batch prediction functionality of the Intrusion Detection System. Users can upload a CSV file containing multiple network records, and the system analyses all entries at once. The resulting output table displays prediction results for each record, enabling efficient detection of intrusions across large datasets.

VI. CONCLUSION

This project successfully demonstrates the design and implementation of an Intrusion Detection System that enhances network security by identifying malicious activities and unauthorized access. By analyzing network traffic patterns and system behavior using machine learning techniques, the system is able to effectively distinguish between normal and intrusion traffic. The use of preprocessing and feature selection improves detection accuracy while reducing false alarms

DOI: 10.17148/IJARCCE.2025.1412151

and computational overhead. The simulation and evaluation results show that the proposed IDS provides reliable and efficient intrusion detection in a controlled environment. Overall, the system offers a practical and scalable security solution that can be further enhanced to address emerging cyber threats in modern network infrastructures.

VII. FUTURE WORK

The proposed intrusion detection system can be further enhanced by extending it to support real-time monitoring of live network traffic instead of relying only on offline datasets. Incorporating advanced deep learning techniques such as convolutional and recurrent neural networks can improve the detection of complex and previously unseen attacks. The system can also be improved by enabling automatic model updates using newly collected data, allowing it to adapt continuously to evolving attack patterns. In addition, integrating the IDS with an intrusion prevention mechanism would allow automatic blocking of malicious traffic, thereby providing proactive security. Future enhancements may also include deployment in cloud, IoT, and distributed environments to improve scalability and coverage while optimizing performance for faster and more efficient intrusion detection.

REFERENCES

[1]. B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE Network*, vol. 8, no. 3, pp. 26–41, May–June 1994, doi: 10.1109/65.283931.

[2]. V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[3]. P. Larson, "Understanding denial-of-service attacks," *IEEE Security & Privacy*, vol. 14, no. 3, pp. 45–52, 2016.

[4]. S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, no. 3, pp. 151–180, 1998.

[5]. S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proc. IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1996, pp. 120–128, doi: 10.1109/SECPRI.1996.502675.

[6]. N. Hubballi, S. Biswas, and S. Nandi, "Sequencegram: n-gram modeling of system calls for program based anomaly detection," in *Proc. 3rd Int. Conf. Communication Systems and Networks (COMSNETS)*, Bangalore, India, 2011, pp. 1–10, doi: 10.1109/COMSNETS.2011.5716416.

[7]. P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019, doi: 10.1109/COMST.2018.2847722.

[8]. M. Azab, M. Alazab, and M. Aiash, "Machine learning based botnet identification traffic," in *Proc. IEEE TrustCom/BigDataSE/ISPA*, Tianjin, China, 2016, pp. 1788–1794, doi: 10.1109/TrustCom.2016.0275.

[9]. J. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," *South African Computer Journal*, no. 56, pp. 136–154, 2015.

[10]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11]. Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[12]. S. Venkatraman and M. Alazab, "Use of data visualization for zero-day malware detection," *Security and Communication Networks*, vol. 2018, pp. 1–13, 2018.

