

AI-Based Network Intrusion Detection System Using Machine Learning.

Manohar C,Dr .Varalakhmi K R

Department of Computer Science and Engineering

R.L.Jallapa institute of technology

Visvesvaraya Technological University, Karnataka, India Email:

manoharchandrappa12@gmail.com

Associate professor

Department of Computer Science and Engineering

R.L.Jallapa institute of technology

Visvesvaraya Technological University, Karnataka, India

Abstract

The rapid expansion of computer networks and internet-based services has significantly increased the complexity and frequency of cyberattacks. Traditional Network Intrusion Detection Systems (NIDS), which primarily rely on signature-based detection techniques, are effective against known threats but often fail to identify previously unseen or evolving attacks. To address this limitation, this paper presents an AI-Based Network Intrusion Detection System (AI-NIDS) that combines machine learning with real-time network traffic monitoring to improve intrusion detection capabilities.

The proposed system captures live network packets using the Scapy library and extracts essential traffic features, including source and destination IP addresses, protocol type, port numbers, and packet length. These features are processed and classified using a Random Forest machine learning model trained on the NSL-KDD dataset. Based on the prediction results, the system distinguishes normal network traffic from malicious traffic. When suspicious activity is detected, an alert is generated, the intrusion details are stored in SQLite and CSV logs, and the monitoring dashboard is updated using Streamlit for real-time visualization.

The experimental evaluation demonstrates that the proposed system provides effective real-time intrusion detection while maintaining efficient performance for continuous network monitoring. The modular architecture enables easy integration, scalability, and future enhancement using advanced machine learning or deep learning techniques. The proposed AI-NIDS offers a practical solution for educational institutions, enterprise networks, and small-to-medium organizations seeking an intelligent and cost-effective approach to network security.

Keywords— Artificial Intelligence, Network Intrusion Detection System, Machine Learning, Random Forest, Network Security, Scapy, Streamlit, NSL-KDD.

I. INTRODUCTION

The increasing dependence on computer networks for communication, cloud computing, online banking, healthcare, education, and industrial automation has made cybersecurity one of the most critical challenges in the digital era. As network infrastructures continue to grow in size and complexity, cybercriminals exploit vulnerabilities through attacks such as denial-of-service, unauthorized access, malware propagation, and reconnaissance activities. These attacks may lead to data theft, financial loss, disruption of services, and compromise of confidential information. Consequently, organizations require intelligent security mechanisms capable of continuously monitoring network activities and identifying malicious behavior before significant damage occurs.

Network Intrusion Detection Systems (NIDS) are widely deployed to enhance network security by monitoring traffic and identifying suspicious activities. Conventional intrusion detection systems generally employ signature-based or rule-based detection techniques. Although these methods provide reliable detection of previously known attacks, they require continuous updates of attack signatures and are ineffective against zero-day attacks or new attack patterns. In addition, they often generate false alarms and require considerable manual effort for maintenance and monitoring.

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have created new opportunities for improving intrusion detection systems. Machine learning algorithms can learn patterns from historical network traffic and identify malicious behavior based on statistical and behavioral characteristics rather than relying solely on predefined signatures. This capability enables AI-driven systems to detect both known and previously unseen attacks with improved accuracy and adaptability.

In this work, an AI-Based Network Intrusion Detection System (AI-NIDS) is proposed to perform intelligent real-time network monitoring. The system utilizes the Scapy library to capture live network packets and extract relevant traffic features. A Random Forest classifier trained using the NSL-KDD dataset analyzes these features and predicts whether the observed traffic is normal or malicious. Detected intrusion events are automatically logged into SQLite and CSV files, while a Streamlit dashboard provides real-time visualization of network activity, alerts, and historical intrusion records.

The proposed system is designed with modularity and scalability in mind, enabling straightforward integration of additional datasets, alternative machine learning algorithms, and advanced visualization capabilities. This architecture supports both academic research and practical deployment scenarios.

A. Motivation

The motivation for this research stems from the increasing sophistication of cyber threats and the limitations of conventional intrusion detection techniques. Existing signature-based systems require frequent updates and are unable to detect previously unseen attacks effectively. Advances in machine learning provide the opportunity to develop adaptive

intrusion detection mechanisms capable of learning complex traffic patterns and improving detection performance. The proposed AI-NIDS aims to leverage these capabilities to provide accurate, real-time, and automated network security monitoring.

B. Problem Statement

Existing Network Intrusion Detection Systems face several limitations when operating in modern network environments. They primarily rely on predefined attack signatures, making them ineffective against zero-day attacks and newly emerging threats. Furthermore, many traditional systems experience high false-positive rates, require continuous manual updates, and provide limited support for intelligent real-time analysis. These challenges highlight the need for an AI-based intrusion detection framework capable of accurately classifying network traffic while supporting continuous monitoring and automated alert generation.

C. Objectives

The primary objectives of the proposed research are:

- To develop an AI-based Network Intrusion Detection System for real-time traffic monitoring.
- To capture and analyze live network packets using Scapy.
- To extract meaningful traffic features for machine learning classification.
- To train and evaluate a Random Forest classifier using the NSL-KDD dataset.
- To identify malicious network activities with high detection accuracy.
- To generate automated intrusion alerts and maintain attack logs.
- To provide an interactive dashboard for real-time visualization of network security events.

D. Research Contributions

The major contributions of this work include:

- Design and implementation of a lightweight AI-based Network Intrusion Detection System.
- Integration of live packet capture with machine learning-based intrusion detection.
- Automated feature extraction and intelligent traffic classification.
- Real-time alert generation and persistent logging using SQLite and CSV.
- Development of a Streamlit dashboard for network monitoring and visualization.
- Evaluation of the proposed framework using the NSL-KDD dataset.

II. RELATED WORK

Network Intrusion Detection Systems (NIDS) have become a significant research area due to the rapid increase in cyber threats and the growing complexity of modern communication networks. Traditional signature-based intrusion detection techniques provide reliable detection of known attacks; however, they are ineffective against unknown or zero-day attacks. To overcome these limitations, researchers have increasingly adopted Machine Learning (ML) and Artificial Intelligence (AI) techniques to improve detection accuracy, reduce false positives, and enable intelligent analysis of network traffic.

Alif Nur Iman and Tohari Ahmad (2020) proposed a Random Forest-based intrusion detection system enhanced with the Boruta feature selection algorithm. Their work focused on optimizing Random Forest parameters, including tree depth and the number of trees, to improve intrusion detection performance on the NSL-KDD dataset. Their study demonstrated that effective feature selection significantly improves classification accuracy while reducing computational complexity.

Yu and Bian (2020) introduced a Few-Shot Learning (FSL) approach for network intrusion detection to address the challenge of limited attack samples. Their model required less than 1% of the NSL-KDD training data while achieving competitive detection performance, demonstrating the potential of deep learning techniques for handling imbalanced intrusion datasets.

Liu et al. (2020) developed a Deep Random Forest model implemented on Apache Spark for large-scale intrusion detection. Their approach evaluated multiple benchmark datasets, including NSL-KDD, UNSW-NB15, CICIDS2017, and CICIDS2018, showing that distributed machine learning frameworks can improve scalability and processing efficiency for network security applications.

Recent research has focused on enhancing machine learning models using explainable artificial intelligence (XAI). In 2024, researchers proposed a Random Forest-based intrusion detection framework integrated with XAI techniques to improve model interpretability. Their work emphasized that explainable models enable security analysts to understand why network traffic is classified as malicious, thereby increasing user trust and supporting informed decision-making.

Another recent study presented an empirical evaluation of Extreme Gradient Boosting (XGBoost) combined with Recursive Feature Elimination (RFE) for intrusion detection. Using the NSL-KDD dataset, the proposed method achieved improved classification performance across attack categories while reducing unnecessary features, highlighting the importance of feature selection in machine-learning-based NIDS.

A 2024 study on network anomaly detection investigated several machine learning algorithms using the NSL-KDD benchmark dataset. The authors evaluated Support Vector Machine (SVM), Logistic Regression, and K-Nearest Neighbour (KNN) classifiers after Random Forest-based feature selection. Their results demonstrated that appropriate feature engineering and model selection can significantly improve detection accuracy and support real-time intrusion monitoring.

Hybrid intrusion detection approaches have also gained attention. A lightweight hybrid RNN-Random Forest model combined recurrent neural networks for feature extraction with Random Forest for classification. Experimental evaluation showed excellent detection performance on both NSL-KDD and CICIDS2017 datasets while reducing computational overhead compared to deeper neural network models.

Despite these advances, several limitations remain. Many proposed systems operate primarily on offline datasets, require substantial computational resources, or focus mainly on classification accuracy without implementing complete real-time monitoring systems that integrate packet capture, alert generation, database logging, and visualization. These

limitations motivate the development of lightweight AI-based intrusion detection systems suitable for practical deployment.

The proposed work addresses these research gaps by integrating Scapy for real-time packet capture, Random Forest for machine learning-based classification, SQLite and CSV for automated logging, and a Streamlit dashboard for live visualization of network traffic and intrusion events. The resulting framework aims to provide accurate, scalable, and user-friendly network intrusion detection while maintaining low computational complexity.

Table I. Comparison of Existing Research

Author & Year	Algorithm	Dataset	Strength	Limitation
Iman & Ahmad (2020)	Random Forest + Boruta	NSL-KDD	Improved feature selection	Offline evaluation only
Yu & Bian (2020)	Few-Shot Learning	NSL-KDD	Works with limited training data	Higher implementation complexity
Liu et al. (2020)	Deep Random Forest	NSL-KDD, CICIDS2017	Scalable distributed processing	Requires Spark infrastructure
XAI-Based IDS (2024)	Random Forest + XAI	NSL-KDD	Improved model interpretability	Additional computational overhead
XGBoost + RFE (2024)	XGBoost	NSL-KDD	High classification performance	More complex feature engineering
Network Anomaly Detection (2024)	SVM, LR, KNN	NSL-KDD	Strong comparative evaluation	No live packet capture
Hybrid RNN-RF (2024)	RNN + Random Forest	NSL-KDD, CICIDS2017	High detection accuracy	More computationally intensive

Research Gap

From the literature survey, the following research gaps are identified:

- Most studies focus on offline evaluation rather than real-time deployment.
- Limited integration of live packet capture with machine learning classification.
- Few systems provide real-time dashboard visualization.
- Automated logging and alert management are often missing.
- Many advanced models require high computational resources.
- Lightweight, real-time, and user-friendly AI-based NIDS solutions remain an active research need.

III. PROPOSED METHODOLOGY

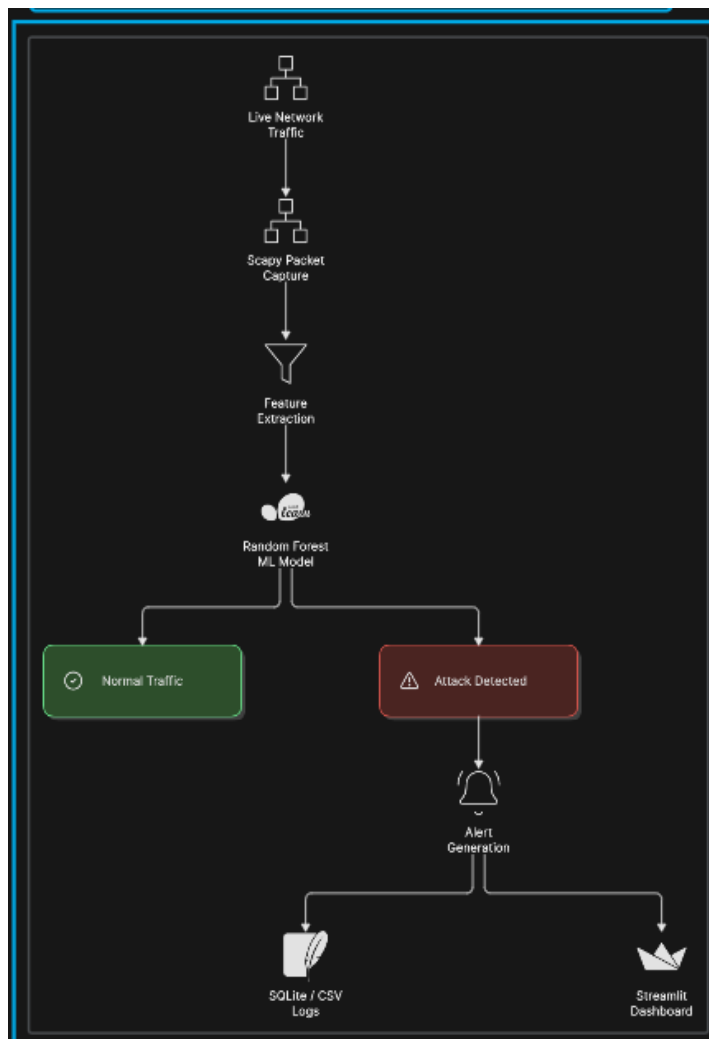
A. Overview

The proposed AI-Based Network Intrusion Detection System (AI-NIDS) is designed to provide intelligent, real-time detection of malicious network activities by integrating live packet capture, machine learning, automated alert generation, and visualization into a single framework. Unlike traditional signature-based intrusion detection systems that rely on predefined attack patterns, the proposed framework employs a Random Forest classifier trained using the NSL-KDD dataset to identify both known and unknown attacks based on traffic characteristics.

The proposed framework consists of seven major stages: packet capture, feature extraction, data preprocessing, machine learning classification, intrusion detection, logging, and dashboard visualization. Each stage contributes to continuous monitoring and accurate classification of network traffic while maintaining low computational overhead.

B. System Architecture

Figure 1. Proposed AI-Based Network Intrusion Detection System Architecture



The architecture consists of the following modules:

1. Live Network Traffic
2. Packet Capture using Scapy
3. Feature Extraction
4. Data Preprocessing
5. Random Forest Classifier
6. Intrusion Detection
7. Alert Generation
8. SQLite & CSV Logging
9. Streamlit Dashboard

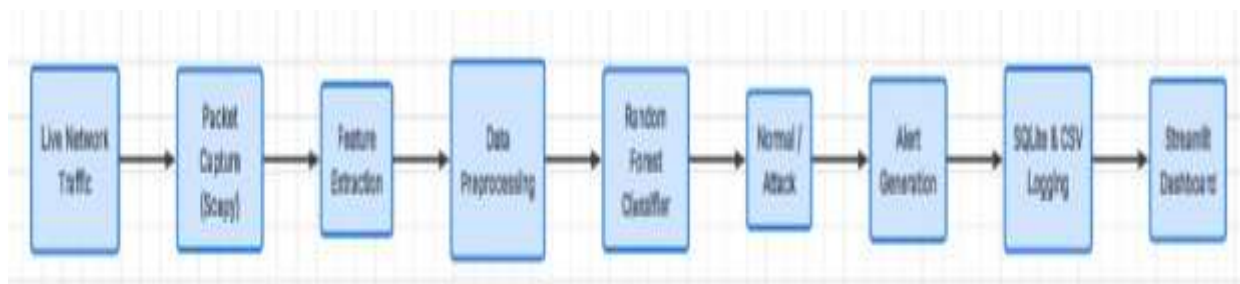
C. Workflow of the Proposed System

The workflow of the proposed AI-NIDS begins with capturing live network packets using the **Scapy** library. Each captured packet is analyzed to extract relevant network features such as source IP address, destination IP address, protocol type, source port, destination port, and packet length. These features are converted into a structured format suitable for machine learning classification.

The extracted features undergo preprocessing before being passed to the trained Random Forest classifier. The classifier predicts whether the observed network traffic represents normal activity or a malicious intrusion. When an intrusion is detected, the system automatically generates an alert, records the event in SQLite and CSV logs, and updates the **Streamlit dashboard** for real-time visualization.

D. Methodology Workflow

Figure 2. Workflow of the Proposed AI-NIDS



E. Packet Capture Module

The first stage of the proposed framework is live packet capture. The Scapy packet manipulation library continuously monitors the selected network interface and captures incoming and outgoing packets in real time. Unlike offline intrusion detection approaches, this module enables continuous observation of actual network traffic, making the framework suitable for practical deployment.

The captured packets include information from multiple protocol layers, enabling the extraction of essential network attributes required for classification.

F. Feature Extraction

After packet capture, relevant traffic features are extracted from each packet. The extracted features include:

Feature	Description
Source IP	Sender IP Address
Destination IP	Receiver IP Address
Protocol	TCP, UDP, ICMP
Source Port	Communication Port
Destination Port	Target Port
Packet Length	Packet Size
Timestamp	Time of Packet Capture

These features provide the Random Forest classifier with meaningful information required for identifying malicious traffic patterns.

G. Data Preprocessing

Before classification, the extracted packet features undergo preprocessing to improve model performance.

The preprocessing stage includes:

- Removal of unnecessary attributes
- Missing value handling
- Label encoding
- Feature normalization
- Feature formatting

These operations ensure that the input data are consistent with the format used during model training.

H. Random Forest Classification

The preprocessed features are supplied to the Random Forest classifier.

Random Forest is an ensemble learning algorithm consisting of multiple decision trees. Each tree independently predicts the traffic class, and the final prediction is obtained using majority voting. This approach improves classification accuracy, reduces overfitting, and provides robust performance for intrusion detection. Ensemble approaches such as Random Forest remain widely used for NIDS because they provide strong accuracy while maintaining efficient prediction time.

The classifier categorizes network traffic into:

- Normal Traffic
- Malicious Traffic

The trained model is stored as:

random_forest_model.pkl

I. Alert Generation

If the classifier identifies malicious traffic, the system immediately generates an intrusion alert.

The alert contains:

- Source IP Address
- Destination IP Address
- Protocol
- Attack Status
- Detection Time

This enables administrators to respond promptly to suspicious activities.

J. Database Logging

All detected intrusion events are automatically stored for future analysis. The logging module maintains records in:

- SQLite Database
- CSV File

Each log entry includes:

- Timestamp
- Source IP
- Destination IP

- Protocol
- Packet Length
- Attack Type

Persistent logging supports incident investigation and historical analysis.

K. Dashboard Visualization

The Streamlit dashboard provides an interactive interface for monitoring network security. The dashboard displays:

- Live Network Traffic
- Attack Alerts
- Packet Statistics
- Historical Logs
- Database Records

This visualization enables administrators to monitor network activity in real time.

Figure 3. Streamlit Dashboard

Detection System

Real-Time Intrusion Monitoring Dashboard

Attack Alerts

	id	src_ip	dst_ip	protocol	attack_type
0	1	192.34.20.166	192.168.0.102	6	Port Scan Detected
1	2	192.34.20.166	192.168.0.102	6	Port Scan Detected
2	3	192.34.20.166	192.168.0.102	6	Port Scan Detected
3	4	192.34.20.166	192.168.0.102	6	Port Scan Detected
4	5	192.34.20.166	192.168.0.102	6	Port Scan Detected
5	6	192.34.20.166	192.168.0.102	6	Port Scan Detected
6	7	192.34.20.166	192.168.0.102	6	Port Scan Detected
7	8	192.34.20.166	192.168.0.102	6	Port Scan Detected

L. Algorithm

Algorithm 1: AI-Based Network Intrusion Detection

Input:

Live Network Traffic

Output:

Attack Detection and Alert

Step 1: Capture live packets using Scapy. Step 2: Extract packet features.

Step 3: Preprocess extracted data.

Step 4: Load trained Random Forest model. Step 5: Predict traffic class.

Step 6:

If traffic == Attack Generate Alert Store Logs Update Dashboard

Else

Continue Monitoring Step 7: Repeat continuously.

M. Advantages of the Proposed Method

The proposed methodology offers several advantages:

- Real-time intrusion detection.
- Intelligent Machine Learning-based classification.
- Automatic alert generation.
- Interactive dashboard visualization.
- Low computational complexity.
- Scalable modular architecture.
- Easy integration with enterprise networks.
- Improved detection performance compared to traditional signature-based IDS.

N. Summary

The proposed AI-Based Network Intrusion Detection System integrates live packet capture, feature extraction, machine learning classification, logging, and visualization into a unified framework. The modular architecture enables efficient real-time monitoring and supports future enhancements such as deep learning models, cloud deployment, and IoT security

integration. This methodology provides a practical and scalable approach for improving network intrusion detection in modern cybersecurity environments.

IV. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

A. Experimental Setup

The proposed AI-Based Network Intrusion Detection System (AI-NIDS) was implemented using Python and evaluated in a Windows-based development environment. The system integrates live packet capture, machine learning-based classification, automated logging, and real-time dashboard visualization. The Random Forest classifier was trained using the NSL-KDD dataset and evaluated through live network traffic monitoring.

The experimental environment was configured to ensure consistent model training, efficient packet capture, and reliable intrusion detection while maintaining low computational overhead.

B. Hardware Configuration

Table 2. Hardware Configuration

Component	Specification
Processor	Intel Core i5/i7 (or Equivalent)
RAM	8 GB
Storage	256 GB SSD or Higher
Operating System	Windows 11 (64-bit)
Network	Ethernet/Wi-Fi

C. Software Configuration

Table 3. Software Configuration

Software	Version	Purpose
Python	3.11	Programming Language
Visual Studio Code	Latest	Development Environment

Scapy	Latest	Live Packet Capture
Scikit-learn	Latest	Machine Learning
Pandas	Latest	Data Processing
NumPy	Latest	Numerical Computation
Streamlit	Latest	Dashboard
SQLite	Built-in	Database
Joblib	Latest	Model Storage

D. Dataset Description

The Random Forest classifier was trained using the NSL-KDD dataset, a widely accepted benchmark for intrusion detection research. The dataset addresses many of the shortcomings of the original KDD Cup 1999 dataset by removing duplicate records and providing a balanced distribution of attack categories.

The dataset contains normal network traffic as well as multiple attack categories including:

- Denial of Service (DoS)
- Probe
- User-to-Root (U2R)
- Remote-to-Local (R2L)

Each record contains several traffic attributes describing the characteristics of a network connection.

E. Data Preprocessing

Before training the classifier, the dataset was preprocessed to improve model performance. Data preprocessing included removing unnecessary attributes, handling missing values, encoding categorical variables, and formatting features into a machine-learning-compatible representation. These steps ensured consistency between the training data and live packet features extracted during execution.

F. Model Training

The Random Forest classifier was trained using the preprocessed NSL-KDD dataset. During training, the algorithm constructed multiple decision trees using randomly selected subsets of

the training data. The final prediction was obtained through majority voting among all decision trees, improving classification robustness and reducing overfitting.

After successful training, the model was serialized and stored for future predictions using the Joblib library.

G. Performance Metrics

The effectiveness of the proposed AI-NIDS was evaluated using standard machine learning performance metrics.

Table 4. Performance Metrics

Metric	Description
Accuracy	Overall prediction correctness
Precision	Correctly predicted attacks among all predicted attacks
Recall	Correctly detected attacks among actual attacks
F1-Score	Harmonic mean of Precision and Recall
Confusion Matrix	Performance summary of classification results

H. Experimental Results

The proposed system was tested using both the NSL-KDD dataset and live network traffic. The trained Random Forest model successfully classified network traffic as either normal or malicious. The system generated alerts for detected intrusions and stored the corresponding records in the database while simultaneously updating the Streamlit dashboard.

The implementation demonstrated stable performance during continuous packet monitoring and effectively supported real-time intrusion detection.

I. Performance Evaluation

The Random Forest classifier achieved a classification accuracy of 97.98%, indicating strong capability in distinguishing normal and malicious network traffic.

J. Discussion

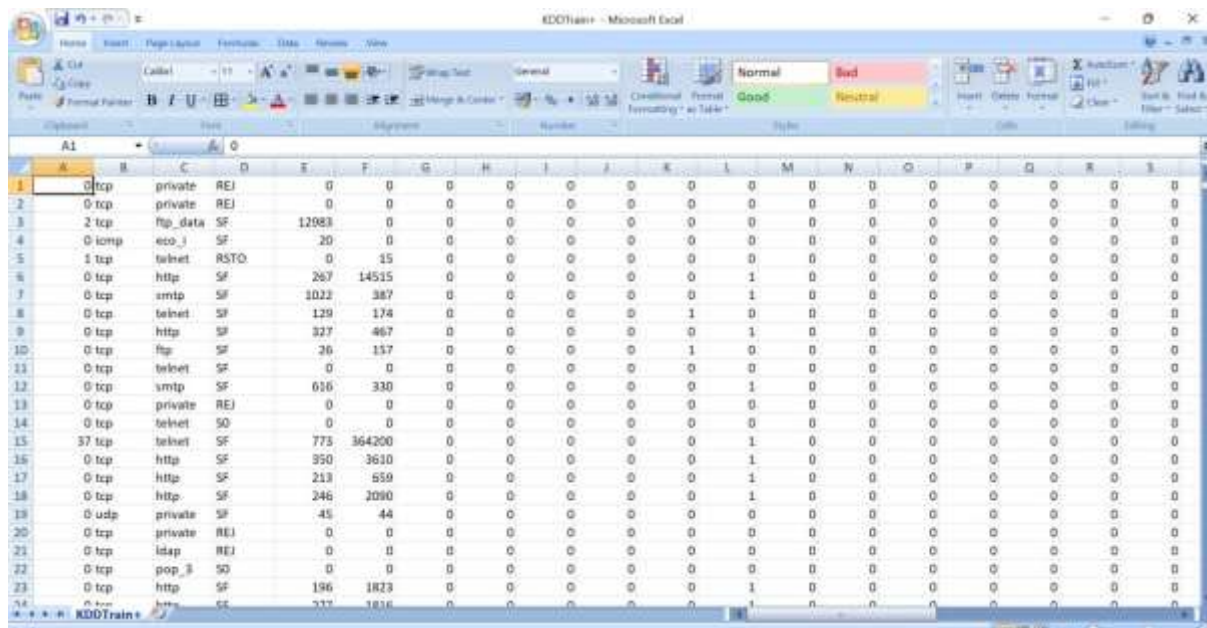
The experimental evaluation indicates that the proposed AI-Based Network Intrusion Detection System effectively combines machine learning with real-time packet monitoring to detect malicious network activities. The Random Forest classifier achieved a classification accuracy of 97.98%, demonstrating reliable performance on the NSL-KDD dataset. The integration of Scapy enabled continuous packet capture, while SQLite and CSV logging supported persistent storage of intrusion records. The Streamlit dashboard provided an intuitive interface for monitoring network activity and attack alerts in real time. Overall, the experimental results validate the practicality and effectiveness of the proposed framework.

K. Summary

This section presented the experimental environment, dataset, preprocessing steps, model training process, and performance evaluation of the proposed AI-Based Network Intrusion Detection System. The results demonstrate that the proposed framework successfully supports real-time intrusion detection with high accuracy while providing automated logging and visualization features.

Output:

Figure 4:KDD dataset



The screenshot shows a Microsoft Excel spreadsheet titled 'KDDTrain+ - Microsoft Excel'. The spreadsheet displays a large dataset with columns labeled A through S. The data is organized into rows, with the first row (A1) containing the following values: A1: 0, B1: tcp, C1: private, D1: REJ, E1: 0, F1: 0, G1: 0, H1: 0, I1: 0, J1: 0, K1: 0, L1: 0, M1: 0, N1: 0, O1: 0, P1: 0, Q1: 0, R1: 0, S1: 0. The subsequent rows (A2 through A24) contain similar data, with some cells containing numerical values (e.g., 12083, 20, 267, 14515, 1022, 387, 129, 174, 327, 467, 26, 157, 0, 0, 616, 330, 0, 0, 773, 364200, 350, 3610, 213, 659, 246, 2090, 45, 44, 0, 0, 0, 0, 196, 1823, 777, 1836). The spreadsheet is displayed in a standard Excel interface with a ribbon at the top and a status bar at the bottom.

Figure 5:detection System

Detection System

Real-Time Intrusion Monitoring Dashboard

Attack Alerts

	id	src_ip	dst_ip	protocol	attack_type
0	1	192.34.20.166	192.168.0.102	6	Port Scan Detected
1	2	192.34.20.166	192.168.0.102	6	Port Scan Detected
2	3	192.34.20.166	192.168.0.102	6	Port Scan Detected
3	4	192.34.20.166	192.168.0.102	6	Port Scan Detected
4	5	192.34.20.166	192.168.0.102	6	Port Scan Detected
5	6	192.34.20.166	192.168.0.102	6	Port Scan Detected
6	7	192.34.20.166	192.168.0.102	6	Port Scan Detected
7	8	192.34.20.166	192.168.0.102	6	Port Scan Detected

Figure 6:Attack Statistics

Attack Statistics

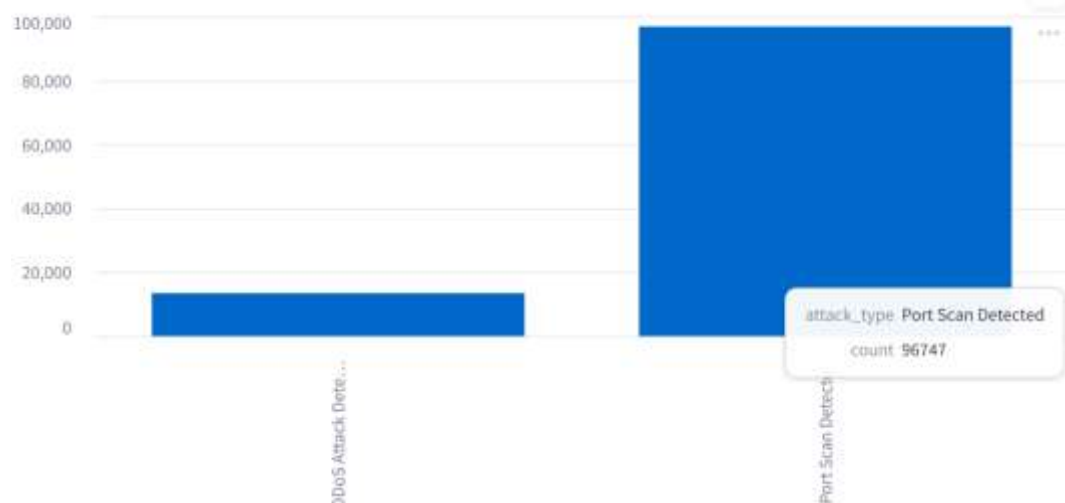


Figure 7: Total Alerts and Live Traffic

Total Alerts

110192

Live Traffic

Packet sniffing is active...

V.CONCLUSION

The increasing complexity of cyber threats has made intelligent intrusion detection systems an essential component of modern network security. Traditional signature-based Network Intrusion Detection Systems (NIDS) are effective in identifying known attacks but often fail to detect previously unseen threats and require frequent signature updates. To overcome these limitations, this research presented an AI-Based Network Intrusion Detection System (AI-NIDS) that integrates machine learning with real-time network traffic monitoring.

The proposed framework employs Scapy for live packet capture, extracts relevant network traffic features, and utilizes a Random Forest classifier trained on the NSL-KDD dataset to classify network traffic as either normal or malicious. Upon detecting suspicious activity, the system automatically generates alerts, records intrusion events in SQLite and CSV logs, and updates an interactive Streamlit dashboard for continuous monitoring.

Experimental evaluation demonstrated that the proposed system achieved a classification accuracy of 97.98%, indicating its effectiveness in identifying malicious network traffic while maintaining efficient real-time performance. The modular architecture allows easy integration of additional machine learning models, datasets, and visualization components. Furthermore, the proposed framework provides a practical and cost-effective solution for educational institutions, enterprise networks, and small-to-medium organizations requiring intelligent network security.

Overall, the proposed AI-NIDS successfully combines machine learning, live packet monitoring, automated logging, and dashboard visualization into a unified platform that enhances network security through accurate and real-time intrusion detection.

REFERENCES

1. Tavallaei, M., Bagheri, E., Lu, W., & Ghorbani, A. A., "A Detailed Analysis of the KDD CUP 99 Data Set," *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.
2. Breiman, L., "Random Forests," *Machine Learning*, 45(1), 5–32, 2001.
3. Scikit-learn Developers, *Scikit-learn Documentation*.
4. Scapy Project Documentation.
5. Streamlit Documentation.
6. Pedregosa, F., et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.
7. McKinney, W., "Data Structures for Statistical Computing in Python," 2010.
8. Harris, C. R., et al., "Array Programming with NumPy," *Nature*, 2020.
9. Python Software Foundation, *Python Documentation*.
10. SQLite Documentation.