iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IX **Month of publication:** September 2026

DOI: https://doi.org/10.22214/ijraset.2026.84969

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IX Sep 2026- Available at www.ijraset.com

AI-Based Network Intrusion Detection System

Shruthi Rampure[1], Vishal[2]
Master of Computer Application, Visvesvaraya Technological University Regional Campus Kalaburagi

Abstract: ***The increasing use of computer networks, cloud platforms, and connected digital services has created a growing need for intelligent and adaptive cybersecurity solutions. Traditional intrusion detection methods that mainly depend on predefined rules and known attack signatures may have difficulty identifying changing or previously unseen network threats. This paper presents an AI-Based Network Intrusion Detection System that applies machine learning to identify malicious network activities from structured traffic data. The proposed system uses the NSL-KDD dataset and a Random Forest classifier to distinguish normal network connections from attack traffic. Data preprocessing operations, including feature transformation, categorical encoding, normalization, and label conversion, are performed before classification. The trained model is integrated with a Flask-based web application that provides secure authentication, dataset uploading, intrusion prediction, visual analytics, prediction logs, and alert generation. An SQLite database is used to maintain user and prediction-related information. The implemented system was tested across authentication, dataset processing, prediction, result visualization, logging, alert generation, and logout functions, with the documented test cases successfully completed. The reported model evaluation includes accuracy, precision, recall, and F1-score, demonstrating the practical application of the proposed approach for automated network traffic analysis and intrusion detection.***

Keywords: Artificial Intelligence, Network Intrusion Detection, Random Forest, NSL-KDD, Machine Learning, Cybersecurity, Flask, Network Security.

I. INTRODUCTION

In the present digital environment, computer networks play an essential role in communication, information exchange, cloud services, and online applications, but the increasing dependence on networked systems also exposes them to threats such as unauthorized access, denial-of-service attacks, malicious activities, and data breaches. Conventional Intrusion Detection Systems mainly depend on predefined rules and known attack signatures, which may have limitations when dealing with modified or emerging attack patterns. To address this challenge, this project presents an AI-Based Network Intrusion Detection System that uses machine learning to analyze network traffic and classify connections as normal or malicious. The proposed system uses the NSL-KDD dataset and applies preprocessing operations such as categorical encoding, feature transformation, normalization, and label conversion before classification. A Random Forest classifier is employed to learn patterns from network traffic and perform intrusion classification. The trained model is integrated with a Flask-based web application that provides secure user authentication, dataset uploading, prediction results, dashboards, prediction logs, and intrusion alerts. The system combines data preprocessing, machine learning, database management, and web-based visualization to provide an accessible platform for automated network traffic analysis. The current implementation focuses on dataset-based intrusion detection, while its modular structure can support future extensions such as real-time monitoring, advanced deep learning models, and broader attack classification.

II. PROBLEM STATEMENT

Modern computer networks are continuously exposed to cyber threats whose patterns and methods change over time. Traditional intrusion detection approaches that depend mainly on predefined rules and known attack signatures may not effectively identify newly developed or modified attacks. In addition, manually examining large volumes of network traffic is time-consuming and can result in missed threats or unnecessary alerts. Therefore, there is a need for an automated and intelligent intrusion detection system that can analyze network traffic, identify suspicious activities, and classify connections as normal or malicious with reliable performance. The proposed system addresses this requirement by applying machine learning techniques to network traffic data and providing a web-based platform for prediction, monitoring, logging, and alert generation..

III. OBJECTIVES OF THE STUDY

The main objective of this study is to develop an AI-based network intrusion detection system capable of identifying and classifying malicious network activities from network traffic data.

©IJRASET: All Rights are Reserved

The study aims to build and evaluate a machine learning model using a suitable network traffic dataset, apply appropriate preprocessing techniques to improve the quality of input data, and use a Random Forest classifier to distinguish normal traffic from attack traffic. It also aims to develop a user-friendly Flask-based web application that allows users to upload datasets, view intrusion detection results, and monitor security-related information through dashboards and logs. In addition, the system is designed to generate clear alerts for detected malicious activities and provide a practical platform for automated network security monitoring.

IV. METHODOLOGY

The methodology of the proposed AI-Based Network Intrusion Detection System follows a structured process for analyzing network traffic and identifying potential intrusions. The process begins with the selection of the NSL-KDD dataset containing labeled network connection records representing normal and attack traffic. The collected data is examined and prepared for machine learning by applying preprocessing operations such as data transformation, categorical feature encoding, numerical normalization, and label conversion. The processed dataset is then divided into suitable training and testing data, and a Random Forest classifier is trained to learn patterns associated with normal and malicious network connections. After model evaluation, the trained classifier is integrated into a Flask-based web application, where authenticated users can upload network traffic files in CSV format for analysis. The uploaded data passes through the same preprocessing pipeline before being supplied to the trained model for classification. The system displays the prediction results through the web interface and maintains prediction information in an SQLite database. Intrusion-related results are presented through dashboards, logs, and alerts, providing a structured approach for network traffic analysis and security monitoring.

V. LITERATURE SURVEY

Research on network intrusion detection has increasingly focused on machine learning, deep learning, ensemble methods, and distributed learning techniques to improve the identification of malicious network activities. Recent studies have explored different approaches for handling complex traffic patterns and improving detection reliability. Althunayyan et al. (2024) investigated a multi-stage intrusion detection approach using hierarchical federated learning and reported strong performance for both known and unseen attacks. Alalwany and Mahgoub (2024) examined ensemble learning techniques for real-time intrusion detection and highlighted the usefulness of combining multiple machine learning models. Kumar et al. (2023) proposed a hybrid intrusion detection approach using GAN-SMOTE with machine learning techniques to address data imbalance and improve attack detection. Lee, Choi, and Kim (2023) developed an intrusion detection approach based on Random Forest for CAN networks, demonstrating the applicability of ensemble-based classification for network security. Lin et al. (2022) explored deep learning networks for identifying cyberattacks and showed the potential of deep learning for complex attack patterns. Yang, Hu, and Yu (2022) investigated federated AI with ConvLSTM for intrusion detection in Internet of Vehicles environments, focusing on distributed learning and detection efficiency. Basavaraj and Tayeb (2022) studied a lightweight intrusion detection framework and compared deep neural networks with conventional machine learning methods. Other research has examined physical-layer identification, hybrid intrusion detection, and federated learning to improve security in connected network environments. Overall, the reviewed studies demonstrate that machine learning and intelligent data-driven techniques can support automated intrusion detection, while challenges such as false positives, computational overhead, unseen attacks, dataset diversity, and real-time deployment remain important areas for further development. The present study builds on these research directions by applying a Random Forest classifier to the NSL-KDD dataset and integrating the detection model into a Flask-based web application for network traffic analysis and monitoring.

VI. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

Existing network intrusion detection systems commonly depend on predefined security rules, fixed signatures, and previously identified attack patterns to monitor network activities. These systems can identify known threats by comparing network traffic with established detection rules, but their effectiveness may be limited when attackers introduce new or modified behaviors. Monitoring large volumes of network traffic manually can also require considerable effort and may result in missed activities or unnecessary alerts. The existing approach therefore has limitations in adapting to changing attack patterns and providing automated analysis of network traffic. These limitations create a requirement for intelligent detection techniques that can learn from network data and support more flexible intrusion identification..

B. *Proposed System*

The proposed system is an AI-Based Network Intrusion Detection System designed to automatically analyze network traffic and identify potential malicious activities. The system uses the NSL-KDD dataset to train a Random Forest classification model capable of distinguishing normal network connections from attack traffic. Before classification, the uploaded data is processed through feature transformation, categorical encoding, normalization, and label conversion to prepare it for machine learning analysis. The trained model is integrated with a Flask-based web application that provides secure user authentication, CSV dataset uploading, intrusion prediction, result visualization, prediction logs, and alert generation. An SQLite database is used to maintain relevant user and prediction information. The system provides a structured platform for automated network traffic analysis and security monitoring, while its modular design allows future integration of real-time monitoring and advanced detection techniques.

VII. FEASIBILITY STUDY

1) Technical Feasibility: The proposed AI-Based Network Intrusion Detection System is technically feasible because it uses established and accessible technologies such as Python, Scikit-learn, Pandas, NumPy, Flask, HTML, CSS, JavaScript, and SQLite. These technologies provide the required support for data preprocessing, machine learning classification, web application development, visualization, and database management. The Random Forest model can be integrated with the Flask application to process uploaded network traffic datasets and generate intrusion predictions.
2) Economic Feasibility: The system is economically feasible because the major development tools, frameworks, and libraries used in the project are open-source and freely available. This reduces software licensing expenses and makes the system suitable for academic development and small-scale deployment. The use of automated intrusion detection can also reduce the effort required for continuous manual analysis of network traffic.
3) Operational Feasibility: The proposed system is operationally feasible because it provides a web-based interface through which users can securely log in, upload network traffic datasets, view prediction results, and monitor intrusion information. Dashboards, prediction logs, and alerts help users understand the detected network activities without requiring complex interaction with the underlying machine learning model.
4) Schedule Feasibility: The project is feasible within an academic development schedule because the system is divided into independent modules such as data preprocessing, machine learning model development, web application development, and database management. These components can be developed systematically and integrated during the final implementation stage. The modular structure also supports easier testing and future enhancement of the system.

VIII. TOOLS AND TECHNOLOGIES USED

1) Python: Python is used as the primary programming language for developing the system. Its simple syntax and extensive library support make it suitable for data processing, machine learning implementation, backend development, and integration of different system components.
2) Scikit-learn: Scikit-learn is used to develop and implement the Random Forest classification model. It also provides functionality for data preprocessing, feature transformation, model training, prediction, and evaluation.
3) NumPy: NumPy is used for numerical computation and array-based operations during data preparation and machine learning processing. It supports efficient handling of numerical network traffic features.
4) Pandas: Pandas is used to load, clean, transform, and analyze structured network traffic data. It also supports handling CSV datasets and preparing data for machine learning.
5) Flask: Flask is used as the web framework for integrating the trained machine learning model with the application. It manages user authentication, dataset uploads, prediction requests, dashboards, analytics, and prediction logs.
6) HTML5: TML5 is used to create the basic structure of the web application, including login pages, dashboards, dataset upload interfaces, result pages, and other application components.
7) CSS3 and Bootstrap: CSS3 and Bootstrap are used to design the application's interface and provide responsive layouts. They help maintain a consistent presentation across different screen sizes.
8) JavaScript: JavaScript is used to provide client-side interaction and support visual elements within the web application. It contributes to an interactive user experience and visualization of security-related information.
9) SQLite: SQLite is used as the database system for storing user information, prediction records, and alert-related data. Its lightweight structure makes it suitable for the implemented application.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

10) NSL-KDD Dataset: The NSL-KDD dataset is used as the network traffic dataset for developing and evaluating the intrusion detection model. It contains labeled network connection records that support the classification of normal and attack traffic.

11) PyCharm: PyCharm is used as the development environment for writing, debugging, testing, and managing the Python-based application. It provides development features that support organized implementation of the system.

IX. SYSTEM DESIGN

A. System Perspective

The AI-Based Network Intrusion Detection System is designed as a modular web-based platform for analyzing network traffic and identifying potential intrusions. The system allows an authorized network administrator to access the application through a web browser, log in securely, and upload network traffic data in CSV format. The uploaded information is handled by the system core, which coordinates preprocessing, machine learning classification, result generation, and data storage. The modular structure separates the major functions of the application and supports organized processing, maintenance, and future enhancement.

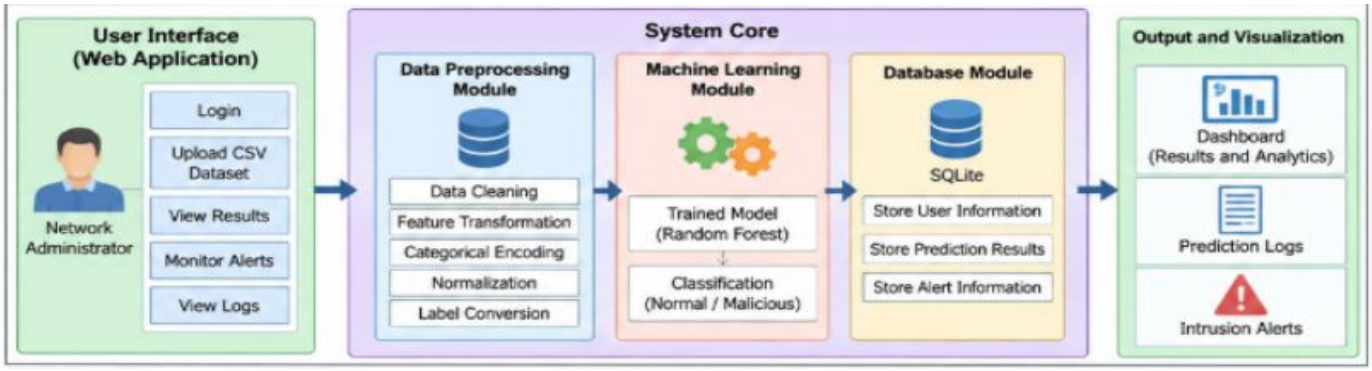


Figure 1: Overall System Architecture of the Proposed AI-Based Network Intrusion Detection System

B. System Architecture

The system architecture consists of the user interface, system core, data preprocessing module, machine learning module, database, and output components. After a dataset is uploaded, the preprocessing module performs data cleaning, feature transformation, categorical encoding, and normalization. The prepared data is then supplied to the Random Forest classifier trained using the NSL-KDD dataset. The classifier categorizes network connections as normal or malicious, and the resulting information is stored in the SQLite database. The results are presented through dashboards, prediction logs, and intrusion alerts.

C. Data Preprocessing Module

The data preprocessing module prepares uploaded network traffic records for machine learning analysis. It performs data cleaning, feature transformation, categorical encoding, numerical normalization, and label conversion. These operations transform the input records into a suitable format for the Random Forest classification model and maintain consistency between the training and prediction processes.

D. Machine Learning Module

The machine learning module applies the Random Forest classification algorithm to the processed network traffic data. The model is trained using labeled records from the NSL-KDD dataset and learns patterns associated with normal and attack traffic. During prediction, the trained model analyzes the processed input records and classifies them as either normal or malicious. The resulting predictions are then passed to the application for display and storage.

E. Database and Output Module

The database and output module manages the storage and presentation of intrusion detection information. SQLite is used to maintain user information, prediction records, and alert-related data. The system displays the generated classification results through dashboards, prediction logs, and intrusion alerts, allowing authorized users to review network activities and previously generated results.

X. SEQUENCE DIAGRAM

The sequence diagram describes the chronological interaction between the network administrator and the major components of the AI-Based Network Intrusion Detection System during intrusion analysis. The process begins when the user accesses the web application and provides login credentials. After successful authentication, the user uploads a network traffic dataset in CSV format. The system receives the file and forwards it to the preprocessing module, where the data is cleaned, transformed, encoded, normalized, and prepared for analysis. The processed records are then passed to the Random Forest machine learning model, which classifies the network connections as normal or malicious. The prediction results are returned to the application and stored in the SQLite database along with the relevant prediction information. Finally, the system presents the classification results through the dashboard and generates intrusion alerts when malicious traffic is identified. The sequence provides a clear representation of the data flow and interaction among authentication, dataset processing, machine learning, database storage, and result presentation components.

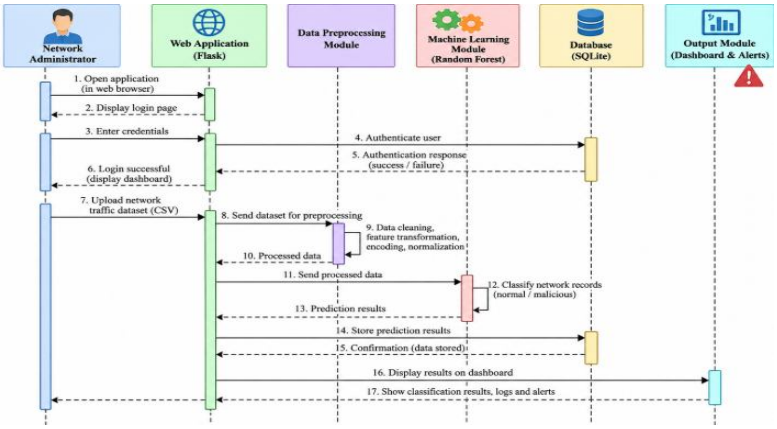


Figure 2: Sequence Diagram of the AI-Based Network Intrusion Detection System

XI. COLLABORATION DIAGRAM

The collaboration diagram illustrates how the major components of the AI-Based Network Intrusion Detection System communicate with one another to complete the intrusion detection process. The interaction begins when the network administrator uses the web interface to upload a network traffic dataset. The request is handled by the controller, which forwards the dataset to the data preprocessing module for cleaning, feature transformation, categorical encoding, and normalization. The processed data is then transferred to the machine learning module, where the trained Random Forest classifier analyzes the network records and generates normal or malicious classifications. The prediction results are returned to the application and stored in the SQLite database along with relevant prediction information. Finally, the web interface retrieves the processed results and presents them through the dashboard, prediction logs, and intrusion alerts. This interaction demonstrates the coordinated communication between the user interface, controller, preprocessing module, machine learning model, database, and output components.

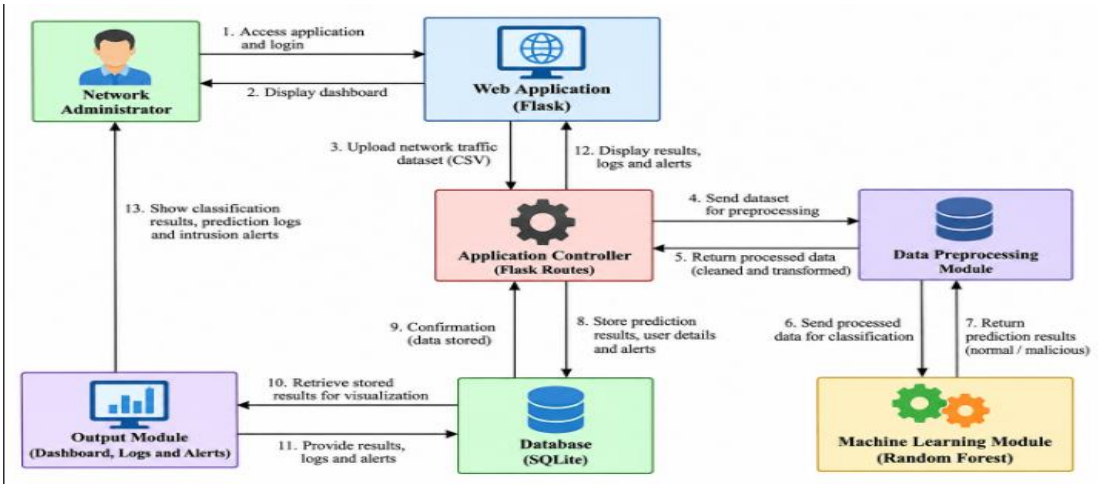


Figure 3: Collaboration Diagram of the AI-Based Network Intrusion Detection System

XII. SYSTEM IMPLEMENTATION

The AI-Based Network Intrusion Detection System is implemented by integrating machine learning, data preprocessing, web application, database, and visualization components into a unified platform. The application is developed using Python and Flask, with the Random Forest model responsible for classifying network traffic as normal or malicious. Users access the system through a web interface where they can register, log in securely, upload network traffic datasets in CSV format, and view the generated detection results. The uploaded dataset is processed using Pandas and the defined preprocessing pipeline before being supplied to the trained machine learning model. The prediction results are then recorded in the SQLite database and presented through the dashboard, prediction logs, and intrusion alerts. The implementation is organized into separate functional components for authentication, dataset processing, prediction, logging, and visualization, which supports easier maintenance and future enhancement.

XIII. RESULTS AND OUTPUT IMAGES

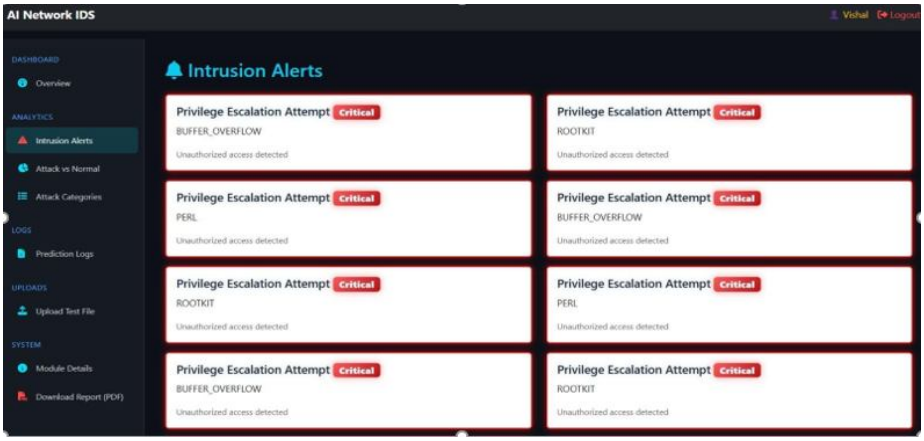


Figure 4: Intrusion Alerts Page

The implemented AI-Based Network Intrusion Detection System produces several outputs through its web-based interface. The main results include the application home page, secure login interface, dataset upload page, intrusion prediction results, model performance information, intrusion alerts, and generated reports. After authentication, the user can upload a network traffic dataset in CSV format for analysis. The system processes the uploaded records and uses the trained Random Forest model to classify network connections as normal or malicious. The prediction results are displayed through the application dashboard, while prediction logs and alerts provide additional information about detected network activities. The documented model evaluation reports an accuracy of 95.12%, precision of 92.87%, recall of 91.94%, and F1-score of 92.90%. The testing results also indicate that the implemented functional test cases, including registration, login, dataset upload, preprocessing, prediction, result display, log storage, alert generation, and logout, were successfully completed.

XIV. CONCLUSION

The AI-Based Network Intrusion Detection System was successfully developed to identify potentially malicious activities within network traffic using machine learning techniques. The system combines data preprocessing, a Random Forest classification model, Flask-based web functionality, and SQLite database management to provide an integrated intrusion detection platform. Network traffic data is processed before classification, allowing the trained model to categorize connections as normal or malicious. The web application provides secure authentication, dataset uploading, prediction results, dashboards, prediction logs, and intrusion alerts, making the detection process more organized and accessible. The documented testing results show that the implemented functional modules operated successfully, including user authentication, dataset processing, prediction, result visualization, database logging, and alert generation. Overall, the project demonstrates the practical integration of machine learning with a web-based security application for automated network traffic analysis and provides a foundation for further development of intelligent network intrusion detection solutions..

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

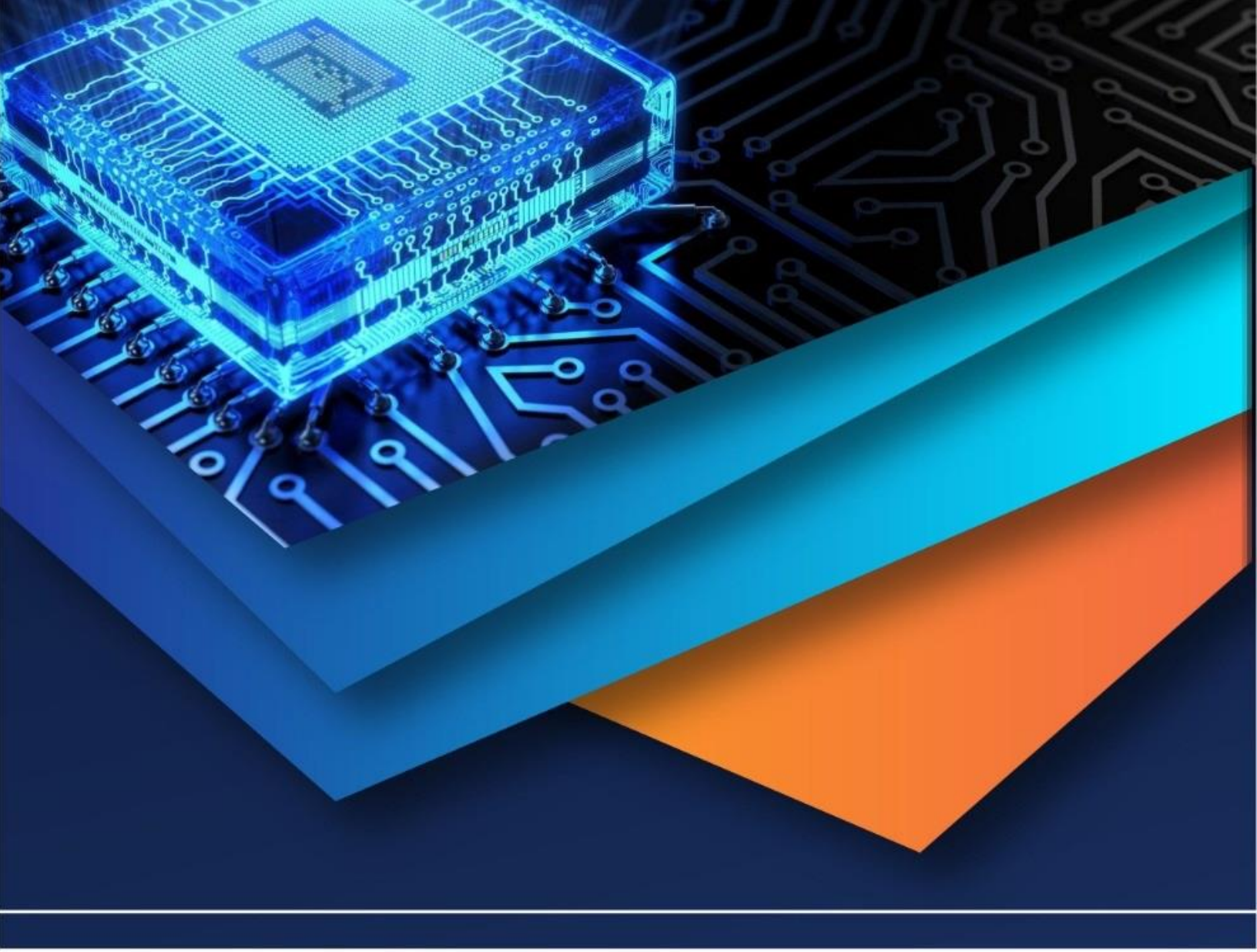
XV. FUTURE ENHANCEMENTS

The proposed AI-Based Network Intrusion Detection System can be further extended to improve its functionality, adaptability, and practical applicability. The current system can be enhanced with real-time network traffic monitoring so that live packets can be analyzed instead of relying only on uploaded datasets. Advanced deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can be incorporated to analyze more complex traffic patterns. The system can also be extended to support multi-class attack classification, enabling identification of specific categories of network attacks. Additional features such as automated response mechanisms, email or SMS notifications, stronger role-based access control, and cloud deployment can further improve security, scalability, and accessibility. These enhancements can help transform the current dataset-based implementation into a more comprehensive network security monitoring platform.

REFERENCES

[1] A. U. Abeykoon, “Prompt Engineering: Techniques and Applications in Conversational AI,” 2024.
[2] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. Chantem, “SINGLE: Single-Frame Based Physical Layer Identification for Intrusion Detection and Prevention on In-Vehicle Networks,” 2019.
[3] M. Althunayyan, A. Javed, and O. Rana, “A Robust Multi-Stage Intrusion Detection System for In-Vehicle Network Security Using Hierarchical Federated Learning,” 2024.
[4] S. Kumar, A. Kumar, H. K., and R. Raghavendra, “Hybrid Intrusion Detection System Using GAN-SMOTE and Machine Learning Techniques for In-Vehicle Networks,” 2023.
[5] A. Falanga and L. Berardinelli, “Automotive Cybersecurity: Data-Driven Intrusion Detection System for In-Vehicle Networks,” 2022.
[6] J. Lee, K. Choi, and S. Kim, “RF-CAN: Random Forest-Based Intrusion Detection System for CAN Networks,” 2023.
[7] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and J.-H. Chen, “Using Deep Learning Networks to Identify Cyber Attacks on In-Vehicle Networks,” 2022.
[8] J. Yang, J. Hu, and T. Yu, “Federated AI-Enabled In-Vehicle Network Intrusion Detection for Internet of Vehicles,” 2022.
[9] L. Xing, K. Wang, H. Wu, H. Ma, and X. Zhang, “FL-MAAE: Intrusion Detection for IoV Using Federated Learning and Memory-Augmented Autoencoder,” 2023.
[10] E. Alalwany and I. Mahgoub, “Ensemble Learning-Based Real-Time Intrusion Detection System for In-Vehicle Networks,” 2024.
[11] J. Alsamiri and K. Alsubhi, “Federated Learning-Based Intrusion Detection Systems in the Internet of Vehicles: A Survey,” 2023.
[12] D. Basavaraj and S. Tayeb, “Towards a Lightweight Intrusion Detection Framework for In-Vehicle Networks,” 2022.
[13] K.-T. Cho and K. G. Shin, “Fingerprinting Electronic Control Units for Vehicle Intrusion Detection (CIDS),” 2016.
[14] M. M. Hasan, M. M. Islam, and M. M. Rahman, “An Intrusion Detection System Framework for In-Vehicle Networks Based on CAN-Bus Data,” 2023.
[15] A. Ahmed and H. H. Salem, “Integrated Vehicle Cybersecurity: Threat Modeling and Intrusion Detection for In-Vehicle Networks,” 2022.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)