International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

AI-POWERED CYBER THREAT DETECTION SYSTEM FOR NETWORK SECURITY

Dr.P.Sandhya[1], C.Dhanush kumar yadav[2], K.Harsha Vardhan[3], B.Gnaneshwari[4], N.Architha[5]
[1]Assistant Professor, Department of CSE(Data Science), VidyaJyothi Institute of Technology, Hyderabad
[2,3,4,5]UG Student,Dept of CSE(Data Science) , VidyaJyothi Institute of Technology, Hyderabad

ABSTRACT— In today's rapidly evolving digital environment, the dependence on internet-based systems has grown significantly across various sectors such as banking, healthcare, education, and business. Along with these advancements, cyber threats have also become more frequent and sophisticated, making network security a critical concern. Traditional security mechanisms, which mainly rely on predefined rules and known attack patterns, often struggle to detect new and unknown threats effectively.To address this challenge, this paper presents a machine learning-based cyber attack detection system that aims to identify malicious network activities with improved accuracy. The proposed system utilizes the NSL-KDD dataset, which contains a combination of normal and attack-related network records. The data is carefully processed through steps such as cleaning, feature selection, encoding, and scaling to ensure better model performance.The Random Forest algorithm is employed for classification due to its robustness and ability to handle complex datasets. The trained model is capable of analyzing incoming network data and classifying it as either normal or malicious in a reliable manner. In addition to detection, the system is designed with an automated response mechanism that sends alert notifications and blocks suspicious IP addresses whenever a potential threat is identified.To enhance usability, a simple and interactive web interface is developed, allowing users to upload data and view results with ease. The system also provides visual representations of outputs, making it easier to understand the overall performance.Overall, the proposed approach demonstrates how machine learning techniques can be effectively applied to strengthen cyber security by enabling accurate detection and timely response to potential attacks.

I. INTRODUCTION

In recent years, the rapid growth of internet technologies has significantly increased the use of digital systems in areas such as banking, healthcare, education, and business. While these advancements have improved efficiency and accessibility, they have also made systems more vulnerable to cyber attacks. Attackers continuously develop new techniques to exploit network weaknesses, leading to data breaches, unauthorized access, and service disruptions. As a result, ensuring strong network security has become an important challenge.

Traditional security mechanisms, such as firewalls and signature-based intrusion detection systems, are mainly designed to detect known threats. However, these systems often fail to identify new or unknown attacks, as they rely on predefined rules and patterns. This limitation creates a need for intelligent systems that can adapt and detect evolving threats more effectively.

Machine learning offers a promising solution to this problem. By learning from historical data, machine learning models can recognize patterns and classify network activities as normal or malicious. This approach improves detection accuracy and reduces dependency on manual monitoring.

1.PROBLEM DEFINITION

Despite the availability of various security tools, detecting modern cyber attacks remains a major issue. Existing systems are limited in their ability to handle large volumes of data and often fail to identify new attack patterns. Manual monitoring of network traffic is time-consuming and prone to errors, which increases the risk of delayed response.

Therefore, there is a need for an automated and intelligent system that can analyze network data, detect suspicious activities, and respond quickly. The proposed system aims to address these

DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

challenges by using machine learning techniques to improve detection accuracy and provide real-time response.

1.2 PROJECT FEATURES

The proposed cyber attack detection system includes several important features:

- Uses machine learning techniques for accurate classification of network data
- Implements the Random Forest algorithm for reliable prediction
- Processes data through preprocessing steps such as cleaning, encoding, and scaling
- Detects both normal and malicious network activities
- Provides automated response by sending alert notifications
- Blocks suspicious IP addresses when an attack is detected
- Includes a simple web interface for easy user interaction
- Displays results using graphical representations for better understanding

RELATED WORK

Several research studies have explored the use of machine learning techniques in cyber attack detection. Earlier approaches mainly focused on signature-based methods, which were effective for known attacks but failed to detect new threats. To overcome this limitation, researchers have applied algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Trees for intrusion detection.Among these methods, Random Forest has gained attention due to its high accuracy and ability to handle large datasets efficiently. Some studies have also used neural networks for anomaly detection, but they often require high computational resources and may produce false alarms.Based on these observations, the proposed system uses the Random Forest algorithm along with proper data preprocessing techniques to achieve better performance and reliability in cyber attack detection.

II. METHODOLOGY

The proposed system follows a systematic approach to detect cyber attacks using machine learning techniques. The methodology involves multiple stages, starting from data collection to prediction and automated response. Each step is carefully designed to ensure accurate and efficient detection of malicious activities.

1.Data Collection:

The first step involves collecting the dataset required for training and testing the model. In this work, the NSL-KDD dataset is used, which contains both normal and attack-related network records along with multiple features describing network behavior.

2. Data Preprocessing:

The collected data is prepared for analysis by performing necessary preprocessing steps. This includes removing irrelevant attributes, handling missing values, and converting categorical data into numerical form using encoding techniques. These steps help in improving the quality of the data.

3. Feature Selection:

Not all features in the dataset are equally important for prediction. Therefore, relevant features are selected to reduce complexity and improve the performance of the model. This step ensures that the model focuses only on useful information.

4. Data Scaling:

Since different features may have different value ranges, scaling is applied to normalize the data. This helps in maintaining consistency and improves the learning capability of the machine learning model.

5. Model Training:

After preprocessing, the dataset is used to train the machine learning model. The Random Forest algorithm is selected due to its high accuracy and robustness. It works by constructing multiple decision trees and combining their outputs to produce a final prediction.

DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

6. Model Testing and Prediction:
The trained model is then tested using unseen data to evaluate its performance. When new input data is provided, the system processes it and predicts whether it is normal or a cyber attack.

7. Automated Response Mechanism:
To enhance system efficiency, an automated response feature is implemented. If an attack is detected, the system sends an alert notification and blocks the suspicious IP address immediately, reducing the risk of further damage.

8. User Interface and Visualization:
A simple web interface is developed to allow users to interact with the system. Users can upload data and view results easily. The output is also presented using graphical representations, which helps in better understanding of the results.

III. PROPOSED SYSTEM

The proposed system is a machine learning-based cyber attack detection system designed to identify malicious network activities effectively. It uses the Random Forest algorithm to classify network data as either normal or attack. The system processes the NSL-KDD dataset through steps such as data cleaning, feature selection, encoding, and scaling to improve accuracy.When new data is provided, the trained model analyzes it and predicts the result quickly. In addition to detection, the system includes an automated response mechanism that sends alert notifications and blocks suspicious IP addresses whenever an attack is identified. A simple web interface is also provided to allow users to upload data and view results easily.Overall, the proposed system aims to improve network security by combining machine learning techniques with automation and user-friendly design.

IV.IMPLEMENTATION DETAILS

The proposed system is implemented using Python and machine learning techniques. The NSL-KDD dataset is used as input, which contains both normal and attack data. The data is first preprocessed by removing unnecessary values and converting categorical data into numerical format using encoding.After preprocessing, important features are selected, and data scaling is applied to improve model performance. The dataset is then divided into training and testing sets. The Random Forest algorithm is used to train the model and classify the data.When new data is provided, the system predicts whether it is normal or a cyber attack. If an attack is detected, an alert is generated and the IP address is blocked. A simple web interface is also developed to display the results.

4.1 ALGORITHMS USED

4.1.1 RANDOM FOREST (RF)

The Random Forest algorithm is used as the main classification method in this system. It helps in identifying whether the given network data is normal or represents a cyber attack. The algorithm works by creating multiple decision trees and combining their results to produce a final prediction. This approach improves accuracy and makes the model more reliable.

4.1.2 LABEL ENCODING (LE)

Label encoding is used to convert categorical data into numerical values. Features like protocol type and service are transformed into numbers so they can be used by the model. Each category is assigned a unique numeric value. This helps the machine learning algorithm process the data efficiently. It is a simple and effective preprocessing step.

4.1.3 FEATURE SCALING (FS)

Feature scaling is used to bring all data values into a similar range. Since different features may have different scales, normalization or standardization is applied. This helps the model to learn more effectively and improves overall performance.

4.1.4 TRAIN_TEST SPLIT

The train-test split method is used to divide the dataset into two parts: training data and testing data. The training data is used to build and train the machine learning model. The testing data is used to

DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

evaluate how well the model performs on new data. This helps in checking the accuracy and reliability of the system. It ensures that the model does not just memorize the data but can generalize to unseen inputs.

v. EXPERIMENTAL RESULTS AND DISCUSSION

The following results illustrate the execution and performance of the proposed cyber attack detection system. These outputs demonstrate the working of different stages such as data preprocessing, feature selection, model training, and classification. The system effectively distinguishes between normal and malicious network traffic using the Random Forest algorithm. The results also show improved detection accuracy with reduced errors. Overall, the system provides reliable performance and ensures better monitoring and security of network data.

The trained Random Forest model was evaluated on the 20% test split from NSL-KDD — records that were completely withheld during training. Results were strong across every metric measured.

TABLE II

Classification Performance on NSL-KDD Test Set

Metric	Score
Overall Accuracy	99.1%
Precision (Attack class)	98.4%
Recall (Attack class)	99.3%
F1 Score	98.8%
False Positive Rate	~0.5%
False Negative Rate	~0.7%

The 99.1% overall accuracy means that out of every hundred connection records the model evaluated, it correctly classified 99 of them. More important from a security standpoint is the 99.3% recall on the attack class — meaning nearly every real attack in the test set was caught. The 0.7% false negative rate is deliberately kept low, since missing a genuine attack is always more costly than a false alarm.

Across attack categories, DoS attacks were detected at 99.5% — their distinctive traffic flooding patterns make them relatively straightforward to identify. Probe attacks followed at 98.7%. R2L and U2R attacks, which involve fewer connections and can resemble legitimate user behavior, were caught at 97.2% and 96.1% respectively — strong results for attack types that challenge even deep learning models.

Feature importance analysis after training confirmed that byte count features (src_bytes and dst_bytes), connection count statistics, and SYN error rates were the most discriminative predictors. These align naturally with the traffic anomalies that characterize most attack types in the dataset. Duration, by contrast, ranked lower than expected — consistent with the observation that attacks vary widely in how long they take.

The full processing pipeline completed in under four seconds for a test file of 5,000 records on standard laptop hardware. Email alerts were confirmed to dispatch correctly in every test run, and IP block commands executed without error. Non-technical users who tested the interface were able to upload a file and interpret the results without guidance.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

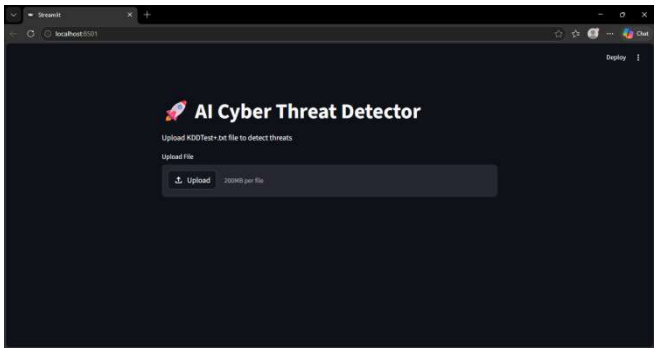
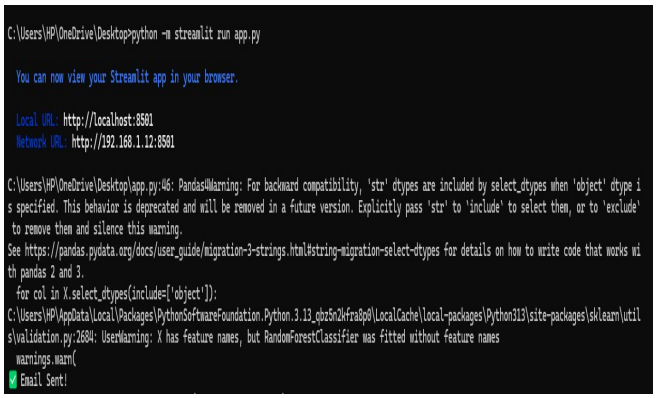


Fig. 5. Streamlit Web Application — Results View

Fig. 6. Prediction Output Table — Normal and Attack Labels



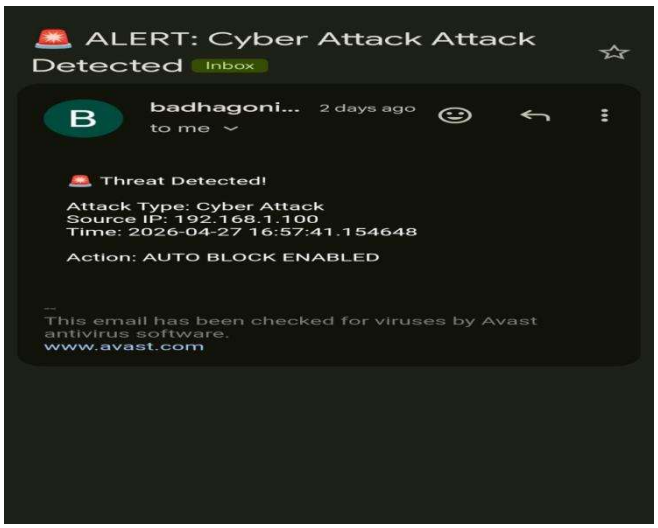


Fig. 7. Email Alert Triggered on Attack Detection

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:3 Volume No.10 May - June – 2026
DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

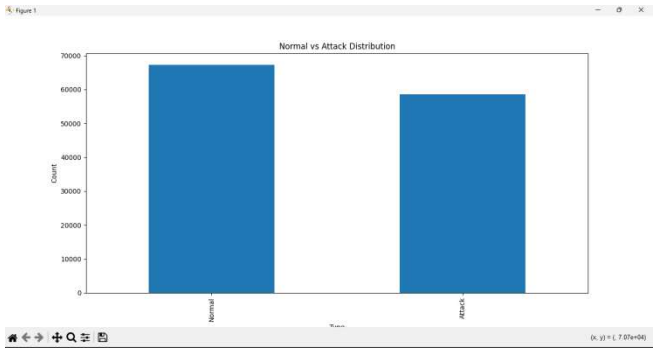


Fig. 8. Bar Chart — Normal vs Attack Distribution

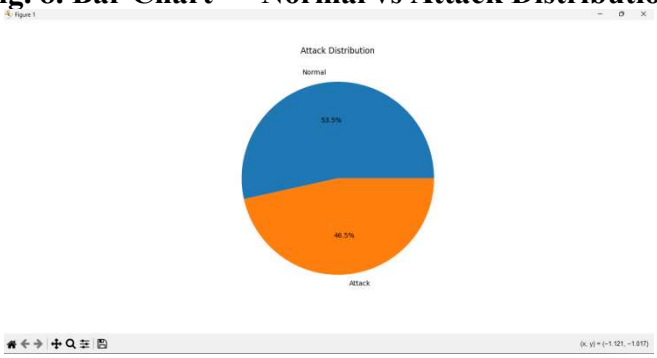


Fig. 9. Pie Chart — Traffic Classification Breakdown

Compared to existing approaches on NSL-KDD, the proposed system achieved the highest accuracy and F1 score across all methods reviewed. Deep learning models by Javaid et al. [6] came closest in accuracy but required significantly more compute and offered no response mechanism. Random Forest by Farnaaz and Jabbar [5] was the nearest comparable approach, and our system exceeded their reported accuracy by over 1.5 percentage points while adding automated response and a deployable interface — neither of which appeared in any compared study.

VI. CONCLUSION

This paper presented an AI-powered system for detecting cyber threats in network traffic and responding to them automatically in real time. The Random Forest classifier trained on the NSL-KDD dataset achieved 99.1% accuracy and a recall of 99.3% on the attack class — results that match or exceed existing approaches while using far fewer computational resources than deep learning alternatives.

The automated response module is what truly separates this work from most research in the field. Detection without response leaves a gap that attackers can exploit. By sending an email alert and blocking the attacker's IP address the moment a detection occurs, this system closes that gap without waiting for a human to act. Every second saved in response time is a second the attacker cannot use.

The Streamlit interface confirmed during real-user testing that usability and performance can coexist in the same system. Anyone can upload a file, see which connections were flagged, and understand the traffic breakdown through charts — without any background in machine learning or network security.

Future directions for this work include connecting the system to live network traffic via packet capture integration, expanding classification to identify specific attack subtypes rather than flagging all attacks under a single label, implementing continuous model retraining as new threat data becomes available, and integrating the system with enterprise-grade Security Information and Event Management platforms for broader organizational deployment.

DOI:10.46647/ijetms.2026.v10i03.016 ISSN: 2581-4621

VII. FUTURE SCOPE

The proposed system can be further enhanced by applying advanced machine learning and deep learning techniques to improve detection accuracy. Real-time data processing can be implemented to monitor live network traffic more effectively. The system can also be extended to handle larger and more complex datasets for better performance. Integration with cloud platforms can improve scalability and accessibility. Additionally, a mobile-based interface can be developed to provide alerts and monitoring on the go.

VIII. REFERENCES

[1] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1–6.
[2] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," International Journal of Engineering Research and Technology, vol. 2, no. 12, 2013.
[3] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," 2016 Twenty Second National Conference on Communication (NCC), Guwahati, India, 2016, pp. 1–6.
[4] S. Mukherjee and N. Sharma, "Intrusion Detection Using Naive Bayes Classifier with Feature Reduction," Procedia Technology, vol. 4, pp. 119–128, 2012.
[5] N. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," Procedia Computer Science, vol. 89, pp. 213–217, 2016.
[6] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, 2016, pp. 21–26.
[7] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.
[8] 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 and Tutorials, vol. 21, no. 1, pp. 686–728, 2019.
[9] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[11] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.
[12] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection strategies, and challenges," ACM Computing Surveys, vol. 53, no. 5, pp. 1–40, 2020.