International Journal of Advanced Trends in Engineering, Science and Technology (IJATEST-ISSN:2456-1126) Volume.11,Issue.2,2026
DOI:10.22413/ijatest/2026/V11/i2/03

Smart Guard AI an Autonomous Machine Learning System for Early Detection and Prevention of Cyber Threats

K B.A Mohammad Faiz, V. Mohammed Faizan, S Feroz Hussain, N. Mohammed Rayan

Abstract: ***Cybersecurity threats are rapidly increasing, posing serious risks to digital systems and sensitive data. This project presents an AI-powered intelligent framework, ThreatGuardian, for detecting and classifying cybersecurity attacks using machine learning algorithms. The system analyzes network data to identify abnormal patterns and potential threats. It uses trained machine learning models to improve detection accuracy and efficiency. A web-based interface developed using Django enables users to monitor and manage threats easily. Overall, it provides a reliable and scalable solution for modern cybersecurity challenges.***

I. INTRODUCTION

The rapid growth of digital technologies and interconnected systems has significantly increased the complexity of cybersecurity challenges in modern network environments. Organizations generate vast amounts of network data, including traffic logs, user activities, and system interactions, which can be analyzed to detect potential threats. The emergence of artificial intelligence and machine learning has enabled the extraction of meaningful patterns from such data, supporting the development of predictive models for identifying cyberattacks and improving system security.

Cybersecurity threat detection has become a critical concern due to the increasing frequency of attacks such as malware, phishing, denial-of-service, and data breaches. These attacks can compromise sensitive information and disrupt critical services. Traditional security mechanisms rely on signature-based detection and manual monitoring, which are often ineffective against evolving and unknown threats. These approaches fail to provide early detection, leading to delayed responses and increased vulnerability of systems.

Consequently, there is a growing need for intelligent, scalable, and automated security systems capable of identifying threats in real time and supporting proactive defense strategies. Machine Learning (ML) techniques provide an effective framework for analyzing complex and high-dimensional network data. These techniques enable systems to learn from historical data and adapt to new attack patterns without explicit programming.

In this work, an intelligent framework named ThreatGuardian is proposed, integrating machine learning algorithms such as Random Forest, Support Vector Machine, Decision Tree, and Neural Networks for accurate threat detection and classification. Ensemble approaches improve prediction accuracy and reduce overfitting, while classification models enable efficient categorization of cyber threats. The system is designed to analyze both structured and unstructured data to enhance detection capability..

A data preprocessing module is implemented to clean, normalize, and transform raw network data into a suitable format for analysis. Feature extraction techniques are applied to identify the most relevant attributes that contribute to threat detection. This improves the efficiency and accuracy of the machine learning models. The system also supports splitting of data into training and testing sets for effective model evaluation.

Despite advancements in cybersecurity analytics, several challenges remain, including handling large-scale datasets, reducing false positives, adapting to new attack patterns, and ensuring real-time performance. Many existing systems lack integration of intelligent learning mechanisms and user-friendly deployment platforms. These limitations reduce their effectiveness in practical environments.

To address these issues, the proposed system combines machine learning models with a web-based interface developed using Django, enabling seamless interaction and real-time monitoring. The interface allows users to upload data, view results, and manage threat detection processes efficiently. Visualization tools such as graphs and dashboards are included to present results clearly.

The system processes input data through multiple stages, including data collection, preprocessing, model training, and prediction. It identifies abnormal patterns and classifies threats into different categories. The results are displayed to the user along with detailed analysis and alerts for suspicious activities.

Additionally, the system is designed to be scalable and adaptable to different environments. It can handle large volumes of network data and can be extended to include additional machine learning models in the future. Security measures are also implemented to ensure data protection and controlled access.

.

II. LITERATURE SURVEY

The application of Machine Learning in cybersecurity has significantly improved the detection and classification of cyber threats. Previous studies focus on algorithms such as Random Forest, Decision Tree, Naïve Bayes, and deep learning models for analyzing network data and identifying malicious patterns. These approaches help in improving detection accuracy and handling large-scale cybersecurity datasets efficiently. Ensemble techniques further enhance performance by combining multiple models for better prediction.

However, existing approaches have several limitations. Many systems rely on traditional methods or single models,

which reduces their ability to detect complex and evolving cyber threats. Issues such as high false positive rates, lack of real-time detection, and limited scalability are common. Additionally, many studies focus only on detection without providing user-friendly interfaces or practical deployment solutions.

To address these challenges, recent research emphasizes the use of hybrid and ensemble machine learning models. Techniques such as Random Forest, AdaBoost, and Bernoulli Naïve Bayes are used to improve accuracy and efficiency. Data preprocessing, feature extraction, and proper model evaluation also play a crucial role in enhancing system performance.

Overall, the literature highlights the importance of integrating multiple machine learning techniques with efficient data handling and user-oriented systems. Such approaches improve threat detection accuracy, reduce manual effort, and provide scalable solutions for modern cybersecurity environments.

III. PROPOSED METHOD

Cybersecurity threat detection has been widely studied using machine learning techniques to identify and classify various types of cyberattacks. Several research works focus on improving detection accuracy and understanding malware behavior through different analytical approaches.

Adel Abusitta et al. (2021) conducted a comprehensive survey on malware classification and composition analysis, highlighting various machine learning techniques used for detecting and categorizing malware. The study emphasizes the importance of feature extraction and analysis in improving detection performance.

Sumit Lad and Amol Adamuthe (2020) proposed a malware classification model using Convolutional Neural Networks (CNN), demonstrating improved accuracy and efficiency compared to traditional methods. The study shows that deep learning techniques can automatically extract features and enhance classification results

Fangtian Zhong and Xiuzhen Cheng (2023) introduced MalFox, a GAN-based framework designed to generate adversarial malware examples. The study highlights the challenges of detecting advanced and evasive malware and shows how deep learning models can improve robustness against such attacks.

Jiaxing Chen et al. (2023) developed a model for analyzing malware propagation over wireless networks, emphasizing how malware spreads across interconnected systems. The study provides insights into predicting and controlling large-scale cyberattacks.

Robert Chun et al. (2019) analyzed malware embedded in PDF files and proposed techniques for identifying malicious content using automated scripts. The study highlights the vulnerabilities in commonly used file formats and the need for improved detection methods.

Overall, these studies demonstrate the effectiveness of machine learning and deep learning techniques in cybersecurity. They also highlight the importance of feature extraction, model selection, and advanced analytical methods for improving threat detection and system security.

IV. MATERIALS USED

4.1 Data Description

The dataset used in this project consists of network traffic and cybersecurity-related attributes that influence threat detection. Features include packet size, protocol type, source and destination IP behavior, and attack labels representing normal and malicious activities.

Each record represents a network instance used to classify potential cyber threats. Before training the models, the dataset is preprocessed by handling missing values, encoding categorical data, and normalizing features. The dataset is then divided into training and testing sets to build and evaluate machine learning models effectively.

4.2 Python Environment and Machine Learning Libraries

In this study, Python was used as the primary programming language for implementing the Machine Learning models and building the student performance prediction system. Python provides a flexible environment for data analysis, model training, and visualization. The system was developed using several open-source Machine Learning and data analysis libraries that support efficient model development and evaluation.

In this project, multiple machine learning algorithms are used for threat detection. The Random Forest algorithm is an ensemble learning technique that builds multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. It is highly effective for handling large datasets and provides reliable classification results.

The AdaBoost (Adaptive Boosting) algorithm is another ensemble method that improves model performance by focusing on misclassified data points. It assigns higher weights to incorrectly classified instances and iteratively improves the model, resulting in better accuracy for complex datasets.

The Bernoulli Naïve Bayes (BNC) classifier is a probabilistic algorithm suitable for binary classification problems. It assumes independence between features and works efficiently with binary or Boolean data. It is fast, computationally efficient, and performs well in detecting patterns related to cyber threats.

These algorithms are trained and evaluated using the dataset to identify the most effective model for cybersecurity threat detection. Their combination ensures improved accuracy, efficiency, and robustness of the proposed system.

4.3 Confusion Matrix

A confusion matrix is used to evaluate the performance of classification models by comparing predicted results with actual outcomes. It consists of four components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). This matrix helps in analyzing how accurately the system detects cyber threats and distinguishes between normal and malicious activities. It also provides insights into model reliability and error distribution.

Thus, in your project, the confusion matrix is presented as a structured table that compares actual vs predicted values to evaluate model effectiveness in threat detection.

4.4 Evaluation Metrics

DOI:10.22413/ijatest/2026/V11/i2/03

The performance of machine learning models is evaluated using standard classification metrics. These metrics help in measuring the accuracy and effectiveness of threat detection by comparing predicted results with actual values.

Evaluation metrics are used to measure how well the machine learning models perform in detecting cybersecurity threats. They are calculated using the values obtained from the confusion matrix, such as True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). These metrics help in comparing different models and selecting the most effective one for the system.

Accuracy is the most basic metric, which measures the overall correctness of the model. It shows how many predictions are correctly classified out of the total predictions. However, accuracy alone may not be sufficient when dealing with imbalanced datasets.

Precision measures how many of the predicted threats are actually correct. It is important when false alarms (false positives) need to be minimized. High precision means fewer incorrect threat alerts.

Recall (Sensitivity) measures how many actual threats are correctly detected by the system. It is important in cybersecurity because missing a real threat (false negative) can be very dangerous. High recall ensures most threats are identified.

F1 Score is the harmonic mean of precision and recall. It provides a balanced measure when both false positives and false negatives are important. It is widely used when there is a need to balance accuracy and detection capability.

Fig – 4: Classification Report

These evaluation metrics together provide a comprehensive understanding of model performance and help in selecting the best algorithm for effective and reliable cybersecurity threat detection.

V. METHODOLOGY

The methodology adopted in this project follows a structured machine learning pipeline to ensure accurate detection and classification of cybersecurity threats. The overall process includes data collection, preprocessing, feature extraction, model training, performance evaluation, and result analysis. Multiple classification algorithms are applied and compared to identify the most effective model for threat detection.

Initially, the dataset is collected and examined for inconsistencies, missing values, and redundant attributes. Data preprocessing techniques such as cleaning, normalization, and encoding are applied to prepare the dataset for analysis. Feature extraction is then performed to select the most relevant attributes that contribute to identifying cyber threats.

After preprocessing, machine learning algorithms including Random Forest, AdaBoost, and Bernoulli Naïve Bayes are implemented. These models are trained using the prepared dataset and tested to evaluate their performance. Each algorithm analyzes patterns in network data to classify activities as normal or malicious.

The models are evaluated using performance metrics such as Accuracy, Precision, Recall, and F1 Score. These metrics help in measuring the effectiveness of each model in detecting cyber threats. The model with the highest accuracy and balanced performance across all metrics is selected as the best-performing model.

Finally, the selected model is integrated into a web-based system for real-time threat detection and monitoring. The methodology ensures a reliable, efficient, and scalable approach for cybersecurity threat analysis.

5.1 Data Preprocessing

Data preprocessing is an essential step in developing an effective cybersecurity threat detection system. In this phase, the dataset containing network traffic and attack-related information is carefully analyzed to ensure data quality and consistency.

Initially, categorical attributes such as protocol type, service type, and attack labels are converted into numerical format using encoding techniques. This transformation is essential because machine learning algorithms require numerical input for effective training and accurate prediction of cybersecurity threats.

Missing values, duplicate records, and inconsistencies in the dataset are identified and handled appropriately to avoid inaccurate predictions. Irrelevant or redundant features that do not significantly contribute to cybersecurity threat detection are removed, which helps improve model efficiency and reduces computational complexity.

The dataset is then normalized or scaled to ensure that all features contribute equally during model training. Feature selection techniques are applied to identify and retain the most relevant attributes for accurate classification of cybersecurity threats, thereby improving model performance and efficiency.

Finally, the dataset is divided into independent variables (features) and the dependent variable (target), where the target represents whether the network activity is normal or malicious. An 80–20 train-test split is applied, with 80% of the data used for training and 20% for testing. This approach ensures proper evaluation of the model's performance on unseen data and improves its generalization capability.

5.2 Model Development and Selection

After preprocessing, multiple machine learning classification algorithms are implemented to identify the most suitable model for detecting cybersecurity threats.

The following algorithms were used:

- Logistic Regression
- BernoliNavie Bayes
- Random Forest

DOI:10.22413/ijatest/2026/V11/i2/03

- Adaboost classifier

Each algorithm is selected based on its learning capability and predictive strength. Logistic Regression is a linear model that performs well when there is a linear relationship between features and the target variable. It is efficient and suitable for binary classification tasks such as identifying normal and malicious network activities. Additionally, it applies regularization techniques to reduce overfitting and improve model generalization.

Decision Tree and Random Forest are tree-based models capable of capturing non-linear relationships in the dataset. Random Forest, being an ensemble learning method, builds multiple decision trees and combines their outputs to reduce variance and improve prediction stability.

AdaBoost is a boosting algorithm that enhances model performance by focusing on misclassified instances and combining multiple weak learners into a strong classifier. Bernoulli Naive Bayes is a probabilistic model suitable for binary classification tasks and performs efficiently on datasets with independent features.

All models are trained using the training dataset and evaluated on the testing dataset to ensure an unbiased comparison of their performance in detecting and classifying cybersecurity threats.

Table-5:Components of Random Forest.

Component	Purpose	Role in Cyber Security Threat Detection
Input Data	Takes network data as input	Network traffic and system logs are fed into the model
Decision Tree	Performs classification using tree structure	Identifies patterns in normal and malicious activities
Feature selection	Selects important features randomly	Focuses on key attributes like protocol, traffic, behavior
Ensemble voting	Combines outputs of multiple trees	Improves accuracy in detecting cyber threats
Final Output	Procedure Classification result	Predicts normal activity or type of cyberattack

5.3 Model Evaluation

A comprehensive evaluation of the developed machine learning models is conducted to assess their classification performance and generalization capability. The evaluation process involves splitting the dataset into training and testing subsets using an 80–20 train-test split strategy. The models are trained on the training dataset and evaluated on the unseen testing dataset to ensure an unbiased assessment of their performance.

To measure the effectiveness of the models, classification evaluation metrics such as accuracy, precision, recall, and F1-score are used. These metrics provide a clear understanding of the model's ability to correctly identify cybersecurity threats, minimize false predictions, and maintain a balance between detection rate and reliability.

Among the evaluated models, the Random Forest classifier demonstrated the strongest performance in detecting cybersecurity threats. It achieved the highest accuracy, indicating its effectiveness in correctly classifying network activities as normal or malicious. This high performance confirms the suitability of Random Forest for handling complex and non-linear patterns in the dataset.

In terms of classification performance, Random Forest also showed better precision, recall, and F1-score compared to other models. These results indicate that the model minimizes false predictions and provides reliable threat detection, ensuring both high accuracy and stability in real-world scenarios.

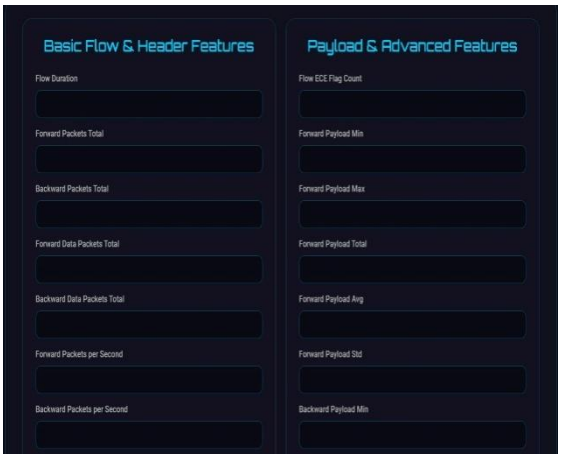
Random Forest and AdaBoost also exhibited strong performance, achieving high accuracy and maintaining good precision, recall, and F1-score values. These ensemble-based models effectively capture complex feature interactions and non-linear patterns present in network traffic data. However, their performance was slightly lower compared to Random Forest, indicating that Random Forest is more suitable for this dataset due to its higher stability and better overall classification performance.

5.4 Model Integration

The final phase of this project involves transforming the selected machine learning model into a functional web-based cybersecurity threat detection system. After evaluating multiple classification algorithms, the Random Forest classifier is identified as the best-performing model. This trained model is then integrated into a Python-based web application developed using the Django framework to enable real-time threat detection.

The trained model is saved using model persistence techniques to ensure that it can be reused without the need for retraining. During application execution, the model is loaded into the backend environment and remains ready to process incoming data and generate predictions.

The web application is designed to accept input data through an interactive user interface. Users provide network-related parameters such as protocol type, service type, packet size, and other relevant traffic features. The input structure is kept consistent with the features used during model training to ensure accurate and reliable threat detection.



DOI:10.22413/ijatest/2026/V11/i2/03

Fig-7: Input page

Once the user submits the form, the backend system validates the input values and performs necessary preprocessing steps, including encoding categorical features and arranging all input attributes in the required format. The processed data is then passed to the trained Random Forest model to predict whether the given network activity is normal or malicious.

The model generates a predicted score representing the expected academic performance of the individual student.

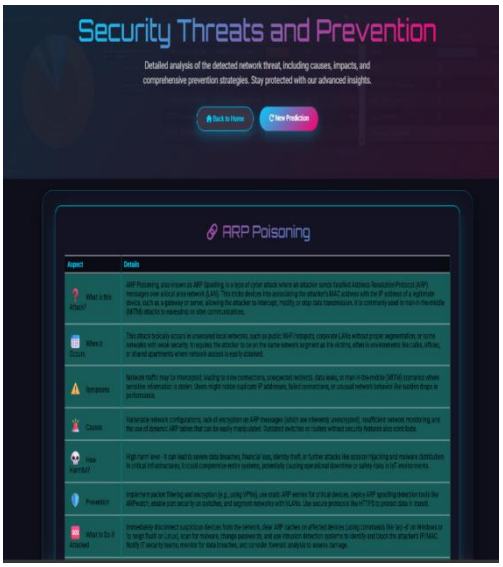


Fig-8: Output page

The result is instantly displayed on the web interface, providing clear and actionable insights such as

- Predicted Result (Normal or Malicious)
- Threat Classification Type (e.g., DoS, Probe, R2L, U2R)
- Risk Level (e.g., Low, Medium, High)
- Detection Confidence Score
- Recommended Security Actions

The system provides appropriate recommendations based on the detected threat type to help users take preventive and corrective measures for enhancing cybersecurity.

Unlike bulk processing systems, this application is designed for real-time, individual prediction of network traffic instances. It does not require CSV file uploads or batch processing; instead, it focuses on analyzing a single input at a time, ensuring simplicity, efficiency, and user-friendliness.

This seamless integration of the trained Random Forest model into a Django-based web application enables efficient real-time threat detection while maintaining high accuracy, scalability, and practical usability.

VI. CONCLUSION

This project successfully developed and implemented a machine learning-based system for detecting and classifying cybersecurity threats. Multiple classification algorithms, including Logistic Regression, Bernoulli Naive Bayes, Random Forest, and AdaBoost, were evaluated to identify the most suitable model for threat detection.

Among the tested models, the Random Forest classifier demonstrated the highest performance in terms of accuracy, precision, recall, and F1-score. Its ability to handle complex and non-linear relationships makes it the most effective and reliable model for detecting malicious network activities.

The selected model was successfully integrated into a Django-based web application that enables real-time analysis of network traffic. The system provides clear outputs such as threat classification, risk level, and recommended security actions, helping users to understand and respond to potential cyber threats effectively.

Overall, the proposed system transforms machine learning concepts into a practical cybersecurity solution that enhances threat detection, improves decision-making, and contributes to a more secure digital environment.

REFERENCES

[1] Adel Abusitta, Miles Q. Li, and Benjamin C. M. Fung, “Malware Classification and Composition Analysis: A Survey of Recent Development,” 2021.
[2] Sumit S. Lad and Amol C. Adamuthe, “Malware Classification with Improved Convolutional Neural Network Model,” 2020.
[3] Fangtian Zhong and Xiuzhen Cheng, “MalFox: Camouflaged Adversarial Malware Example Generation,” 2023.
[4] Robert Chun, Mike Wu, and Navneet Goel, “Malware Analysis on PDF,” 2019 .
[5] Jiaxing Chen et al., “Modeling and analyzing malware propagation over wireless networks,” 2023.
[6] Sophie Martin, Daniel Clark. “Hybrid AI Moadels that combines DL and Traditional techniques,” 2025.