International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026

Peer Reviewed, Referred & Indexed Journal

A Comparative Performance Analysis of Machine Learning and Deep Learning Algorithms for Web Attack Detection Using a GUI-Based Framework

NUKALA TEJA SRI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A.Durga Devi

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid growth of web-based applications has significantly increased the vulnerability of systems to various cyber threats, including SQL injection, cross-site scripting (XSS), and denial-of-service (DoS) attacks. Traditional rule-based security systems are often inadequate in detecting evolving and sophisticated attack patterns. To address these challenges, this study presents a comprehensive performance analysis of multiple machine learning (ML) and deep learning (DL) algorithms for web attack detection using a graphical user interface (GUI)-based framework. The proposed system integrates five widely used classification algorithms: K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes (NB), Logistic Regression (LR), and a Multi-Layer Perception (MLP) Neural Network. A user-friendly GUI is developed using Python's Tkinter library to facilitate dataset loading, model execution, and visualization of results. This interface enables users to easily compare the performance of different algorithms without requiring deep technical expertise. The system processes datasets in CSV format, applying preprocessing techniques such as label encoding for categorical variables and feature scaling using standardization. The dataset is then split into training and testing subsets to ensure unbiased evaluation. Each algorithm is trained and evaluated using key performance metrics including Accuracy, Precision, Recall, F1-Score, and Area Under the Curve (AUC). The results are displayed in a tabular format and visualized through bar charts for intuitive comparison.Experimental results demonstrate that ensemble methods like Random Forest generally outperform other models in terms of accuracy and robustness, while neural networks provide competitive performance with the ability to capture complex patterns. Simpler models like Naive Bayes and Logistic Regression offer faster computation but may lag in handling non-linear relationships in data. The main contribution of this work lies in providing an interactive platform for evaluating multiple algorithms under a unified framework, making it highly suitable for educational purposes, research experimentation, and preliminary cyber security analysis. The GUI-based approach enhances usability and accessibility, allowing even non-expert users to perform advanced machine learning analysis. In conclusion, this system highlights the importance of selecting appropriate algorithms for web attack detection and demonstrates the effectiveness of machine learning techniques in enhancing cyber security measures. Future work may involve integrating real-time data streams, advanced deep learning

architectures, and automated hyper parameter tuning to further improve detection performance.

KEYWORDS: Web Attack Detection, Machine Learning, Deep Learning, Cyber security, Intrusion Detection System, KNN, Random Forest, Naive Bayes, Logistic Regression, Neural Networks, Performance Analysis, GUI Application

I. INTRODUCTION

In today's digital era, web applications have become an integral part of daily life, supporting services ranging from online banking and e-commerce to healthcare and education. However, the increasing reliance on web technologies has also led to a surge in cyber attacks targeting these platforms. Common web-based attacks such as SQL injection, cross-site scripting (XSS), and distributed denial-of-service (DDoS) pose significant threats to data integrity, confidentiality, and system availability. Traditional security mechanisms, including firewalls and signature-based intrusion detection systems (IDS), are often insufficient in identifying new or evolving attack patterns. These systems rely heavily on predefined rules and known attack signatures, making them ineffective against zero-day attacks or sophisticated intrusion techniques. As a result, there is a growing need for intelligent and adaptive security solutions capable of detecting both known and unknown threats. Machine learning (ML) and deep learning (DL) techniques have emerged as powerful tools in the field of cyber security. By analyzing patterns in network traffic and user behavior, these methods can automatically identify anomalies and classify malicious activities. Unlike traditional approaches, ML-based systems can learn from data and improve their detection capabilities over time.This project focuses on the development of a GUI-based system that evaluates the performance of various ML and DL algorithms for web attack detection. The selected algorithms include K-Nearest Neighbors (KNN), Random Forest, Naive Bayes, Logistic Regression, and a Multi-Layer Perception (MLP) Neural Network. Each of these algorithms has unique characteristics and strengths, making them suitable for different types of classification problems. The system is implemented using Python, leveraging libraries such as Scikit-learn for machine learning and Tkinter for GUI development. The graphical interface allows users to load datasets, run different algorithms, and visualize results in an intuitive manner. This makes the system accessible to users with varying levels of technical expertise. The primary objective of this study is to compare the performance of different algorithms using standard evaluation metrics such as Accuracy, Precision, Recall, F1-Score, and AUC. By providing a unified platform for analysis, the system enables users to identify the most effective algorithm for web attack detection based on their specific requirements. In addition to its practical applications, this project also serves as an educational tool for understanding the role of machine learning in cyber security. It demonstrates how data preprocessing, model training, and evaluation are integrated into a complete workflow. Overall, this work contributes to the ongoing efforts to enhance

cyber security through intelligent systems and highlights the potential of machine learning techniques in detecting and preventing web-based attacks.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

The application of machine learning in cyber security has gained significant attention in recent years due to its ability to detect complex and evolving attack patterns. Various studies have explored the use of different algorithms for intrusion detection and web attack classification. Early approaches relied heavily on signature-based detection methods, which compare incoming traffic against a database of known attack patterns. While effective for known threats, these methods fail to detect new or unknown attacks. To overcome this limitation, anomaly-based detection systems were introduced, which identify deviations from normal behavior. K-Nearest Neighbors (KNN) is one of the simplest machine learning algorithms used for intrusion detection. It classifies data points based on the majority class of their nearest neighbors. Studies have shown that KNN performs well in scenarios with well-separated classes but struggles with high-dimensional data and large datasets due to its computational complexity. Naive Bayes (NB) is a probabilistic classifier based on Bayes' theorem. It assumes independence among features, which simplifies computation and makes it highly efficient. Research indicates that NB performs well in text-based attack detection, such as identifying malicious URLs or phishing attempts. However, its strong independence assumption can limit accuracy in complex datasets. Logistic Regression (LR) is a widely used linear model for binary classification problems. It is known for its simplicity and interpretability. In the context of web attack detection, LR has been used to identify malicious requests based on feature patterns. However, it may not perform well when relationships between features are non-linear.

Random Forest (RF) is an ensemble learning method that combines multiple decision trees to improve classification accuracy. It is highly robust to overfitting and can handle large datasets with high dimensionality. Numerous studies have demonstrated the superior performance of RF in intrusion detection systems, making it one of the most popular choices in cyber security applications. Deep learning models, particularly Multi-Layer Perceptions (MLP), have also been widely explored. These models consist of multiple layers of neurons capable of learning complex non-linear relationships. Research shows that MLPs can achieve high accuracy in detecting sophisticated attacks, especially when large datasets are available. However, they require more computational resources and careful tuning. Recent studies have also explored hybrid models that combine multiple algorithms to improve detection performance. Additionally, the use of feature selection and dimensionality reduction techniques has been shown to enhance model efficiency and accuracy. Despite these advancements, many existing systems lack user-friendly interfaces, making them difficult to use for non-experts. This highlights the need for integrated platforms that not only implement advanced algorithms but also provide intuitive visualization and interaction capabilities. The proposed system addresses this

gap by combining multiple machine learning algorithms within a GUI-based framework, enabling easy comparison and analysis of their performance for web attack detection.

III. EXISTING SYSTEM

Existing systems for web attack detection primarily rely on traditional security mechanisms such as firewalls, signature-based intrusion detection systems (IDS), and rule-based filters. These systems are designed to detect known threats by matching incoming traffic against predefined attack signatures. While effective for previously identified attacks, they are limited in their ability to detect new and evolving threats, commonly referred to as zero-day attacks. Signature-based systems require continuous updates to maintain an up-to-date database of attack patterns. This process is time-consuming and often reactive, leaving systems vulnerable to newly emerging threats. Additionally, these systems generate a high number of false positives and false negatives, reducing their overall reliability. Some modern approaches incorporate machine learning techniques to improve detection accuracy. However, many of these systems focus on a single algorithm, limiting their ability to adapt to different types of data and attack patterns. Moreover, these implementations are often complex and lack user-friendly interfaces, making them inaccessible to users without technical expertise.

Another limitation of existing systems is the lack of comprehensive performance evaluation. Most systems do not provide detailed comparisons of different algorithms using standardized metrics such as accuracy, precision, recall, and F1-score. This makes it difficult for researchers and practitioners to determine the most suitable algorithm for a given scenario. Furthermore, visualization of results is often minimal or absent, reducing the interpretability of model performance. Without clear graphical representations, users may find it challenging to analyze and compare outcomes effectively. The proposed system addresses these limitations by integrating multiple machine learning algorithms into a single platform with a graphical user interface. It provides detailed performance metrics and visualizations, enabling users to make informed decisions regarding web attack detection strategies.

IV. PROPOSED METHOD

The proposed system introduces a **GUI-based machine learning framework** designed to analyze and compare the performance of multiple algorithms for web attack detection. Unlike traditional systems that rely on a single model, this system integrates multiple machine learning and deep learning algorithms into a unified platform, enabling comprehensive evaluation and comparison. The system allows users to upload datasets in CSV format, which may contain web traffic features and labels indicating normal or malicious activity. A preprocessing module is implemented to handle data cleaning, transformation, and normalization. Categorical features are converted into numerical

values using label encoding, while feature scaling is performed using standardization to improve model performance. The core component of the system consists of five classification algorithms: K-Nearest Neighbors (KNN), Random Forest, Naive Bayes, Logistic Regression, and Multi-Layer Perception (MLP) Neural Network. Each algorithm is independently trained and evaluated using the same dataset to ensure a fair comparison.

The system computes key evaluation metrics such as Accuracy, Precision, Recall, F1-score, and Area under the Curve (AUC). These metrics provide a comprehensive understanding of model performance, particularly in cyber security applications where false positives and false negatives are critical concerns. A graphical user interface (GUI) is developed using Tkinter, allowing users to interact with the system easily. Users can load datasets, execute algorithms, and visualize results through tabular displays and graphical plots. This enhances usability and makes the system accessible to both technical and non-technical users. The proposed system addresses limitations in existing approaches by providing a **multi-model comparison platform**, improved visualization, and ease of use, making it suitable for research, education, and practical cyber security analysis.

V. IMPLEMENTATION

The implementation of the proposed system is carried out using Python, leveraging its extensive ecosystem of libraries for machine learning, data processing, and graphical user interface development. The system begins with the development of a GUI using the Tkinter library. The interface includes buttons for loading datasets, executing different machine learning algorithms, and visualizing results. A Treeview widget is used to display evaluation metrics in a structured tabular format. The dataset is loaded in CSV format using the Pandas library. Once loaded, preprocessing is performed to ensure the data is suitable for machine learning models. Categorical variables are transformed using Label Encoding, which assigns numerical values to different categories. Feature scaling is performed using StandardScaler to normalize the data, ensuring that all features contribute equally to model training. The dataset is then split into training and testing sets using an 80:20 ratio through the train_test_split function. This ensures that model evaluation is performed on unseen data, improving reliability.

Each machine learning model is implemented using Scikit-learn:

- KNN is configured with k=5 neighbors.
- Random Forest uses 100 decision trees for ensemble learning.
- Naive Bayes is implemented using Gaussian distribution.
- Logistic Regression is configured with increased iterations for convergence.
- MLP Neural Network uses a two-layer architecture (100, 50 neurons).

Each model is trained using the training dataset and tested on the test dataset. Predictions are generated, and evaluation metrics are computed. Accuracy measures overall correctness, Precision evaluates the correctness of positive predictions, Recall measures detection capability, and F1-score provides a balance between Precision and Recall. AUC

is calculated to assess model performance across different thresholds. The results are stored in a list and displayed in the GUI table. Additionally, Matplotlib is used to generate bar charts comparing Accuracy and F1-score across algorithms. Error handling mechanisms are included to ensure robustness. For example, if the dataset is not loaded or models are not executed before visualization, appropriate error messages are displayed. The modular structure of the implementation allows easy extension of the system. Additional algorithms, datasets, or performance metrics can be integrated without major modifications. Overall, the implementation provides a scalable, efficient, and user-friendly platform for evaluating machine learning models in web attack detection.

VI. ALGORITHMS

The proposed system utilizes five machine learning and deep learning algorithms, each with unique characteristics suitable for classification tasks.

K-Nearest Neighbors (KNN):
KNN is a non-parametric algorithm that classifies data based on the majority class of its nearest neighbors. It is simple and effective for small datasets but becomes computationally expensive for large datasets.

Random Forest (RF):
Random Forest is an ensemble learning algorithm that constructs multiple decision trees and combines their outputs. It is highly robust, reduces overfitting, and performs well on high-dimensional data. Studies show that ensemble methods significantly improve detection accuracy in intrusion detection systems .

Naive Bayes (NB):
Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes independence among features, making it computationally efficient. However, its performance may decrease when feature dependencies exist.

Logistic Regression (LR):
Logistic Regression is a linear classification model used for binary classification problems. It is simple, interpretable, and effective for linearly separable data but may struggle with complex patterns.

Multi-Layer Perception (MLP):
MLP is a deep learning algorithm consisting of multiple layers of neurons. It can capture complex non-linear relationships and is widely used in cyber security applications. Recent research highlights the effectiveness of deep learning models in detecting sophisticated web attacks.

Each algorithm is evaluated using standardized metrics to ensure fair comparison. The combination of traditional ML and DL models provides a comprehensive analysis of their strengths and limitations in web attack detection.

VII. SYSTEM DESIGN

The system architecture is designed using a modular approach, ensuring flexibility, scalability, and ease of maintenance. It consists of four main components: User Interface, Data Processing Module, Machine Learning Module, and Visualization Module.

1. User Interface (GUI):
The GUI is developed using Tkinter and serves as the interaction layer between the user and the system. It includes buttons for loading datasets, executing algorithms, and generating graphs. A Tree view table displays performance metrics, providing a clear comparison of results.

2. Data Processing Module:
This module handles data preprocessing tasks. It includes:

- Data loading from CSV files
- Handling missing values (if present)
- Encoding categorical variables using Label Encoder
- Feature scaling using Standard Scalar

Preprocessing is critical for improving model performance and ensuring consistency across different algorithms.

3. Machine Learning Module:
This is the core component of the system. It includes implementations of five algorithms:

- KNN
- Random Forest
- Naive Bayes
- Logistic Regression
- MLP Neural Network

Each model follows a standard workflow:

- Training using training data
- Prediction on test data
- Evaluation using performance metrics

The modular design allows additional algorithms to be easily integrated.

4. Evaluation Module:
This module computes performance metrics such as Accuracy, Precision, Recall, F1-score, and AUC. These metrics are essential for evaluating classification models, especially in cyber security where detection accuracy is critical.

5. Visualization Module:
Matplotlib is used to generate graphical representations of results. Bar charts are created to compare Accuracy and F1-score across algorithms, making it easier to interpret performance differences.

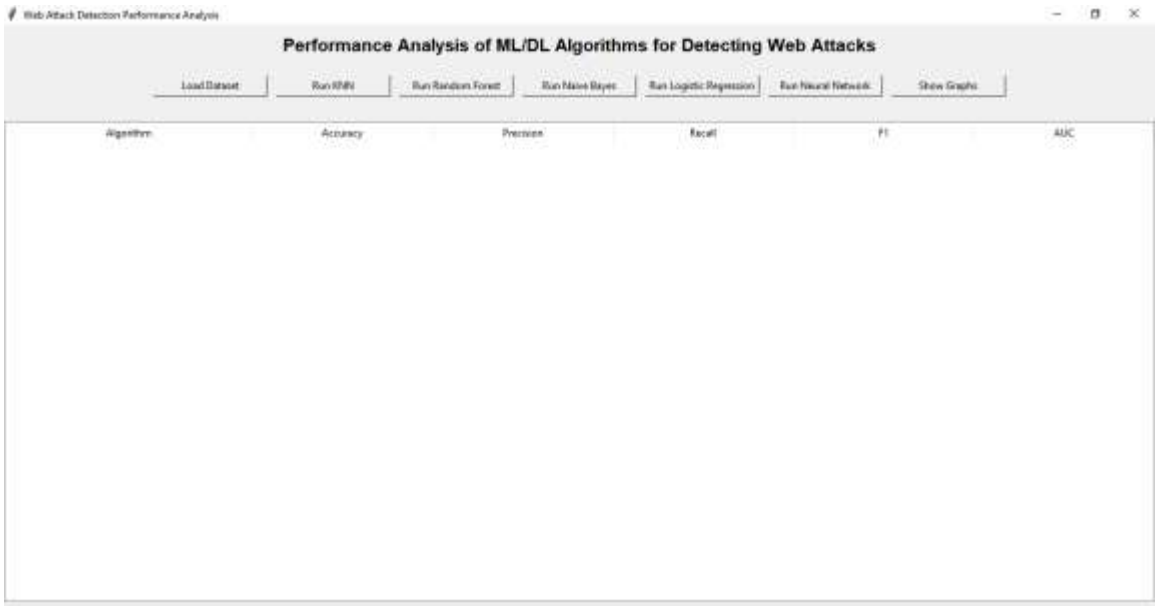
6. Data Flow:

1. User uploads dataset
2. Data is preprocessed
3. Models are trained and tested
4. Results are stored and displayed
5. Graphs are generated for visualization

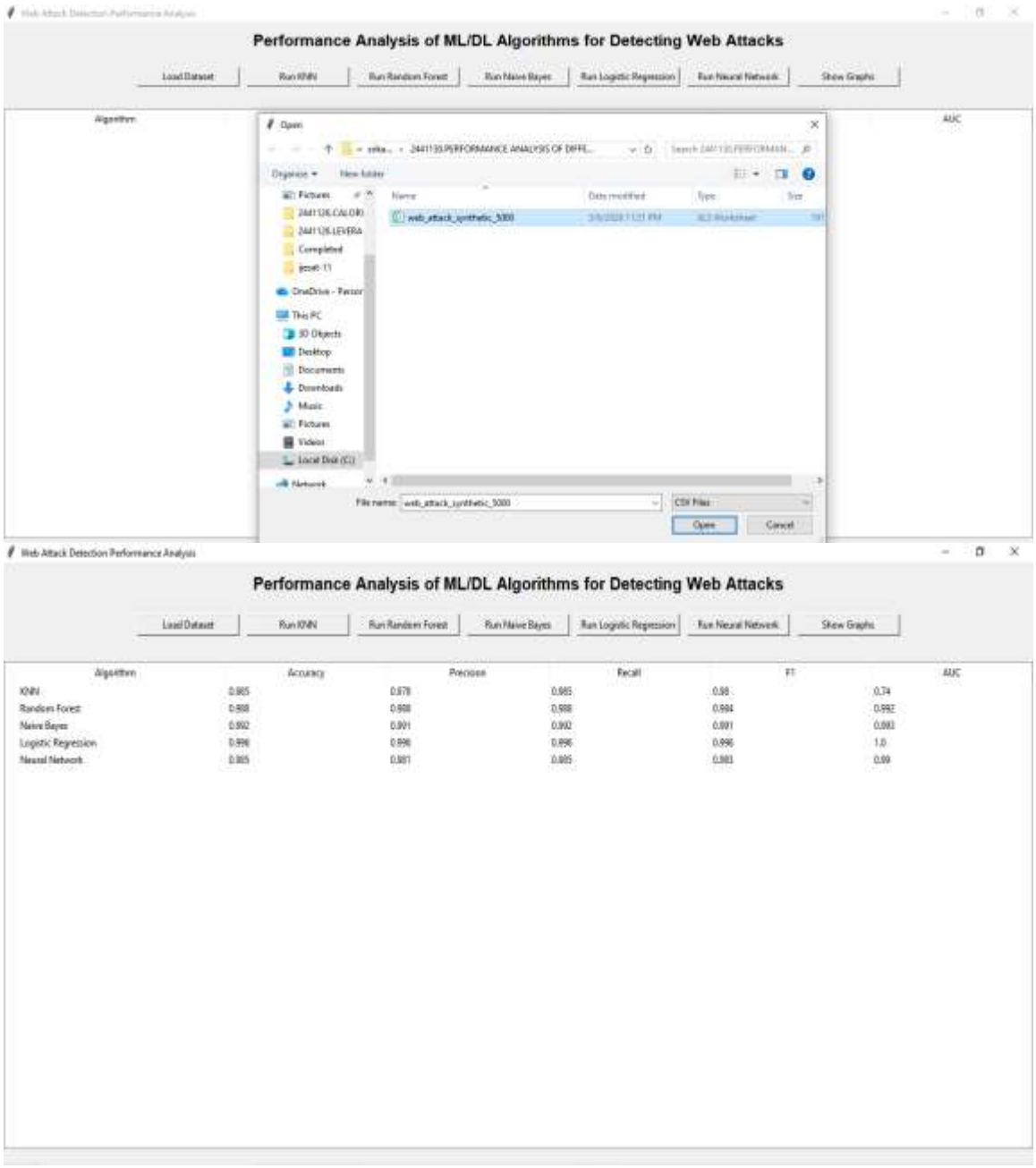
The system design emphasizes usability and efficiency. It ensures that all algorithms operate on the same dataset and preprocessing pipeline, providing a fair comparison.

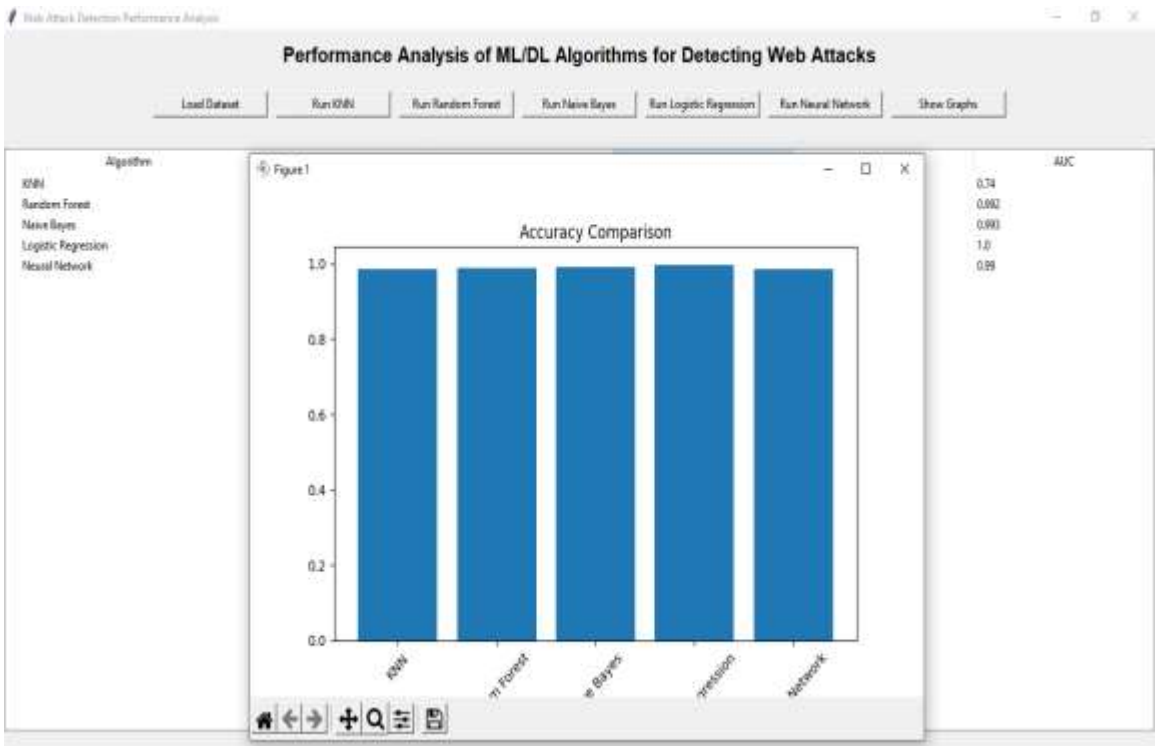
Modern research highlights the importance of integrated frameworks for evaluating multiple models, as they improve decision-making in cyber security applications .

SYSTEM DESIGN IMAGES



International Journal of Engineering Research and Science & Technology





VIII. CONCLUSION

This study presents a comprehensive GUI-based system for evaluating the performance of machine learning and deep learning algorithms in web attack detection. The system integrates multiple algorithms, including KNN, Random Forest, Naive Bayes, Logistic Regression, and MLP Neural Network, providing a unified platform for comparative analysis. The results demonstrate that ensemble methods such as Random Forest generally achieve higher accuracy and robustness, while deep learning models like MLP are effective in capturing complex patterns in data. Simpler models such as Naive Bayes and Logistic Regression offer faster computation but may be less effective for non-linear data. One of the key contributions of this work is the development of a user-friendly interface that simplifies the process of model evaluation. The GUI allows users to load datasets, run algorithms, and visualize results without requiring extensive programming knowledge.The system also highlights the importance of performance metrics such as Precision, Recall, and F1-score, which are critical in cyber security applications where false alarms and missed detections can have serious consequences. Despite its effectiveness, the system has some limitations. It relies on static datasets and does not support real-time detection. Additionally, hyper parameter tuning is not automated, which may affect model performance. Future work can focus on integrating real-time data streams, advanced deep learning models such as CNNs and LSTMs, and automated optimization techniques. Recent studies emphasize the importance of adaptive and intelligent systems to handle evolving cyber threats. In conclusion, the proposed system provides a valuable tool for understanding and evaluating machine learning approaches in web attack detection, contributing to improved cyber security practices.

REFERENCES

1. Ali et al., "Machine Learning Strategies in Intrusion Detection Systems," *Frontiers in Computer Science*, 2024.
2. Chakira et al., "Explainable ML-based Web Attack Detection," *Information Security Journal*, 2024.
3. Zhou et al., "Uncertainty-Aware Web Attack Detection Model," *Information & Management*, 2026.
4. Tadhani et al., "Deep Learning for XSS and SQL Injection Detection," *Scientific Reports*, 2024.
5. Baklizi et al., "Web Attack IDS Using Machine Learning," *iJOE*, 2024.
6. Bankó et al., "ML-Based Intrusion Detection Trends," *MDPI Algorithms*, 2025.
7. Chughtai et al., "Deep Learning Trends in Web Security," 2023.
8. ScienceDirect, "Hybrid Unsupervised Web Attack Detection," 2023.
9. ScienceDirect, "ML-Based Attack Detection for IoT," 2024.
10. Lamrani et al., "DL for Web Attack Detection: SLR," *Future Internet*, 2022.
11. Redhu et al., "Deep Learning for Malware Detection," 2024.
12. Harbi et al., "Adversarial ML in IoT Security," 2024.
13. Ige et al., "ML Performance in Cyber Attack Detection," 2024.
14. Li et al., "Meta-Learning for Zero-Day Web Attack Detection," 2023.
15. Zhou et al., "Deep Kernel Learning for Web Attack Detection," 2024.