RESEARCH ARTICLE OPEN ACCESS

Enhanced Explainable AI Method To Detect Threat in E-Commerce Website

Mr. A. Vivekanandhan *, R. Srinithi**, M. Deva Tharshini**, M. Nithya Sri**, S. Ramya**

*Assistant Professor -Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal, Mayiladuthurai,
**UG Students - Computer Science and Engineering, A.V. C. College of Engineering, Mannampandal, Mayiladuthurai,
India

Abstract: The rapid growth of e-commerce platforms has significantly increased the risk of cyberattacks such as Distributed Denial of Service (DDoS), brute-force login attempts, and fake account creation. Traditional rule-based security mechanisms often fail to detect sophisticated and evolving attack patterns. This paper proposes an intelligent cyberattack detection system that combines machine learning and explainable artificial intelligence techniques to improve detection accuracy and transparency. The proposed system uses the LightGBM algorithm to classify network traffic and identify malicious activities in real time. To enhance interpretability, SHAP is used to explain model predictions by highlighting the contribution of each feature to the final decision. The system is integrated with an e-commerce web application deployed through Nginx, where server logs are continuously monitored. Detected attacks are displayed on a real-time dashboard along with detailed explanations and recommended mitigation strategies. Experimental results demonstrate that the proposed system achieves high detection accuracy while providing meaningful insights into model decisions.

Keywords: Cyber Security, E-Commerce Security, Machine Learning, Explainable AI, LightGBM, SHAP, Intrusion Detection System

I. Introduction

E-commerce platforms have become an essential component of modern digital infrastructure, enabling online transactions, product browsing, and customer interactions. However, the increasing reliance on web-based systems has also made them prime targets for cyberattacks.

Common attacks such as DDoS, brute-force login attempts, and automated bot activities can disrupt services, compromise sensitive user data, and lead to financial losses. Traditional security mechanisms rely heavily on static rules and signatures, which are insufficient to detect dynamic and evolving threats.

To address these limitations, machine learning techniques have been widely adopted in intrusion detection systems. However, many machine learning models operate as black boxes, making it difficult to understand the reasoning behind their predictions.

This paper introduces a hybrid approach that integrates machine learning with explainable artificial intelligence to provide both accurate detection and interpretability.

II. Related Work

Reference [1] discusses the role of Explainable Artificial Intelligence in improving transparency and trust in AI-driven systems used in e-commerce platforms. The study explains how XAI techniques can help users understand how artificial intelligence models make decisions in recommendation systems and business analytics. By providing interpretable insights, the system improves user confidence and transparency in automated decision-making processes. However, the research mainly focuses on explaining AI decisions and does not address real-time cyber threat detection in e-commerce environments.

Reference [2] explores the integration of Explainable Artificial Intelligence in anomaly detection systems for threat management in e-commerce platforms. The authors highlight the importance of combining anomaly detection techniques with explainable models to improve

transparency and analysis of security threats. The study demonstrates how XAI methods can identify abnormal patterns in online transactions and user activities. However, the proposed approach mainly focuses on anomaly detection concepts and lacks a practical real-time deployment framework for monitoring and preventing cyberattacks.

Reference [3] proposes an advanced Temporal Convolutional Network framework for intrusion detection in Electric Vehicle (EV) charging infrastructure. The system uses deep learning models to analyze temporal patterns in network traffic and detect potential cyber intrusions. The results show that temporal deep learning models can significantly improve the accuracy of intrusion detection systems. However, the framework is specifically designed for EV charging infrastructure and cannot be directly applied to web-based or e-commerce systems without further adaptation.

Reference [4] introduces an Explainable AI-based recommender system designed for e-commerce applications. The system provides personalized product recommendations while also offering explanations for the recommendations generated by the model. This improves user satisfaction and transparency in recommendation systems. However, the proposed approach mainly focuses on recommendation systems and does not address cybersecurity threats or malicious activity detection in e-commerce platforms.

Reference [5] examines explainable recommender systems in online shopping environments. The study demonstrates how interpretable machine learning models can provide clear explanations for product recommendations and user behavior analysis. This helps improve user trust and decision-making in e-commerce platforms. Nevertheless, the research does not consider cyber threat detection or real-time monitoring of malicious user activities in web applications.

Reference [6] provides a comprehensive survey on Explainable Artificial Intelligence techniques used to interpret machine learning models. The study discusses several XAI methods including SHAP, LIME, and feature attribution techniques that help understand how complex models generate predictions. The survey highlights the importance of explainability in building trustworthy AI systems. However, the work is mainly a theoretical overview and does not focus on implementing real-time cyber threat detection systems for web applications.

From the analysis of these studies, it is evident that existing research mainly focuses on explainability in AI systems and recommendation models, while only limited work addresses real-time cyber threat detection in e-commerce environments. Furthermore, many existing solutions either lack real-time monitoring capabilities or do not integrate explainability with practical security detection mechanisms. Therefore, the proposed system aims to develop an enhanced Explainable AI-based threat detection framework that combines rule-based detection, machine learning classification, and SHAP-based explanations to improve transparency and cybersecurity monitoring in e-commerce websites.

III. Proposed System and Methodologies

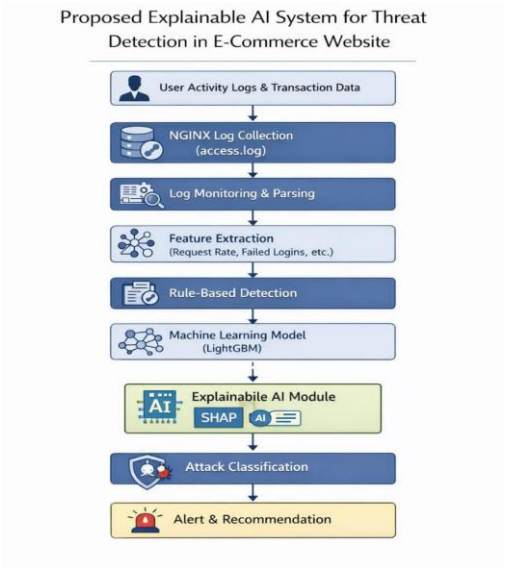


Figure1: Architecture of the proposed system

The proposed system focuses on real-time web attack detection and analysis using a hybrid approach that combines machine learning with Explainable Artificial Intelligence (XAI). The system continuously monitors web traffic from an e-commerce application deployed using Nginx and detects malicious activities such as Distributed

Denial of Service (DDoS), brute-force login attempts, and abnormal user behaviour.

The methodology consists of several stages including log data collection, preprocessing, feature extraction, attack prediction using a machine learning model, classification, and explainable AI analysis. The system not only detects attacks but also explains the reasoning behind each detection using interpretable techniques.

The overall workflow enables the system to analyse both traffic behaviour patterns and feature-level contributions influencing the prediction.

3.1 Dataset Preparation

To train and evaluate the proposed cyberattack detection system, a custom dataset was developed using web server access logs collected from an e-commerce application deployed on NGINX server. The dataset includes both normal user activity and simulated attack traffic, enabling the system to learn real-world behaviour patterns.
Dataset Source:
Google Drive Repository:
https://drive.google.com/drive/folders/1fecrl3XeRNA2RthcRayHgwpOuqKSd9h4?usp=sharing

Dataset Description
The dataset consists of structured log entries obtained from server access logs. Each log record represents a single HTTP request and contains essential information such as client IP address, timestamp, request method, requested resource, and response status code.

The dataset captures both legitimate user interactions and abnormal activities generated during simulated attack scenarios, ensuring diversity in traffic patterns.

Dataset Categories
The dataset is divided into two primary classes:

- **Normal Traffic**
  Represents genuine user activities such as browsing products, accessing pages, and performing standard interactions within the application.
- **Attack Traffic**
  Represents malicious behaviours, including:
  - High-frequency requests indicating Distributed Denial of Service (DDoS) attacks
  - Repeated login attempts indicating brute force attacks
  - Automated request patterns generated by bots

Data Representation
Each log entry is converted into a structured format suitable for machine learning processing. The dataset can be represented as:
$D = \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$
where:

- X_i represents the feature vector extracted from the i^{th} log entry
- $y_i$represents the corresponding label (Normal or Attack)
- n represents the total number of samples

Data Splitting
The dataset is divided into three subsets to ensure proper model training and evaluation:

- **Training Set** – used to train the model
- **Validation Set** – used for tuning hyperparameters
- **Testing Set** – used to evaluate final performance

Mathematically:
$D = D_{train} \cup D_{val} \cup D_{test}$
where all subsets are mutually exclusive.

Data Labelling
Each log record is labelled based on traffic behaviour:

$y_i = 0$, if normal traffic
1, if attack traffic

Labelling is performed using predefined conditions such as abnormal request frequency, repeated failed logins, and unusual request patterns.

Significance of the Dataset
The constructed dataset plays a critical role in enabling the proposed system to learn patterns of both normal and malicious traffic. It provides a realistic environment for training the model and ensures accurate detection of cyberattacks in real-time scenarios.

3.2 Log Preprocessing

The preprocessing stage converts raw log data into structured format suitable for analysis.

Log Parsing
Log entries are parsed using regular expressions:
$L_i = (IP_i, T_i, M_i, U_i, S_i)$

where:

- IP_i=IP address
- T_i =timestamp
- M_i =HTTP method
- U_i =requested URL
- S_i =status code

Data Cleaning

- Removal of incomplete entries
- Standardization of timestamp format

Session Formation
Requests are grouped based on IP address within a time window:
$S(IP) = \{L_1, L_2, \ldots, L_n\}$

3.3 Feature Extraction

Feature extraction converts log data into numerical representations for model input.
The feature vector is defined as:

$X = [x_1, x_2, x_3, x_4, x_5, x_6]$

where:

- x_1 = Request Rate
- x_2 = Failed Login Attempts
- x_3 = Login Indicator
- x_4 = Registration Indicator
- x_5 = HTTP Status Code
- x_6 = POST Method Indicator

Request Rate Calculation

$x_1 = \frac{N_r}{\Delta t}$

where:

- N_r = number of requests
- Δt = time window (seconds)

Failed Login Count

$x_2 = \sum_{i=1}^{n} II(\text{failed login})$

These features help capture abnormal behaviour patterns.

3.4 Attack Detection using LightGBM

The system uses LightGBM for prediction.
LightGBM is a gradient boosting framework that builds decision trees sequentially.
Model Representation
$F(x) = \sum_{k=1}^{K} f_k(x)$

where:

- f_k= decision tree
- K= number of trees

Prediction Probability
$P(y=1|x) = \frac{1}{1+e^{-F(x)}}$

3.5 Attack Classification

Based on prediction and feature thresholds, attacks are classified into:
DDoS Attack
$x_1 > \vartheta_1$
High request rate within short duration
Brute Force Attack
$x_2 > \vartheta_2$
High number of failed login attempts
Fake Account Creation
$x_4 > \vartheta_3$
High registration frequency
Normal Traffic
$P(y=1|x) < \theta$
Decision Rule

3.6 Real-Time Monitoring

The system continuously monitors logs in real-time.
$L_t \rightarrow X_t \rightarrow F(X_t) \rightarrow C_t$
where:

- L_t = log entry
- X_t = feature vector
- $F(X_t)$ = prediction
- C_t = classification

Process Flow

1. Read new log entry
2. Extract features
3. Predict using model
4. Classify attack

This enables live detection of malicious activities.

3.7 Explainable AI Using SHAP

To improve transparency and interpretability, the proposed cybersecurity framework integrates Explainable Artificial Intelligence (XAI) techniques. These techniques provide insights into why a particular request is classified as an attack. The framework uses SHAP (SHapley Additive Explanations) for feature-level interpretation.

I.SHAP (SHapley Additive Explanations)
SHAP is a feature attribution method based on Shapley values derived from cooperative game theory. It explains the contribution of each feature toward the final prediction made by the machine learning model.
The SHAP explanation model can be mathematically represented as:
$f(x) = \emptyset_0 + \sum_{i=1}^{n} \emptyset_i x_i$

where:

- f(x) = model output (prediction score)
- $\emptyset_0$ = base value (average prediction of the model)
- $\emptyset_i$ = contribution of feature i
- x_i = value of feature i
- n = total number of input features

Working of SHAP in the Proposed System

In the proposed attack detection system, SHAP is applied to analyse how input features derived from web traffic logs influence the classification result.

The system extracts features such as:

- Request rate (number of requests per second)
- HTTP request method (GET/POST)
- URL access patterns
- Failed login attempts
- Registration attempts
- Response status codes

Each feature is assigned a SHAP value indicating its impact on the prediction.
Interpretation of SHAP Values

- Positive SHAP value ($\emptyset_i > 0$)
  → Increases the probability of an attack
- Negative SHAP value ($\emptyset_i < 0$)
  → Decreases the probability of an attack
- Magnitude of SHAP value
  → Represents the strength of influence of that feature

Attack Explanation Generated by SHAP
For every detected attack, the system provides a detailed explanation based on SHAP values. The explanation highlights the most influential features responsible for the detection.

Example explanation:

- High request rate indicates abnormal traffic behaviour
- Frequent POST requests suggest automated attack scripts
- Multiple failed login attempts indicate brute force attack
- Rapid repeated access to same URL shows possible DDoS activity

Advantages of SHAP in the Proposed System

- **Feature Contribution Analysis**
  Identifies which features most strongly influence the attack detection
- **Local Interpretability**
  Explains individual predictions for each request
- **Global Interpretability**
  Provides overall understanding of model behaviour
- **Improved Transparency**
  Helps users understand why an IP is flagged as malicious
- **Trustworthy Decision Making**
  Enhances confidence in the AI-based security system

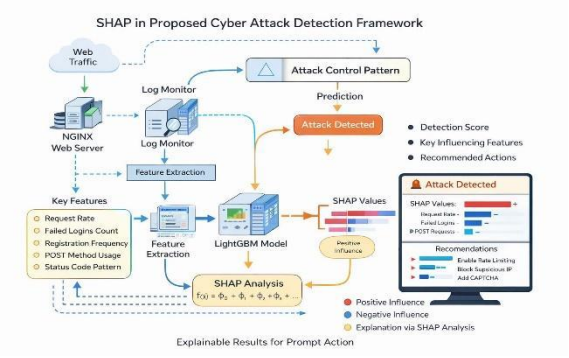
Integration with Real-Time Monitoring
In the proposed framework, SHAP is integrated with the real-time log monitoring system. Whenever an attack is detected:

1. Feature vector is generated from incoming log data
2. The trained model predicts the attack probability
3. SHAP computes feature contributions
4. The system displays:
   - Attack type
   - Detection score
   - Key influencing features
   - Recommended mitigation actions

Outcome of Explainable AI Integration

By incorporating SHAP into the detection pipeline, the system not only identifies cyberattacks but also explains the reasoning behind each decision. This enables:

- Faster incident analysis
- Better defensive strategies
- Reduced false positives
- Enhanced usability for system administrators



3.8 Alert Generation and Visualization

When an attack is detected, alerts are generated.

Output includes:

- Attack type
- Source IP
- Time of detection
- Prediction score

Traffic Pattern Explanation

Traffic=$\frac{Requests}{Time}$

Example:

"35 requests detected in 10 seconds from a single IP"

Explanation Section

Provides:

- Feature-based reasoning
- SHAP-based contribution analysis

Recommendation Section

- Enable rate limiting
- Block IP
- Add CAPTCHA

3.9 System Workflow

Logs ⟹ Preprocessing ⟹ Feature Extraction ⟹ Model ⟹ Classification ⟹ SHAP ⟹ Alert

The proposed system provides several advantages:

- Real-time detection
- High accuracy using LightGBM
- Explainable predictions using SHAP
- Multi-attack detection capability
- Easy deployment with web servers

Final Outcome

The proposed system provides:

- Efficient real-time attack detection
- Transparent and explainable results
- Improved web application security
- Interactive monitoring dashboard

IV. Results and Discussion

The proposed system was evaluated using real-time web traffic generated from an e-commerce application deployed using Nginx. Both normal user behaviour and simulated attack scenarios such as DDoS, brute-force login attempts, and fake account creation were tested.

The system continuously monitored server logs, extracted relevant features, and predicted the probability of attack for each incoming request. Based on these predictions, the system classified traffic into different attack categories.

Example Results:

Traffic Type	Score	Prediction
Normal User Activity	0.82	Normal
DDoS Attack	0.18	Attack Detected
Brute Force Login Attack	0.25	Attack Detected

Table 1:Demonstration of the output

The results demonstrate that the system is capable of accurately distinguishing between normal and malicious traffic patterns.

Observations:

- The system successfully detected high request rate patterns associated with DDoS attacks.
- Multiple failed login attempts were correctly classified as brute-force attacks.
- Abnormal registration frequency was identified as fake account creation attacks.
- The model provided real-time predictions, enabling immediate detection.

Explainability Analysis:

The integration of SHAP allowed the system to explain predictions by identifying the most influential features:

- High request rate significantly increased attack probability
- Failed login attempts contributed strongly to brute-force detection
- POST request patterns indicated automated behaviour

The system also provided detailed explanations, such as:

"25 requests detected within 10 seconds from a single IP, indicating abnormal traffic behaviour."

These explanations improve system transparency and help administrators understand the reason behind each alert.

V. Conclusion

This research presented an explainable machine learning framework for detecting cyberattacks in web applications. The proposed system integrates real-time log monitoring with a classification model based on LightGBM to identify malicious traffic patterns.

The system extracts behavioural features such as request rate, failed login attempts, and access patterns from server logs and uses them to classify different types of attacks including DDoS, brute-force, and fake account creation. By combining rule-based logic with machine learning, the system achieves accurate and efficient detection.

To enhance transparency, Explainable Artificial Intelligence techniques such as SHAP were incorporated to provide feature-level explanations for each prediction. The system highlights the key factors influencing attack detection and presents them in an interpretable manner.

Experimental results demonstrate that the proposed framework can effectively distinguish between normal and malicious traffic while providing meaningful explanations. The system also supports real-time monitoring and alert generation, making it suitable for practical deployment in web applications.

VI. Future Enhancements

Although the proposed system achieves effective real-time attack detection, several enhancements can further improve its performance and scalability.

1. Advanced Behavioural Analysis

Future work can include deeper user behaviour analysis by tracking session patterns over longer durations to detect stealthy or slow-rate attacks.

2. Integration of Deep Learning Models

The system can be extended using deep learning techniques such as recurrent neural networks (RNNs) or transformers to capture complex temporal patterns in web traffic.

3. Multivariate Feature Expansion

Additional features such as:

- User agent analysis
- Geo-location tracking
- Device fingerprinting

can be incorporated to improve detection accuracy.

4. Automated Response System

Future versions can implement automatic mitigation techniques such as:

- Dynamic IP blocking
- Intelligent rate limiting
- Real-time firewall rule updates

5. Deployment in Distributed Environments

The system can be scaled to monitor multiple servers simultaneously using cloud-based architectures.

6. Real-Time Dashboard Enhancement

Advanced visualization dashboards with:

- Graph-based traffic monitoring
- Historical attack analysis
- Predictive alerts

can improve usability and decision-making.

References

[1] Sarkar, A., et al., "Explainable AI in E-Commerce: Enhancing Trust and Transparency in AI-Driven Decisions," 2025.

[2] Villegas-Ch, W., et al., "Integrating Explainable Artificial Intelligence in Anomaly Detection for Threat Management in E-Commerce Platforms," 2024.

[3] Benfarhat, M., et al., "Advanced Temporal Convolutional Network Framework for Intrusion Detection in Electric Vehicle Charging Stations," 2023.

[4] Sreenivasa, K., et al., "XAI-Based E-Commerce Recommender System," 2025.

[5] Dunyamaliyev, I., "Explainable Recommender Systems in E-Commerce," 2025.

[6] Adadi, A., and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.