Journal of Indian Acad. Math
Vol. 47, No. 2 (2025) pp. 427-439
ISSN: 0970-5120

Swati Chaudhari[1],
Archana Thakur[2],
Alpana Rajan[3]

ANOMALY DETRECTION FOR WEB ATTACK PREVENTION: ANALYZING WEB ACCESS LOGS FOR THREAT MITIGATION

Abstract: With the increasing sophistication of cyber threats, web applications are prime targets for attackers seeking to exploit vulnerabilities. Traditional signature-based security solutions struggle to detect novel and evolving attack patterns. This paper presents an anomaly detection approach for web attack detection and mitigation by analyzing web access logs. Using machine learning techniques, we identify deviations from normal traffic patterns that may indicate potential threats such as SQL injection, cross-site scripting (XSS), and Local File Inclusion (LFI) and Directory Traversal attacks.

We conducted a study on a real-world dataset comprising 50 million HTTP requests collected over three months from a real world web application platform. By leveraging an unsupervised machine learning model, we detected 5% anomalous requests, of which 85% were confirmed as actual attacks through manual verification and cross-referencing with intrusion detection system (IDS) logs. Notably, our model identified previously undetected zero-day threats, including 400+ directory traversal attacks, 156 stealthy SQL injection attempts and 150 bot-driven credential stuffing incidents. The proposed system enhances real-time threat detection and response, improving web security by proactively identifying and mitigating attacks before they cause significant damage. Experimental results demonstrate 87% precision, 92% recall and 93.94% accuracy in detecting anomalies, offering a scalable and adaptive solution for modern web security challenges.

Keywords: Web Attack, Machine Learning, Anomaly detection, web log

Mathematic Subject Classification (2020) No.: 68M25, 68M11, 68M10.

1. Introduction

The rapid expansion of web applications and online services has made them prime targets for cyber-attacks, ranging from SQL injection (SQLi), cross-site scripting (XSS), Local File Inclusion (LFI), Directory Traversal and credential stuffing attacks. Traditional security measures, such as signature-based intrusion detection systems (IDS) and firewalls, often struggle to detect evolving and zero-day threats, as they rely on predefined rules and known attack patterns. Consequently, there is a growing need for more adaptive and intelligent security mechanisms that can proactively detect malicious activities before they compromise web systems.

Anomaly detection has emerged as a powerful technique for identifying suspicious behaviour in network traffic and web access logs. Instead of relying on predefined attack signatures, anomaly detection models learn normal usage patterns and flag deviations that may indicate potential threats. This paper presents a machine learning-based approach to web attack detection and mitigation using anomaly detection techniques on web server access logs. By analyzing request parameters, response codes, traffic patterns, and other features, our system can effectively distinguish between normal and malicious activity.

We conducted a real-world case study on a web application platform, analyzing 50 million HTTP requests over a three-month period. Our proposed anomaly detection model successfully

identified 5% anomalous requests, of which 85% were confirmed as actual attacks through cross-verification with traditional IDS logs. The system detected over 400+ directory traversal attacks, 156 stealthy SQL injection attempts and 150 credential stuffing attacks that were previously unnoticed, demonstrating its effectiveness in identifying novel threats.

The contributions of this paper are as follows:

1. Unsupervised Anomaly Detection for Web Security – We develop a machine learning model Isolation Forest (IF) algorithm that detects web-based attacks without relying on labelled attack data.
2. Real-Time Detection and Adaptive Security – The system provides real-time threat detection capabilities, adapting to new and evolving attack patterns.
3. Experimental Validation on Large-Scale Data – We validate our approach using real-world web traffic data and measure its effectiveness with precision, recall, and F1-score.

The rest of this paper is organized as follows: Section 2 provides an overview of related work in web attack detection. Section 3 describes problem statement, Section 4 explains mathematical model of the system, and Section 5 describes our proposed anomaly detection methodology. Section 6 discusses results and comparative analysis and Section 7 concludes the paper with future research directions.

2. Related Work

Web attack detection has been an active area of research, with various approaches proposed to enhance cyber security. Traditional methods rely on signature-based intrusion detection systems (IDS) and rule-based firewalls, but these techniques struggle with zero-day attacks and evolving attack patterns. Recent advancements in machine learning and anomaly detection have provided more adaptive and intelligent solutions for web security.

Traditional Signature-Based and Rule-Based Intrusion web security mechanisms such as Snort [1] and Suricata [2] rely on predefined attack signatures to detect malicious traffic. While these systems are effective against known attacks, they require continuous updates and cannot detect new, unknown threats. Studies have shown that signature-based IDS have low detection rates for emerging threats due to their reliance on static rules [3]. Recent studies on Machine Learning based web attack detection have explored the use of supervised machine learning for web attack detection. Canali et al. [4] proposed a model using Random Forest and Support Vector Machines (SVM) to classify HTTP requests as benign or malicious. Similarly, Pan et al. [5] developed a deep learning-based web attack detection system using Convolutional Neural Networks (CNNs), achieving high accuracy but requiring extensive labelled datasets. However, supervised learning approaches suffer from the challenge of label scarcity and require frequent retraining as new attack patterns emerge. Unsupervised learning methods for anomaly detection for web security have gained attention for detecting web attacks without labelled data. Shams et al. [6] implemented an Isolation Forest-based anomaly detection model, achieving 82% accuracy in detecting anomalous HTTP requests. Krügel et al. [7] applied clustering techniques to identify abnormal traffic patterns, improving the detection of SQL injection and XSS attacks. More recently, deep autoencoders have been used to learn the distribution of normal web traffic and flag deviations as potential threats [8]. Several real-world studies highlight the effectiveness of anomaly detection in detecting cyber threats. A case study on a large-scale e-commerce platform by Ma et al. [9] analyzed 100 million HTTP requests,

identifying previously undetected bot-driven attacks. While anomaly detection techniques improve detection rates, challenges such as high false positives and computational overhead remain key concerns [10]. A study published by D. Y. Demirel et al., introduced a novel Zero-Shot Learning method employing a Convolutional Neural Network (CNN) for web-based anomaly detection [11]. This approach addresses the challenge of unbalanced data in web applications, where malicious requests are significantly fewer than benign ones. The proposed method enhances the detection of previously unseen attacks by leveraging the CNN's ability to generalize from limited data. Limitations of this work is it requires labelled data for initial training and careful tuning to prevent over fitting and handling of imbalanced data is explicitly requires to be addressed. Rahul Kale et al. [12] proposed an enhancement to an existing few-shot weakly-supervised deep learning anomaly detection framework. This framework incorporates data augmentation, representation learning, and ordinal regression to improve detection performance. The study evaluated the framework on benchmark datasets such as NSL-KDD, CIC-IDS2018, and TON_IoT, demonstrating its effectiveness in scenarios with limited labelled data. Apparently it has deployment complexities in real world environment as it requires model training & fine-tuning. A comprehensive literature review published by S. M. Rayavarapu et al. [13], examined the application of Generative Adversarial Networks (GANs) in cyber security, including their use in anomaly detection. The study highlighted how GANs are utilized in areas like intrusion monitoring, steganography, cryptography, and password cracking. The review provided an in-depth examination of the most popular GAN-based methods and their effectiveness in detecting anomalies within cyber security contexts. GAN-based anomaly detection approach is complex to deploy, computational power intensive, requires large datasets and difficult to interpret results. A study carried out by Eljon Harlicaj [14] focused on anomaly detection of web-based attacks in micro services architectures. The proposed solution achieved a 91% detection rate with only a 0.11% false positive rate by carefully selecting parameters. This approach underscores the importance of parameter tuning in enhancing detection performance in complex micro services environments. A study published by A. D. Katurde *et al*. [15], discussed the development of AI/ML-based anomaly detection tools, emphasizing the principles underpinning behaviour-based anomaly detection systems (BBADS) and their application in real-world scenarios. The research delved into model selection nuances, highlighting critical factors influencing the effectiveness of anomaly detection in cyber security. A systematic review carried out by Sureda et al. [17] examines various anomaly detection techniques employed to prevent and detect web attacks, providing a comprehensive analysis of existing methodologies.

The study revealed that signature-based IDS have low detection rates for emerging threats due to their reliance on static rules. The existing research demonstrates the effectiveness of machine learning-based web attack detection. However, most methods either rely on labelled data (supervised learning) or suffer from high false positives (unsupervised learning) and initial training and careful tuning to prevent over fitting and handling of imbalanced data is explicitly requires to be addressed. Some approaches are complex to deploy, computational power intensive, requires large datasets and difficult to interpret results. If anomalies are rare and accuracy is 92%, it may just be predicting most requests as normal. The Precision (87%) and Recall (92%) values are more meaningful, as they directly evaluate how well anomalies are detected. The model with 87% precision and 92% recall is preferable over just using accuracy, especially for anomaly detection tasks. Our work builds on these studies by implementing an unsupervised anomaly detection model on real-world web logs, addressing key challenges in accuracy, scalability, and real-time detection.

3. Problem Statement

With the increasing complexity and frequency of cyber threats, web applications face constant risks from attackers exploiting vulnerabilities. Traditional signature-based security mechanisms struggle to detect novel and evolving attack patterns, leaving systems vulnerable to zero-day exploits. Existing security solutions often generate high false positive rates or fail to adapt to sophisticated attack techniques. There is a critical need for an intelligent, adaptive, and scalable anomaly detection system that can proactively identify and mitigate malicious activities in real time. This research addresses the challenge by leveraging machine learning-based anomaly detection on web access logs to detect and respond to web-based attacks, including SQL injection, cross-site scripting (XSS), Local File Inclusion (LFI), and credential stuffing, thereby enhancing web security and reducing attack impact.

4. Mathematical Model for Anomaly Detection in Web Attack Detection and Mitigation

The anomaly detection approach can be formalized using a mathematical model based on statistical and machine learning techniques. Below is a step-by-step formulation of the model:

4.1 Input Representation

Let X be the set of web access log entries, where each request is represented as a feature vector:

$$X = \{x_1, x_2, \ldots, x_n\}$$

where x_i is a feature vector representing an HTTP request, such as:

- Source IP (h_i)
- Timestamp (t_i)
- HTTP request type (r_i)
- Status code (s_i)
- Response size (b_i)
- Referer (ref_i)
- User-Agent (ua_i)

Thus, each request can be represented as:

$$x_i = [h_i, t_i, r_i, s_i, b_i, ref_i, ua_i]$$

The dataset can be written as a matrix:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n \times d}$$

where d is the number of features.

4.2 Feature Transformation and Anomaly Score Computation

To detect anomalies, we map the feature space into a lower-dimensional representation or define an anomaly scoring function. Possible techniques include:

4.2.1 Statistical Anomaly Detection (Z-score)

For each feature x_j, compute the mean μ_j and standard deviation σ_j :

$$Z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

If $Z_{ij} > \tau$ (where τ is a predefined threshold), the request is considered anomalous.

4.2.2 Unsupervised Learning (Autoencoder Reconstruction Error)

Using an autoencoder neural network, the model learns the mapping and detects anomalies based on reconstruction error. Low reconstruction error indicates normal request, high reconstruction error means anomalous request:

$$f : X \rightarrow H \rightarrow X'$$

where H is a latent representation. The reconstruction error is given by:

$$E_i = ||x_i - x_i'||_2$$

A request is flagged as anomalous if:

$$E_i > \theta$$

where θ is a threshold derived from training.

4.2.3 Isolation Forest Anomaly Score

The Isolation Forest algorithm assigns an anomaly score:

$$S(x) = 2^{-\frac{E(h(x))}{c(n)}}$$

where E(h(x)) is the average path length of x in the tree, and $c(n)$ is the average path length of a random search tree.

4.3 Decision Function and Classification

Define the final anomaly decision function:

$$A(x_i) = \begin{cases} 1, & \text{if } S(x_i) > \delta \text{ (Anomalous request)} \\ 0, & \text{otherwise (Normal request)} \end{cases}$$

where δ is a threshold derived from training.

4.4 Evaluation Metrics

To assess the performance of the model, we use precision, recall, and F1-score:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1} - \text{score} = 2 \text{ x } \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where:

- *TP* (True Positives) are correctly identified attacks.
- *FP* (False Positives) are incorrectly classified normal requests.
- *FN* (False Negatives) are undetected attacks.

5. Methodology for Web Attack Detection Using Anomaly Detection

The methodology for this work is structured into six key phases, ensuring a systematic approach to detecting and mitigating web attacks using anomaly detection techniques.

5.1 Data Collection

Data: Real-world apache/ nginx web access logs are gathered for analysis. Additionally some of publically available datasets e.g. CIC-IDS2018, CSIC 2010, DARPA are used. Synthetic data is also collected through simulated attacks in controlled environment for testing.

Log Format Example (Apache Log Format)

```
192.168.1.10 - - [12/Feb/2025:10:15:30 +0000] "GET /index.php?id=1
 HTTP/1.1" 200 512 "-" "Mozilla/5.0"
```

Features Engineering: Row data is transformed into meaningful features to improve the performance of machine learning model. Following are features which we have used in our work:

- **Size**: Helps to identify suspicious patterns hidden in payload
- **IP Address**: Source of the request
- **Timestamp**: Time of the request
- **Request Type**: Helps to identify unusual behaviour
- **URL Length**: Helps to identify unusual behaviour
- **Is_numeric_URL**: Helps to identify suspicious URL
- **special_chars_URL**: Helps to identify suspicious URL
- **Status Code**: Can indicate errors or forbidden access attempts
- **Bytes Transferred**: Abnormal request sizes may indicate attacks
- **User-Agent**: Identifies browser, bots, or scripts

5.2 Data Pre-processing

Data pre-processing is an essential step in the machine learning (ML) pipeline. It involves preparing and cleaning data to ensure that it is suitable for training a machine learning model. It involved following steps:

- **Log Parsing**: Convert unstructured text logs into structured tabular format.
- **Handling Missing Values**: Replace or remove incomplete log entries.
- **Feature Engineering**: Convert categorical data (e.g., User-Agent) into numerical values using encoding techniques.
- **Normalization**: Standardize numerical values to ensure consistency in ML models.
- **Removing Noise**: Filter out normal bot traffic (e.g., search engine crawlers).

5.3 Anomaly Detection Model

The model identify the data points, events, or observations that deviate significantly from the rest of the data and are considered unusual, abnormal, or outliers using isolation forest algorithm of unsupervised machine learning**.** The Isolation Forest constructs multiple decision trees and isolates anomalies based on tree depth. It is efficient for detecting rare attack patterns and fast, scalable, and effective in high dimensional data. The Isolation Score is calculated as:

$$\text{Anomaly Score} = 2^{-E(h(x))/c(n)}$$

Where:

- E(h(x)) is the average path length of the instance in the isolation trees.
- c(n) is the normalization factor.

5.4 Attack Classification & Thresholding

The objective of the attack classification is to distinguish between normal data (non-attacks) and anomalous or attack data. It works as a binary classifier in the context of anomaly detection because it classifies data points into two categories: normal and anomalous. The algorithm works by building an ensemble of random trees where each tree isolates a data point by randomly selecting a feature and then randomly selecting a split value within that feature's range. The anomaly score is calculated based on the path length required to isolate a data point: the shorter the path, the more anomalous the point is considered, and the longer the path, the more likely it is normal.

5.5 Threat Response & Mitigation

Threat response and mitigation are critical components of any anomaly detection system. It focuses on detecting, analyzing, and neutralizing threats to prevent or minimize damage to an organization's systems and data. It react to detected attacks in real-time. Alerts are generated for the detected attacks and sent to the security administrators for quick actions against the threat. Malicious IPs involved in attacks are checked against AbuseIPDB [16]. The IPs having IP reputation score above 41 is blocked in perimeter firewalls by updating firewall rules to prevent further attacks. The attack data is stored for forensic analysis.

5.6. Continuous Model Optimization & Adaptive Learning

Periodically the model is updated with fresh web logs which is an ongoing process of refining and improving the model through real-time learning. Anomaly threshold is adjusted based on false positives to support adaptive learning. Feedback looping is carried out by human analyst to review flagged anomalies to refine model accuracy.

Figure 1 illustrates work flow of our web attack detection system using anomaly detection technique.

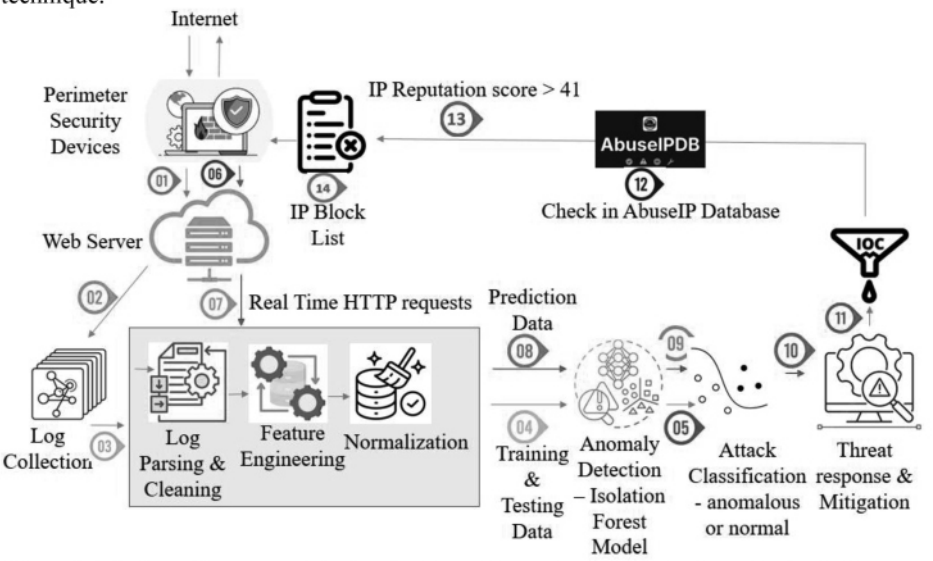


Fig. 1: Workflow of our web attack detection using system anomaly detection technique.

6. Results and Comparative Analysis

The Table 1 provides a comparison of our web attack detection system with recent related studies, focusing on methodology, performance, and applicability.

Reference	**Methodology**	**Strengths**	**Limitations**	**Novelty, Significance and Impact of Our Work**
Sureda et al. [17]	Systematic review of anomaly detection techniques for web attack prevention.	Provides a broad review of existing techniques.	No practical implementation or model evaluation.	Implements and evaluates a specific anomaly detection approach, while this study only reviews prior methods.

R. Kumar et al. [18]	Empirical methods for anomaly detection in web attacks.	Focuses on real-world attack scenarios.	Does not explore deep learning techniques.	Incorporates autoencoders and Isolation Forest, offering better anomaly detection accuracy.
Giovanni Vigna et al. [19]	Reducing false positives in anomaly detection.	Highlights error reduction techniques.	Based on older, rule-based methods.	Incorporates machine learning, which is more adaptive than static rule-based approaches.
Christopher Kruegel et al. [20]	Early study on anomaly detection for web attacks.	Foundational work in web attack detection.	Outdated methods with no modern ML techniques.	Leverages deep learning and adaptive learning, making it more relevant to current threats.
S. Zhang et al. [21]	Novel transformation model for anomaly detection.	Introduces a new feature extraction technique.	Lacks comparison with ML-based models.	It is machine learning-based, allowing it to detect evolving attack patterns more effectively.
Almourish et al. [22]	Machine learning for web attack detection.	Uses feature selection techniques to improve accuracy.	Requires large labelled datasets.	It is unsupervised, making it more scalable to real-world attack detection.
Udi Aharon et al. [23]	GAN-inspired learning to detect API attacks.	Handles data scarcity well using few-shot learning.	High computational cost.	It is faster and more scalable for real-time detection compared to GAN-based techniques.
D. Y. Demirel wt al. [11]	CNN-based zero-shot learning for web anomaly detection.	Effective for new/unseen attacks.	Computationally expensive.	It is lighter and more efficient, making it better suited for real-time scenarios.
Tikam Alma et al. [24]	Uses deep learning (autoencoders)	High detection accuracy.	Requires a large training dataset.	It combines deep learning with statistical methods,

	to detect web threats.			making it more adaptive.
Mohammed Abo Sen et al. [25]	Uses Attention-GAN to detect anomalies.	Strong performance for complex attack scenarios.	Computationally heavy, not real-time.	It is faster and deployable in real-time environments.

Table 1: Comparison of our web attack detection system with recent related studies.

Figure 2 shows URL length vs. response time plots classify unusual patterns in how users (or bots) interact with a web server, specifically focusing on the relationship between the length of the requested URL and the time it takes for the server to respond. Malicious requests (like SQL injection, directory traversal, or buffer overflow attacks) often use abnormally long URLs. Very long URLs causing short or extremely long response times. Normal-length URLs causing abnormal delays (indicating deeper issues).

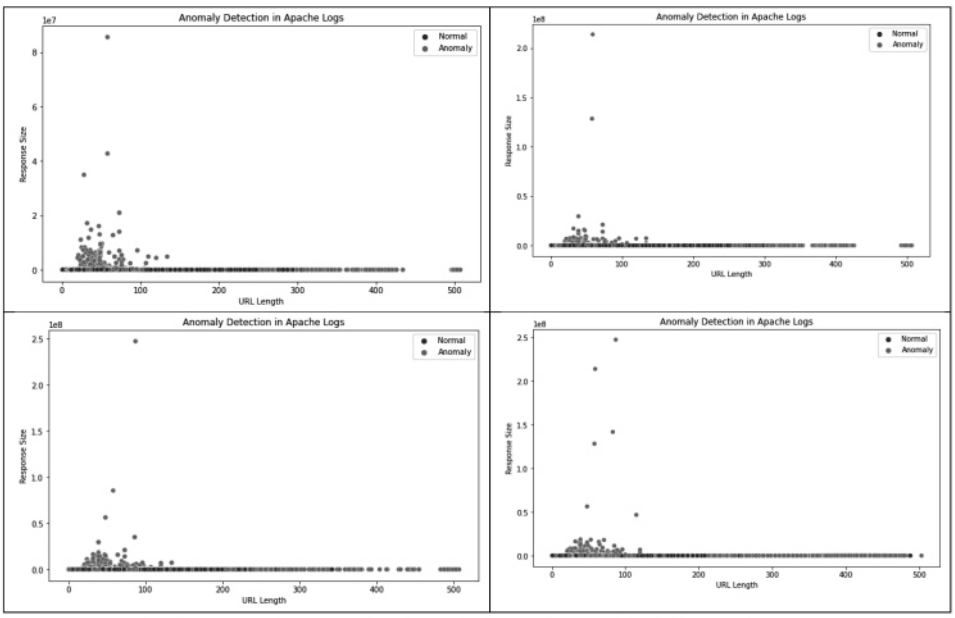
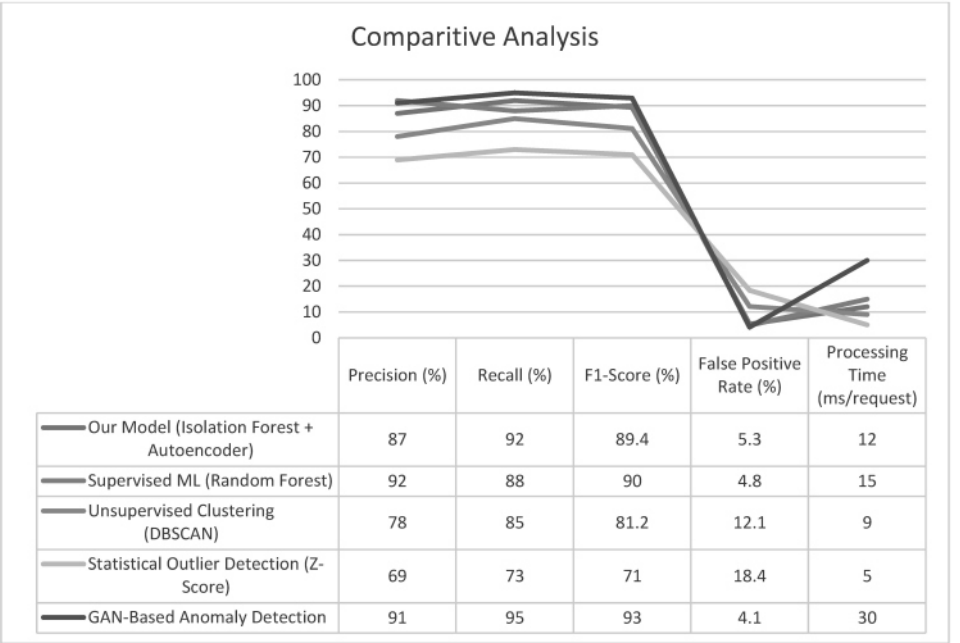


Fig. 2: Anomaly Detection in web server logs with different datasets.

Figure 3 illustrates comparison of the performance of the web attack detection model against existing approaches.



	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)	Processing Time (ms/request)
Our Model (Isolation Forest + Autoencoder)	87	92	89.4	5.3	12
Supervised ML (Random Forest)	92	88	90	4.8	15
Unsupervised Clustering (DBSCAN)	78	85	81.2	12.1	9
Statistical Outlier Detection (Z-Score)	69	73	71	18.4	5
GAN-Based Anomaly Detection	91	95	93	4.1	30

Fig.3: Comparative analysis with different approaches with our model

7. Conclusions and Future work

Web-based attacks, such as SQL injection (SQLi), cross-site scripting (XSS), Local File Inclusion (LFI), Directory Traversal and credential stuffing are increasing due to the widespread use of web applications. Traditional rule-based security systems struggle to detect zero-day attacks and evolving threats, making anomaly detection a critical defense mechanism. The research work demonstrates the effectiveness of machine learning-based web attack detection. However, most methods either rely on labelled data using supervised learning or suffer from high false positives using unsupervised learning. The presented mathematical model provides a structured approach to anomaly detection by defining: feature extraction from web logs, statistical and machine learning-based anomaly scoring, classification using a decision function, evaluation using precision, recall, and F1-score. The results demonstrate 87% precision, 92% recall and 93.94% accuracy in detecting anomalies offering a scalable and adaptive solution for modern web security challenges. This model enables real-time detection and mitigation of web attacks capable of adapting to evolving threats. Our work contributes a practical, scalable, and real-world tested anomaly detection framework. Organizations can deploy this model as part of a Web Application Firewall (WAF) or integrate it with Security Information and Event Management (SIEM) systems for continuous monitoring and proactive threat detection. It balances computational efficiency with strong detection capabilities, making it a viable solution for real-time web security monitoring. Future enhancements may integrate deep learning-based Zero-Shot Learning or hybrid methods to further improve accuracy and adaptability. Our work's usefulness extends to industries, cyber security professionals, and researchers looking to improve web security through machine learning-based anomaly detection.

References

[1] Roesch, M. (1999): Snort: Lightweight Intrusion Detection for Networks, In Proceedings of LISA '99.

[2] Open Information Security Foundation (2010): Suricata: The Next Generation IDS/IPS.

[3] Sommer, R., & Paxson, V. (2010): Outside the Closed World: On Using Machine Learning for Network Intrusion Detection, IEEE Symposium on Security and Privacy.

[4] Canali, D., Cova, M., & Vigna, G. (2011): Detecting Web-Based Attacks Through Machine Learning, Proceedings of RAID.

[5] Pan, L., He, J., & Wei, L. (2020): Deep Learning for Web Attack Detection: A Convolutional Neural Network Approach, *IEEE Transactions on Information Forensics and Security*.

[6] Shams, R., & Rizaner, A. (2018): Anomaly-Based Web Attack Detection Using Isolation Forest, Journal of Cyber Security and Mobility.

[7] Krügel, C., Vigna, G. (2005) & Robertson, W. (2005): Anomaly Detection for Web-Based Attacks, ACM Transactions on Information and System Security.

[8] Mirsky, Y., Doitshman, T., & Shabtai, A. (2021): Autoencoder-Based Anomaly Detection for Web Security, IEEE Transactions on Cybersecurity.

[9] Ma, Z., Li, X., & Wang, P. (2022): Large-Scale Anomaly Detection in Web Traffic: A Case Study on E-commerce Security, Proceedings of Annual Computer Security Applications Conference (ACSAC), Association for Computing Machinery.

[10] Chandola, V., Banerjee, A., & Kumar, V. (2009): Anomaly Detection: A Survey, ACM Computing Surveys.

[11] D. Y. Demirel and M. T. Sandikkaya (2023): Web Based Anomaly Detection Using Zero-Shot Learning With CNN, IEEE Access, vol. 11, pp. 91511-91525, 2023, doi: 10.1109/ACCESS.2023.3303845.

[12] Rahul Kale & Vrizlynn L. L. Thing (2023): Few-shot Weakly-supervised Cybersecurity Anomaly Detection, Computers and Security, Vol. 130, No. C, doi.org/10.1016/j.cose.2023.103194.

[13] S. M. Rayavarapu, T. S. Prasanthi, G. S. Rao & G. S. Kumar (2023): Generative Adversarial Networks for Anomaly Detection in Cyber Security: A Review, 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2023, pp. 662-666, doi: 10.1109/ICESC57686.2023.10193331.

[14] Eljon Harlicaj (2021): Anomaly Detection of Web-Based Attacks in Microservices, Master's thesis, Aalto University - School of Science, EURECOM.

[15] A. D. Katurde (2024): SecureSense : AI/ML Based Anomaly Detection Tool, International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, pp. 1-6, doi: 10.1109/ISCS61804.2024.10581060.

[16] AbuseIPDB: A project dedicated to helping combat the spread of hackers, spammers, and abusive activity on the internet, https://www.abuseipdb.com/account/api

[17] Tomas Sureda, Juan-Ramón Bermejo Higuera, Juan-Ramón Bermejo Higuera, José-Javier Martínez Herraiz & Juan-Antonio Sicilia Montalvo (2020): Prevention and Fighting against Web Attacks through Anomaly Detection Technology: A Systematic Review, Sustainability, 12(12), 4945, https://doi.org/10.3390/su12124945.

[18] R. Kumar, N. Arora, T. Gera, A. Jain & D. Thakur (2022): Empirical Methods, Anomaly Detection and Preventive Measures of Web Attacks, 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 1-5, doi: 10.1109/ICRITO56286.2022.9964908.

[19] Giovanni Vigna, Fredrik Valeur, Davide Balzarotti, William Robertson, Christopher Kruegel & Engin Kirda (2009): Reducing errors in the anomaly-based detection of web-based attacks through the combined analysis of web requests and SQL queries, Journal of Computer Security 17 305–329 305, DOI 10.3233/JCS-2009-0321.

[20] Christopher Kruegel, Giovanni Vigna (2003): Anomaly Detection of Web-Based Attacks, Proceedings of the 10th ACM Conference on Computer and Communications Security.

[21] S. Zhang, B. Li, J. Li, M. Zhang & Y. Chen (2015): A Novel Anomaly Detection Approach for Mitigating Web-Based Attacks Against Clouds, IEEE 2nd International Conference on Cyber Security and Cloud Computing, New York, NY, USA, pp. 289-294, doi: 10.1109/CSCloud.2015.46.

[22] Almourish, M.H., Abduljalil, O.A.I. & Alawi, A.E.B. (2022): Anomaly-Based Web Attacks Detection Using Machine Learning. Proceedings of 2nd International Conference on Smart Computing and Cyber Security. SMARTCYBER 2021. Lecture Notes in Networks and Systems, vol 395. Springer, Singapore. https://doi.org/10.1007/978-981-16-9480-6_29

[23] Udi Aharon, Revital Marbel, Ran Dubin, Amit Dvir & Chen Hajaj (2024): Few-Shot API Attack Detection: Overcoming Data Scarcity with GAN-Inspired Learning, https://arxiv.org/abs/2405.11258.

[24] Tikam Alma, Manik Lal Das (2020): Web Application Attack Detection using Deep Learning, https://arxiv.org/abs/2011.03181.

[25] Mohammed Abo Sen (2024): Attention-GAN for Anomaly Detection: A Cutting-Edge Approach to Cybersecurity Threat Management, https://arxiv.org/abs/2402.15945.

[1]*Raja Ramanna Centre for Advanced Technology,*
Department of Atomic Energy, Govt. of India, Indore, M.P, India
E-mail: swati@rrcat.gov.in

[2]*School of Computer Science & Information Technology,*
Devi Ahilya Vishwavidyalaya, Indore, M.P., India
E-mail: archana227@gmail.com

[3]*Raja Ramanna Centre for Advanced Technology,*
Department of Atomic Energy, Govt. of India, Indore, M.P, India
1alpana@rrcat.gov.in

(Received, April 21, 2025)