iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.78241

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

AI-Powered Web Application Firewall with a Real-Time Security Operations Center (SOC) Dashboard

Shanmukha Prashanth Palli, Mrs.D. Arpitha Rani
Dept. of Cybersecurity)Malla Reddy University Hyderabad, India

Abstract: ***Web applications have become critical components of modern digital infrastructure, making them frequent targetsof cyberattacks such as SQL injection, cross-site scripting, command injection, and directory traversal. Traditional Web Application Firewalls rely primarily on signature-based rule matchingandoftenfailtodetectobfuscatedorpreviouslyunseen attacks. This paper proposes an AI-enhanced Web Application Firewall that combines signature-based detection with machine learning-basedanomalydetectionforintelligentreal-timerequest inspection. The proposed framework analyzes incoming HTTP requests, performs payload decoding and feature extraction, and classifies suspicious traffic through a hybrid detection pipeline. The system is implemented using Flask, Scikit-learn, LightGBM, and a web-based dashboard for monitoring security events. Experimentalresultsindicatethatthehybridapproachimproves detection accuracy and reduces false positives compared to a standalone rule-based model. The proposed solution offers a scalable and adaptive framework for securing modern web applications.***

Keywords: AI-Powered Web Application Firewall, Web Application Security, SQL Injection Detection, Cross-Site Scripting (XSS) Protection, Machine Learning-Based Anomaly Detection, Hybrid Rule-Based and AI Security Engine, Real-Time Security Operations Center (SOC) Dashboard, HTTP Request Threat Analysis, Automated IP Banning Mechanism, Cyberattack Monitoring and Logging.

I. INTRODUCTION

Web applications have become the foundation of e-commerce, digital governance, financial systems, healthcare platforms, and cloud services. As their adoption increases,they continue to attract sophisticated attacks at the application layer. Common threats include SQL injection (SQLi), cross-site scripting (XSS), command injection, directory traversal, andmaliciouspayloadobfuscation .Theseattackstargetweak-nesses in input validation, session handling, request parsing, and backend execution.

Traditional firewalls and network intrusion detection sys-tems focus primarily on lower network layers and are often insufficient for analyzing HTTP request content. Web Ap-plication Firewalls (WAFs) were introduced to bridge thisgap by inspecting requests before they reach the protected application. However, many WAF solutions rely mainly on static signature rules. Although such systems are effective for known attack patterns, they are less reliable when confronting zero-day threats, encoded payloads, or adversarial variations.

Machinelearninghasemergedasapromisingdirection for modern cybersecurity because it can learn discriminatory patterns from malicious and benign traffic. In this work, a hy-brid framework is proposed in which deterministic rule-based filtering is combined with machine learning-based anomaly detection.Thisdesignimprovesrobustnessbyquicklycatching knownthreatswhilealsoidentifying suspicioustrafficpatterns that do not explicitly match a predefined rule.

Theprimarycontributionsofthispaperareasfollows:

- DesignofahybridAI-enhancedWAFarchitecturefor intelligent web request inspection.
- Integration of payload decoding, rule-based attack detec-tion, and machine learning-based anomaly detection.
- Development of a practical prototype using Python Flask and web monitoring components.
- Presentation of a results section that supports application screenshots and system output analysis.

II. LITERATURE REVIEW

Web application security has attracted significant research attention because HTTP-based systems remain among the most attacked components of modern digital infrastructure.

TraditionalWAFplatformsarelargelybasedonrulematching and signature engines, which remain effective for known exploit strings but often struggle with evolving payload struc-tures and adversarial obfuscation [1]. Early anomaly-based approachesdemonstratedthatmodelingthestructureofHTTP requests can improve attack detection beyond static rule eval-uation [2].

Foundational surveys in intrusion detection categorized se-curity systems into misuse detection and anomaly detection, showingthatanomaly-basedmodelsprovidebroaderdetection coverage but may introduce more false positives if not care-fully tuned [3]. Later taxonomies refined these classifications and emphasized the importance of combining multiple detec-tion strategies for practical deployments [4]. This observation laid the groundwork for hybrid web security models that integrate deterministic rules with adaptive learning.

Machine learning has since become central to intelligent intrusion detection. Studies on malicious web request classifi-cation have shown that supervised learning techniques suchas decision trees, support vector machines, and ensemble methodscanidentifysuspicioustrafficpatternswithimproved accuracy[5].Deeplearningapproacheshavefurtherimproved representationlearningforcomplexpayloadsbyautomatically discovering hidden nonlinear patterns from request data [6]. These methods are especially useful when payloads are ob-fuscated or differ from previously observed signatures.

Another important area of prior work concerns benchmark datasets and realistic traffic evaluation. Analyses of widely used security datasets have highlighted both their usefulness and their limitations, particularly regarding representativeness and modern attack diversity [7]. This has motivated the devel-opment of application-specific datasets and traffic simulations for web-layer detection experiments.

Cloud-based WAF research has also emphasized scalability and centralized threat handling. Cloud-native WAF architec-tures can combine distributed monitoring, automated updates, and threat intelligence integration to improve protection for large-scale services [8]. However, scalability alone does not fullysolvethechallengeofintelligentclassification,especially when real-time request analytics are required.

Hybrid frameworks that combine rule-based filtering with machine learning classifiers have shown strong promise in recent studies. Such systems use signature engines for fast detectionofknownexploitswhileforwardingmoreambiguous requests to a learning-based component for deeper analysis[9]. This layered approach reduces the load on the classifier and improves operational reliability. Recent intelligent WAF research has also focused on adaptive systems capable of analyzing request behavior, parameter structure, and payload entropy for zero-day attack identification [10].

Despite these advances, many existing approaches either prioritize detection accuracy without sufficient deployment practicality or provide strong filtering logic without meaning-ful monitoring interfaces. The system proposed in this paper addresses this gap by combining signature-based inspection, machinelearning-basedanomalydetection,payloaddecoding, and dashboard-oriented monitoring within a single prototype framework.

III. PROBLEM STATEMENT

Conventional Web Application Firewalls depend heavilyon predefined rule sets to detect malicious requests. This introduces several limitations. First, rule-based engines are highly effective only when attack signatures are already known.Second, attackerscanevadedetectionusingencoding, polymorphism, and payload restructuring. Third, excessive rule sensitivity may increase false positive rates, affecting legitimate users. Finally, maintaining large and frequently updated rule sets is operationally expensive.

Therefore, there is a need for an intelligent and adaptive WAFsystemthatcandetectbothknownandunknownattack patterns while maintaining low false positive rates and practi-cal deployment simplicity.

IV. PROPOSED SYSTEM ARCHITECTURE

Theproposedframeworkconsistsofthefollowingmain modules:

- HTTPRequestInspectionEngine
- PayloadDecodingandNormalizationModule
- Signature-BasedDetectionModule
- FeatureExtractionModule
- MachineLearningDetectionEngine
- DecisionEngine
- LoggingandMonitoringDashboard

Incoming HTTP requests are intercepted by the inspection engine, which forwards the payload to the normalization module.

The payload is decoded using URL decoding and Base64 decoding where applicable.
The normalized data is then processed by the signature-based detection engine to identify explicit malicious patterns such as SQL injection and XSS payloads. If no conclusive rule-based verdict is obtained, features are extracted and sent to the machine learning clas-sifier. The final decision engine allows, blocks, or logs the request. Security events are stored for administrative review and dashboard visualization.

V. METHODOLOGY

The proposed hybrid detection process is organized as a multi-stage pipeline.

A. RequestCapture

Incoming HTTP requests are captured before reaching the applicationserver.Relevantdatasuchasquerystrings,payload content, and headers are extracted for inspection.

B. PayloadDecodingandNormalization

Obfuscated payloads are normalized using decoding tech-niques. This stage helps reveal hidden malicious strings that may bypass naive string matching.

C. Signature-BasedDetection

The normalized request is compared against known mali-ciouspatternsusingregularexpressionsandrule-basedheuris-tics. This stage provides fast detection for clear attack signa-tures.

D. FeatureExtraction

If the payload is not decisively flagged by rules, statistical and structural features are extracted. These include payload length, number of special characters, token distribution, en-tropy, and suspicious symbol frequency.

E. MachineLearningClassification

The extracted features are passed into a trained classifier. The classifier predicts whether the request is benign or mali-cious based on patterns learned from training data.

F. DecisionandLogging

The final decision engine combines the signature and ma-chine learning outputs. Based on the verdict, the request is allowed, blocked, or logged for further review.

VI. WORKFLOW AND DETECTION PIPELINE

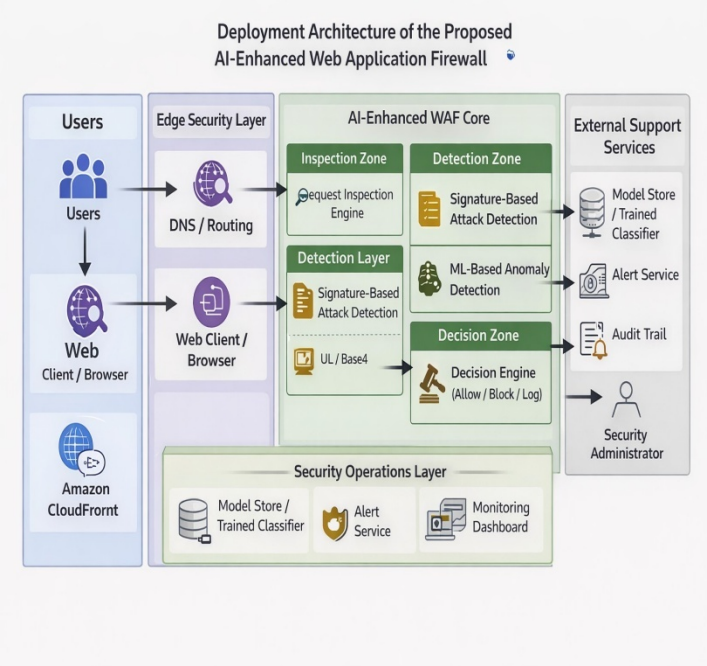


Fig. 1.System workflow of the proposed AI-enhanced WAF showingrequest inspection, payload decoding, rule-based filtering, machine learningclassification, and final action

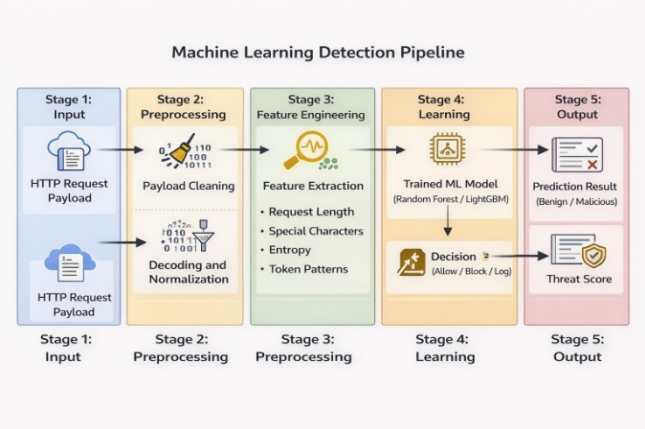


Fig. 2.Machine learning detection pipeline used to preprocess requestpayloads, extract features, classify traffic, and generate threat scores.

Fig. 1 illustrates the overall workflow of the proposed sys-tem. Requests first pass through inspection and normalization stages,followedbysignaturematching.Ifnodefinitiverulehit occurs, the traffic is analyzed by the machine learning module before final action is taken.

Fig. 2 illustrates the machine learning pipeline, including payloadcleaning,normalization,featureextraction,prediction, and threat scoring. This layered flow improves the capability of the system to detect suspicious but previously unseen payloads.

VII. IMPLEMENTATION

The prototype was implemented using Python-based tech-nologies. Flask was used as the backend framework to handle request processing and web interface integration. NumPy and Pandas were used for preprocessing and data handling, while Scikit-learn and LightGBM were considered for classification tasks. HTML, CSS, and JavaScript were used to support the frontend monitoring interface.

The application includes a request analysis interface where input strings or request payloads can be tested for malicious intent. The system generates security outcomes such as *allow*, *block*, and *log*, along with associated analytics.

VIII. EXPERIMENTAL SETUP

The system was evaluated using a mix of benign web requests and crafted malicious payloads representing SQL injection, cross-site scripting, command injection, and traver-sal attacks. The goal of the experiment was to compare the effectiveness of three detection modes:

- Rule-baseddetectiononly
- Machinelearning-baseddetectiononly
- Hybriddetectioncombiningboth

Thefollowingevaluationmetricswereconsidered:

- Accuracy
- Precision
- Recall
- FalsePositiveRate

IX. RESULTS AND DISCUSSION

The hybrid model achieved better overall performance than standalone rule-based detection because it preserved fast sig-nature matching while improving the detection of suspicious requests that lacked direct signature matches. The system was also able to maintain better operational balance by reducing unnecessary blocking of benign traffic.

TABLEI

COMPARATIVEPERFORMANCEOFDETECTIONAPPROACHES

DetectionMethod	Accuracy	FalsePositiveRate
Rule-BasedWAF	82%	High
MLModel	91%	Medium
HybridDetectionSyste m	96%	Low

TableIshowsthatthehybriddetectionsystemoutperformed the standalone approaches in both accuracy and false positive control. This validates the importance of combining determin-istic filtering with adaptive learning.

A. Application Screenshots and Interface Evaluation

ThedevelopedAI-enhancedWebApplicationFirewallpro-totype includes several user-facing and administrative inter-faces that demonstrate the operational capabilities of the proposed system.

TheinterfaceshowninFig.3isdisplayedwhenthe WAF detects a malicious request. It provides a reference ID, timestamp,clientIPaddress,requestpath,andseveritylevel.

Fig.3.Blocked-accessresponsepagegeneratedbytheWebApplicationFirewall when a suspicious request is denied.

Sensitivetechnicalinformationisintentionallyhiddentoavoid exposing internal security mechanisms while still providing enough context for user awareness.

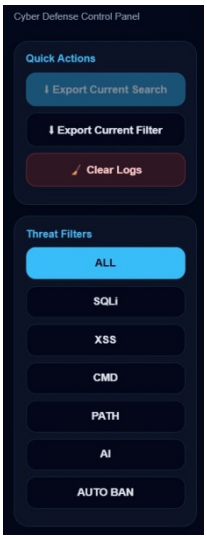


Fig.4.Liveeventfeedshowingrequestmetadata,rule-basedscores,AIdetection scores, and final security decisions.

Fig. 4 illustrates the live security event feed used for monitoring incoming traffic. Each row represents a processed request containing information such as event ID, timestamp, client IP, HTTP method, request path, detection scores, AI flags, and the final decision. This feature enables administra-tors to observe real-time security events and quickly identify potential threats.

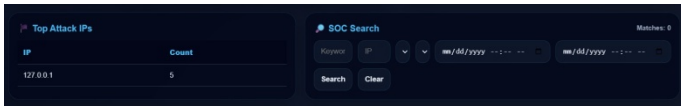


Fig.5.TopattackIPanalysisandSOCsearchinterfaceforsecurityinvestigation.

The interface in Fig. 5 presents the most active attack sourcesalongwithaSecurityOperationsCenter(SOC)search tool. Administrator scanfiltersecurityeventsusingparameters such as keyword, IP address, or time range. This functionality supports incident investigation and threat analysis.

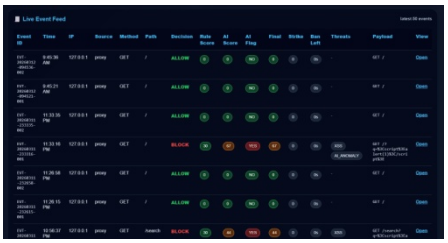


Fig.6.Quick-actionpanelandthreatfilteringinterfaceusedforadministrativecontrol.

Fig. 6 shows the administrative quick-action panel that enables export of search results, filter-based log export, and log clearing operations. The threat filtering module allows administrators to focus on specific attack types such as SQL injection, cross-site scripting (XSS), command injection, and path traversal.

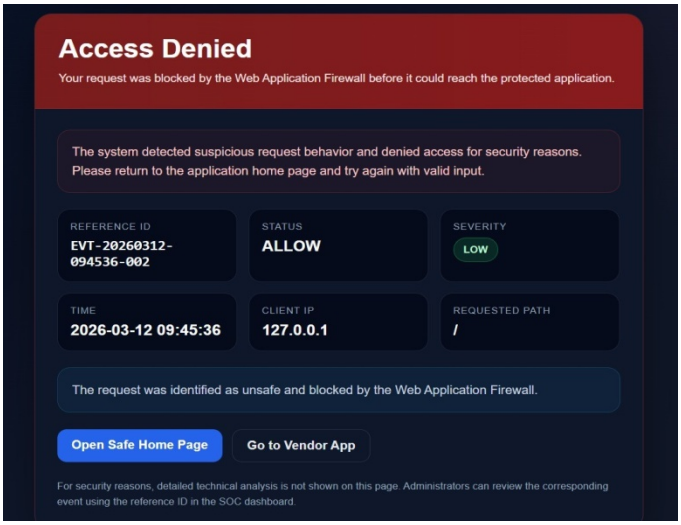


Fig.7.SOCmonitoringdashboarddisplayingeventstatistics,detectioncounts, and threat analytics.

ThemonitoringdashboardshowninFig.7summarizes key security metrics including total events, blocked requests, allowed requests, and AI detections. Additional visualization panels display threat trends and attack type distribution, en-abling administrators to quickly understand the current secu-rity posture of the system.

X. CONCLUSION

This paper presented an AI-enhanced Web Application Firewall that integrates signature-based attack detection with machine learning-based anomaly detection. The hybrid design improves attack detection capability while maintaining prac-tical deployment behavior for web request inspection. The implementation demonstrates that combining traditional rules with learned classifiers can enhance protection against both explicit and previously unseen web attacks.

XI. FUTURE WORK

Futureimprovementstothissystemmayinclude:

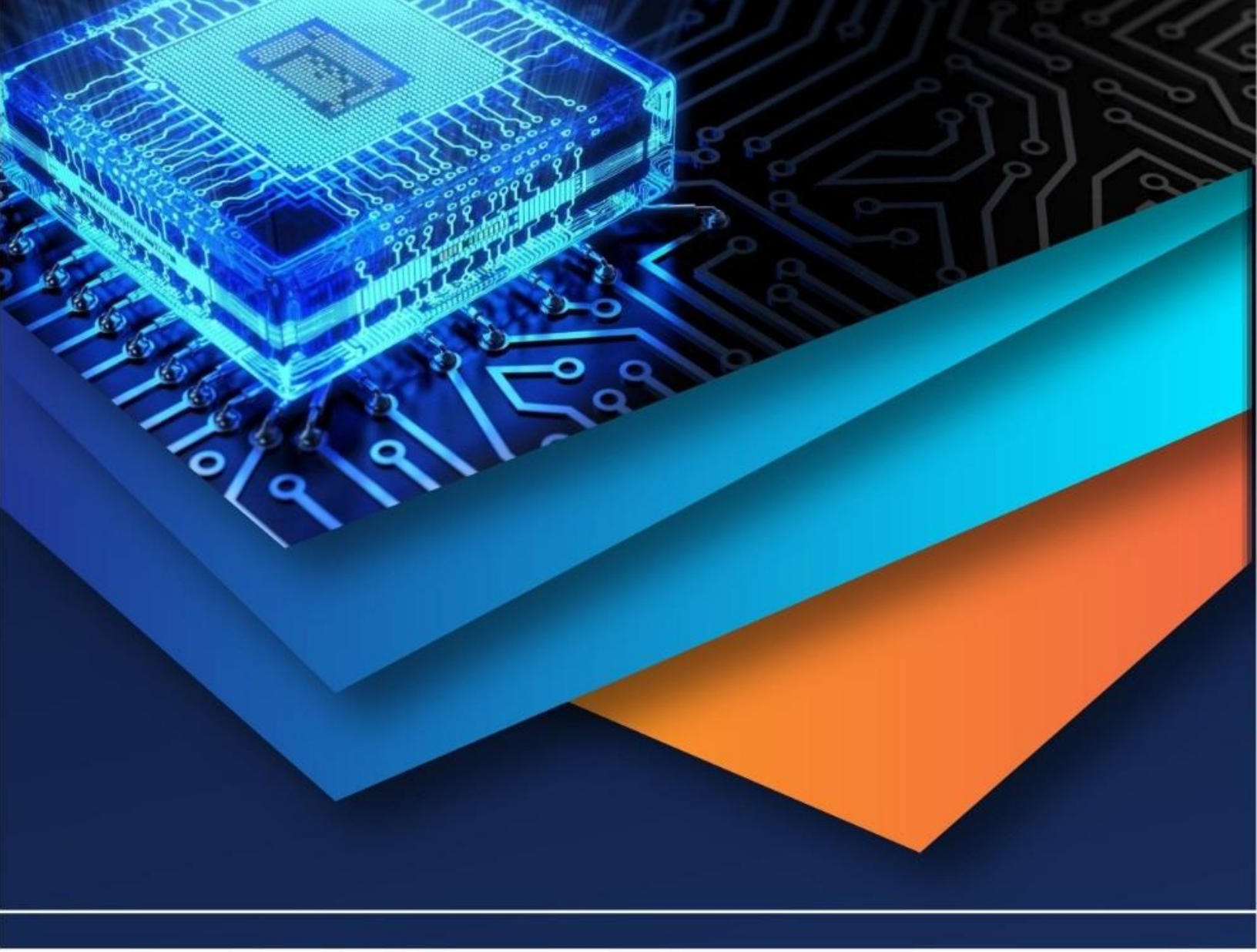
- Integrationofdeeplearningmodelsforricherpayload representation
- Inclusionofreal-timethreatintelligencefeeds
- Deploymentasadistributedorcloud-nativesecurity service
- AdvanceddashboardvisualizationsandSOC-styleana-lytics
- Largerreal-worlddatasetsforbroaderevaluation

XII. ACKNOWLEDGMENT

The authors would like to thank the department, mentors, and project reviewers for their guidance and support duringthe development of this work.

REFERENCES

[1] OWASP Foundation, “OWASP Top 10: The ten most critical webapplication security risks,” 2023.
[2] C. Kruegel and G. Vigna, “Anomaly detection of web-based attacks,”in Proc. 10th ACM Conf. Computer and Communications Security,Washington, DC, USA, 2003, pp. 251–261.
[3] S. Axelsson, “Intrusion detection systems: A survey and taxonomy,”Technical Report 99-15, Chalmers University of Technology, 2000.
[4] H. Debar, M. Dacier, and A. Wespi, “A revised taxonomy for intrusiondetection systems,” Annales des Telecommunications, vol. 55, no. 7–8,pp.361–378,2000.
[5] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, “Using machinelearning to detect malicious web requests,” IEEE Security & Privacy,vol. 16, no. 3, pp. 80–87, 2018.
[6] Y. Kim, W. Lee, and Y. Kim, “Deep learning based intrusion detectionsystem for web applications,” IEEE Access, vol. 7, pp. 123–135, 2019.
[7] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A detailed analysisof the KDD Cup 99 data set,” in Proc. IEEE Symp. ComputationalIntelligenceforSecurityandDefenseApplications,Ottawa,ON,Canada,2009, pp. 1–6.
[8] B. Liu, Y. Xiao, and H. Deng, “Cloud-based web application firewallarchitecture,”IEEETrans.CloudComputing,vol.8,no.2,pp.567–578,2020.
[9] J. S. Park, H. Kim, and D. Shin, “Hybrid intrusion detection systemusing machine learning and rule-based detection,” in Proc. IEEE Int.Conf. Information Security, 2018, pp. 1–6.
[10] R. Singh and P. Sharma, “Intelligent web application firewall usingmachine learning for zero-day attack detection,” in Proc. Int. Conf.Advances in Cyber Security, 2021, pp. 45–50.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY