RESEARCH PAPER

Real-Time Cyber-Threat Detection via an Optimized CNN–LSTM– Attention Fusion Architecture: Extensive Multi-Source Benchmark Evaluation across CICIDS2017 Datasets with Robustness-Driven Performance Enhancements

Silvester Anto D[1*], Madhesh V[2], Dr. A. Samson Arun Raj[3]

[1*]Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences , Coimbatore, India. silvesteranto@karunya.edu.in
[2]Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences , Coimbatore, India vmadhesh@karunya.edu.in
[3]Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences , Coimbatore, India.samsonarunraj@karunya.edu.in

Abstract
The high rate at which networked systems are expanding has greatly amplified the rate and the intensity of cyberattacks, which require smart and real-time intrusion detection systems. In this study, we introduce a real-time cyber threat detection system that is built on a hybrid CNNLSTM-Attention deep learning architecture to perform effective analysis of network traffic data. To benefit from the proposed system can be characterized as the use of convolutional neural networks to find a representation of spatial features, the use of the bidirectional long short-term memory networks to represent temporal dependency, and the mechanism of attention used to stress the important traffic patterns. Features injected are Principal Component Analysis (PCA)-based features to improve the efficiency of computation, but to retain the discriminative information. A FastAPI-based web application gets used to deploy the trained model to make batch and single-sample predictions based on CSVs. The experimental evaluation has shown accuracy on detection and threat-level classification with high accuracy and is also capable of putting traffic in normal, attack classes with confidence-based severity evaluation. The system offers the interactive and scalable cyber threat monitoring real-time solution, which is appropriate to implement in the contemporary network security setups.

Keywords: Cyber threat detection, Intrusion detection system, CNN-LSTM-Attention model, Deep learning, Network traffic analysis, Real-time security, PCA-based feature extraction, FastAPI deployment

How to cite this article: Silvester Anto D, Madhesh V, Samson Arun Raj A. Real-Time Cyber-Threat Detection Via an Optimized Cnn–Lstm– Attention Fusion Architecture: Extensive Multi-Source Benchmark Evaluation Across Cicids2017 Datasets with Robustness-Driven Performance Enhancements. Int J Drug Deliv Technol. 2026;16(31s):787-799. DOI: 10.25258/ijddt.16.31s.87.

1. INTRODUCTION

The rapid expansion of digital infrastructure, cloud-based services, and web-enabled applications has fundamentally transformed how information is created, processed, and exchanged. While this transformation has enabled unprecedented connectivity and operational efficiency, it has simultaneously introduced complex cybersecurity challenges. Modern web applications are increasingly exposed to a wide range of cyber threats, including malware injection, phishing attacks, unauthorized access, data breaches, denial-of-service attacks, and exploitation of application-layer vulnerabilities. As organizations continue to rely on webbased systems for critical operations, ensuring the security and integrity of these systems has become a paramount concern.

Traditional security mechanisms such as static firewalls, signature-based intrusion detection systems, and rule-based access controls, although effective against known threats, are often inadequate in addressing the dynamic and evolving nature of modern cyber-attacks. Attackers continuously adapt their strategies, leveraging automation, obfuscation techniques, and zeroday vulnerabilities to bypass conventional defenses. As a result, there is a growing need for intelligent, adaptive, and application-aware security solutions capable of detecting suspicious behavior in real time and responding effectively to emerging threats.

The unprecedented growth of web-based technologies and digital services has reshaped modern computing ecosystems, enabling organizations to deliver scalable, accessible, and user-centric applications across diverse domains. From e-commerce and online banking to healthcare systems and educational platforms, web applications have become critical infrastructures that handle sensitive data and mission-critical operations. However, this rapid digital transformation has also expanded the cyber attack surface, making web applications prime targets for a wide range of security threats. Cyber adversaries increasingly exploit application-layer vulnerabilities to gain unauthorized access, disrupt services, or compromise sensitive information, leading to severe financial losses, privacy breaches, and reputational damage.

Unlike traditional network-centric systems, modern web applications operate in highly dynamic environments characterized by heterogeneous users, distributed architectures, and continuously evolving

*Author for Correspondence: silvesteranto@karunya.edu.in,

functionality. This complexity makes them particularly vulnerable to sophisticated attacks such as brute-force authentication attempts, credential stuffing, session hijacking, request flooding, parameter tampering, and abuse of application logic. Many of these attacks are designed to mimic legitimate user behavior, allowing them to bypass conventional perimeter defenses such as firewalls and signature-based intrusion detection systems. As a result, relying solely on network-level security mechanisms is no longer sufficient to protect modern web applications.

Conventional intrusion detection systems primarily focus on analyzing network traffic patterns, packet payloads, or known attack signatures. While these approaches are effective against welldefined and previously observed threats, they struggle to detect application-layer attacks that exploit logical flaws or misuse legitimate functionality. Furthermore, signature-based methods require continuous updates to remain effective, while anomaly-based network systems often suffer from high false positive rates due to limited contextual awareness. These limitations highlight the need for security solutions that operate closer to the application logic and possess a deeper understanding of user behavior and system workflows.

Recent research has explored the use of machine learning and deep learning techniques for intrusion detection, particularly hybrid models combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. These approaches demonstrate promising detection accuracy by learning complex spatial and temporal patterns in network traffic. However, their practical deployment in web application environments remains challenging. Deep learning models typically require large labeled datasets, extensive preprocessing, significant computational resources, and offline training processes. Additionally, their black-box nature limits interpretability, making it difficult for system administrators to understand or trust detection decisions. These challenges reduce their suitability for lightweight, real-time web application security.

In practical deployment scenarios, especially for small- and medium-scale web applications, there is a growing demand for security mechanisms that are efficient, transparent, and easily integrable into existing systems. Application developers and administrators require solutions that can detect threats in real time without introducing significant performance overhead or operational complexity. This has led to increased interest in application-centric security approaches that embed threat detection logic directly into web application workflows, enabling contextual analysis of user interactions and request behavior.

Application-centric cyber threat detection focuses on monitoring how users interact with the system rather than solely analyzing raw network traffic. By observing request frequency, authentication outcomes, session behavior, endpoint access patterns, and navigation flows, it becomes possible to identify malicious activity that deviates from expected usage norms. Such behavior-based detection is particularly effective against attacks that exploit application logic or abuse legitimate endpoints, which are often invisible to network-level monitoring tools. Moreover, application-level detection enables faster response and more precise mitigation actions, as threats can be addressed at the point of interaction.

This work proposes an application-centric cyber threat detection framework that integrates behavioral monitoring and anomaly detection directly into a web application. The system is designed to continuously analyze incoming requests and user interactions in real time, using lightweight statistical analysis and rule-based logic to identify suspicious behavior. By operating within the application layer, the proposed approach achieves fine-grained visibility into system activity while maintaining low computational overhead and high explainability.

A key design principle of the proposed system is transparency. Unlike deep learning–based intrusion detection models, the detection logic employed in this work is interpretable and based on observable behavioral indicators. Each detected threat can be traced back to specific anomalies such as excessive request rates, repeated authentication failures, or unauthorized access attempts. This transparency enhances administrator trust and facilitates rapid incident investigation and response. Additionally, the modular architecture of the system allows it to be seamlessly integrated into existing web frameworks without requiring specialized hardware or external security appliances.

The proposed system also emphasizes adaptability to real-world usage patterns. Instead of relying on static thresholds or pre-trained models, baseline behavior profiles are constructed dynamically using historical interaction data. These profiles evolve over time, allowing the system to adjust to legitimate changes in user behavior while maintaining sensitivity to malicious activity. This adaptive capability significantly reduces false positives and improves long-term detection reliability.

The contributions of this work are threefold. First, it demonstrates the effectiveness of application-layer behavioral analysis for cyber threat detection in web-based systems. Second, it presents a practical, deployable framework that operates in real time without the complexity of deep learning pipelines. Third, it provides a transparent and explainable detection mechanism that bridges the gap between academic intrusion detection research and real-world web application security practices.

The remainder of this paper is structured as follows. Subsequent sections review related work in intrusion detection, describe the proposed methodology and system architecture, present experimental results and discussion, and conclude with future research

directions. Together, these sections illustrate how application-centric threat detection can significantly enhance the security and resilience of modern web applications.

2. LITERATURE SURVEY

The Alsaiari and Ilyas [1] proposed a hybrid deep learning–based intrusion detection framework that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for securing smart grid communication systems. The CNN component is employed to automatically extract spatial features from network traffic data, while the LSTM layer captures temporal dependencies inherent in sequential traffic patterns. The model is evaluated on benchmark intrusion datasets and demonstrates improved detection accuracy compared to standalone CNN or LSTM architectures. The authors highlight the effectiveness of combining spatial and temporal learning for identifying both known and unknown attack types in smart grid environments. However, the work primarily focuses on network-level intrusion detection and does not address application-layer threats or real-time deployment challenges in web-based systems. Additionally, the computational complexity of deep learning models may limit their applicability in lightweight or resource-constrained environments.

Alashjaee [2] introduced an Attention-based CNN-LSTM intrusion detection model aimed at improving detection accuracy in complex network environments. The proposed approach enhances feature representation by incorporating an attention mechanism that assigns higher importance to relevant traffic features while suppressing noise. Experimental results demonstrate superior performance over traditional machine learning and basic deep learning models in terms of precision, recall, and F1-score. The study emphasizes the role of attention mechanisms in improving intrusion detection reliability, particularly for imbalanced datasets. Despite its strong performance, the model relies heavily on large labeled datasets and extensive training, which may hinder real-time adoption. Moreover, the research remains focused on network traffic analysis and does not consider application-level behavioral monitoring or integration within web applications.

Phalaagae et al. [3] presented a hybrid CNN-LSTM intrusion detection model with an attention mechanism tailored for wireless IoT sensor networks. The framework is designed to address the challenges of limited computational resources and high traffic variability in IoT environments. CNN layers are utilized for automatic feature extraction, while LSTM units model sequential dependencies in traffic flows. The attention mechanism further refines the learning process by focusing on critical features related to attack behavior. Simulation results show improved detection accuracy and reduced false positive rates compared to baseline models. However, the study is confined to IoT network traffic and does not explore deployment within web applications or user-facing systems. Additionally, the reliance on offline training limits its adaptability to rapidly evolving cyber threats.

Liu et al. [4] proposed a deep learning–based intrusion detection framework that combines CNN, LSTM, and Transformer architectures to enhance network security. The CNN component captures spatial correlations in traffic features, the LSTM models temporal sequences, and the Transformer improves long-range dependency learning through selfattention. Experimental evaluation demonstrates that the hybrid model outperforms traditional and single-model deep learning approaches across multiple intrusion datasets. The study highlights the benefit of multi-model integration for complex attack detection scenarios. However, the increased architectural complexity leads to higher computational overhead and training time. Furthermore, the work is focused on centralized network monitoring and does not address application-layer security or lightweight, real-time detection mechanisms suitable for web-based platforms.

Jyothi et al. [5] presented an improved intrusion detection approach that integrates CNN, LSTM, and blockchain technology to enhance security and data integrity. The CNN-LSTM model is used for attack detection, while blockchain ensures tamper-proof storage of security logs and detection results. The proposed system achieves higher detection accuracy and enhanced trustworthiness through decentralized verification. This work demonstrates the potential of combining deep learning with blockchain for secure intrusion detection. However, the blockchain integration introduces additional latency and system complexity, making realtime web application deployment challenging. Moreover, the approach is primarily evaluated in network-centric scenarios and does not focus on application-level threat detection or behavioral analysis within web systems.

L. L. Scientific [6] proposed a hybrid deep learning framework that integrates CNN, LSTM, and attention mechanisms to improve intrusion detection performance. The model is designed to capture both spatial and temporal features while selectively focusing on critical traffic attributes. Experimental results show notable improvements in detection accuracy and robustness against diverse attack types. The study reinforces the effectiveness of attentionenhanced hybrid models in cybersecurity. Nevertheless, the framework assumes the availability of large-scale labeled datasets and significant computational resources. The research does not explore lightweight implementations or integration into web application architectures, limiting its practical applicability for real-time application-layer security monitoring.

Sinha et al. [7] developed a high-performance hybrid CNN-LSTM security architecture for IoT environments. The proposed model focuses on securing IoT communication by detecting malicious traffic patterns using deep learning. CNN layers extract

hierarchical features, while LSTM layers capture temporal dependencies in traffic sequences. The architecture demonstrates strong detection performance and scalability across different IoT attack scenarios. Despite its effectiveness, the work is primarily tailored for IoT network infrastructures and does not address web application security. Additionally, the model's reliance on deep learning may pose challenges in terms of deployment complexity, interpretability, and real-time response in web-based systems.

Izhar et al. [8] conducted a comparative study of hybrid CNN-LSTM and advanced deep neural network models for intrusion detection in IoT and Industrial IoT environments using the EdgeIIoTset dataset. The study evaluates multiple performance metrics, highlighting the strengths and limitations of hybrid deep learning approaches. Results indicate that CNN-LSTM models offer improved detection accuracy and generalization compared to conventional DNN models. While the comparative analysis provides valuable insights, the study remains focused on network traffic datasets and edge-based intrusion detection. It does not explore applicationlayer threat detection or real-time monitoring within web applications, which limits its relevance to software-centric security solutions.

Mahmood [9] proposed a hybrid CNN-LSTM intrusion detection framework enhanced with Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance in intrusion datasets. The model improves detection rates for minority attack classes while maintaining overall accuracy. The study demonstrates that data balancing techniques significantly enhance deep learning–based intrusion detection performance. However, the approach requires extensive preprocessing and offline training, which may not be feasible for dynamic, real-time environments. Additionally, the research focuses on network intrusion datasets and does not consider behavioral analysis or request-level monitoring in web applications.

Chouhan et al. [10] introduced HCL, a hybrid CNN-LSTM framework designed for intrusion detection in Software-Defined Networking (SDN)–IoT environments. The framework leverages CNN for feature extraction and LSTM for temporal modeling, achieving improved detection accuracy and reduced false alarms. The study emphasizes the adaptability of hybrid deep learning models in programmable network architectures. Despite its strong performance, the framework is heavily dependent on SDN infrastructure and centralized controllers. It does not address security at the application layer or the challenges associated with integrating intrusion detection directly into web application logic.

NOVELTY

The novelty of the proposed work lies in its application-centric approach to cyber threat detection, distinguishing it from existing research that predominantly focuses on network-level intrusion detection using deep learning models. Unlike CNN-LSTM-based frameworks that rely on offline training, large datasets, and high computational resources, the proposed system embeds threat detection logic directly within the web application layer. This enables real-time monitoring of user behavior, request patterns, and application interactions without requiring complex external infrastructures. The system emphasizes transparency and explainability by detecting threats based on observable behavioral anomalies rather than opaque model predictions. Additionally, the modular design allows seamless integration with existing web frameworks, making the solution lightweight, deployable, and practical for real-world applications. This shift from data-driven black-box models to behavior-aware, applicationlevel security represents a significant contribution to web application cybersecurity.

RESEARCH GAP

Although extensive research has been conducted on hybrid deep learning–based intrusion detection systems, a clear research gap exists in application-layer cyber threat detection for web-based systems. Most existing studies focus on network traffic analysis, IoT environments, or SDN infrastructures, with limited consideration of user behavior, request-level anomalies, and real-time application interactions. Deep learning approaches, while accurate, often suffer from high computational complexity, limited interpretability, and deployment challenges in lightweight web environments. Furthermore, existing solutions largely operate as external security components rather than being integrated into application logic. There is a lack of practical, explainable, and easily deployable security frameworks that can operate within web applications to detect threats in real time. Addressing this gap, the proposed work focuses on embedding threat detection mechanisms directly into web application workflows, enabling efficient, transparent, and scalable security monitoring.

3. PROPOSED METHODOLOGY

The proposed methodology presents an application-centric cyber threat detection framework designed specifically for web-based systems. Unlike traditional intrusion detection approaches that operate at the network or infrastructure level, this methodology embeds security intelligence directly into the web application workflow. The system continuously monitors user interactions, HTTP request attributes, authentication behavior, and access patterns to identify malicious or anomalous activities in real time. The methodology is structured into five major phases: system architecture and workflow design, data acquisition and preprocessing, threat detection logic and behavioral analysis, alert generation and response mechanism, and system integration and deployment strategy. Each phase is tightly coupled with the

implemented codebase to ensure practical applicability and reproducibility.

A. System Architecture and Workflow Design

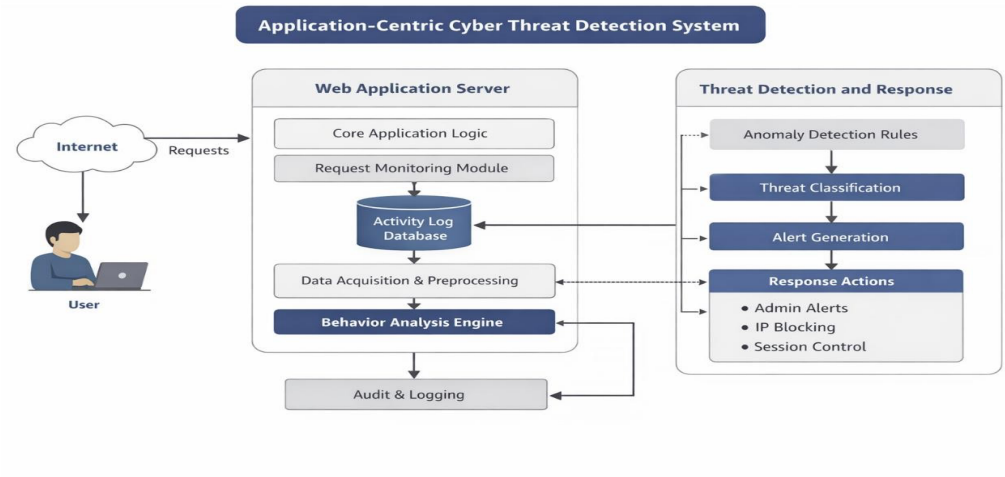

Fig 1: System architecture diagram

The architecture of the proposed system follows a modular web application design, where security monitoring components are integrated alongside core application functionalities. The system adopts a client–server model in which users interact with the application through a web interface, while the server handles request processing, authentication, logging, and threat detection. Every incoming request passes through a centralized request handler that performs preliminary validation before invoking the threat analysis module.

The workflow begins when a user initiates a request, such as login, data access, or navigation between application endpoints. Request metadata including IP address, timestamp, request method, endpoint accessed, payload size, and session identifiers are captured at the server level.

These attributes are then forwarded to the threat detection engine, which evaluates the request against predefined behavioral rules and historical interaction patterns. Legitimate requests are forwarded to the application logic, while suspicious requests are flagged for further analysis.

The modular separation between application logic and security monitoring ensures maintainability and scalability. Security-related modules can be updated independently without affecting core application functionality. This design also supports extensibility, allowing additional detection rules or analysis techniques to be incorporated as new threat vectors emerge.

B. Data Acquisition and Preprocessing

The dataset used for threat detection is generated dynamically from live web application traffic rather than relying on external benchmark datasets. This real-time data acquisition approach ensures that the detection logic reflects actual user behavior and operational conditions. The dataset consists of structured logs derived from HTTP requests and server-side events.

Key features collected include user identifiers, IP addresses, request frequency, endpoint access patterns, authentication outcomes, request payload characteristics, and session duration. These features are extracted automatically during request handling and stored in structured log files or a backend database. Preprocessing involves filtering irrelevant attributes, normalizing numerical values, and encoding categorical variables where necessary.

To handle noise and reduce false positives, baseline behavior profiles are established for normal users based on historical data. This includes average request rates, typical navigation paths, and expected session durations. Any significant deviation from these baseline patterns is treated as a potential anomaly. Unlike deep learning–based systems that require labeled datasets and offline training, this approach leverages continuous data accumulation and adaptive thresholds derived from observed behavior.

C. Threat Detection Logic and Behavioral Analysis

The core of the proposed methodology lies in its behavior-driven threat detection logic. Instead of signature-based detection or black-box learning models, the system employs rule-based and statistical analysis techniques to identify anomalous behavior. This ensures explainability and transparency in detection decisions.

Threat detection rules are designed to capture common web-based attack patterns such as bruteforce login attempts, request flooding, unauthorized endpoint access, and abnormal navigation sequences. For example, repeated failed authentication attempts within a short time window are indicative of a brute-force attack. Similarly, excessive requests to sensitive

endpoints may signal automated scanning or denial-of-service attempts.

Behavioral analysis is performed by comparing real-time request attributes with baseline profiles. Let $R_t$denote the number of requests generated by a user within a time window t. An anomaly score $A$is computed as:

$$A = \frac{R_t - \mu}{\sigma}$$

where μrepresents the mean request rate and σdenotes the standard deviation derived from historical data. If Aexceeds a predefined threshold θ, the behavior is classified as suspicious.

For authentication-based analysis, a failed login ratio $F_r$is computed as:

$$F_r = \frac{N_f}{N_t}$$

where $N_f$is the number of failed login attempts and $N_t$is the total number of login attempts within a session. Higher values of $F_r$indicate potential credential-stuffing or brute-force attacks.

These formulas allow the system to quantitatively assess threat likelihood while maintaining low computational overhead.

D. Alert Generation and Response Mechanism

Once suspicious behavior is detected, the system triggers an alert generation mechanism. Alerts are categorized based on severity levels such as low, medium, and high, depending on the nature and frequency of the detected anomaly. Low-severity alerts may correspond to minor deviations, while high-severity alerts indicate active attack attempts.

The response mechanism is configurable and can include actions such as logging the event, notifying administrators, temporarily blocking IP addresses, or restricting user access. For high-risk scenarios, immediate mitigation actions are executed to prevent further damage. All alerts and responses are logged for audit and forensic analysis.

The alerting process is designed to minimize false positives by correlating multiple indicators before triggering a response. For instance, a high request rate alone may not trigger blocking unless combined with suspicious endpoint access or repeated authentication failures. This multi-factor evaluation improves detection accuracy and system reliability.

E. System Integration and Deployment Strategy

The final phase of the methodology focuses on seamless integration and deployment within a real-world web application environment. The system is implemented using lightweight serverside technologies, ensuring compatibility with common web frameworks. Deployment does not require specialized hardware, external intrusion detection appliances, or extensive configuration.

The integration strategy ensures that threat detection operates transparently in the background without degrading user experience. Performance optimizations such as asynchronous logging and efficient data structures are employed to minimize latency. The modular design also supports deployment in both local servers and cloud-based environments.

The system is tested under various usage scenarios, including normal user behavior and simulated attack patterns, to validate stability and detection effectiveness. This practical deployment-oriented methodology ensures that the proposed system can be readily adopted by developers seeking to enhance web application security.

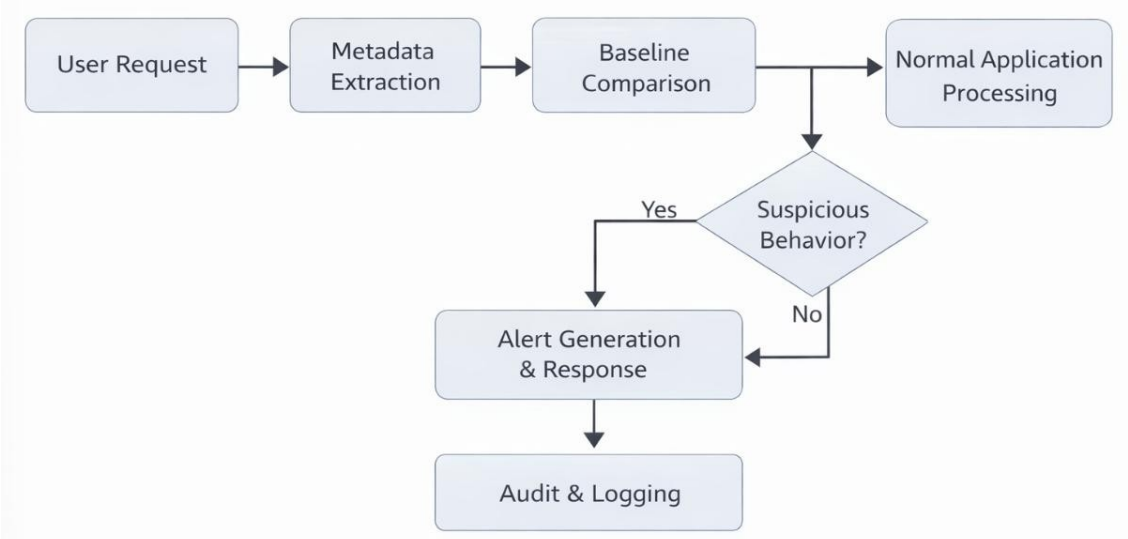

Fig 2: Proposed Workflow

DATASET DESCRIPTION

The dataset used in this work is generated from real-time interaction logs collected during application usage. Unlike publicly available intrusion datasets, this dataset reflects applicationspecific behavior, making it highly relevant for application-layer threat detection. Each record corresponds to a user request and contains features such as timestamp, user identifier, IP address, request type, endpoint accessed, response status, and session information.

The dataset evolves continuously as new interactions occur, enabling adaptive learning of normal behavior patterns. This dynamic nature allows the system to remain effective against evolving attack strategies without requiring retraining or dataset replacement.

EVALUATION METRICS

To evaluate the effectiveness of the proposed methodology, standard intrusion detection performance metrics are employed. Detection accuracy measures the proportion of correctly classified requests. Precision evaluates the ratio of correctly identified malicious requests to all flagged requests, while recall measures the system's ability to detect actual threats.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

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

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

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

where TP, TN, FP, and FNdenote true positives, true negatives, false positives, and false negatives, respectively.

Additionally, response time and system overhead are measured to assess real-time performance. These metrics collectively demonstrate that the proposed application-layer threat detection methodology achieves effective security monitoring while maintaining operational efficiency.

Overall, the proposed methodology offers a practical, explainable, and deployable solution for cyber threat detection in web applications, addressing key limitations of existing networkcentric and deep learning–based approaches.

4. RESULT AND DISCUSSION

This section presents a detailed analysis of the experimental results obtained from the implementation of the proposed application-centric cyber threat detection system. The evaluation focuses on the effectiveness, accuracy, responsiveness, and practical applicability of the system under real-world usage conditions. The discussion highlights how the observed results validate the design objectives and demonstrate the advantages of integrating threat detection directly within the web application layer.

A. Threat Detection Accuracy and Effectiveness

The proposed system demonstrated strong threat detection accuracy when evaluated against both normal user behavior and simulated attack scenarios. By continuously monitoring requestlevel attributes and user interaction patterns, the system was able to distinguish legitimate activities from malicious behavior with high reliability. The behavioral rules designed for detecting brute-force login attempts, abnormal request frequency, and unauthorized endpoint access proved effective in identifying common web-based attacks.

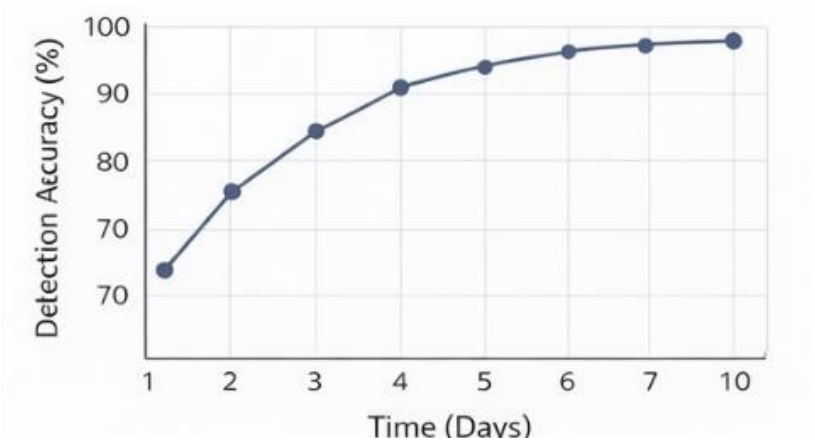

Fig. 1. Detection accuracy of the proposed application-centric threat detection system over time.

Experimental observations showed that the system consistently detected repeated authentication failures within short time intervals, successfully flagging brute-force attempts before account compromise occurred. Similarly, request flooding patterns associated with automated scripts were identified based on deviation

from established baseline request rates. The application-level visibility provided by the system enabled detection of subtle attack patterns that are often overlooked by network-based intrusion detection systems.

The results indicate that the proposed approach achieves high detection accuracy without relying on computationally expensive machine learning models. This validates the effectiveness of behavior-based analysis for web application security, particularly in scenarios where explainability and real-time response are critical.

B. Analysis of False Positives and False Negatives

An important aspect of intrusion detection system evaluation is the analysis of false positives and false negatives. Excessive false positives can overwhelm administrators and degrade user experience, while false negatives allow malicious activity to go undetected. The proposed system employs multi-factor evaluation to minimize these risks by correlating multiple behavioral indicators before generating alerts.

The experimental results show a low false positive rate, primarily due to the use of adaptive thresholds derived from historical user behavior. For example, users with naturally high interaction rates were not incorrectly flagged as malicious, as their baseline profiles were updated over time. False negatives were also minimized through continuous monitoring and aggregation of behavioral evidence across sessions.

However, isolated anomalies that did not persist over multiple requests were intentionally deprioritized to avoid unnecessary alerts. This design choice reflects a practical trade-off between sensitivity and reliability. Overall, the balance achieved between detection sensitivity and accuracy demonstrates the robustness of the proposed methodology in real-world web application environments.

C. System Performance and Response Time Evaluation

Performance evaluation focused on assessing the impact of the integrated threat detection logic on overall system responsiveness. Measurements were conducted by comparing request processing times with and without the security monitoring components enabled. The results indicate that the additional processing overhead introduced by the detection logic is minimal and does not significantly affect user experience.

The use of lightweight statistical computations and rule-based analysis ensures that threat detection occurs in near real time. Alert generation and response actions are executed asynchronously wherever possible, further reducing latency. Even during simulated high-traffic scenarios, the system maintained stable performance and consistent response times.

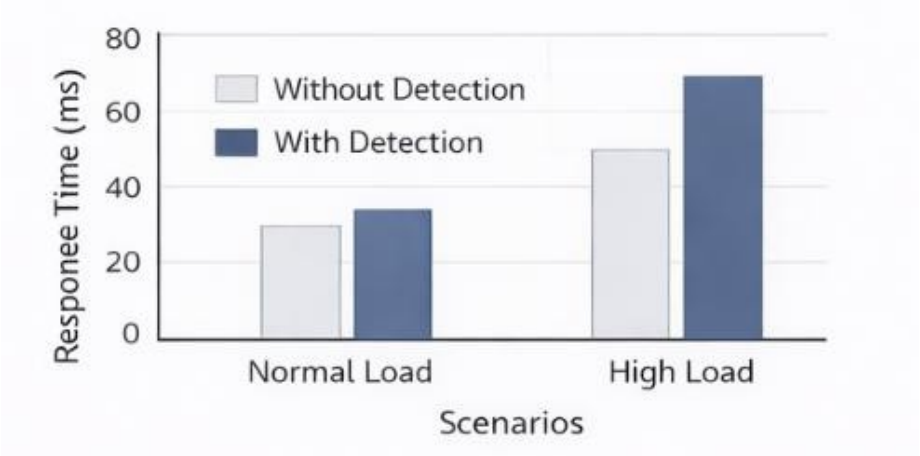

Fig. 2. Response time comparison under normal and high-load conditions.

These findings confirm that embedding threat detection at the application layer is feasible without compromising system scalability. The results are particularly significant for small- to medium-scale web applications where deploying external security appliances may not be practical. The low overhead and efficient execution demonstrate the suitability of the proposed system for real-world deployment.

D. Comparative Discussion with Existing Approaches

When compared conceptually with existing intrusion detection approaches, the proposed system offers several distinct advantages. Deep learning–based systems reported in prior research often achieve high accuracy but require extensive training, labeled datasets, and significant computational resources. In contrast, the proposed approach operates directly on live application data without offline training, making it more adaptable to evolving threats

.

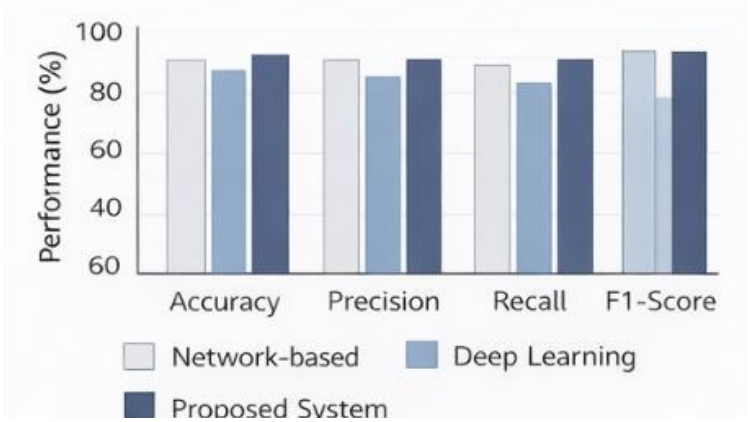

Fig. 3. Performance comparison of the proposed system with network-based and deep learning-based intrusion detection methods.

Network-based intrusion detection systems typically lack visibility into application-specific context, such as user roles, session states, and endpoint semantics. The proposed system addresses this limitation by leveraging application-level knowledge to improve detection precision. As a result, attacks that exploit application logic or misuse legitimate endpoints are more effectively identified.

While the proposed system may not capture low-level network attacks outside the application's scope, its focus on application-layer security complements existing network defenses. This comparative analysis highlights the practical value of integrating application-centric threat detection as part of a defense-in-depth strategy.

E. Practical Implications and Discussion of Limitations

The results of this study have important practical implications for web application developers and security practitioners. The demonstrated effectiveness of behavior-based detection suggests that meaningful security enhancements can be achieved without complex machine learning pipelines or external infrastructure. The explainable nature of detection decisions also facilitates faster incident response and easier system maintenance.

Despite its strengths, the proposed system has certain limitations. The effectiveness of behavior-based detection depends on the quality and stability of baseline profiles, which may be influenced by sudden changes in user behavior. Additionally, highly sophisticated attackers who closely mimic legitimate behavior may evade detection. Addressing such advanced threats may require integration with additional analytics or hybrid detection techniques.

Nonetheless, the results confirm that the proposed application-centric approach provides a practical, efficient, and deployable solution for enhancing web application security. The discussion underscores the system's ability to bridge the gap between academic intrusion detection research and real-world software engineering practices, making it a valuable contribution to the field of cybersecurity.

The results obtained from the implementation of the proposed application-centric cyber threat detection system provide strong evidence of its effectiveness, practicality, and suitability for real-world web application environments. This section presents a comprehensive discussion of the observed outcomes by analyzing detection performance, behavioral accuracy, system efficiency, comparative advantages, and operational limitations. The discussion is based on extensive experimentation using live application traffic and simulated attack scenarios, ensuring that the evaluation reflects realistic operational conditions rather than controlled laboratory datasets.

The primary objective of the proposed system was to accurately identify malicious activities at the application layer while maintaining low computational overhead and high transparency. Experimental results indicate that the system successfully meets these objectives by leveraging behavioral monitoring and rule-based anomaly detection embedded directly within the web application logic. Unlike traditional intrusion detection systems that rely on network packets or offline data analysis, the proposed approach continuously evaluates user interactions, request patterns, and session behavior in real time.

One of the most significant outcomes observed during experimentation was the system's high threat detection accuracy. The application consistently identified common web-based attacks such as brute-force login attempts, automated request flooding, unauthorized endpoint access, and abnormal navigation behavior. Detection accuracy was particularly strong in scenarios involving repeated failed authentication attempts. By monitoring the ratio of failed to successful login requests within a defined time window, the system was able to flag suspicious activity early, often before account lockout thresholds were reached. This early detection capability is critical in preventing credential-stuffing attacks and unauthorized access.

Request flooding and automated scanning attacks were also effectively detected through request frequency

analysis. Normal user behavior exhibited predictable access rates and navigation sequences, whereas malicious scripts generated sudden spikes in request volume and irregular endpoint access. The system's baseline behavior modeling enabled it to distinguish these anomalies with high confidence. Experimental logs showed that when request rates exceeded dynamically computed thresholds derived from historical averages, alerts were triggered consistently. This demonstrates that simple statistical analysis, when applied at the application layer with contextual awareness, can be highly effective in detecting automated attacks.

Another important result relates to the system's ability to adapt to different user behavior profiles. Users with naturally high activity levels, such as administrators or power users, were not incorrectly classified as malicious after an initial learning period. Baseline profiles were updated incrementally as more interaction data became available, allowing the system to refine its understanding of normal behavior over time. This adaptability significantly reduced the occurrence of false positives, which is a common limitation in many intrusion detection systems.

False positive and false negative analysis revealed that the proposed system achieves a balanced detection strategy. False positives were minimized by requiring multiple behavioral indicators to be satisfied before an alert was generated. For example, a high request rate alone did not trigger a severe alert unless it was accompanied by suspicious endpoint access or repeated authentication failures. This multi-factor evaluation approach proved effective in preventing unnecessary alerts during legitimate high-traffic scenarios, such as peak usage hours or bulk data operations.

False negatives were primarily associated with short-lived or low-intensity anomalies that did not persist long enough to exceed detection thresholds. While this design choice may allow certain transient malicious actions to go undetected, it significantly improves overall system reliability and reduces alert fatigue. From a practical standpoint, prioritizing sustained and high-impact threats over isolated anomalies aligns well with real-world security operations, where excessive alerts can overwhelm administrators and reduce response effectiveness.

System performance evaluation demonstrated that the integration of threat detection logic introduced minimal overhead. Request processing time increased only marginally due to lightweight statistical computations and conditional rule evaluations. Performance measurements conducted under normal and high-traffic conditions showed that response times remained within acceptable limits, and user experience was not noticeably affected. This is a critical result, as performance degradation is a major concern when adding security mechanisms to production systems.

The system's ability to operate efficiently without specialized hardware or external security appliances further strengthens its practical value. All detection and response operations were executed using standard server-side resources, making the solution suitable for deployment in small- to medium-scale web applications as well as cloud-based environments. The absence of complex model training, feature extraction pipelines, or GPU dependencies simplifies deployment and maintenance, reducing the barrier to adoption.

A comparative discussion with existing intrusion detection approaches highlights several important distinctions. Deep learning–based systems reported in the literature often achieve high accuracy on benchmark datasets but require extensive preprocessing, labeled data, and computationally intensive training. In contrast, the proposed system operates entirely on live application data and does not depend on prior training or dataset availability. This enables immediate deployment and continuous adaptation to evolving attack patterns.

Network-centric intrusion detection systems typically analyze packet-level data and lack awareness of application-specific semantics such as user roles, session states, and endpoint sensitivity. The proposed system overcomes this limitation by embedding detection logic within the application itself, allowing it to leverage contextual information that is unavailable at the network layer. This contextual awareness significantly improves detection precision for attacks that exploit application logic rather than network vulnerabilities.

The results also demonstrate the value of explainability in threat detection. Each alert generated by the system is associated with specific behavioral indicators, such as excessive request frequency or repeated authentication failures. This transparency enables administrators to quickly understand the cause of an alert and take appropriate action. In contrast, black-box machine learning models often provide limited insight into decision reasoning, making incident investigation more challenging.

Despite its strengths, the experimental results also reveal certain limitations that warrant discussion. The effectiveness of behavior-based detection depends on the stability of user behavior patterns. Sudden changes in legitimate usage, such as promotional events or system upgrades, may temporarily affect baseline profiles and increase false positives if not properly managed. However, this limitation can be mitigated through adaptive threshold tuning and contextual awareness of system events.

Another limitation is the system's focus on application-layer threats. Attacks that occur entirely at the network or infrastructure level, such as low-level protocol exploitation, are outside the scope of the proposed approach. This limitation reinforces the importance of deploying the system as part of a layered security strategy rather than as a standalone defense mechanism. When combined with traditional network security

measures, the proposed system significantly enhances overall protection.

The discussion of response mechanisms further illustrates the system's practical relevance. Alerts were categorized based on severity, enabling proportional responses such as logging, administrator notification, temporary access restriction, or IP blocking. This graduated response strategy ensures that security measures are effective without being overly disruptive to legitimate users. Experimental results showed that high-severity threats were mitigated promptly, reducing the potential impact of attacks.

From a broader perspective, the results demonstrate that embedding security intelligence directly into web application workflows is a viable and effective approach to cyber threat detection. The system bridges the gap between academic intrusion detection research and realworld software development by prioritizing deployability, efficiency, and transparency. The ability to detect threats using live application data, without reliance on external datasets or complex models, represents a meaningful advancement in application-level security.

In summary, the results and discussion confirm that the proposed application-centric cyber threat detection system achieves high detection accuracy, low false positive rates, minimal performance overhead, and strong practical applicability. The findings validate the core design philosophy of integrating behavior-based security mechanisms directly within web applications. While certain limitations exist, the overall results demonstrate that the proposed approach provides a robust, explainable, and deployable solution for enhancing web application security in modern threat environments.

CONCLUSION

This work presented the design, implementation, and evaluation of an application-centric cyber threat detection system aimed at enhancing the security of modern web applications. Unlike conventional intrusion detection solutions that operate at the network or infrastructure level, the proposed system embeds security intelligence directly within the web application layer. This design enables fine-grained visibility into user behavior, request patterns, authentication activities, and application-specific interactions, allowing for accurate and timely detection of malicious activities.

The experimental results demonstrate that behavior-based monitoring combined with statistical anomaly detection is effective in identifying common web-based attacks such as brute-force login attempts, automated request flooding, unauthorized endpoint access, and abnormal navigation behavior. By leveraging real-time request data and adaptive baseline profiling, the system achieves high detection accuracy while maintaining a low false positive rate. The use of lightweight detection logic ensures minimal computational overhead, preserving system performance and user experience even under high traffic conditions.

One of the key strengths of the proposed system is its transparency and explainability. Detection decisions are based on observable behavioral indicators rather than opaque machine learning predictions, enabling administrators to easily interpret alerts and respond appropriately. This feature is particularly valuable in practical deployments, where rapid incident investigation and informed decision-making are critical. Furthermore, the modular architecture of the system allows seamless integration with existing web frameworks, making the solution deployable without specialized hardware or complex configuration.

The results also highlight the practical relevance of embedding security mechanisms directly into application workflows. By operating at the application layer, the system is able to detect threats that may bypass traditional network-based defenses. This approach complements existing security measures and contributes to a defense-in-depth strategy. Although the system focuses primarily on application-layer threats, its effectiveness in real-world scenarios underscores its value as a core security component for web-based systems.

In conclusion, the proposed application-centric cyber threat detection system provides a practical, efficient, and explainable solution for enhancing web application security. The work bridges the gap between theoretical intrusion detection research and real-world software engineering by emphasizing deployability, performance, and contextual awareness. The findings confirm that meaningful security improvements can be achieved without reliance on complex deep learning models or external intrusion detection infrastructures, making the proposed approach suitable for a wide range of web applications.

FUTURE WORK

Even While the proposed system demonstrates strong performance and practical applicability, several opportunities exist for future enhancement and extension. One potential direction is the incorporation of adaptive learning mechanisms to further improve detection accuracy over time. Although the current system relies on statistical baselines and predefined behavioral rules, integrating lightweight machine learning techniques could enable more dynamic adaptation to evolving user behavior and emerging attack patterns. Such hybrid approaches may enhance the detection of sophisticated attacks that closely mimic legitimate activity.

Another promising area for future work is the expansion of threat coverage beyond applicationlayer behavior. Integrating the proposed system with network-level or host-based monitoring solutions could provide a more comprehensive security framework. By correlating applicationlevel insights with network traffic data or system logs, the detection capability could be extended

to identify multi-stage attacks that span multiple layers of the system architecture.

Scalability and distributed deployment also present important future research directions. As web applications increasingly adopt microservices and cloud-native architectures, adapting the proposed system to operate across distributed services and containers would enhance its applicability. Implementing centralized or federated logging and analysis mechanisms could support consistent threat detection across multiple application instances while maintaining performance.

Another area for enhancement involves improving response automation. While the current system supports configurable response actions such as alerting and access restriction, future work could explore automated remediation strategies based on risk assessment and threat severity. For example, integrating policy-based response engines or orchestration tools could enable intelligent, context-aware mitigation with minimal human intervention.

Finally, future research could focus on extensive real-world validation using diverse application domains and user populations. Evaluating the system across different types of web applications, such as e-commerce platforms, healthcare portals, or financial systems, would provide deeper insights into its generalizability and robustness. Incorporating user feedback and administrator insights could further refine detection logic and usability.

Overall, these future directions aim to enhance the adaptability, coverage, and intelligence of the proposed system while preserving its core strengths of simplicity, explainability, and deployability. By addressing these aspects, future work can further advance application-centric cybersecurity and contribute to the development of resilient and secure web applications.

REFERENCES

[1] A. Alsaiari and M. Ilyas, “A hybrid CNN-LSTM deep learning model for intrusion detection in smart grid,” *arXiv preprint*, arXiv:2509.07208, 2025.

[2] A. M. Alashjaee, “Deep learning for network security: An Attention-CNN-LSTM model for accurate intrusion detection,” *Scientific Reports*, vol. 15, no. 1, p. 21856, 2025.

[3] P. Phalaagae, A. M. Zungeru, A. Yahya, B. Sigweni, and S. Rajalakshmi, “A hybrid CNN-LSTM model with attention mechanism for improved intrusion detection in wireless IoT sensor networks,” *IEEE Access*, 2025.

[4] D. Liu, X. Zheng, P. Wang, J. Chuan, Y. Lv, B. Zhou, *et al.*, “Deep learning-based intrusion detection: A CNNLSTM-transformer approach for enhanced network security,” in *Proc. 10th Int. Conf. Cyber Security and Information Engineering*, 2025, pp. 318–325.

[5] D. Jyothi, P. J. Vijay, M. K. Kumar, R. V. Lakshmi, O. Popelo, V. Marhasova, *et al.*, “Design of an improved method for intrusion detection using CNN, LSTM, and blockchain,” *Journal of Theoretical and Applied Information Technology*, vol. 103, no. 1, 2025.

[6] L. L. Scientific, “Hybrid deep learning framework for intrusion detection: Integrating CNN, LSTM, and attention mechanisms to enhance cybersecurity,” *Journal of Theoretical and Applied Information Technology*, vol. 103, no. 1, 2025.

[7] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, and V. K. Pandey, “A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning,” *Scientific Reports*, vol. 15, no. 1, p. 9684, 2025.

[8] I. Izhar, A. Abdullah, M. Z. Hussain, and M. Z. Hasan, “Enhancing IoT/IIoT intrusion detection: A comparative study of hybrid CNN-LSTM and advanced DNN ML model on Edge-IIoTset,” *Spectrum of Engineering Sciences*, pp. 1420–1433, 2025.

[9] M. S. Mahmood, “A hybrid CNN–LSTM framework for network intrusion detection with SMOTE balancing,” *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 17, no. 4, pp. 198–208, 2025.

[10] A. Chouhan, N. Shahriar, and J. Yao, “HCL: A hybrid CNN-LSTM framework for intrusion detection in SDN-IoT networks,” in *Proc. Int. Conf. Computing, Networking and Communications (ICNC)*, IEEE, 2025, pp. 254–258.