International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3298-3302
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0422

An Adaptive Hybrid Intelligence System for API Abuse and Data Exposure Detection

Padma Latha P[1], Harini K[2] , Gokila J[3], Deepa Priya V[4]
[1,2,3]UG – Information Technology, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu
[4] Assistant Professor, Information Technology, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu
Email ID: *padmalathapothiraj@gmail.com[1], harinik180906@gmail.com[2], gokilajayachandran493@gmail.com[3], deepapriyakcet@gmail.com[4]*

Abstract
As application programming interfaces (APIs) have become the primary communication layer for digital services, they are increasingly targeted by sophisticated cyber attackers seeking direct access to sensitive data. Traditional security mechanisms, which rely on static rules and predefined signatures, often fail to detect evolving attack patterns and zero-day threats. This paper proposes an intelligent, lightweight middleware security layer designed to mitigate API abuse and prevent unauthorized data exposure in real-time. The proposed system addresses the critical limitations of conventional API security approaches — such as the inability to detect low-and-slow request patterns and Broken Object Level Authorization (BOLA) flaws — by implementing a hybrid architecture. This multi-staged pipeline transitions from deterministic filtering to behavioral analysis. It first employs a rule engine to block high-frequency attacks and validate authentication with minimal latency. For more advanced threats, an artificial intelligence (AI) engine utilizes the Isolation Forest algorithm to identify subtle anomalies in request patterns and endpoint access, even when valid credentials are used. Finally, the system inspects outgoing response payloads to detect and prevent mass data exposure caused by authorization flaws or backend misconfigurations. By integrating an adaptive feedback loop for continuous model refinement, demonstrates enhanced resistance to zero-day attacks compared to static systems. The final solution provides a scalable and industrially applicable framework for safeguarding cloud-based ecosystems. Experimental deployment using a Python-based stack comprising Fast API, Scikit-learn, MongoDB, and Redis confirms that combining fast rule-based filtering with AI-driven anomaly detection effectively protects modern applications from sophisticated API abuse and sensitive data leakage.
Keywords: *Anomaly detection; Cloud security; Data exposure; Hybrid intelligence; Isolation Forest*

1. Introduction
In recent years, the use of APIs has grown rapidly, especially with the rise of cloud-based applications and microservices. APIs allow different software systems to communicate with each other, making them a core part of modern web and mobile applications. However, because APIs are exposed to the internet, they have also become a common target for attackers (Kaul & Khurana, 2021; Zoppi et al., 2021). This has raised serious concerns about how to protect API-based systems from different types of cyber threats. One of the biggest challenges in API security is detecting attacks that have never been seen before, commonly known as zero-day attacks. Traditional security systems rely on known attack signatures or fixed rules, so they are not able to detect new or unknown threats (Zoppi et al., 2021). This is a major limitation, as attackers continuously change their methods to bypass detection systems. To overcome this issue, researchers have started using machine learning and AI techniques for threat detection. These approaches can learn from data and identify unusual patterns without depending on predefined rules (Farzaan et al., 2025; Sharma & Grover, 2024). For example, unsupervised methods like Isolation Forest can detect anomalies by learning normal API behaviour and identifying deviations (Pu et al., 2021). Some recent studies have also used generative AI to create synthetic attack data, which

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3298-3302
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0422

helps solve the problem of limited real-world datasets (Pawana et al., 2024). However, several challenges still remain. Machine learning models alone tend to generate a high number of false positives and may not always perform efficiently in real-time environments (Cauchideesan & Poravj, 2023; Sharma & Grover, 2024). On the other hand, rule-based systems are fast and accurate for known attacks but cannot detect new threats. Most existing research focuses on either rule-based or machine learning approaches separately, and only a few studies attempt to combine both methods into a single system (Huang et al., 2023; Zoppi et al., 2021). To address these limitations, this paper proposes a hybrid system. The main idea is to combine rule-based detection with an AI-based anomaly detection model, allowing the system to detect both known and unknown attacks. Unlike previous approaches, this system focuses on integrating both methods into a single framework while maintaining real-time performance.

The objectives of this study are to:

1. Build a hybrid framework that combines rule-based and AI-based detection for API security.
2. Monitor API traffic in real time and detect anomalies with minimal delay.
3. Evaluate the system performance in terms of accuracy and efficiency. [6-10]
4. Demonstrate that the hybrid approach performs better than individual methods.

1.1.Background on API Security and AI-Based Detection

APIs play an important role in connecting different services and platforms in modern digital systems. Since APIs are typically accessible over the internet, they are exposed to a wide range of security threats. Basic security measures such as authentication and encryption provide a level of protection, but they are not always sufficient against advanced attacks (Kaul & Khurana, 2021).AI-based detection methods are increasingly being explored to improve API security, as they can analyse large volumes of traffic data and automatically identify suspicious patterns (Farzaan et al., 2025).

1.2.Limitations and Research Gap

Despite significant progress, existing systems still have several limitations. Rule-based systems can only detect known attacks, while machine learning models often suffer from high false positive rates and require careful tuning (Huang et al., 2023; Zoppi et al., 2021).Another important challenge is maintaining real-time performance without increasing latency in API responses (Cauchideesan & Poravj, 2023).The proposed hybrid system aims to address these issues by combining the strengths of both approaches, providing a more balanced and effective solution for API security. [1-5]

2. Method

This section describes the design and implementation of the proposed Hybrid system. The system is built using Python and follows a modular architecture to allow independent testing of each component. The overall goal is to detect both known and unknown API-based attacks by combining a rule-based filtering layer with an AI-driven anomaly detection model. Table 1Shows Dataset Summary

Table 1. Dataset Summary

Parameter	Value
Total Records	12,000
Normal Requests	7,440
Attack Requests	4,560
Number of Features	6
Dataset Type	Synthetic

Table 2 Feature Description

Feature Name	Description
Request rate	Number of requests per time interval
Endpoint count	Number of unique endpoints accessed
Token reuse	Reuse of authentication tokens
Payload size	Size of API request
Time gap	Time between consecutive requests
behavior core	Combined behavioral Metric

2.1.System Architecture

The proposed Hybrid system follows a multi-

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3298-3302
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0422

layered architecture. Initially, incoming API requests are processed through a rule-based filtering layer that detects known attack patterns such as high request rates and invalid tokens. If the request passes this layer, feature extraction is performed to convert request data into numerical format. These features are then passed to the AI-based anomaly detection model. The Isolation Forest model analyzes the request behavior and classifies it as normal or anomalous. Finally, the response module logs the results and takes appropriate action such as blocking or allowing the request. Table 3.

Table 3 Model Configuration

Parameter	Value
Algorithm	Isolation Forest
Dataset Size	12,000 records
Features Used	6
Contamination (Experiment 1)	0.1
Contamination (Experiment 2)	0.3
Programming Language	Python
Libraries Used	Scikit-learn, Pandas, NumPy

2.2.Dataset Preparation

A synthetic dataset was generated to simulate API request behavior. The dataset contains 12,000 records with both normal and attack patterns. Features such as request frequency, endpoint usage, token reuse, payload size, and time gaps were included. The dataset was preprocessed to remove missing values and normalized before training. Labels were used only for evaluation and not during training, as the model follows an unsupervised approach.

2.3.Feature Extraction

Feature extraction was performed to convert raw API request data into meaningful numerical values. Six features were selected based on API behaviour patterns, including request rate, endpoint count, token reuse, payload size, time gap, and behaviour score. These features help the model understand request patterns and detect anomalies effectively.

2.4.Model Training

The Isolation Forest algorithm was used for anomaly detection due to its efficiency in handling high-dimensional data. The model was trained using two different contamination values (0.1 and 0.3) to analyse the effect on detection performance. The model learns normal behaviour patterns and identifies anomalies based on deviations from these patterns. The performance of the model was evaluated using standard metrics such as accuracy, precision, recall, F1 score, and false positive rate. A confusion matrix was also used to analyze the classification results in detail.

3. Results and Discussion

3.1.Results

The experiments were conducted to evaluate the performance of the proposed hybrid system. Three configurations were compared: a standalone rule-based model, a standalone AI-based model using Isolation Forest, and the combined hybrid model. Additionally, two contamination settings (0.1 and 0.3) were tested for the Isolation Forest to observe the effect on detection accuracy. The results are presented in Figures 1, 2, and 3 below. Table 4.

Table 4 Performance Comparison

Metric	IF (0.1)	IF (0.3)
Accuracy	72%	89.13%
Precision	100%	95.22%
Recall	26.32%	75.18%
F1 Score	41.67%	84.02%
False Positive Rate	0%	2.31%

Table 5 Confusion Matrix (Contamination = 0.3)

Metric	Value
True Negatives (TN)	7,268
False Positives (FP)	172
False Negatives (FN)	1,132
True Positives (TP)	3,428

3.2.Discussion

The experimental results highlight the behaviour and performance trade-offs of the anomaly detection model under different configurations. When the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3298-3302
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0422

contamination parameter was set to 0.1, the model achieved very high precision and zero false positive rate. This indicates that all detected attacks were correct, and no normal requests were misclassified. However, the recall was significantly low, meaning that a large number of attack instances were not detected. This shows that the model was overly conservative and only identified the most obvious anomalies.

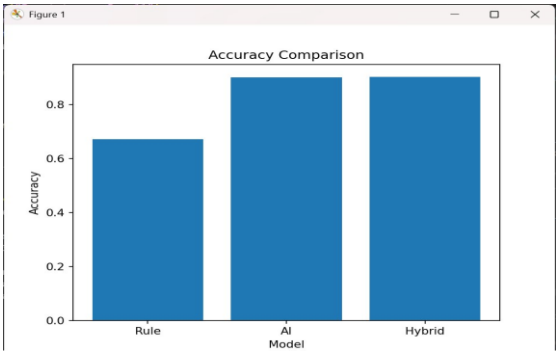

Figure 1 Accuracy Comparison Across Rule-Based, AI-Based, And Hybrid Models

When the contamination value was increased to 0.3, the model showed a significant improvement in recall, detecting a much larger portion of attack requests. Although this led to a slight increase in the false positive rate, the overall performance improved considerably, as reflected by higher accuracy and F1 score. This demonstrates that tuning the contamination parameter plays a crucial role in balancing detection capability and false alarms.

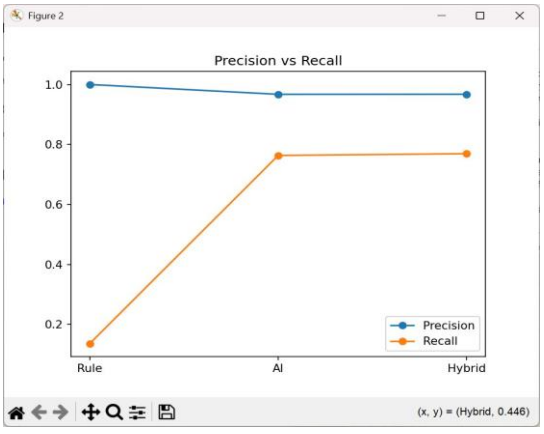

Figure 2 Precision vs Recall Comparison Across Rule-Based, AI-Based, and Hybrid Models.

The results clearly show the inherent trade-off between precision and recall in anomaly detection systems. A lower contamination value favors precision but misses many attacks, while a higher value improves detection coverage at the cost of a few false positives. This trade-off is especially important in real-time API security, where both accuracy and responsiveness are critical.

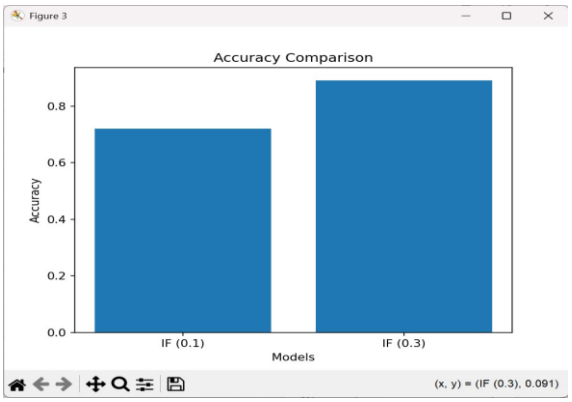

Figure 3 Accuracy Comparison Between Isolation Forest with Contamination 0.1 and 0.3.

The hybrid architecture further strengthens the system by combining rule-based detection with AI based anomaly detection. The rule-based layer efficiently handles known attack patterns with minimal delay, while the AI model detects more complex and previously unseen threats. This combination helps overcome the limitations of using either approach individually. Overall, the proposed hybrid system demonstrates that a hybrid approach can achieve a better balance between detection accuracy, false positives, and real-time performance. These findings confirm that integrating rule-based and AI-based methods is an effective strategy for securing modern API-driven applications.

Conclusion

This paper addressed the problem of detecting API abuse and data exposure, especially in cases where traditional security systems fail to identify unknown or zero-day attacks. Existing rule-based approaches are limited to known attack patterns, while standalone machine learning models often suffer from false positives and performance issues in real-time environments. To overcome these limitations, the proposed hybrid system combines rule-based detection with AI-based anomaly detection using the

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 05 May 2026
Page No: 3298-3302
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0422

Isolation Forest algorithm. The results confirm that this hybrid approach improves detection performance by balancing precision and recall. The experimental analysis also showed that parameter tuning, particularly the contamination value, plays an important role in improving the effectiveness of anomaly detection. The system was able to detect both known and unknown attacks more efficiently compared to individual methods, while maintaining acceptable false positive rates. This makes it suitable for real-time API security in modern cloud-based applications. Overall, the study confirms that integrating rule-based and AI-based techniques provides a more reliable and scalable solution for API protection. In future work, the system can be further improved by incorporating adaptive learning techniques and testing with real-world datasets to enhance robustness and practical applicability.

Acknowledgements

We would like to express our sincere gratitude to our guide and faculty members of the Department of Information Technology, Kamaraj College of Engineering and Technology, for their continuous support and guidance throughout this work. We also thank our institution for providing the necessary resources and environment to carry out this project successfully. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1]. Kaul, D., & Khurana, R. (2021). AI to detect and mitigate security vulnerabilities in APIs: Encryption, authentication, and anomaly detection in enterprise-level distributed systems. Eigenpub Review of Science and Technology, 5(1), 34–62.

[2]. Zoppi, T., Ceccarelli, A., & Bondavalli, A. (2021). Unsupervised algorithms to detect zero-day attacks: Strategy and application. IEEE Access, 9, 90603–90615. Doi: 10.1109/ACCESS.2021.3090957.

[3]. Farzaan, M. A. M., Ghanem, M. C., El-Hajjar, A., & Ratnayake, D. N. (2025). AI-powered system for an efficient and effective cyber incidents detection and response in cloud environments. IEEE Transactions on Machine Learning in Communications and Networking.Doi:10.1109/TMLCN.2025.3564912.

[4]. Pu, G., Wang, L., Shen, J., & Dong, F. (2021). A hybrid unsupervised clustering-based anomaly detection method. Tsinghua Science and Technology, 26(2), 146–153.

[5]. Chatera, M., Borgi, A., Slama, M. T., Sfar-Gandoura, K., & Landoulsi, M. I. (2022). Fuzzy isolation forest for anomaly detection. Procedia Computer Science, 207, 916–925.Doi: 10.1016/j.procs.2022.09.147.

[6]. Kovačević, A., Putnik, N., & Tošković, O. (2020). Factors related to cyber security behavior. IEEE Access, 8, 125140–125151. Doi: 10.1109/ACCESS.2020.3007867.

[7]. Sharma, R., & Grover, J. (2024). Enhancing anomaly detection in cybersecurity using hybrid machine learning approaches.

[8]. Cauchideesan, P., & Poravj, K. (2023). API rate limiting and latency trade-offs in real-time systems.

[9]. Pawana, S., et al. (2024). Generative AI for cybersecurity dataset generation and anomaly detection.

[10]. Huang, X., et al. (2023). Hybrid intrusion detection systems combining rule-based and machine learning approaches.