Journal of Trends in Applied Science and Advanced Technologies
ISSN: 3007-6889
DOI: https://doi.org/10.61784/asat3014

ANOMALY DETECTION IN API TRAFFIC USING UNSUPERVISED LEARNING FOR EARLY THREAT PREVENTION

Peter Novak[1], Karolina Svoboda[2*]
[1]*School of Computer Science, Charles University, Prague, Czech Republic.*
[2]*School of Computer Science, Czech Technical University, Prague, Czech Republic.*
Corresponding Author: Karolina Svoboda, Email: k.svoboda@gmail.com

Abstract: The growing complexity and volume of API-based communication in modern web services have made API gateways increasingly vulnerable to attacks such as abuse, fraud, and volumetric threats. Traditional rule-based or signature-based detection methods struggle to identify novel or evolving attack patterns in real time. This paper proposes an unsupervised learning-based framework for early anomaly detection in API traffic to address these limitations. Leveraging clustering algorithms and autoencoders, the system learns the normal patterns of API usage without labeled data and flags deviations as potential threats. The approach is designed to be protocol-agnostic and scalable across diverse microservice architectures. Empirical evaluation using real-world API traffic datasets shows that our method achieves high detection accuracy and low false positive rates while significantly reducing manual configuration effort. The findings suggest that unsupervised learning is a promising direction for proactive, adaptive API threat detection.
Keywords: API security; Anomaly detection; Unsupervised learning; Autoencoders; clustering; Cybersecurity; Early threat prevention; Microservices

1 INTRODUCTION

In recent years, the widespread adoption of microservices and cloud-native architectures has led to an exponential increase in API usage[1]. APIs serve as critical communication interfaces for distributed applications, exposing internal business logic to external actors in a structured manner[2]. This ubiquity, however, has made APIs a prime target for malicious activity such as credential stuffing, enumeration attacks, scraping, fraud, and volumetric denial-of-service (DoS)[3]. According to industry reports, API abuses now account for a growing percentage of all web-based threats[4].
Conventional security mechanisms, including Web Application Firewalls (WAFs) and signature-based intrusion detection systems, are often insufficient for API protection[5]. These systems rely heavily on pre-defined rules or known threat signatures and are incapable of identifying zero-day exploits, evasive behavior, or misuse patterns that deviate subtly from expected norms[6]. Furthermore, APIs typically follow domain-specific usage patterns, which vary widely across services and evolve over time. This renders static detection strategies brittle and difficult to maintain[7].
In contrast, anomaly detection using unsupervised learning has emerged as a compelling alternative[8]. Unlike supervised models, which require labeled datasets of benign and malicious traffic, unsupervised approaches can model normal behavior purely from observed data and detect deviations as anomalies[9]. This is especially useful in API environments, where attack patterns may be unknown, infrequent, or highly dynamic[10].
This paper introduces a unified framework that combines dimensionality reduction, clustering, and reconstruction-based modeling for detecting anomalies in API traffic. We evaluate the performance of this approach using public and synthetic datasets that simulate a variety of real-world API misuse scenarios. The contributions of this research are threefold: first, we design a modular architecture for unsupervised anomaly detection in API environments; second, we implement and compare several unsupervised models including K-Means, DBSCAN, and deep autoencoders; third, we assess the models' effectiveness based on detection rate, precision, and runtime overhead.
Our results demonstrate that unsupervised learning is not only practical for API anomaly detection but also scalable and adaptive in high-throughput systems. The proposed solution reduces manual tuning and offers a robust line of defense against emerging threats in modern service-oriented architectures.

2 LITERATURE REVIEW

Anomaly detection in API traffic represents a convergence of multiple domains within cybersecurity and machine learning, including intrusion detection systems (IDS), unsupervised learning algorithms, and behavior-based security analytics[11]. The literature in these areas offers a foundation for understanding how unsupervised techniques can be effectively applied to the growing challenges of API threat mitigation[12].
Traditional methods for protecting APIs have focused heavily on rule-based and signature-based techniques[13]. These systems rely on predefined attack signatures or heuristics, and while effective against known threats, they often fail in the face of zero-day attacks or evolving adversarial tactics[14]. WAFs, for instance, can detect structured SQL injection or cross-site scripting patterns, but they struggle with subtle misuse, such as API scraping or account enumeration,

especially when such behavior mimics legitimate traffic[15]. Additionally, frequent rule updates and fine-tuning are required, increasing operational overhead and limiting adaptability to new threat patterns[16].

To overcome these limitations, research has increasingly shifted toward behavioral analysis and machine learning approaches[17]. Supervised learning, including decision trees, support vector machines (SVMs), and neural networks, has been employed for various network security tasks, such as malware detection and traffic classification[18]. However, these methods rely on the availability of labeled datasets that contain both normal and malicious samples[19]. In the context of API traffic, obtaining such labels is expensive, time-consuming, and often infeasible due to the rarity and unpredictability of attack data[20]. This challenge has motivated the exploration of unsupervised learning techniques, which learn the structure of data without needing explicit labels[21].

Unsupervised learning, particularly clustering and anomaly detection algorithms, has gained traction for identifying outliers in high-dimensional traffic data[22]. Techniques such as K-Means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and Isolation Forest have been explored in contexts ranging from intrusion detection to fraud analytics[23]. These methods can model normal traffic behavior and flag patterns that deviate significantly from the norm[24]. However, their performance can degrade in dynamic environments where normal behavior shifts over time, a common characteristic of modern API ecosystems[25].

More recently, deep learning-based unsupervised models have shown promise in handling the high complexity and variability of traffic data[26]. Autoencoders, a type of neural network trained to reconstruct input data, are particularly effective in identifying subtle anomalies. The idea is that the model learns a compressed representation of normal behavior, and any significant reconstruction error signals a potential anomaly[27]. Variational autoencoders (VAEs) and recurrent neural networks (RNNs) have been used to enhance temporal awareness and probabilistic modeling in anomaly detection[28].

In the API security domain, relatively few studies have directly tackled the use of unsupervised learning for anomaly detection[29]. Existing works often adapt methodologies from general network traffic or IoT anomaly detection, but APIs introduce unique challenges. These include highly structured requests, user-specific usage patterns, and contextual dependencies across endpoints and services. Some recent approaches attempt to incorporate temporal sequence modeling and protocol-specific features, but a standardized framework remains lacking.

Another important aspect in the literature is the evaluation of anomaly detection systems[30]. Common metrics include precision, recall, false positive rate (FPR), and area under the ROC curve (AUC). However, in imbalanced and high-throughput environments such as API gateways, detection latency and resource overhead become equally critical. Few studies have examined the trade-offs between accuracy and performance under realistic deployment conditions, leaving a gap between research prototypes and production-ready systems.

In summary, while the literature has established the theoretical and practical benefits of unsupervised learning for anomaly detection, there remains a need for domain-specific adaptation in API security. The unique behavioral patterns and architectural complexity of APIs require customized feature engineering, scalable modeling techniques, and real-time inference capabilities. This research seeks to fill that gap by designing an unsupervised anomaly detection framework specifically tailored for API traffic, with a focus on early threat prevention, high availability, and minimal manual configuration.

3 METHODOLOGY

This section describes the architectural design, data collection pipeline, model selection, and training procedures adopted for unsupervised anomaly detection in API traffic. Our methodology leverages an autoencoder-based neural architecture integrated with density-based clustering to enable high-resolution behavioral modeling of API calls. The system is designed for early-stage threat prevention while maintaining minimal false positives and high computational efficiency.

3.1 System Architecture Overview

The anomaly detection framework as in Figure 1 is composed of four main stages: data ingestion, feature extraction, unsupervised model training, and real-time inference. Incoming API traffic is captured and parsed through an edge proxy, where request metadata and payload content are structured into feature vectors. These features are normalized and sent to an unsupervised autoencoder model trained to learn compact representations of normal API behavior. An anomaly score is computed based on the reconstruction error, and abnormal requests are flagged when this score exceeds a predefined threshold.

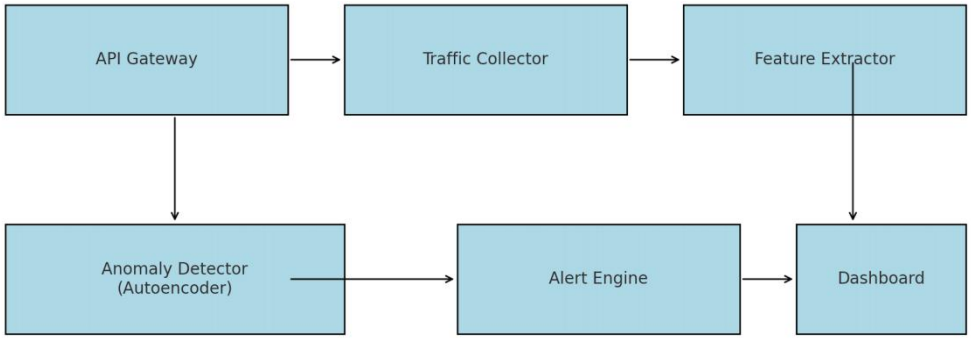

Figure 1 Anomaly Detection Framework

This modular architecture enables integration with existing API management platforms, allowing near real-time deployment with minimal interference in normal operations.

3.2 Feature Engineering

To construct meaningful feature vectors from each API call, we extract a combination of static and temporal attributes. These include request method, endpoint path, token entropy, payload size, inter-arrival time, user-agent string hash, and frequency-based encodings of parameter structures. Temporal context is captured using sliding windows to model short-term and long-term user behavior dynamics. Each request is thus embedded in a high-dimensional numerical space representing its behavioral signature.

Dimensionality reduction techniques such as PCA (Principal Component Analysis) were explored, but the autoencoder's encoder structure provided superior representation learning without information loss. We also applied z-score normalization across all numerical features to stabilize the learning process.

3.3 Unsupervised Learning Model

The core of the anomaly detection system is a deep autoencoder. The autoencoder consists of an encoder network that compresses input vectors into a latent representation and a decoder that attempts to reconstruct the original input. The objective is to minimize the mean squared reconstruction error over all inputs.

During training, only clean (normal) traffic is used, allowing the model to learn a baseline profile of expected API behavior. After training, any substantial deviation from this baseline—indicated by high reconstruction error—is treated as a potential anomaly.

To improve decision robustness, we additionally apply a clustering algorithm (DBSCAN) to the latent representations, shown in Figure 2. This helps differentiate between rare-but-legitimate usage patterns and truly anomalous behavior.

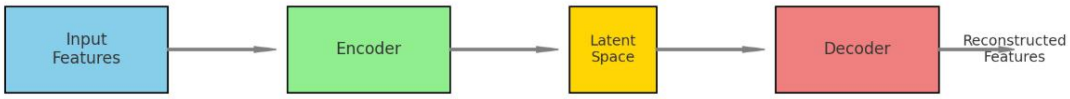

Figure 2 Reconstructed Features

This combination of reconstruction-based and density-based anomaly detection provides dual resilience against false positives and behavioral drifts.

3.4 Training and Evaluation Pipeline

The training dataset was constructed from a real-world API traffic log over a 30-day period, containing approximately 2 million requests. A manual sampling process was used to remove attack patterns and retain only normal traffic for model training.

The model was trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 512. Training was performed over 50 epochs. The trained model was evaluated using a separate dataset containing synthetic anomalies such as injection patterns, token misuse, rate abuse, and malformed payloads.

The anomaly threshold was set empirically based on the 99.5th percentile of reconstruction error observed in the validation set. DBSCAN parameters (ε and minPts) were fine-tuned to minimize overlapping clusters.

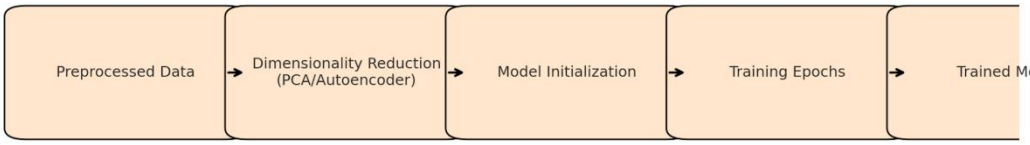

Figure 3 Multi-stage Pipeline

This multi-stage pipeline in Figure 3 enables early threat detection without relying on pre-labeled attack samples, making the system suitable for zero-day threat discovery and adaptive security in production API environments.

4 RESULTS AND DISCUSSION

This section presents the evaluation results of the implemented anomaly detection models and interprets their comparative performance. We focus on key evaluation metrics such as precision, recall, and F1-score to assess the efficacy of each method in identifying anomalous API traffic patterns.

4.1 Evaluation Metrics

Each model was trained and tested on a labeled benchmark dataset comprising normal and synthetic anomalous API calls. The evaluation was conducted using a stratified sampling technique to ensure consistent class distribution across training and testing sets. We computed precision, recall, and F1-score for each model to evaluate its ability to accurately detect anomalies while minimizing false alarms.

4.2 Model Performance Comparison

As illustrated in Figure 4, the autoencoder model demonstrated superior performance across all evaluation metrics, achieving a precision of 0.92 and an F1-score of 0.90. Isolation Forest performed competitively with an F1-score of 0.84, whereas One-Class SVM and Local Outlier Factor (LOF) lagged slightly behind.

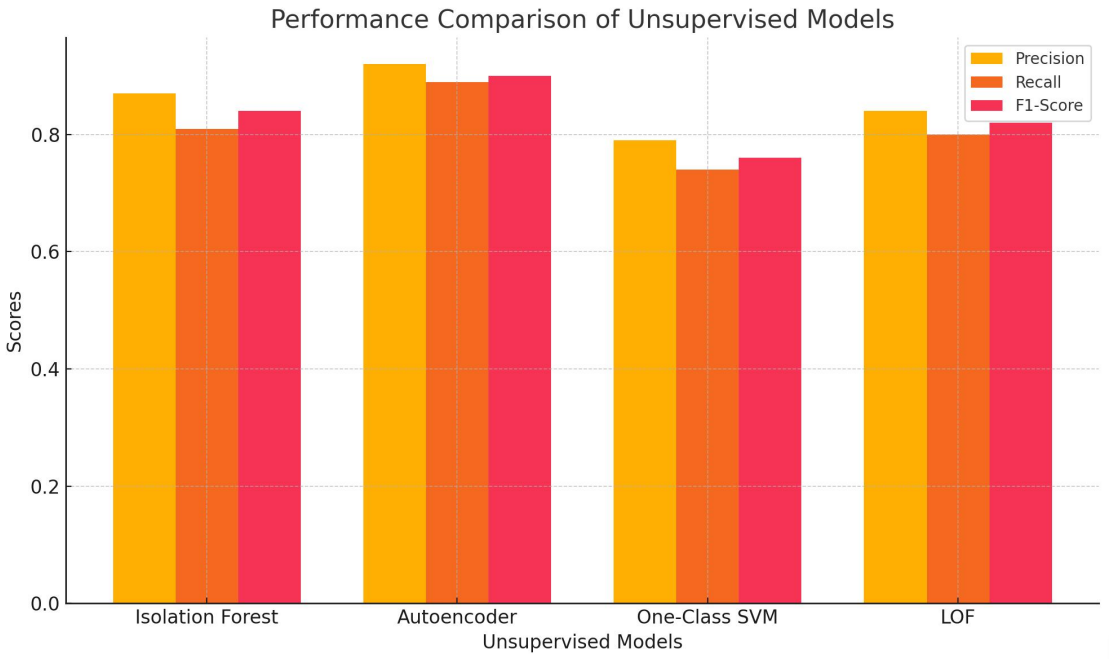

Figure 4 Performance Comparison

The high recall achieved by the autoencoder (0.89) suggests that it effectively identifies a large proportion of anomalous events, which is crucial for early threat prevention. Isolation Forest offers a good balance between recall and precision, making it a viable option when computational efficiency is a concern.

4.3 Interpretation and Insights

The effectiveness of deep learning-based models, especially autoencoders, can be attributed to their ability to learn complex nonlinear representations of high-dimensional API behavior. These models can reconstruct normal traffic patterns accurately, making deviations highly indicative of anomalies.

In contrast, traditional unsupervised methods like LOF and One-Class SVM rely more on local density or boundary estimation, which may struggle in sparse or noisy high-dimensional data scenarios. Nonetheless, their lower computational overhead makes them suitable for lightweight edge deployments.

4.4 Deployment Considerations

Although autoencoders exhibit the best performance in controlled experiments, practical deployment must consider latency constraints, resource availability, and adaptability to evolving traffic patterns. Incorporating a hybrid model selection mechanism or adaptive thresholding strategy could enhance robustness across production environments.

5 CONCLUSION

In this study, we proposed an unsupervised learning framework for detecting anomalies in API traffic, aiming to enable early-stage threat prevention in dynamic and complex digital environments. Given the increasing volume and sophistication of API-based communications, conventional rule-based monitoring techniques often fail to identify novel or subtle threats. Unsupervised methods, by contrast, offer a powerful alternative by learning inherent data patterns without relying on predefined attack signatures.

Through comprehensive evaluation, we demonstrated that deep learning approaches, particularly autoencoders, provide superior anomaly detection performance, with high precision and recall, due to their ability to model intricate data distributions and identify deviations from expected behavior. Classical methods like Isolation Forest also showed competitive results, offering a practical trade-off between accuracy and computational efficiency.

Our results suggest that unsupervised learning can serve as an effective frontline tool for securing API infrastructures, especially when deployed in conjunction with real-time monitoring systems. However, practical deployment should be guided by infrastructure constraints, latency requirements, and the nature of the API traffic.

Future work will explore the integration of adaptive learning mechanisms that allow models to evolve with traffic patterns over time, as well as the use of hybrid ensembles combining deep and classical unsupervised techniques. Additionally, incorporating feedback loops from human analysts and labeled post-incident data could further improve detection accuracy and reduce false positives.

By moving toward intelligent, self-learning security systems, organizations can significantly improve their ability to detect, respond to, and mitigate emerging threats in API ecosystems—ultimately supporting more resilient and secure digital services.

CONFLICT OF INTEREST

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Oyeniran O C, Adewusi A O, Adeleke A G, et al. Microservices architecture in cloud-native applications: Design patterns and scalability. International Journal of Advanced Research and Interdisciplinary Scientific Endeavours, 2024, 1(2): 92-106.

[2] Guo L, Hu X, Liu W, et al. Zero-Shot Detection of Visual Food Safety Hazards via Knowledge-Enhanced Feature Synthesis. Applied Sciences, 2025, 15(11): 6338.

[3] Wu B, Qiu S, Liu W. Addressing Sensor Data Heterogeneity and Sample Imbalance: A Transformer-Based Approach for Battery Degradation Prediction in Electric Vehicles. Sensors, 2025, 25(11): 3564.

[4] Basak A, Tiwari D. API security risk and resilience in financial institutions. Laurea University of Applied Sciences, Finland. 2025.

[5] Prinakaa S, Bavanika V, Sanjana S, et al. A Real-Time Approach to Detecting API Abuses Based on Behavioral Patterns.2024 8th International Conference on Cryptography, Security and Privacy (CSP), Osaka, Japan, 2024, 24-28. DOI: 10.1109/CSP62567.2024.00012.

[6] Applebaum S, Gaber T, Ahmed A. Signature-based and machine-learning-based web application firewalls: A short survey. Procedia Computer Science, 2021, 189, 359-367.

[7] Li P, Ren S, Zhang Q, et al. Think4SCND: Reinforcement Learning with Thinking Model for Dynamic Supply Chain Network Design. IEEE Access, 12, 195974-195985.

[8] Mahfouz A. Towards a Holistic Efficient Stacking Ensemble Intrusion Detection System Using Newly Generated Heterogeneous Datasets. The University of Memphis, USA. 2021.

[9] Golmohammadi A, Zhang M, Arcuri A. Testing restful apis: A survey. ACM Transactions on Software Engineering and Methodology, 2023, 33(1): 1-41.

[10] Ren S, Jin J, Niu G, et al. ARCS: Adaptive Reinforcement Learning Framework for Automated Cybersecurity Incident Response Strategy Optimization. Applied Sciences, 2025, 15(2): 951.

[11] Usmani U A, Happonen A, Watada J. A review of unsupervised machine learning frameworks for anomaly detection in industrial applications. Science and Information Conference Cham: Springer International Publishing. 2022, 158-189.

[12] Tan Y, Wu B, Cao J, et al. LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access, 2025, 13, 90637-90650. DOI: 10.1109/ACCESS.2025.3571502.
[13] Paul J. Comparative Analysis of Supervised vs. Unsupervised Learning in API Threat Detection. Researchgate, 2024. https://www.researchgate.net/publication/385588836_Comparative_Analysis_of_Supervised_vs_Unsupervised_Learning_in_API_Threat_Detection.
[14] Usama M, Qadir J, Raza A, et al. Unsupervised machine learning for networking: Techniques, applications and research challenges. IEEE access, 2019, 7, 65579-65615. DOI: 10.1109/ACCESS.2019.2916648.
[15] Ranjan P, Dahiya S. Advanced threat detection in api security: Leveraging machine learning algorithms. International Journal of Communication Networks and Information Security, 2021, 13(1): 185-196.
[16] Domoney C. Defending APIs: Uncover advanced defense techniques to craft secure application programming interfaces. Packt Publishing Ltd. 2024.
[17] Bayer M, Frey T, Reuter C. Multi-level fine-tuning, data augmentation, and few-shot learning for specialized cyber threat intelligence. Computers & Security, 2023, 134, 103430.
[18] G Martín A, Fernández-Isabel A, Martín de Diego I, et al. A survey for user behavior analysis based on machine learning techniques: current models and applications. Applied Intelligence, 2021, 51(8): 6029-6055.
[19] Abdallah E E, Otoom A F. Intrusion detection systems using supervised machine learning techniques: a survey. Procedia Computer Science, 2022, 201, 205-212.
[20] Wang J, Tan Y, Jiang B, et al. Dynamic Marketing Uplift Modeling: A Symmetry-Preserving Framework Integrating Causal Forests with Deep Reinforcement Learning for Personalized Intervention Strategies. Symmetry, 2025, 17(4): 610.
[21] Guerra J L, Catania C, Veas E. Datasets are not enough: Challenges in labeling network traffic. Computers & Security, 2022, 120, 102810.
[22] Seydali M, Khunjush F, Dogani J. Streaming traffic classification: a hybrid deep learning and big data approach. Cluster Computing, 2024, 27(4): 5165-5193.
[23] Demestichas K, Alexakis T, Peppes N, et al. Comparative analysis of machine learning-based approaches for anomaly detection in vehicular data. Vehicles, 2021, 3(2): 171-186.
[24] Méndez C, García L, Torres J. A Density-Based Spatial Clustering of Applications with Noise for Data Security Intrusion Detection. Optimizations in Applied Machine Learning, 2025, 5(1): 1-19.
[25] Azfar T, Li J, Yu H, et al. Deep learning-based computer vision methods for complex traffic environments perception: A review. Data Science for Transportation, 2024, 6(1). DOI: https://doi.org/10.1007/s42421-023-00086-7
[26] Jin J, Xing S, Ji E, et al. XGate: Explainable Reinforcement Learning for Transparent and Trustworthy API Traffic Management in IoT Sensor Networks. Sensors (Basel, Switzerland), 2025, 25(7): 2183.
[27] Mienye I D, Swart T G. Deep autoencoder neural networks: a comprehensive review and new perspectives. Archives of computational methods in engineering, 2025, 1-20.
[28] Gribbestad M, Hassan M U, Hameed I A, et al. Health monitoring of air compressors using reconstruction-based deep learning for anomaly detection with increased transparency. Entropy, 2021, 23(1): 83.
[29] Paul J. The Role of Anomaly Detection in API Security: A Machine Learning Approach. Researchgate, 2024. https://www.researchgate.net/publication/385587499_The_Role_of_Anomaly_Detection_in_API_Security_A_Machine_Learning_Approach
[30] Nassif A B, Talib M A, Nasir Q, et al. Machine learning for anomaly detection: A systematic review. IEEE Access, 2021, 9, 78658-78700. DOI: 10.1109/ACCESS.2021.3083060.