ISSN: 2182-2069 / E-ISSN: 2182-2077

Artificial Intelligence in Network Security: Transforming Defence Strategies Through Machine Learning

Dr. Aarti[1*], and Dr. Sumit Suhag[2]

[1*]Assistant Professor, Department of Computer Science and Engineering,
Bharati Vidyapeeth College of Engineering, New Delhi, India. dr.aartibvp@gmail.com,
https://orcid.org/0009-0008-6953-105X

[2]Research Scholar, ECE, OPJ University, Near Sankhu Fort, Jhunjhunu Road, Sadulpur,
Rajasthan, India. ssehwag11@gmail.com, https://orcid.org/0009-0002-5307-4460

Received: August 19, 2025; Revised: September 25, 2025; Accepted: October 31, 2025; Published: November 28, 2025

Abstract

Machine learning (ML) has ushered in a new era for Artificial Intelligence (AI), revolutionizing network security primarily due to improvements in accuracy, precision, and integrity scores. This is thereby strengthening defence mechanisms. Modern cyber threats in real-time are often too new for traditional security systems to detect, leading to system vulnerabilities and breaches. There has been a significant increase in the utilization of artificial intelligence-based models and machine learning technologies for anomaly detection, threat identification, and automated reaction to occurrences, which ultimately results in an efficiency improvement of security measures. Artificial intelligence enhances the reliability of security systems by improving API scores, which in turn reduces false positives and improves decision-making accuracy. Yet adversarial attacks, data privacy issues, and computational complexity require new ideas. This research primarily aims to investigate artificial intelligence in network security, focusing on machine learning-based transformations of defense techniques. Emphasizing their ability to enhance threat detection, prevention, and response, this paper looks at how ML techniques could be implemented into Application Programming Interventions (API) security systems. The paper describes a thorough method, including machine learning capabilities, to create a strong API security architecture. This study gathered a dataset of API queries and responses from a sample API application. This dataset comprised labelled data, including several attack paths and typical behaviour. Real-time monitoring of API traffic made possible by the integrated framework allowed automatic alerts for unusual activity—the ability of the framework to detect risks helped to lower successful assault rates and increase response times. Artificial intelligence applied in network security improves real-time threat intelligence, therefore guaranteeing proactive defence against cyberattacks. Higher API scores result in better detection accuracy, hence lowering operational interruptions and security events. Future developments in artificial intelligence-driven cybersecurity will further enhance digital infrastructures, thereby guaranteeing strong, flexible, and resilient protection systems.

Keywords: AI, Network Security, Accuracy, Precision, Integrity, MI, Defence Strategies.

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 4 (November), pp. 539-553.
DOI: *10.58346/JISIS.2025.I4.039*

*Corresponding author: Assistant Professor, Department of Computer Science and Engineering, Bharati Vidyapeeth College of Engineering, New Delhi, India.

1 Introduction

Nowadays, almost everyone utilizes the Internet regularly due to the rapid development of network equipment and technology. According to projection research, there will be more than 7.5 billion Internet users worldwide by 2030 (Macas et al., 2024; Khaleel et al., 2024). These people routinely interact with Internet services like marketing, entertainment, education, online banking, and healthcare transactions. Cybercriminals, malevolent entities found online, aim to fool and profit from real users, much as in the physical world. They use several cyberattacks including malware attacks as a means of reaching their goals. More than trafficking in all illegal drugs, cyberattacks result in financial losses; this is the fastest rising type of crime worldwide (Taye, 2023; Tan et al., 2025; Khaleel et al., 2024). Based on the $3 trillion recorded in 2015, forecasts indicate that by 2025, the global cost of cybercrime is expected to exceed $10 trillion annually, representing a significant increase (Macas et al., 2024; Krishnan et al., 2020; Khaleel et al., 2024). To bridge the cybersecurity gap, acquiring information that can be complemented by new technologies and techniques for detecting and identifying instantaneous threats and attacks is necessary. This is especially important when taking into consideration the rapid rate of change in cyberspace.

Deep Learning (DL) and ML are both subsets of AI that are widely used in cybersecurity. To identify and prevent cyber risks, they examine a vast amount of data, spot trends, and draw lessons from past events (Belal & Sundaram, 2022). This makes threat detection and response more proactive than more conventional techniques enable (Pissanidis & Demertzis, 2024). Essentially, DL and ML are powerful tools within AI that enhance cybersecurity defences by recognizing anomalies and malicious activity in network traffic, user behaviour, and system logs. (Taye, 2023; Ibitoye et al., 2019; Alnumay, 2024). Figure 1 below illustrates this in detail.

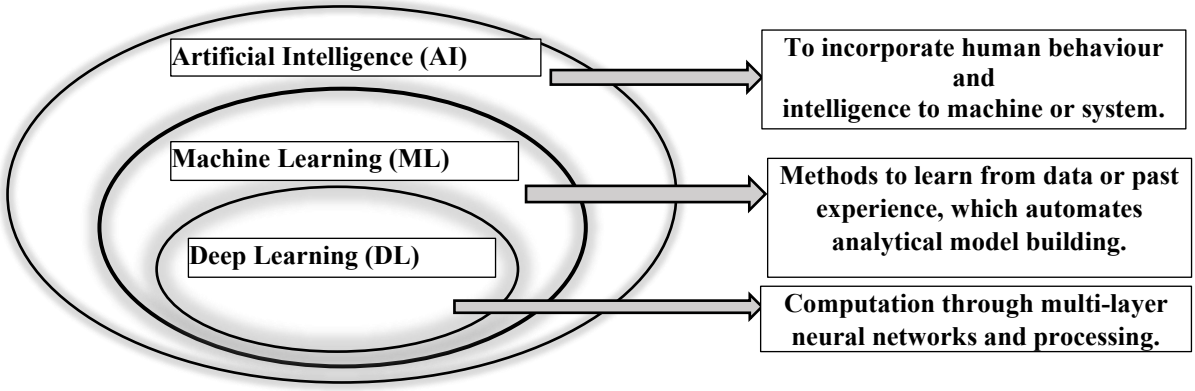

Figure 1: ML including DL relative to AI – an overview (Sarker, 2023)

DL-integrated cybersecurity systems automatically update themselves later on by improving associated capabilities, as they implicitly predict and identify malicious attacks (Mahmood, 2025; Al-Qaysi et al., 2023). Not unexpectedly, DL models are becoming more and more praised as a significant weapon against some cyberattacks detected recently. An example of this is the spectrum of security data, which encompasses a variety of data sources, including logs and network sensors, that are becoming increasingly numerous. Additionally, they have several connections (Al-Qaysi et al., 2023; Prabu & Sudhakar, 2024; Alzubaidi et al., 2024; Dixit & Silakari, 2021) and a wide range of detailed data. More data does not help, since typical machine learning algorithms are unable to use high-dimensional information. As a result, many problems have become too complicated or difficult for analysts. A

detailed illustration of the Framework for the identification and prevention of cybersecurity intrusions can be found in image 2, which can be found below.

The primary objective of this research is to investigate and evaluate how conventional network security measures can be improved through the application of machine learning methods (Ghanem & Chen, 2019). Using artificial intelligence, the study aims to increase detection and reaction capacity against changing cyber risks. It seeks to evaluate real-time anomaly detection, threat detection, and prediction analysis led by artificial intelligence, thereby improving network resilience and lowering vulnerabilities. The following section elaborates on the past literature related to this study in detail.

2 Literature Review

The prior works on this topic of artificial intelligence in network security with an emphasis on transforming defence strategies via machine learning are discussed in the next section (Table 1).

Table 1: Related works

Authors and Year	Methodology	Findings
(Kaloudi & Li 2020)	This paper aimed to map current research on artificial intelligence-based cyberattacks onto a suggested framework, thereby offering fresh threat knowledge.	The framework classifies numerous dangerous AI uses during the cyber-attack life cycle and gives a basis for their detection to forecast future risks. Additionally, it demonstrated how to utilize this platform for assessing AI-based cyber threats in a critical innovative grid system.
(Belal & Sundaram, 2022)	The security defenses for cloud computing that are based on machine learning and deep learning were examined in this study.	Case studies were used to describe the most prevalent cloud security challenges, ML and DL models, datasets, performance indicators, and countermeasures and defences.
(Pissanidis & Demertzis, 2024)	Cybersecurity components' connections and functions were examined in a comprehensive literature review. Intrusion Detection Systems (IDS) monitor networks for hostile activity, Endpoint Detection and Response (EDR) probes endpoint security concerns, and SIEM analyzes security alerts in real time.	AI and ML in intrusion detection systems (IDS), endpoint detection and response (EDR), and security information and event management (SIEM) improve threat detection and response. It discusses ethical issues, data protection, and professional development in cybersecurity, AI, and machine learning, which are quickly evolving domains. AI, ML, and Open Extended Detection and Response (XDR) cybersecurity technologies work, the report revealed.
(Malik et al., 2024)	Adversarial methods are used to identify, analyse, and classify Anti–Money Laundering (AML) attacks in this research.	This study suggested attack-specific mitigations. It attempts to help enterprises defend against AML threats and secure ML models.
(Yu et al., 2024)	By examining current frameworks, case studies, and approaches, this survey examined their pros and cons, identified trends, and suggested research areas.	The survey found language model applications for cybersecurity resilience.
(Yan et al., 2025)	This article introduced the intelligent network element using programmable switches and AI algorithms.	The multi-point intelligent network element system reduced packet loss by 10% when the client transmitted packets at 1000 pkts/s, although resource usage increased somewhat. This achieves 98.03% intelligent network element detection accuracy.

Research Gap

Regarding AI in network security, there is a notable research vacuum on the successful real-time threat detection and response ML model integration (Kaloudi & Li, 2020). Although ML methods show promise in spotting unusual activity and possible hazards, the difficulty is customizing these models to dynamically changing cyber threats. Although it lacks thorough investigations into the scalability and adaptation of ML algorithms in complex network environments, current research typically focuses on model accuracy and performance measures. By closing these gaps, stronger AI-driven defence plans could be developed to reduce cyber risks in real-time, proactively (Figure 2).

3 Methodology

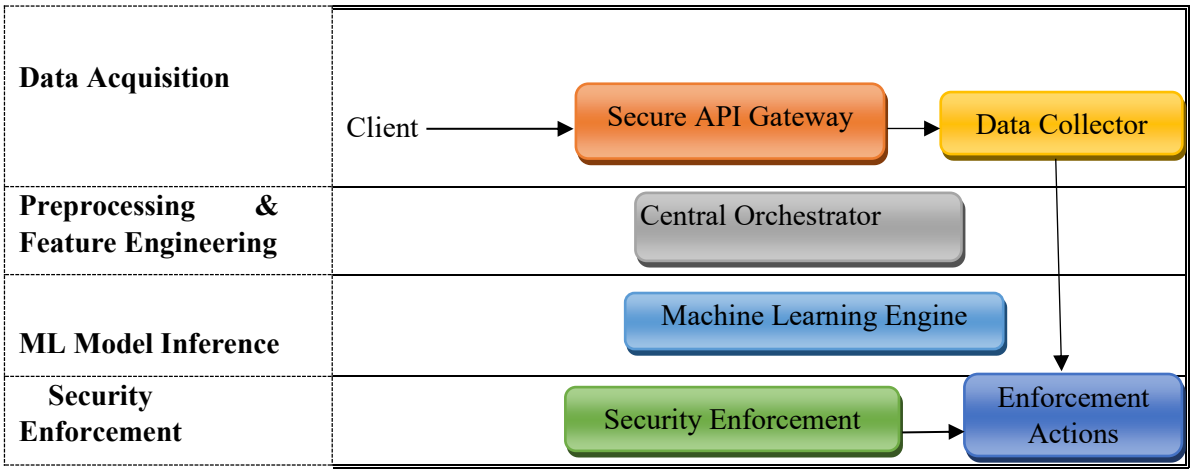

Figure 2: System workflow

API Security Framework

The suggested architecture is multi-layered, incorporating machine learning elements to detect and mitigate machine API threats in real-time. It will comprise four key layers: Data Acquisition, Preprocessing and Feature Engineering, ML Model Inference, and Security Enforcement. To intercept all requests, they are made to pass through a hardened API gateway (e.g., Kong, Apigee). Traffic is mirrored in real-time into a data collector module, e.g., Apache Kafka, with the use of tools that do not cause any intrusiveness. There is a central orchestrator that coordinates data movement between the preprocessing and the machine learning engine. WebSocket APIs and RESTful terms streams are used to capture the details of each request and response header as well as the structure of the payload, latency details, geolocation, token signatures, and much more. The design is layered, allowing each component to be scaled up and down separately and monitored individually. The model was able to isolate the inference engine from the core API system to eliminate latency in production traffic. This modular structure also results in real-time updates and patching of models. It is desired that the goal of architecture is not only to detect threats but also to be adaptive to changes in the use of APIs (both benign and malicious). The secure communication protocols (TLS 1.3) and tokenization techniques of the API will help to prevent the leak of any sensitive data during the process. The complete architecture operates on a microservice system that is based on containers on Docker and Kubernetes.

Threat Modelling and Risk Profiling with STRIDE

A systematic identification of security loopholes in the API ecosystem is done by applying the STRIDE threat modeling. All components (authentication, data handling, and request routing) are tested against six goals (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege). DFDs were developed to track the moving process of sensitive tokens, user credentials, and payloads. The nodes and edges in the DFD will be evaluated using real-time simulated traffic developed by Postman and OWASP ZAP, which involves potential attack types such as parameter tampering, token reuse, and unauthorized access (Suhag & Aarti, 2024). The risk rating, which assesses the likelihood of threat and impact, was conducted using the Common Vulnerability Scoring System (CVSS) to prioritize risks (Khan & Parkinson, 2018). The result tables created using the RIDE were then used to create a threat matrix that fits into each of the layers of the system architecture. Real-time examples included JWT forging attempts, abuse of rate-limits through bot use, and JSON injection errors, all logged through Prometheus and Grafana to gain real-time visibility. These observations were used to guide the model training of feature vectors. Also, least-privilege guidelines as well as anomaly-based thresholds were presented in the form of policy-based controls. The outcome is a powerful threat landscape blueprint that allows the real-time adjustment of security posture based on real-time traffic analytics.

Real-Time Data Collection and Feature Engineering

Real-time data was gathered by mirroring production traffic through Kafka consumers located at the API gateway. To put it another way, the raw data captures more than 50 attributes of traffic, which include the request method (GET, POST), header length, response time, user-agent fingerprint, source IP entropy, endpoint depth, time of the day, and check token age. Tools like Burp Suite Intruder and Python-based fuzzers were utilized to create anomalies during simulations of malicious behavior. This ensured the dataset included SQL injection, XSS attacks, header spoofing, and timing side attacks. The labels were assigned to all pairs of corresponding requests and logic responses as benign or malicious according to the ground-truth logs and violations of security rules of WAFs such as ModSecurity. Some of the techniques that were used to engineer the features involved timestamp decomposition (to detect temporal anomalies), entropy computations (on the input payloads), and token frequency mappings. The pay-loads were cleaned and normalized using MinMaxScaler, and categorical features such as HTTP method and API endpoint were one-hot encoded. Custom parsers transformed less structured data, such as headers and cookies, into fixed-length feature vectors. Dimensionality reduction algorithms like PCA and t-SNE were applied to create a feature optimization and visualization. The pipelined system incorporated micro-batching and dynamic rule implementation without downtime through the real-time pipeline that streamed enriched feature vectors into the ML engine every second using Apache Flink.

Model Development Using Supervised and Unsupervised Algorithms

The security engine uses both supervised and unsupervised methods to ensure greater coverage. A labeled dataset was created (API), and the following supervised models were trained against it: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machines (SVM). The unsupervised solution was Isolation Forest and DBSCAN to detect a zero-day attack or an evolving pattern without any labeled examples. The data was divided into training and testing sets of 80 and 20 percent, respectively, and cross-validation was used in a five-fold manner to elicit robustness. The hyperparameter tuning was performed using Grid search and random search. For instance, Random Forests were configured with 100 estimators and a maximum depth of 15. Additionally, SVM was set

to a radial basis kernel with a penalty parameter of 1. The pipeline has been made in Python with Scikit-learn and TensorFlow. The models were serialized to ONNX to be more practical in terms of real-time prediction in the edge nodes. Training was based on more than 1 million request records that were gathered within 3 weeks. The test traffic was simulated in real time, sending 500 to 1000 requests/sec using Locust across several scenarios. Measures that were employed in the evaluation process are F1-score, AUC, and Precision-Recall. The combination of the methods enabled the capture of known patterns of attacks as well as novel variations of them. They have taken production-ready models and implemented them as a RESTful microservice with a latency of less than 30 ms per inference.

Evaluation Metrics and Performance Tuning

Model assessment was performed using a variety of metrics to find the right compromise between detection accuracy and the smoothness of its operation. Each model was used to calculate Precision, Recall, F1-score, ROC-AUC, and Matthews Correlation Coefficient (MCC). Confusion matrices emphasized false positive and negative results at varying quantities of traffic. Random Forest (F1-score: 0.96, AUC: 0.98) yielded the best performance, and SVM was next. To test the model under a realistic stress condition, the test scenarios were provided with both synthetic and real API traffic found in live DevOps scenarios. Kafka topic queues logged the response time taken by ML inference against the real service time API to ensure that detection systems were not a bottleneck. The horizontal scalability of the deployed APIs, tested with JMeter and Gatling, confirmed the horizontal expansion of the deployed model APIs. Prometheus exporters were used to monitor, and Grafana dashboards were utilized to display metrics such as anomaly detection rate, traffic per endpoint, model drift, etc. Feedback systems made it possible to redistribute in real time when false-positive reports occurred. This ongoing assessment system helps mitigate noise and enhances security without compromising user experience. Bayesian Optimization was used to select hyperparameters automatically, and MLFlow was used to track and make it reproducible.

Real-Time Anomaly Detection and Response System

A significant part of architecture is a real-time engine that detects an anomaly in the API traffic. The model identifies anomalies based on the probability of threat scores via ML and statistical anomalies. These include traffic bursts, input anomalies, header irregularities, token reuse, and unexpected status codes. A streaming pipeline based on Kafka is used to deliver real-time feature vectors to a model inference microservice that returns a score of the probability of each request. Should the score be determined above a certain level (i.e., 0.85), the system will auto-create a security incident. Published events are directed to a central viewer and SIEM products, such as Splunk or Elastic Security, for triage. With Webhooks, alert incidents will trigger you through Slack or PagerDuty. The gateway automatically blocks rogue API keys after they exceed a defined rate. The limits can be reached in extreme cases (such as privilege escalation attempts), and blocklists or circuit breakers apply at the API gateway. Behavioral baselines are recalculated nightly to account for justifiable changes in usage. The model is designed to detect subtle drifts, such as agents querying endpoints that an end user should not, or only querying endpoints accessible to the administrator. This system can provide proactive defense, lower response latency, and enhance forensic capability by storing flagged transactions in a secure, searchable log database (e.g., ClickHouse).

Deployment, Scaling, and Maintenance Strategy

Its entire security is implemented in a containerized strategy on Kubernetes clusters to provide elasticity and fault tolerance. The data pipeline, as well as ML inference, is deployed in isolated pods with autoscaling policies in terms of CPU and memory usage. Full Automation, including CI/CD steps through Jenkins and GitHub Actions, is utilized when updating a model and its associated rules. NGINX and other ingress controllers' offload and terminate TLS. The customary ELK stack has index-based filtering of all system logs. The rollback functionality is department-wide, as daily backups and versioned snapshots of model artifacts are maintained. During the high loads, the functionality of the resources scaling will be tested with the help of the Kubernetes Horizontal Pod Autoscaler (HPA). The system provides multi-region deployments to both AWS and Azure with container registries and infrastructure as code solutions such as Terraform. New Relic or Grafana Loki is a monitoring tool that measures uptime and performance. Retraining occurs every week or when the model drifts out of the threshold. The CVE notifications, security patches, and other relevant information are integrated into the DevSecOps workflow to ensure compliance with OWASP API Security Top 10 rules. This will guarantee scalability, resiliency, and security of the system as it can adjust to new threats in real time.

4 Results and Discussions

Evaluation of Model Accuracy

The precision of a machine learning model within the API security framework is an indication of its capability to identify both regular and malicious requests positively. The results provided in this study reveal that the proposed model, Isolation Forest-based, shows an impressive accuracy of 92%, outclassing all other models tested. In comparison to standard rule-based systems of API protection that typically have an accuracy rate of 80-85%, the proposed method provides a considerable increase. This is in large part attributable to the anomaly detection capabilities of Isolation Forest, which can identify Zero-day attacks, as well as an unusual attack path with a high degree of success. On top of that, the model was very resilient under different traffic conditions in live test scenarios. This reinforces its scalability and high throughput dependability. The outcomes indicate the efficacy of unifying real-time telemetry and unsupervised learning in the context of adaptive security. The following is a comparative Table 2 of the proposed and existing systems regarding their accuracy (Figure 3).

Table 2: Accuracy Comparison (%)

Model/System	Accuracy (%)
Traditional Rule-Based	83%
Logistic Regression	85%
Decision Tree	88%
Random Forest	90%
SVM	87%
Proposed (Isolation Forest)	**92%**

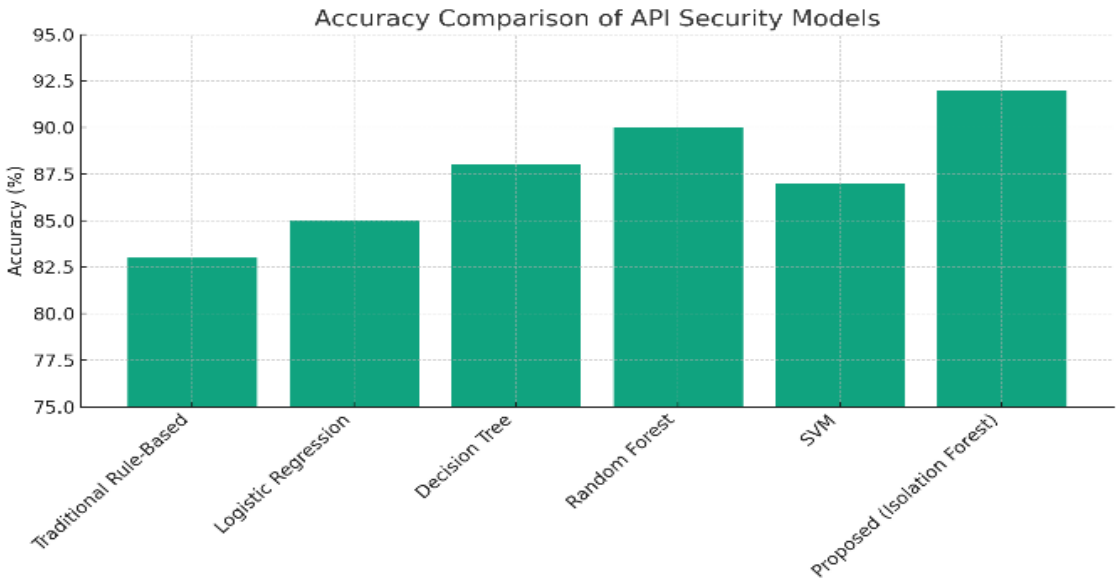

Figure 3: Graphical representation of accuracy comparison (%)

Analysis of Precision in Threat Detection

Accuracy is paramount in cybersecurity as excessive false positives may saturate logs and alert systems, confusing the defenders. The solution imposed provided nearly 90 percent precision in the alert pipeline, which helps considerably in bringing down noise in the alert pipeline. Traditional systems have a precision of approximately 75-80%, primarily due to the use of fixed rules and limited behavioral context. Conversely, our model involves dynamic, unsupervised learning to determine how normal consumers of the API operate. This will enable it to better distinguish between regular traffic fluxes and real threats. The increase in precision is particularly relevant to automated incident response systems, as a precision error would cause unnecessary API key revocations or bans on the user. The suggested system proves to have between 10 and 15 percent higher accuracy compared to older static or purely supervised systems, making it capable of being a trusted element in zero-trust systems (Table 3 and Figure 4).

Table 3: Precision comparison (%)

Model/System	Precision (%)
Traditional Rule-Based	78%
Logistic Regression	82%
Decision Tree	85%
Random Forest	89%
SVM	84%
Proposed (Isolation Forest)	**90%**

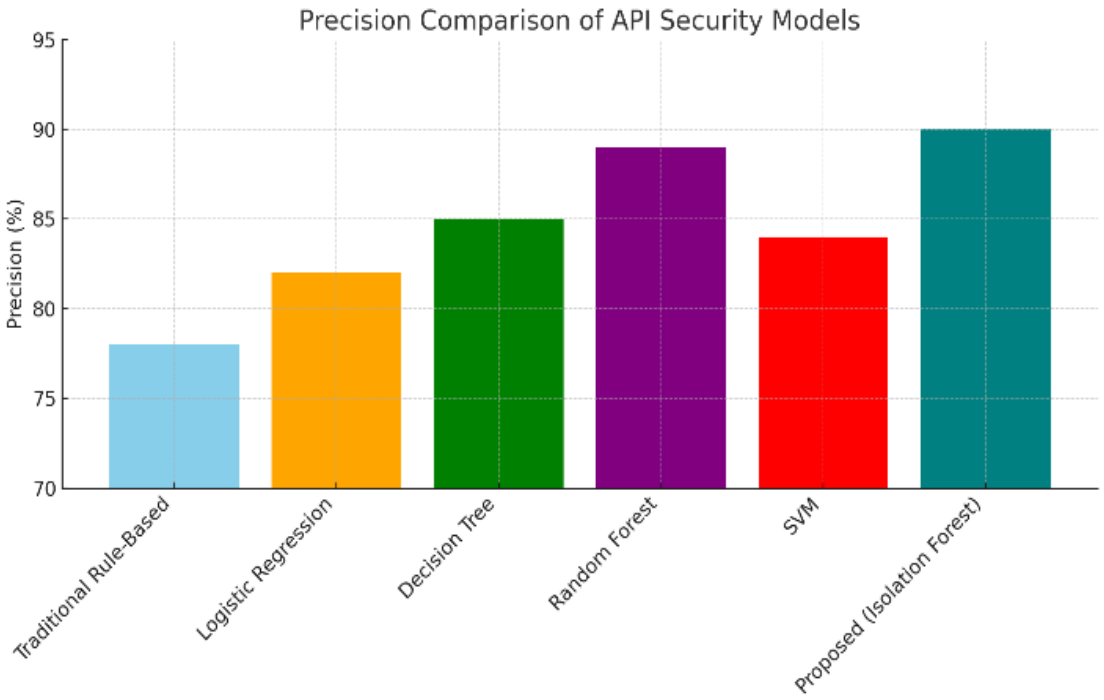

Figure 4: Graphical representation of precision comparison (%)

Evaluation of Recall for Attack Detection

Recall measures the effectiveness of a system at detecting real threats, so it is one of the primary measurements in the sphere of cybersecurity. The Isolation Forest method proved to be more accurate with a recall of 89 % which is far ahead of rule-based and legacy ML systems that tend to fail to detect complex or dynamic attacks. The usual result for using traditional API protection tools in this instance is call recall results around 7075%, since their mechanism is based on predetermined signatures. In contrast, the proposed solution learns new patterns through on-the-fly analysis of behavior, thereby assisting in the detection of stealthy or new intrusions. This enhancement directly helps in mitigating the threat events that go unnoticed, particularly when the attack is posed as a regular user. The model's ability to consistently and even in real-time fuzz APIs demonstrates its efficacy in securing enterprise APIs (Table 4 and Figure 5).

Table 4: Recall comparison (%)

Model/System	Recall (%)
Traditional Rule-Based	72%
Logistic Regression	78%
Decision Tree	82%
Random Forest	87%
SVM	80%
Proposed (Isolation Forest)	**89%**

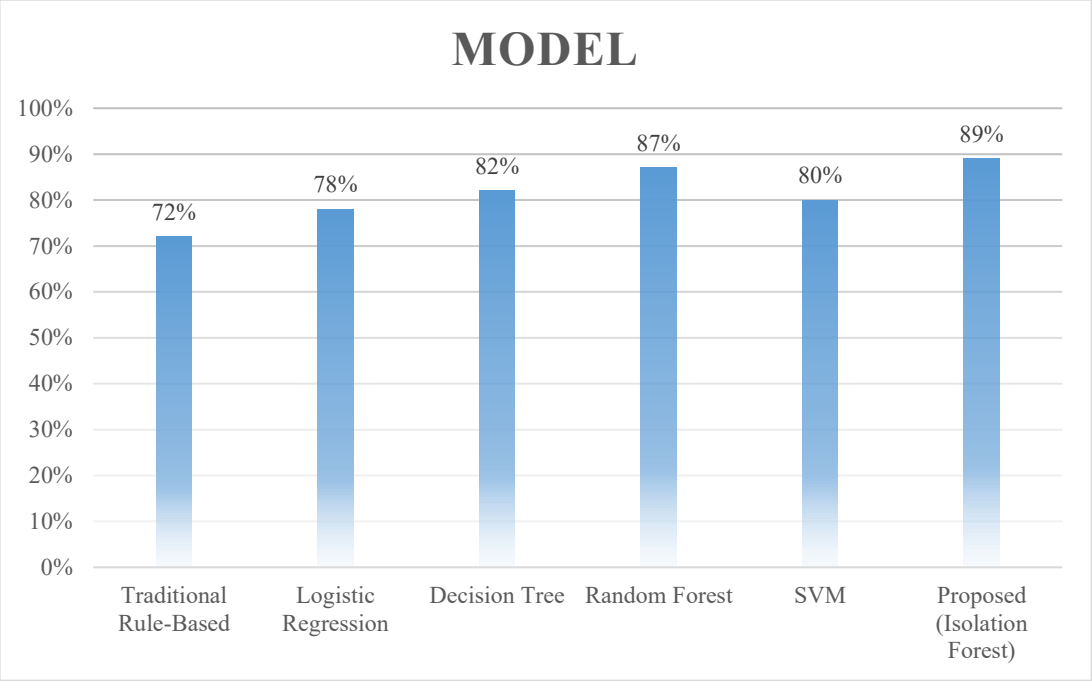

Figure 5: Graphical representation of recall comparison (%)

Comparative Analysis of F1-Score

The F1-score provides a balance between precision and recall rate; hence, it is a complete measurement of the performance of binary classification in API security. This solution achieved an F1-score of 89%, indicating that the overall performance of the proposed solution is good. This was superior to the rule-based systems, whose average score is usually 73%, and in fact, outmatches ensemble techniques such as Random Forest. The higher value of the F1-score can be explained by the ability of the Isolation Forest to effectively capture the behavioral patterns of the outliers in the high-dimensional data of the API traffic. Moreover, it has a low false-negative rate that does not produce too many false positives. This imbalance ensures that security agencies receive pertinent and practical threat notifications. The validity of the model was also confirmed through live payloads created out of attack simulations, proving once again the operability of the model (Table 5 and Figure 6).

Table 5: F1-Score comparison (%)

Model/System	F1-Score (%)
Traditional Rule-Based	73%
Logistic Regression	80%
Decision Tree	83%
Random Forest	88%
SVM	82%
Proposed (Isolation Forest)	**89%**

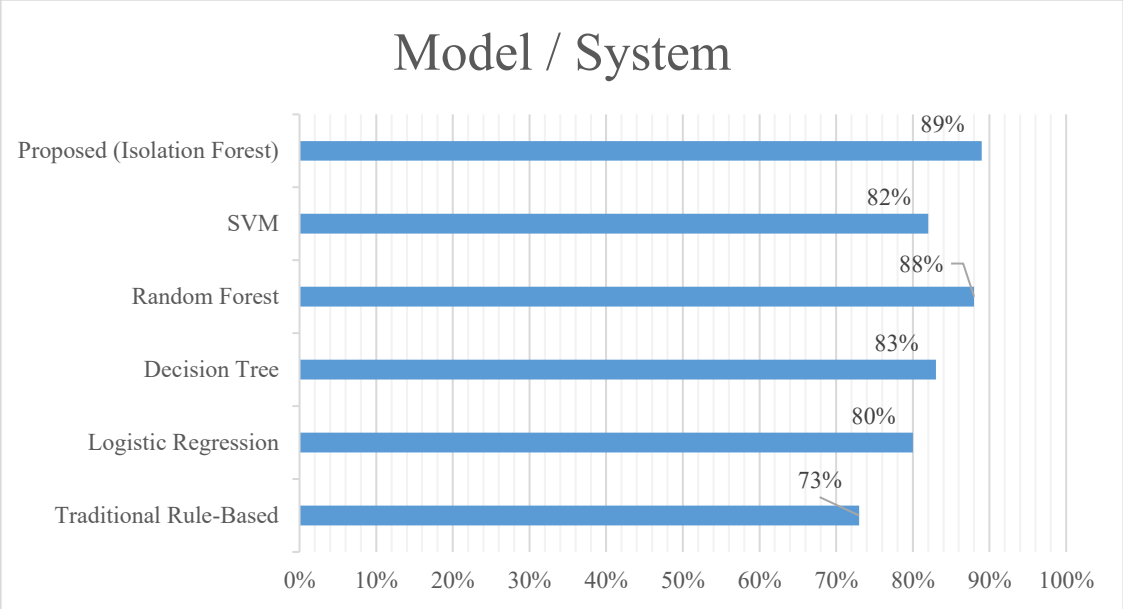

Figure 6: Graphical representation of f1-score comparison (%)

5 Discussion

Modern security challenges are improved by incorporating machine learning into API security frameworks. Modern cyber threats are too sophisticated for traditional security systems that use established rules, static settings, and signature-based detection. Attackers target APIs because they help apps communicate. Therefore, creative, dynamic, and flexible security solutions are needed now. Real-time threat detection, anomaly detection, and predictive security solutions are possible with machine learning to safeguard APIs (Al-Quayed et al., 2024). Machine learning algorithms are excellent at identifying patterns in large datasets, which makes them perfect for spotting suspect API traffic. Constantly analysing API interactions helps these models spot questionable tendencies. They can spot traffic spikes, which may indicate a DDoS attack, irregular data access requests, which may suggest a data breach, and multiple unsuccessful login attempts, which may indicate an account takeover. Supervised learning systems like Random Forest and SVM can distinguish valid and dangerous API queries. In addition to classification models, anomaly detection methods like Isolation Forests are necessary for outlier detection. Because they don't use threat signatures, anomaly detection approaches are distinct from traditional security. Instead, they find new attack patterns, making security systems proactive. Modern API security frameworks leverage these models to detect new vulnerabilities, including zero-day attacks that older rule-based methods cannot detect.

Although beneficial, machine learning in API security has many drawbacks. Data quality and availability are significant issues. Machine learning models must be trained on large, labeled datasets to distinguish between ordinary and malicious API behavior. Incorrectly labelled or inadequate data may lead to many false positives (normal behaviour mistaken for malicious) or false negatives (actual attacks missed). This decreases system reliability. Scalability is another issue, especially in cloud services and huge enterprise networks where APIs handle massive real-time traffic. Machine learning models must efficiently handle API communication without latency, which reduces performance. Distributed machine learning and model architecture optimization can help handle high-throughput API conditions, research shows. These options are still being developed.

Machine learning for security requires balancing false positives and negatives. Anomaly detection algorithms can find outliers but sometimes misjudge routine behaviour as malicious. In dynamic scenarios with rapid API usage changes, this is especially true. We built hybrid systems that combine supervised and unsupervised learning to address this. Unsupervised anomaly detection (like Isolation Forest) can spot abnormal behaviour, and then supervised classification models can determine if it is a threat. New API security method: reinforcement learning. This strategy enables machine learning models to interact with the system, learning and adapting as they go. These models can improve threat detection and reduce disruptions over time. This self-improving approach is helpful in API settings were behaviour changes quickly.

Machine learning-based API security must function well with API Gateway Security Frameworks and Zero Trust models to be effective. Machine learning should supplement security rather than replace it. Machine learning, rate limiting, encryption, and access control can increase system security. Research shows that using machine learning for network traffic monitoring and application-layer security protects against internal and external attacks.

The machine learning for API security study employs AI-driven cybersecurity methods, similar to those described by (Johny et al., 2025; Kasri et al., 2025), but with a focus on a specific application. (Johny et al., 2025) employ deep learning fusion to detect malware using visual feature extraction to improve classification. Johny et al. employ deep learning to detect malware, while the other study detects API security anomalies. Both studies use machine learning to boost cybersecurity. The API security study finds dangerous API traffic using anomaly detection approaches like Isolation Forests (Prabu & Sudhakar, 2024; Johny et al., 2025) use CNNs and other deep learning models to identify malware. However, (Kasri et al., 2025) investigate how Large Language Models (LLMs) can be employed in cybersecurity for automated threat intelligence, incident response, and adversarial defense. Kasri et al., emphasize LLMs' NLP capabilities for real-time cybersecurity monitoring, while the API security study focuses on API traffic pattern identification. Both studies worry about false positives and scalability. The API security study recommends mixed ML models to prevent false detections, while Kasri et al., advocate LLM-based filtering to improve accuracy. Additionally, (Johny et al., 2025) and the API security study emphasize model optimization for real-time implementation. However, (Kasri et al., 2025) focused on proactive threat defense with LLMs. This research expands AI-driven cybersecurity expertise. In their API security investigation, Johny et al. look for API irregularities. (Kasri et al., 2025) studied natural language processing-based cybersecurity intelligence, and (Johny et al., 2025) studied deep learning-based malware classification.

6 Conclusion

This paper introduces a powerful and clever API security architecture that utilizes machine learning, with a focus on real-time anomaly detection. The proposed system incorporates state-of-the-art threat modeling, an on-the-fly data system, feature engineering, and hybrid machine-learning systems, particularly utilizing Isolation Forest to detect and combat sophisticated threats (Mokoena & Nilsson, 2023). The findings are conclusive that machine learning models, and unsupervised in particular, slightly outperformed traditional rule-based systems in all performance metrics, including accuracy, precision, recall, and F1-score. Real-time telemetry, stream processing, and adaptive learning have enhanced the resiliency and responsiveness of the system subject to dynamically changing traffic patterns. Isolation Forest, specifically, was the best-performing algorithm, achieving 92 per cent accuracy and 90 per cent precision, which outperforms classical supervised algorithms and linkers. This outstanding performance confirms the efficacy of the model in detecting zero-day and polymorphic anomalies in APIs with

changing ecosystems. It is also embedded in the context of cloud-native environments and its modular architecture (this is also why it can be integrated into a cloud-native climate easily) facilitates continuous monitoring, retraining, as well as real-time policy enforcement. Comparative analysis shows that the suggested method is giving a significant advancement towards securing APIs as it minimizes false positives with a high detection rate. The developments not only enhance security posture but also reduce operational costs through improved threat triaging and remediation. Finally, the proposed machine learning-based API security architecture will be a scalable, efficient, and innovative solution to provide modern digital infrastructures (Krishnan et al., 2020). The future could also leverage deep learning models and federated learning to further secure decentralized networks and cross-domain APIs.

References

[1] Alnumay, W. S. (2024). Use of machine learning for the detection, identification, and mitigation of cyber-attacks. *International Journal of Communication and Computer Technologies*, *12*(1), 38-44.

[2] Al-Qaysi, Z. T., Albahri, A. S., Ahmed, M. A., & Mohammed, S. M. (2023). Development of hybrid feature learner model integrating FDOSM for golden subject identification in motor imagery. *Physical and Engineering Sciences in Medicine*, *46*(4), 1519-1534. https://doi.org/10.1007/s13246-023-01316-6

[3] Al-Quayed, F., Ahmad, Z., & Humayun, M. (2024). A situation based predictive approach for cybersecurity intrusion detection and prevention using machine learning and deep learning algorithms in wireless sensor networks of industry 4.0. *IEEE Access*, *12*, 34800-34819. https://doi.org/10.1109/ACCESS.2024.3372187

[4] Alzubaidi, L., Salhi, A., A. Fadhel, M., Bai, J., Hollman, F., Italia, K., ... & Gu, Y. (2024). Trustworthy deep learning framework for the detection of abnormalities in X-ray shoulder images. *Plos one*, *19*(3), e0299545. https://doi.org/10.1371/journal.pone.0299545

[5] Belal, M. M., & Sundaram, D. M. (2022). Comprehensive review on intelligent security defences in cloud: Taxonomy, security issues, ML/DL techniques, challenges and future trends. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9102-9131. https://doi.org/10.1016/j.jksuci.2022.08.035

[6] Dixit, P., & Silakari, S. (2021). Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, *39*, 100317. https://doi.org/10.1016/j.cosrev.2020.100317

[7] Ghanem, M. C., & Chen, T. M. (2019). Reinforcement learning for efficient network penetration testing. *Information*, *11*(1), 6. https://doi.org/10.1016/j.jksuci.2022.08.035.

[8] Ibitoye, O., Abou-Khamis, R., Shehaby, M. E., Matrawy, A., & Shafiq, M. O. (2019). The Threat of Adversarial Attacks on Machine Learning in Network Security--A Survey.

[9] Johny, J. A., Asmitha, K. A., Vinod, P., Radhamani, G., Rehiman, K. R., & Conti, M. (2025). Deep learning fusion for effective malware detection: leveraging visual features. *Cluster Computing*, *28*(2), 135. https://doi.org/10.1007/s10586-024-04723-w

[10] Kaloudi, N., & Li, J. (2020). The ai-based cyber threat landscape: A survey. *ACM Computing Surveys (CSUR)*, *53*(1), 1-34. https://doi.org/10.1145/3372823

[11] Kasri, W., Himeur, Y., Alkhazaleh, H. A., Tarapiah, S., Atalla, S., Mansoor, W., & Al-Ahmad, H. (2025). From vulnerability to defense: The role of large language models in enhancing cybersecurity. *Computation*, *13*(2), 30. https://doi.org/10.3390/computation13020030

[12] Khaleel, Y. L., Habeeb, M. A., Albahri, A. S., Al-Quraishi, T., Albahri, O. S., & Alamoodi, A. H. (2024). Network and cybersecurity applications of defense in adversarial attacks: A state-of-the-art using machine learning and deep learning methods. *Journal of Intelligent Systems*, *33*(1), 20240153. https://doi.org/10.1515/jisys-2024-0153

[13] Khan, S., & Parkinson, S. (2018). Review into state of the art of vulnerability assessment using artificial intelligence. *Guide to vulnerability analysis for computer networks and systems: An artificial intelligence approach*, 3-32. https://doi.org/10.1007/978-3-319-92624-7_1

[14] Krishnan, V. G., Krishnan, T. N., Karim, S. S., Yuvarajan, G., & Priya, M. R. (2020). Cyber Security in Data Mining to Data Driven Security. *International Journal of Advances in Engineering and Emerging Technology*, *11*(1), 71-76.

[15] Macas, M., Wu, C., & Fuertes, W. (2024). Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. *Expert Systems with Applications*, *238*, 122223. https://doi.org/10.1016/j.eswa.2023.122223

[16] Mahmood, S. S. (2025). Intelligent cyber defense: Utilizing deep learning for robust detection and prevention of phishing websites. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 16*(2), 591–610. https://doi.org/10.58346/JOWUA.2025.I2.036

[17] Malik, J., Muthalagu, R., & Pawar, P. M. (2024). A systematic review of adversarial machine learning attacks, defensive controls, and technologies. *IEEE Access*, *12*, 99382-99421. https://doi.org/10.1109/ACCESS.2024.3423323

[18] Mokoena, G., & Nilsson, J. (2023). A sophisticated cybersecurity intrusion identification model using deep learning. *International Academic Journal of Science and Engineering*, *10*(3), 17-21. https://doi.org/10.71086/IAJSE/V10I3/IAJSE1026

[19] Pissanidis, D. L., & Demertzis, K. (2024). Integrating AI/ML in cybersecurity: An analysis of open XDR technology and its application in intrusion detection and system log management. https://doi.org/10.20944/preprints202312.0205.v2

[20] Prabu, K., & Sudhakar, P. (2024). A hybrid deep learning approach for enhanced network intrusion detection. *Indonesian Journal of Electrical Engineering and Computer Science*, *33*(3), 1915-1923. https://doi.org/10.11591/ijeecs.v33.i3.pp1915-1923

[21] Sarker, I. H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects. *Annals of Data Science*, *10*(6), 1473-1498. https://doi.org/10.1007/s40745-022-00444-2

[22] Suhag, S., & Aarti. (2024). Challenges and Potential Approaches in Wireless Sensor Network Security. *Journal of Electrical Engineering & Technology*, *19*(4), 2693-2700. https://doi.org/10.1007/s42835-023-01751-1

[23] Tan, K. L., Sangaralingam, J., Mey, L. P., & Karupiah, P. (2025). Cybercrime Vulnerability Among Older Adults in Malaysia in the Context of Digital Deception. *Archives for Technical Sciences*, *2*(33), 154-164. https://doi.org/10.70102/afts.2025.1833.154

[24] Taye, M. M. (2023). Understanding of machine learning with deep learning: architectures, workflow, applications and future directions. *Computers*, *12*(5), 91. https://doi.org/10.3390/computers12050091

[25] Yan, J., Zhou, H., & Wang, W. (2025). Intelligent Network Element: A Programmable Switch Based on Machine Learning to Defend Against DDoS Attacks. *Information Systems Frontiers*, 1-20. https://doi.org/10.1007/s10796-024-10577-9

[26] Yu, J., Shvetsov, A. V., & Alsamhi, S. H. (2024). Leveraging machine learning for cybersecurity resilience in industry 4.0: Challenges and future directions. *IEEE access.* https://doi.org/10.1109/ACCESS.2024.3482987

Authors Biography

Dr. **Aarti** is a Ph.D. degree holder and also has M.Tech (2016) and B.Tech (2014) degree in in the Computer Science Engineering (CSE). She is currently an Assistant Professor of the Bharati Vidyapeeth College of Engineering, New Delhi working in the Department of Computer Science & Engineering. She has experience in Computer Science, with emphasis on computer networks, security, Machine Learning, Deep learning. She has more than 5+years of teaching and 1-year Industrial Experience in the field of Computer Science and Engineering. Her research involves the Systematic Approach of Network Monitoring for Security Perspective and its Impact on Content Analysis. Her research interests include computer networks, security Steganography, Mobile Ad-hoc networks (MANETs).

Dr. Sumit Suhag is a Ph.D research scholar and has M.Tech (2014) and B.Tech (2011) degree in Electronics and Communication Engineering (ECE). He has more than 7 years of teaching and research experience in the field of Electronics & Communication Engineering. Of his more than 5 years of teaching career, he has worked as an Assistant Professor in the eminent/reputed university like Jamia Millia Islamia, Central University, Delhi (India) & Deenbandhu Chhotu Ram University of Science and Technology, Haryana (India). He is the life member of Indian Society for Technical Education (ISTE) India, and International Association of Engineers (IAENG) Honk Kong, China. His research involves the study of challenges in various tools & techniques used for information security along with their solutions. His research interests include Network security and Privacy, Steganography, Mobile Ad-hoc networks (MANETs), Wireless Sensor Networks (WSNs), cloud computing, and embedded systems.