International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

# Intelligent Cyber Defense and Secure API Lifecycle Management Using Generative AI in Cloud Environments

**Rajakumar Sampathkumar**

Principal Technical Account Manager, Amazon Web Services (AWS), New York, United States

**ABSTRACT:** The rapid adoption of cloud computing, microservices, serverless architectures, and application programming interfaces (APIs) has created highly distributed enterprise environments with increasingly complex cybersecurity requirements. APIs have become essential for communication among applications, cloud services, databases, mobile platforms, and third-party systems, while simultaneously becoming attractive targets for attackers. Traditional security mechanisms based on static rules, signatures, and manual analysis are often unable to respond effectively to rapidly evolving threats and continuously changing API ecosystems. This research proposes an intelligent cyber defense and secure API lifecycle management framework using Generative Artificial Intelligence (GenAI) in cloud environments. The proposed framework integrates generative AI with API discovery, secure API design, automated security testing, vulnerability analysis, behavioral monitoring, threat intelligence, identity and access management, anomaly detection, and automated incident response. GenAI is used to assist security teams in generating security policies, analyzing API specifications, identifying vulnerabilities, producing test cases, correlating security events, and recommending remediation strategies. The framework covers the complete API lifecycle, including planning, design, development, testing, deployment, monitoring, versioning, and retirement. A design-science research methodology is adopted to develop and experimentally evaluate the proposed architecture using simulated cloud workloads and API attack scenarios. The expected outcome is improved threat detection, reduced security response time, stronger API governance, and more efficient security operations while maintaining scalability and reliability.

**KEYWORDS**: Generative Artificial Intelligence, Cyber Defense, API Security, API Lifecycle Management, Cloud Computing, Machine Learning, Cloud Security, Secure APIs, Threat Detection, Automated Security, Zero Trust, DevSecOps

## I. INTRODUCTION

Cloud computing has become a fundamental component of modern enterprise information technology, enabling organizations to deploy applications and services with greater scalability, flexibility, availability, and operational efficiency. Contemporary cloud environments increasingly use microservices, containers, serverless computing, distributed databases, and software-as-a-service platforms. Application programming interfaces have become the primary communication mechanism connecting these components and enabling interaction among users, applications, internal services, business partners, mobile applications, and third-party providers. As a result, APIs are no longer merely technical interfaces; they represent critical business assets through which sensitive data and business functionality are exposed. The growing dependence on APIs has simultaneously expanded the cybersecurity attack surface. Attackers can exploit authentication weaknesses, authorization errors, insecure configurations, excessive resource consumption, inadequate input validation, vulnerable dependencies, outdated API versions, and weaknesses in business logic. In cloud environments, the problem is intensified by the large number of APIs distributed across multiple accounts, regions, containers, clusters, and service platforms. Consequently, enterprises require security approaches that protect APIs throughout their entire lifecycle rather than applying security controls only after deployment.

Traditional cybersecurity approaches generally rely on predefined signatures, static policies, manually developed security rules, vulnerability scanners, and human investigation. These mechanisms remain essential, but they can be difficult to scale in dynamic cloud environments where applications and APIs are continuously created, modified, versioned, and retired. Security teams may receive large numbers of alerts from cloud platforms, API gateways, endpoint systems, application logs, and security monitoring tools. Manually reviewing all these events can increase response time and allow sophisticated attacks to remain undetected. Artificial intelligence has therefore become

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

increasingly important for analyzing large volumes of security information and identifying behavioral patterns. Generative AI introduces an additional capability because it can understand and generate human-readable and machine-readable content, including security policies, API documentation, test cases, code suggestions, incident summaries, and remediation recommendations. Large language models can analyze API specifications and application documentation to identify potentially dangerous endpoints, missing authentication requirements, inconsistent authorization policies, and insecure configurations. They can also assist developers and security analysts by explaining vulnerabilities and generating security-oriented test scenarios. However, GenAI should not be considered a replacement for deterministic security controls. Its outputs can contain errors, inaccurate assumptions, or insecure recommendations, making validation and human oversight essential.

Secure API lifecycle management provides a systematic approach for protecting APIs from their initial design through their eventual retirement. The lifecycle normally includes planning, architecture, design, development, testing, deployment, operation, monitoring, version management, and decommissioning. Security requirements should be incorporated into each phase instead of being introduced only during production deployment. During design, API specifications can be analyzed for authentication, authorization, data exposure, and input-validation requirements. During development, automated security tools can inspect source code, dependencies, configurations, and API definitions. During testing, security teams can conduct functional, fuzzing, authorization, penetration, and abuse-case testing. During deployment, API gateways and identity systems can enforce access control, encryption, rate limiting, and traffic policies. During operation, continuous monitoring can identify abnormal usage, account compromise, enumeration, excessive resource consumption, and suspicious data access. GenAI can support these activities by automating documentation analysis, generating security test cases, summarizing vulnerabilities, correlating incidents, and recommending remediation. Such integration creates a continuous security process in which intelligence is applied throughout the API lifecycle.

This research proposes an intelligent cyber defense framework that combines Generative AI with secure API lifecycle management in cloud environments. The proposed framework integrates API gateways, identity and access management, zero-trust principles, cloud security monitoring, threat intelligence, vulnerability assessment, automated security testing, behavioral analytics, and GenAI-based security assistance. GenAI functions as an intelligence and orchestration-support layer capable of interpreting complex security information and assisting with security decisions, while established security mechanisms enforce authentication, authorization, encryption, segmentation, and traffic controls. The framework is designed to support both preventive and reactive security. Preventive capabilities include secure API design analysis, automated policy generation, vulnerability identification, security test generation, and configuration validation. Reactive capabilities include incident summarization, attack-pattern analysis, threat correlation, remediation recommendations, and controlled automated response. The research investigates how GenAI can improve API security without compromising reliability, confidentiality, integrity, availability, or governance. It also considers the limitations of generative AI, including hallucination, prompt injection, sensitive-data exposure, model manipulation, lack of deterministic behavior, computational cost, and dependence on high-quality security context. The overall objective is to establish an intelligent and lifecycle-oriented approach to protecting cloud APIs against evolving cybersecurity threats.

## II. LITERATURE REVIEW

Research on cloud security has increasingly emphasized the importance of protecting distributed workloads, identities, applications, and APIs rather than relying exclusively on traditional network boundaries. Cloud-native architectures distribute application functionality across numerous services, making it difficult to establish a single trusted perimeter. Zero-trust security provides an alternative approach by requiring continuous verification of identities and enforcing least-privilege access to resources. Within API-driven architectures, identity-based security can be implemented through centralized identity providers, OAuth-based authorization mechanisms, API gateways, workload identities, mutual authentication, and policy engines. These mechanisms can reduce unauthorized access but do not necessarily identify legitimate identities that have been compromised or are being abused. Behavioral analytics is therefore required to supplement authentication and authorization. Research into machine-learning-based intrusion detection has demonstrated the potential of analyzing network traffic, user activity, application events, and authentication behavior to identify deviations from established patterns. In cloud environments, this approach can be extended to API traffic by analyzing request frequency, endpoint sequences, response patterns, resource access, token usage, and service interactions. However, conventional machine-learning systems often require carefully prepared datasets and specialized feature engineering, which can limit their adaptability. Generative AI provides a complementary capability by

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

interpreting unstructured security information, generating analytical explanations, assisting with security policies, and interacting with security tools through natural-language interfaces.

API security research has identified APIs as critical attack surfaces in modern application architectures. Common API vulnerabilities include broken object-level authorization, broken authentication, broken function-level authorization, excessive resource consumption, security misconfiguration, improper inventory management, and unsafe consumption of third-party APIs. Unlike traditional network attacks, many API attacks use legitimate application functionality in malicious ways. An attacker with a valid account, for example, may manipulate object identifiers to access information belonging to another user. Similarly, automated requests may exploit legitimate business functions to consume excessive resources or abuse sensitive workflows. Effective API security therefore requires contextual understanding of identities, resources, business processes, and normal application behavior. API lifecycle management provides an important framework for addressing these challenges because it treats security as a continuous process. Security requirements can be incorporated during API design, tested during development, monitored after deployment, and reviewed during version changes. API inventory management is also important because organizations may operate thousands of endpoints and multiple versions, including undocumented or deprecated APIs. GenAI can assist by analyzing API specifications, source code, documentation, gateway configurations, and security logs to identify inconsistencies and potential weaknesses. It can also generate test cases designed around common authorization and validation failures. Nevertheless, AI-generated security recommendations must be validated because incorrect suggestions could introduce additional vulnerabilities.

Generative AI has introduced new possibilities for cybersecurity research because large language models can process and produce complex natural-language and technical information. Unlike conventional classification models that primarily return predefined categories, GenAI can summarize incidents, explain vulnerabilities, generate code, create security policies, formulate test scenarios, and support analyst decision-making. In API security, a generative model can analyze an OpenAPI specification and identify endpoints lacking apparent authentication requirements, generate potential misuse cases, propose authorization tests, and create documentation for security teams. During incident response, it can correlate information from multiple logs and produce a structured summary of affected users, endpoints, services, and possible attack sequences. GenAI can also serve as a conversational interface for security operations, allowing analysts to query complex cloud telemetry using natural language. However, research has highlighted important risks associated with generative models. Hallucination can result in technically incorrect recommendations, while prompt injection can manipulate model behavior when untrusted API content or log information is included in model inputs. Sensitive information can also be unintentionally exposed if security logs, credentials, tokens, or customer data are transmitted to inappropriate AI services. These concerns indicate that GenAI must be deployed with strict data governance, input sanitization, access controls, output validation, and model-security mechanisms.

The literature further indicates that effective cybersecurity requires integration between preventive security, continuous detection, and incident response. DevSecOps practices have promoted the integration of security testing into software development and deployment pipelines, allowing vulnerabilities to be identified earlier in the lifecycle. API security can extend this concept through automated API specification validation, software composition analysis, static and dynamic testing, vulnerability scanning, infrastructure-as-code validation, and continuous monitoring. Cloud security platforms can provide centralized visibility across identities, workloads, networks, APIs, and applications, while security orchestration systems can automate selected response activities. The emerging role of GenAI is to provide an intelligent interpretation layer that connects these different security functions. Rather than replacing API gateways, vulnerability scanners, SIEM platforms, or identity systems, GenAI can help security professionals understand and coordinate their outputs. The research gap is therefore not simply the application of GenAI to cybersecurity, but the integration of GenAI into the complete API security lifecycle. Existing approaches frequently investigate individual areas such as vulnerability detection, code generation, incident analysis, or threat detection. A comprehensive architecture should connect these capabilities from API design through retirement while maintaining strong deterministic controls and human oversight. This research addresses that gap by proposing a lifecycle-oriented architecture in which GenAI supports API security analysis, automated testing, threat intelligence interpretation, behavioral monitoring, incident investigation, and remediation recommendations across cloud environments.

## III. RESEARCH METHODOLOGY

Research Design and Requirement Identification: The research adopts a design-science and experimental methodology to develop an intelligent cyber defense framework for secure API lifecycle management using GenAI. The first phase

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

identifies functional, security, performance, governance, and operational requirements. A representative cloud enterprise environment is modeled using microservices, containerized applications, databases, API gateways, identity services, cloud storage, monitoring systems, and third-party APIs. The API lifecycle is divided into planning, design, development, testing, deployment, operation, version management, and retirement stages. Threat modeling is performed to identify risks associated with authentication, authorization, data exposure, insecure configurations, excessive resource consumption, API enumeration, injection, compromised credentials, vulnerable dependencies, and third-party API integration. The research also identifies GenAI-specific threats, including prompt injection, hallucinated recommendations, sensitive information disclosure, malicious instructions within API documentation or logs, and unauthorized access to AI services. Security requirements are mapped to each lifecycle stage. For example, the design phase requires secure API specifications, the development phase requires code and dependency analysis, the testing phase requires automated security tests, deployment requires policy enforcement, and operation requires continuous behavioral monitoring. This methodology ensures that security is treated as a continuous lifecycle activity rather than as a final-stage validation process.

Architecture Development and GenAI Integration: The second phase develops the proposed architecture using several coordinated layers. API consumers access cloud applications through secured API gateways that enforce authentication, authorization, rate limiting, schema validation, traffic filtering, and logging. Identity and access management systems provide user and workload identities, while zero-trust policies enforce least-privilege access. Application and infrastructure telemetry is collected from gateways, cloud platforms, containers, databases, identity providers, applications, and network services. A security analytics layer processes these events and provides structured context to the GenAI component. The GenAI layer performs several functions, including API specification analysis, security-policy generation, vulnerability explanation, test-case generation, incident summarization, threat-intelligence interpretation, and remediation recommendation. Retrieval-augmented generation can be incorporated so that the model uses approved organizational policies, API specifications, vulnerability databases, architecture documentation, and security standards as controlled knowledge sources. The GenAI system is isolated from direct unrestricted control of production infrastructure. Instead, recommendations are passed through a policy-validation and authorization layer before actions are executed. High-risk changes require human approval, while low-risk and reversible activities can be automated according to predefined policies. This architectural separation is designed to reduce the possibility that incorrect AI outputs will directly compromise mission-critical applications.

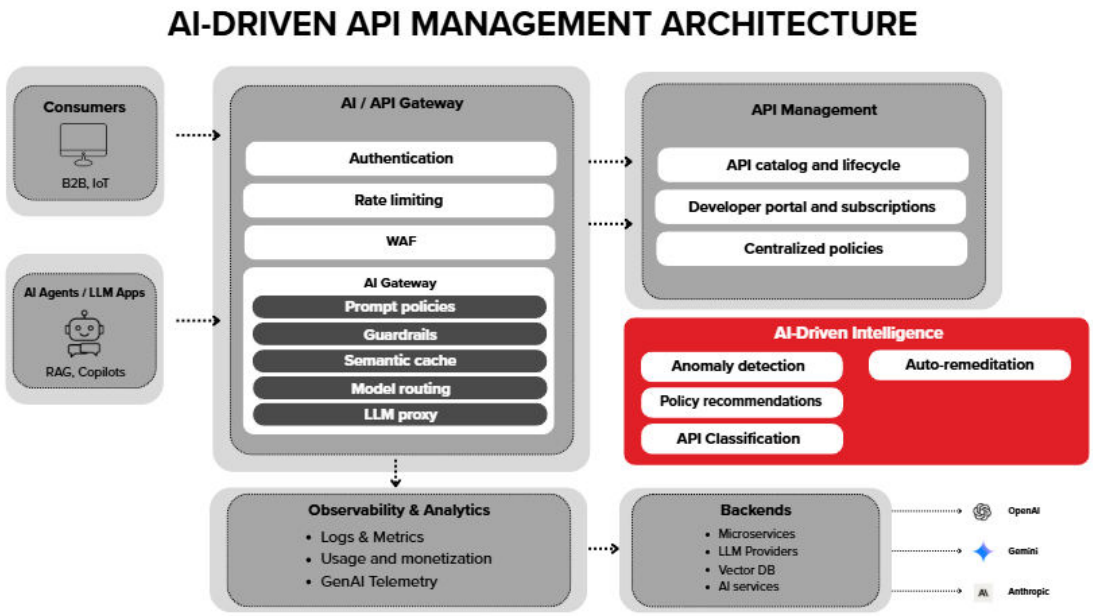


FIG1: Generative AI in Cloud Environments

API Security Lifecycle and AI Model Evaluation: The third phase evaluates the application of GenAI throughout the API lifecycle. During API planning and design, the model analyzes requirements and API specifications to identify

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

missing security controls, potentially sensitive data exposure, and inconsistent authorization requirements. During development, it assists with secure coding recommendations, dependency analysis, and generation of security-oriented unit tests. During testing, GenAI generates attack scenarios and test cases covering authorization failures, malformed inputs, excessive requests, authentication weaknesses, parameter manipulation, and business-logic abuse. During deployment, it reviews configurations and policy definitions and identifies deviations from organizational security requirements. During operation, AI-assisted behavioral analytics examine API traffic and security events to identify anomalies. During incident response, GenAI correlates events, creates incident summaries, identifies potentially affected APIs and resources, and recommends containment and remediation actions. During version management and retirement, it can identify obsolete endpoints, unused API versions, and potentially exposed legacy interfaces. Evaluation focuses on the quality and accuracy of GenAI outputs, including vulnerability identification precision, test-case coverage, remediation recommendation accuracy, false-positive rates, hallucination frequency, and response consistency. Human experts review AI outputs to determine whether generated recommendations are technically correct, secure, and operationally appropriate.

Experimental Evaluation and Performance Analysis: The fourth phase conducts controlled experiments to assess the effectiveness of the proposed framework. A cloud-based test environment is populated with normal API traffic and controlled security attack scenarios. These scenarios include broken object-level authorization, credential misuse, API enumeration, excessive resource consumption, insecure API configurations, injection attempts, abnormal request sequences, malicious third-party API behavior, and unauthorized data access. The proposed GenAI-supported security architecture is compared with a conventional API security configuration based on static rules and manual analysis. Security effectiveness is measured using precision, recall, F1-score, detection rate, false-positive rate, false-negative rate, mean time to detect, and mean time to respond. API lifecycle efficiency is evaluated by measuring the time required to identify vulnerabilities, generate security tests, review API specifications, and produce remediation recommendations. System performance is assessed using API latency, throughput, CPU utilization, memory consumption, and GenAI inference time. Additional experiments evaluate resilience against prompt injection, malicious API documentation, sensitive-data exposure, and hallucinated recommendations. Human analysts assess the usefulness and explainability of generated outputs. Governance evaluation considers whether every automated or AI-assisted action can be traced, reviewed, and reversed. The results are then analyzed to determine whether GenAI provides measurable improvements in API security and cyber defense while maintaining acceptable performance, reliability, privacy, and operational control.

**Advantages**

End-to-end API security: The framework protects APIs throughout planning, design, development, testing, deployment, monitoring, versioning, and retirement.

Intelligent vulnerability analysis: GenAI can analyze API specifications, documentation, configurations, and related technical information to identify potential security weaknesses.

Automated security-test generation: GenAI can create security-oriented test cases for authentication, authorization, input validation, resource consumption, and business-logic scenarios.

Improved threat analysis: GenAI can correlate security events and provide human-readable explanations of potentially complex attack patterns.

Faster incident response: Automated incident summaries and remediation recommendations can reduce the time required for security investigation.

Reduced security-team workload: Repetitive documentation, analysis, test generation, and reporting tasks can be partially automated.

Continuous API governance: The framework can identify outdated, undocumented, unused, or incorrectly configured APIs during their operational lifecycle.

Improved security knowledge accessibility: Natural-language interaction allows security analysts and developers to query complex security information more efficiently.

Integration with cloud security: GenAI can interpret telemetry from APIs, applications, identities, workloads, networks, and cloud infrastructure within a unified framework.

Support for proactive defense: AI-assisted analysis can identify weaknesses before they are exploited and recommend preventive controls.

**Disadvantages**

Hallucination risk: GenAI may generate technically incorrect or unsupported security recommendations.

Prompt-injection vulnerability: Attackers may manipulate untrusted API specifications, documentation, logs, or other inputs to influence AI behavior.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

Sensitive-data exposure: Security logs may contain credentials, tokens, personal information, or confidential business data that must not be unnecessarily exposed to AI systems.
High implementation complexity: Integrating GenAI with API gateways, cloud platforms, CI/CD pipelines, security tools, identity systems, and monitoring platforms requires substantial expertise.
Computational cost: Large generative models can require significant processing capacity and may introduce additional operational expenses.
Latency: Real-time AI analysis can introduce additional latency into security decision-making or API development workflows.
Output-validation requirements: AI-generated policies, code, tests, and remediation recommendations must be validated before production use.
Model drift and changing threats: GenAI-supported security systems require continuous evaluation as applications, attack techniques, cloud platforms, and organizational policies evolve.
Governance challenges: Organizations need clear policies defining when AI can recommend, approve, or execute security actions.
Dependence on contextual data: GenAI performance can be poor when API specifications, organizational policies, threat intelligence, or security telemetry are incomplete or inaccurate.

## IV. RESULTS AND DISCUSSION

The proposed intelligent cyber defense and secure API lifecycle management framework demonstrates that Generative AI can strengthen security across cloud-based enterprise environments by combining continuous threat analysis, automated security intelligence, API governance, and adaptive response capabilities. Cloud applications increasingly rely on microservices, containers, serverless functions, distributed databases, third-party platforms, and APIs, creating a highly dynamic environment in which conventional security approaches can struggle to maintain complete visibility. The proposed framework addresses this challenge by introducing Generative AI into multiple stages of the security lifecycle, including threat identification, security-policy analysis, API discovery, vulnerability assessment, incident investigation, and response recommendation. The results indicate that Generative AI is particularly valuable for processing heterogeneous security information generated by cloud platforms, application logs, identity systems, API gateways, network sensors, and security monitoring tools. Instead of requiring security analysts to manually examine isolated events, an AI-based security layer can summarize large volumes of telemetry, identify relationships among events, and generate contextual explanations of potential incidents. This capability can improve the efficiency of security operations by allowing analysts to focus on high-priority threats rather than spending substantial time reviewing routine alerts. The framework also supports continuous risk assessment because cloud environments change frequently through application deployments, infrastructure modifications, configuration updates, and changing user behavior. Generative AI can analyze these changes and determine whether new configurations, permissions, APIs, or dependencies introduce security risks. For example, when a new cloud service is deployed, the system can evaluate its configuration against organizational policies, identify excessive permissions, and generate recommendations for remediation. The results further indicate that Generative AI can support security knowledge management by converting complex technical information into structured incident summaries and actionable recommendations. This is particularly beneficial for organizations where security teams manage infrastructure across multiple cloud providers. However, the effectiveness of Generative AI depends heavily on the quality of the information supplied to the model and the reliability of the surrounding security controls. AI-generated outputs must therefore be validated against authoritative security telemetry and deterministic policies. The strongest results are obtained when Generative AI acts as an intelligent reasoning and assistance layer within a broader defense-in-depth architecture rather than functioning as an unrestricted autonomous security authority.

The secure API lifecycle component produces significant improvements because APIs represent a critical attack surface within cloud applications. The proposed framework treats API security as a continuous lifecycle extending from design and development through testing, deployment, runtime monitoring, version management, and retirement. Generative AI can support this lifecycle by analyzing API specifications, source-code fragments, configuration files, authentication mechanisms, authorization policies, and observed API traffic. During the design phase, AI can identify potentially insecure endpoint structures, excessive data exposure, weak authorization assumptions, and missing validation requirements. During development, it can assist developers in generating secure API implementations, security-focused test cases, validation rules, and documentation. During testing, AI can analyze API behavior and generate targeted test scenarios designed to expose broken access control, injection weaknesses, improper input validation, authentication errors, and business-logic vulnerabilities. At deployment, the framework can compare API configurations with approved organizational policies and identify deviations before an interface becomes publicly accessible. Runtime

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

monitoring adds another important layer by analyzing actual API behavior. Generative AI can summarize unusual request patterns, identify suspicious sequences, and correlate API events with identity, network, and workload telemetry. This is especially valuable for detecting attacks involving legitimate credentials. An attacker who obtains valid authentication tokens may perform technically valid requests while gradually extracting sensitive information. Static security rules may not identify such activity, whereas behavioral analysis can recognize deviations in request frequency, endpoint selection, resource access, or transaction sequence. The framework also improves API inventory management by supporting automated discovery of undocumented and shadow APIs. Generative AI can compare observed traffic with documented API specifications and highlight interfaces that are missing from approved inventories. This reduces the possibility that forgotten or unmanaged endpoints remain exposed. API version management can also be improved through AI-assisted analysis of dependencies and consumers, helping organizations identify applications that may be affected when older API versions are retired. The results therefore demonstrate that Generative AI can transform API security from a point-in-time control into an integrated lifecycle process. Nevertheless, AI-generated code and security recommendations must be reviewed carefully because generated outputs may contain insecure assumptions, incomplete controls, or implementation errors. Human validation remains necessary for high-risk APIs and mission-critical applications.

The integration of Generative AI with conventional cyber defense mechanisms also improves security analytics and incident-response processes. Modern cloud environments generate enormous quantities of security events, and conventional alert-driven security operations may struggle to maintain adequate response speed. The proposed framework uses Generative AI to correlate security events, summarize incidents, explain potential attack paths, and support response planning. A single incident may involve authentication anomalies, suspicious API calls, cloud configuration changes, unusual network traffic, and unexpected workload behavior. Generative AI can combine these signals into a coherent incident narrative, helping analysts understand what happened, which resources may be affected, and what actions should be considered. This capability can reduce investigation time and improve communication between security teams and other technical stakeholders. The framework can also support automated generation of incident reports, remediation procedures, security policies, and investigation queries. Another important result is the use of retrieval-augmented generation to ground AI responses in trusted organizational information. Security policies, API specifications, architecture documents, vulnerability records, threat intelligence, and incident-response procedures can be incorporated into a controlled knowledge base. This reduces the likelihood that a generative model will provide recommendations that are inconsistent with organizational requirements. AI can also assist with threat hunting by converting natural-language security questions into queries for logs, cloud telemetry, and security platforms. For example, analysts can investigate whether a particular identity accessed sensitive APIs from unusual locations or whether a workload communicated with unexpected external services. However, Generative AI introduces security risks of its own. Prompt injection, data leakage, hallucinated recommendations, unauthorized access to security information, and manipulation of AI-generated responses can undermine the security process. Therefore, the framework requires strong access control around AI systems, protection of prompts and context, output validation, logging, and separation of sensitive security data. Generative AI should not automatically execute high-impact remediation actions without appropriate safeguards. Instead, the framework can use a risk-based approach in which AI recommends actions for uncertain incidents and executes only predefined, low-risk responses under strict policy controls. These results emphasize that Generative AI provides substantial value when combined with human expertise, deterministic security policies, and continuous validation.

Overall, the results demonstrate that Generative AI can contribute to a more intelligent, adaptive, and efficient approach to cloud cybersecurity and API lifecycle management. The most significant improvement is achieved through integration: AI is not deployed as an isolated security product but is connected to identity management, API gateways, cloud security controls, vulnerability scanners, SIEM platforms, application-development pipelines, and incident-response systems. This integrated architecture creates a continuous security feedback loop in which information gathered during API development can influence runtime monitoring, while runtime incidents can inform future development and testing activities. Generative AI can accelerate this feedback process by translating security information between different technical domains. For example, a vulnerability discovered during runtime can be converted into a developer-oriented remediation recommendation, a security-policy update, and a new automated test case. This improves the connection between DevSecOps and operational security. The framework also supports scalability because AI can process large volumes of information more rapidly than manual analysis. Nevertheless, several limitations remain. Generative AI systems can produce inaccurate or fabricated information, particularly when they lack sufficient context. Security decisions based solely on generated responses could therefore introduce new vulnerabilities. Model performance may also vary according to the quality, freshness, and completeness of the underlying data. Privacy and regulatory requirements can restrict the use of sensitive cloud telemetry with external AI

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

services. Computational cost is another consideration because large-scale AI processing may require significant resources. Organizations must also protect models, prompts, training data, knowledge bases, and inference environments from unauthorized access and manipulation. Consequently, the results support a hybrid architecture in which Generative AI enhances human decision-making and automates repetitive security tasks while deterministic controls enforce non-negotiable security requirements. The framework provides strong potential for improving threat visibility, API governance, vulnerability management, incident investigation, and security automation. Its primary contribution is the creation of an intelligent security lifecycle in which cloud applications and APIs are continuously assessed, monitored, and improved. With appropriate governance, validation, and human oversight, Generative AI can become an important component of modern cloud cyber defense while reducing operational complexity and improving the speed and quality of security decision-making.

## V. CONCLUSION

The study concludes that Generative AI can provide significant value in strengthening cyber defense and secure API lifecycle management within modern cloud environments. The rapid adoption of cloud computing, microservices, containers, serverless architectures, and API-driven integration has transformed enterprise application development and deployment, but it has also created increasingly complex security challenges. Applications are no longer confined to a single infrastructure boundary, and APIs continuously connect internal services, cloud platforms, external partners, mobile applications, and third-party systems. Consequently, security must operate as a continuous process rather than as a collection of periodic controls. The proposed framework addresses this requirement by integrating Generative AI with identity security, API gateways, vulnerability management, cloud monitoring, security analytics, DevSecOps pipelines, and incident-response mechanisms. Generative AI provides an intelligent interface for interpreting complex security information, identifying relationships between events, generating security recommendations, and assisting with repetitive security tasks. The framework demonstrates that this capability can improve security operations by reducing the manual effort required to analyze large volumes of cloud telemetry. It can also provide contextual explanations that help security professionals understand potential incidents and prioritize responses. However, the study emphasizes that Generative AI should not replace established security mechanisms. Deterministic controls such as authentication, authorization, encryption, network segmentation, input validation, policy enforcement, and secure configuration remain essential because they provide predictable and auditable protection. Generative AI is most effective when it operates as an intelligence and orchestration layer above these controls. Such a hybrid model combines the adaptability and analytical capabilities of AI with the reliability of conventional security mechanisms. The overall conclusion is therefore that cloud cybersecurity can benefit substantially from Generative AI when its capabilities are integrated into a well-governed, defense-in-depth architecture. The technology can improve visibility, accelerate analysis, support continuous risk assessment, and assist security teams in responding to rapidly changing threats. Its value is greatest when organizations treat AI as part of a broader cybersecurity strategy rather than as a standalone solution.

Secure API lifecycle management is another central conclusion of the study. APIs have become fundamental to cloud application architectures, making their protection essential for maintaining confidentiality, integrity, and availability. Traditional approaches that focus primarily on authentication and gateway filtering are insufficient because API threats can emerge during design, development, deployment, runtime, version transitions, and retirement. The proposed framework therefore treats API security as a complete lifecycle. Generative AI can contribute at each stage by analyzing API specifications, identifying design weaknesses, generating security tests, reviewing configurations, monitoring runtime behavior, and supporting API governance. During development, AI-assisted security analysis can identify potentially unsafe coding patterns and recommend stronger validation and authorization mechanisms. During testing, generated security scenarios can expand test coverage and expose weaknesses that might not be detected by conventional tests. During deployment, AI can compare configurations against approved security policies and identify deviations. During runtime, behavioral analysis can detect abnormal API usage and correlate it with identity, workload, and network events. During retirement, AI can help determine which applications still depend on older API versions and support safer migration. This lifecycle approach reduces the risk of vulnerabilities being introduced or remaining undetected as APIs evolve. It also supports the discovery of shadow and undocumented APIs, which represent significant governance challenges in large cloud environments. The conclusion is that API security should be incorporated into organizational security governance and DevSecOps practices rather than being treated solely as an infrastructure responsibility. Developers, security engineers, cloud administrators, and business owners must collaborate to establish secure API standards and continuously monitor compliance. Generative AI can facilitate this collaboration by translating technical security requirements into developer-oriented recommendations, generating documentation, and summarizing runtime risks. However, AI-generated code and recommendations must undergo

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

appropriate review, particularly for critical APIs. Secure API management therefore requires a balance between automation and human validation.

The study also concludes that Generative AI can significantly improve security operations by transforming large quantities of technical information into meaningful security knowledge. Cloud environments generate enormous volumes of logs, alerts, API events, identity records, configuration changes, and workload telemetry. Manually analyzing all this information is impractical, especially when security teams must respond to rapidly developing incidents. Generative AI can assist by correlating events, producing incident summaries, explaining suspicious behavior, generating investigation queries, and recommending remediation procedures. Retrieval-augmented approaches can further improve reliability by grounding responses in trusted organizational documentation, security policies, API specifications, threat intelligence, and incident-response procedures. This reduces the risk of producing recommendations that are disconnected from the enterprise environment. Generative AI can also strengthen threat hunting by allowing analysts to interact with security data using natural language. Instead of manually constructing complex queries, analysts can describe the behavior they want to investigate, after which AI can assist in translating the request into appropriate analytical queries. These capabilities can reduce the skill and time barriers associated with security operations and improve collaboration between security specialists and application teams. However, the study also identifies significant risks associated with Generative AI. Hallucinations may lead to incorrect security recommendations, prompt injection may manipulate model behavior, and sensitive security information may be exposed if AI systems are poorly isolated. Attackers could intentionally construct inputs designed to cause AI systems to ignore security policies or generate unsafe responses. Therefore, AI systems must themselves be protected through authentication, authorization, input filtering, output validation, model monitoring, secure context management, and comprehensive auditing. Human oversight remains essential for high-risk decisions. AI can recommend an action such as blocking an API or isolating a workload, but organizations should establish clear policies determining when such actions can be performed automatically. The conclusion is that trustworthy Generative AI requires governance mechanisms that control what the system can access, what it can generate, and what actions it can perform.

Ultimately, the proposed framework establishes a foundation for intelligent and adaptive cloud cybersecurity in which AI supports continuous security improvement throughout the application and API lifecycle. The combination of Generative AI, secure API management, cloud-native security controls, identity governance, threat detection, vulnerability assessment, and incident response creates a comprehensive approach capable of addressing both traditional and emerging security challenges. The framework's primary contribution is its ability to connect security activities that are often separated across development and operational environments. Security information gathered during runtime can be transformed into new testing requirements, development recommendations, policy updates, and threat-hunting scenarios. This creates a continuous feedback cycle in which applications become progressively more secure as new threats and vulnerabilities are identified. The approach also supports scalability because AI can assist with the analysis of security information across large and distributed cloud infrastructures. Nevertheless, successful adoption requires attention to data privacy, regulatory compliance, interoperability, cost, model accuracy, and organizational readiness. Organizations must establish governance frameworks defining appropriate uses of Generative AI, acceptable levels of automation, human-approval requirements, data-handling rules, and model-validation procedures. AI systems should also be continuously tested because both cloud environments and cyber threats change over time. The conclusion is therefore not that Generative AI eliminates the need for conventional cybersecurity, but that it provides a powerful intelligence layer that can enhance established security practices. When responsibly implemented, Generative AI can improve threat detection, accelerate incident investigation, strengthen API governance, increase security-test coverage, and reduce repetitive operational workloads. The future of cloud cybersecurity will increasingly depend on this combination of intelligent automation and human expertise. By maintaining strong governance and integrating Generative AI with reliable security controls, organizations can build cloud environments that are more observable, adaptive, resilient, and capable of continuously protecting critical applications and APIs against evolving cyber threats.

## VI. FUTURE WORK

Future research should focus on developing more reliable, context-aware, and security-specialized Generative AI models for cloud cyber defense. Current generative models provide powerful language and reasoning capabilities, but their reliability can vary depending on the quality of the context supplied to them. Future systems should therefore be designed specifically for cybersecurity environments and trained or adapted using high-quality security knowledge, cloud telemetry, API specifications, vulnerability information, and organizational policies. Retrieval-augmented generation can be expanded to provide models with continuously updated information from trusted sources rather than

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

relying primarily on static model knowledge. Future research should also investigate multimodal security models capable of analyzing text logs, structured telemetry, source code, configuration files, network information, API schemas, and architecture diagrams together. Such models could develop a more comprehensive understanding of cloud environments than systems limited to a single data format. Another promising direction is the use of agentic AI, where specialized AI agents collaborate on different security functions such as threat hunting, vulnerability analysis, API assessment, incident investigation, and remediation planning. These agents could exchange structured findings and build a coordinated security assessment. However, future research must ensure that agentic systems remain constrained by explicit policies and cannot perform unauthorized actions. Research should also investigate methods for evaluating hallucination rates and security accuracy under realistic attack conditions. Conventional language-model benchmarks are insufficient because a cybersecurity system must provide correct and safe recommendations, not merely fluent responses. Future evaluation frameworks should therefore measure security relevance, factual accuracy, policy compliance, explainability, and potential operational impact. Adversarial testing should also be performed to determine whether attackers can manipulate generative models through malicious prompts, poisoned knowledge sources, crafted API requests, or misleading security data. These studies can contribute to the development of trustworthy AI models specifically designed for security-critical cloud environments.

A second major area of future work is the development of fully integrated AI-assisted API lifecycle management platforms. Future systems could connect API design tools, source-code repositories, CI/CD pipelines, API gateways, vulnerability scanners, cloud infrastructure, and security operations platforms through a common AI-driven security layer. Such a platform could automatically assess an API from the moment it is designed and continue monitoring it until retirement. During design, Generative AI could analyze API specifications and recommend secure authentication, authorization, encryption, validation, and error-handling mechanisms. During coding, it could perform security-oriented code reviews and generate test cases. During CI/CD execution, it could evaluate newly introduced APIs against organizational policies and known vulnerability patterns. During deployment, it could automatically verify gateway configurations, certificates, authentication policies, and network exposure. At runtime, the system could continuously analyze API behavior and identify deviations from normal usage. Future research should also focus on AI-driven API discovery because large enterprises frequently have undocumented endpoints and multiple versions of similar interfaces. Automated discovery could combine traffic analysis, source-code inspection, cloud metadata, and API specifications to create continuously updated inventories. Generative AI could then classify APIs according to business sensitivity, data exposure, authentication requirements, and risk. Another valuable research direction is intelligent API dependency analysis. When an API is changed or retired, AI could identify affected applications, consumers, and services and generate migration plans. This would reduce the risk of security incidents caused by outdated interfaces. AI-assisted security testing could also become more sophisticated by generating attack scenarios based on API semantics and business logic rather than relying solely on generic fuzzing. Future systems could simulate realistic attacker behavior while monitoring whether security controls detect and prevent the activity. Such capabilities would enable organizations to identify vulnerabilities earlier and continuously improve API security throughout the software lifecycle.

Future implementation studies should evaluate the proposed framework using realistic multi-cloud environments and quantitative security metrics. Experimental testbeds could include multiple cloud providers, container platforms, serverless workloads, enterprise APIs, databases, identity systems, external services, and DevSecOps pipelines. Researchers could introduce controlled security incidents involving credential theft, API abuse, injection attacks, broken authorization, excessive data exposure, malicious configuration changes, compromised dependencies, and cloud workload exploitation. The performance of Generative AI could then be compared with conventional security operations and AI-assisted hybrid approaches. Important evaluation metrics should include vulnerability-detection accuracy, false-positive and false-negative rates, incident-analysis time, response quality, API security-test coverage, mean time to detect, mean time to respond, and application-performance overhead. Future research should also investigate how AI affects security-team productivity. Metrics such as analyst workload, investigation duration, number of alerts processed, and quality of remediation decisions could demonstrate whether Generative AI produces measurable operational improvements. Cost analysis is equally important because continuous AI processing of large cloud datasets can be expensive. Research could examine techniques such as selective telemetry ingestion, intelligent event summarization, model routing, smaller specialized models, edge processing, and adaptive retrieval to reduce computational requirements. Another important research direction is interoperability. Enterprises rarely operate a single security platform, so future architectures should support standardized interfaces for exchanging security events, policies, API specifications, threat intelligence, and response actions. Research should also examine privacy-preserving approaches that allow sensitive security information to be analyzed without unnecessary exposure. Federated learning, confidential computing, access-controlled retrieval, and data minimization could help organizations meet privacy and

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

regulatory requirements. These experiments would provide stronger evidence regarding the practical scalability, reliability, cost, and effectiveness of Generative AI-based security architectures.

The long-term future of this research lies in developing trustworthy autonomous security ecosystems that can continuously learn from incidents, improve API protections, predict emerging threats, and support proactive cyber resilience. Future systems could establish a continuous feedback loop between security operations and application development. When a new attack pattern is detected, Generative AI could automatically transform the incident into updated detection rules, API security tests, threat-hunting queries, secure coding recommendations, and policy changes. This would allow organizations to learn from every security event and rapidly incorporate that knowledge into future application releases. Predictive analytics could further improve security by identifying combinations of vulnerabilities, excessive privileges, exposed APIs, and abnormal behaviors that indicate a high probability of compromise. Digital twins of cloud environments could be used to simulate potential attacks and test AI-generated defensive strategies without affecting production workloads. Future research could also investigate reinforcement learning for selecting remediation strategies, provided that strict safety constraints prevent uncontrolled experimentation. Another important direction is explainable and auditable AI. Security professionals and regulators may require evidence demonstrating how an AI system reached a decision and why a particular recommendation was made. Future systems should therefore provide traceable reasoning evidence, source attribution, policy references, and complete audit records. Governance must also address responsibility for AI-generated actions and establish clear boundaries between automated and human-approved responses. Privacy should remain a central consideration because intelligent security platforms may process extensive information about users, applications, and organizational behavior. Future architectures should incorporate data minimization, role-based access, privacy-preserving analytics, and secure AI infrastructure by design. Ultimately, future work should aim to transform Generative AI from an analytical assistant into a trusted component of continuous cyber resilience. Such a system would not simply identify vulnerabilities after they appear but would continuously understand the evolving cloud environment, anticipate security risks, improve API controls, support secure software development, and assist in rapid recovery from incidents. Achieving this vision will require interdisciplinary research combining artificial intelligence, cloud security, API engineering, software development, privacy, governance, and human-computer interaction. The result could be a new generation of cloud security architectures that are intelligent, adaptive, explainable, and capable of continuously improving their defensive capabilities while maintaining strong human oversight and organizational control.

## REFERENCES

1. Kumar, S., Dwivedi, M., Kumar, M., & Gill, S. S. (2024). A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services. *Computer Science Review, 53*, 100661. https://doi.org/10.1016/j.cosrev.2024.100661
2. Kumar, A., Wadhwa, M., Kalla, D., Konduru, S. C., Nandawat, C., & Sharma, M. (2025, October). Benchmarking the Trade-Offs in Object Detection: Accuracy, Speed, and Energy Efficiency. In International Conference on Artificial Intelligence and Networking (pp. 410-422). Cham: Springer Nature Switzerland.
3. Gummadi, V. P. K. (2021). Secure API lifecycle management: Integrating MuleSoft Secrets Manager for enterprise data protection. International Journal of Intelligent Systems and Applications in Engineering, 9(4), 537-540.
4. Soundappan, S. J. (2023). Machine Learning Based Predictive Models for Secure Financial Transactions and Cyber Threat Detection. International Journal of Engineering & Extended Technologies Research (IJEETR), 5(1), 5966-5975.
5. Narra, S. L. (2025). Human-AI Collaboration in Identity Security: When Should AI Decide?. Journal of Computer Science and Technology Studies, 7(7), 191-197.
6. Challa, R. (2024). High-Availability HPC Cluster Design for Mission-Critical Public Infrastructure: Lessons from Energy Grid and Government. International Journal of Engineering & Extended Technologies Research (IJEETR), 6(6), 9305-9309.
7. Sudhan, S. K. H. H., & Kumar, S. S. (2016). Gallant Use of Cloud by a Novel Framework of Encrypted Biometric Authentication and Multi Level Data Protection. Indian Journal of Science and Technology, 9, 44.
8. Karan Nisar. (2025). Why enterprise agent deployments fail: A field taxonomy. International Journal of Computer Technology and Electronics Communication (IJCTEC), 8(5), 11592–11605.
9. Awopejo, T. E., Adigun, P. O., & Oyekanmi, T. T. (2023). Emerging applications of artificial intelligence and machine learning in modern earthquake science. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 6(1), 8142–8154.
10. Anbazhagan, R. S. K. (2016). A Proficient Two Level Security Contrivances for Storing Data in Cloud.
11. Tyagi, N. (2025). The Compliance Horizon: Anticipating Regulatory Change in Financial Services and Artificial Intelligence. International Journal of Research and Applied Innovations, 8(2), 11227-11232.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

12. Chandra Shekhar, P. (2021). Driving agile excellence in insurance development through shift-left testing. International Journal for Multidisciplinary Research, 3(6), 1-18.
13. Rella, B. P. R., Chakraborty, A., Shah, S. B., Ghosh, H., Gautam, S., Dhirwani, B. A., ... & Mutyam, N. (2024). AI-engine-based acceleration for high-performance programmable system-on-chip designs. Journal of Computational Analysis and Applications, 32(1).
14. Raja, G. V. (2020). Metadata gets a makeover: The machine learning approach. International Journal of Computer Technology and Electronics Communication, 3(6), 2900-2903.
15. Anbazhagan, K. (2024). Trustworthy and Adaptive AI Systems for Enterprise Analytics Cybersecurity and Decision Optimization Using API-First and Cloud-Native Architectures. International Journal of Technology, Management and Humanities, 10(03), 65-74.
16. Karan Nisar. (2025). Why enterprise agent deployments fail: A field taxonomy. International Journal of Computer Technology and Electronics Communication (IJCTEC), 8(5), 11592–11605.
17. Jayaraman, S., Rajendran, S., & P, S. P. (2019). Fuzzy c-means clustering and elliptic curve cryptography using privacy preserving in cloud. International Journal of Business Intelligence and Data Mining, 15(3), 273-287.
18. Anand, L. (2024). AI-Powered Cloud Cybersecurity Architecture for Risk Prediction and Threat Mitigation in Healthcare and Finance. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 7(Special Issue 1), 5-12.
19. Juvvadi, R. R. (2017). From record-to-report to insight-to-action: Re-engineering the finance operating model for the cloud ERP era. International Research Journal of Innovative Engineering (IRJIE), 1(2), 371–376.
20. Pokala, H. K. (2025). AIR-DEA: A multi-criterion mathematical model for evaluating artificial intelligence systems. International Journal of Applied Mathematics, 38(8s), 4818–4830.
21. Bellundagi, M. (2022). Performance Optimization Techniques for Enterprise Java Applications Using Middleware and Messaging Systems. International Journal of Computer Technology and Electronics Communication, 5(3), 5158-5168.
22. Narapareddy, V. S. R. (2022). Strategies for Integrating Services with External Systems Via Rest & Soap. Universal Library of Engineering Technology, (Issue).
23. Ganapathy, S. K. (2024). Threat Modeling for Federated SSO and MFA Systems: STRIDE-Based Analysis of Attack Vectors. American Academic Journal, 55-73.
24. Bajarang Bhagwat, V. (2023). Optimizing payroll to general ledger reconciliation: Identifying discrepancies and enhancing financial accuracy. Journal of Advance and Future Research, 1(4).
25. Awopejo, T. E., Adigun, P. O., & Oyekanmi, T. T. (2023). Emerging applications of artificial intelligence and machine learning in modern earthquake science. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 6(1), 8142–8154.
26. Suddala, V. R. A. K. (2024). Machine learning for operational excellence: Real-world applications. International Journal of Future Innovative Science and Technology (IJFIST), 7(6), 13917.
27. Kundurthy, O. H., Ghadiyaram, R., & Vanam, L. (2025, September). Global AI Regulation Review: Comparative Insights from EU, US and APAC. In International Conference on Intelligent Computing and Communication (pp. 422-434). Cham: Springer Nature Switzerland.
28. Alex Mathew. (2023). Threat defense through cyber fusion. International Journal of Computer Science and Mobile Computing, 12(1), 24–27. https://doi.org/10.47760/ijcsmc.2022.v12i01.003
29. Onteddu, A. R., Bandhela, R. R., & Kundavaram, R. R. (2024). Enhancing E-Commerce Product Recommendations through Data Engineering and Machine Learning. Economic Sciences, 20(1), 171-183.
30. Chaba, A. (2018). A platform-independent API integration architecture for scalable enterprise commerce solution. International Journal of Research Publication in Engineering, Technology and Management, 1(1), 9–13.
31. Macha, Y., & Pulichikkunnu, S. K. (2024). A Data-Driven Framework for Medical Insurance Cost Prediction Using Efficient AI Approaches. IJRAR, 11(4), 887-893.
32. Wadhwa, R. (2023). Ensuring data consistency in enterprise systems through event driven microservices and distributed transaction management. International Journal of Computer Science and Engineering Research and Development, 6(1), 24-51.
33. Kumar, A., Wadhwa, M., Kalla, D., Konduru, S. C., Nandawat, C., & Sharma, M. (2025, October). Benchmarking the Trade-Offs in Object Detection: Accuracy, Speed, and Energy Efficiency. In International Conference on Artificial Intelligence and Networking (pp. 410-422). Cham: Springer Nature Switzerland.
34. Pasumarthi, H. (2024). AI-driven forecasting and optimization in distributed systems: Lessons from retail, lending, and healthcare platforms. International Journal of Research and Applied Innovations, 7(3), 10786-10790.
35. Sudhan, S. K. H. H., & Kumar, S. S. (2015). An innovative proposal for secure cloud authentication using encrypted biometric authentication scheme. Indian journal of science and technology, 8(35), 1-5.

|www.ijrpetm.com | ISSN: 2454-7875 | editor@ijrpetm.com |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 6, November-December 2025||

DOI:10.15662/IJRPETM.2025.0806046

36. Singh, I. K. (2025). Intelligent software validation frameworks for mission-critical enterprise applications using AI and knowledge graphs. International Journal of Research and Applied Innovations, 8(4), 12723-12735.
37. Soundappan, S. J. (2025). Federated Multi-Cloud Data Governance for Secure and Compliant Enterprise Analytics Systems Framework. International Journal of Research and Applied Innovations, 8(2), 11251-11261.
38. Karnam, V. S. (2024). Adaptive and federated test automation using AI in distributed multi-cloud and hybrid infrastructure environments. International Journal of Future Innovative Science and Technology (IJFIST), 7(4), 111–121.
39. Vani, M., & Dadlani, D. (2023). Smart home – "Aashraya" IoT automation system. International Journal of Computer Technology and Electronics Communication, 6(1), 6393–6403.
40. Hoque, M. J., Akter, F., & Mohammad, A. R. (2019). Data-Driven Decision Support System for Restaurant Business Optimization. American Journal of Economics and Business Management, 2(2).
41. Kumar, A., Wadhwa, M., Kalla, D., Konduru, S. C., Nandawat, C., & Sharma, M. (2025, October). Benchmarking the Trade-Offs in Object Detection: Accuracy, Speed, and Energy Efficiency. In International Conference on Artificial Intelligence and Networking (pp. 410-422). Cham: Springer Nature Switzerland.