NextGen Scientific Publication
American International Journal of Computer Science and Technology
Volume 7 Issue 5, Pg. No. 13-27, AIJCST-V7I5P102, 2025
https://doi.org/10.63282/3117-5481/AIJCST-V7I5P102

Original Article

Adaptive DevSecOps: Integrating AI-Driven Threat Detection in Continuous Delivery Pipelines

***Guru Pramod Rusum[1], Kiran Kumar Pappula[2]**
[1,2] *Independent Researcher. USA.*

Abstract:

DevOps and Continuous Delivery (CD) have transformed the landscape of software development at a rapid rate, creating new security concerns that cannot be effectively resolved by older approaches. DevSecOps became a paradigm shift or model in which security was inserted into the software development process. Nevertheless, the growing complexity of microservices, cloud-native environments, and the changing threat environments all require more flexible and smarter solutions. The paper suggests an adaptive DevSecOps model powered by AI that incorporates real-time threat detection in CI/CD pipelines and allows for the prevention and mitigation of security threats proactively, without affecting the speed of development. We propose a system that integrates static and dynamic analysis that uses machine learning, behavioural anomaly detection and continuous learning techniques to secure pipelines. Model training and inference occur with logs, code repositories, and configuration data as input, and the model trains over time via feedback loops. The effectiveness and feasibility of the suggested approach have been confirmed by the experiments, which achieved extremely high detection accuracy (up to 97 percent) with low false positive rates and minimal latency. Real-world case study demonstrates system scalability and responsiveness in real-life scenarios. The integration issues, ethics, and challenges of adoption in the industry are also mentioned in the paper. Lastly, it presents research directions to be pursued in the future, which lie within the areas of explainability, adaptive response orchestration, and federated learning in collaborative threat intelligence.

Keywords:

Adaptive Devsecops, AI-Driven Security, Threat Detection, CI/CD Pipelines, Machine Learning, Anomaly Detection.

Article History:

Received: 15.07.2025

Revised: 17.08.2025

Accepted: 29.08.2025

Published: 06.09.2025

1. Introduction

The growing use of cloud-native applications and the increasing need for digital services have transformed the way software is created, distributed, and supported. [1-3] Organizations have adopted DevOps, which is a process that has combined the development and operations staff to increase efficiency in the software delivery process to satisfy the needs of fast feature delivery and ensure continuous improvements are made. In today's fast-changing world, traditional security methods like post-development testing and manual audits are no longer sufficient. DevSecOps, a method that includes security at every stage of the software development life

cycle (SDLC), aims to address this gap. However, implementing DevSecOps can be challenging due to the evolving threat landscape, which includes automation, AI, and new ways to exploit old guards. Traditional methods like manual review and static security rules are less likely to find vulnerabilities that don't change, while advanced attack vectors change quickly. The paper proposes a DevSecOps model that can adapt over time and incorporates AI-based threat detection into the CI/CD pipeline. This integration not only increases awareness of threats but also keeps security practices current, enabling teams to adapt to new situations. The paper analyzes the benefits and implementation difficulties of such a system, along with a case study demonstrating its practical effects in a software development context. It suggests that machine learning and AI technologies can be used to identify weak spots and learn from past events to better deal with threats.

2. Background and Related Work

2.1. DevSecOps Evolution and Security Challenges

DevSecOps is a big change in how software is made because it adds security to the software development lifecycle (SDLC). This integration makes security a team effort by bringing together the development, operations, and security teams. Microservices, containerization, and cloud-native applications are all examples of modern IT ecosystem trends that call for better security systems. These architectures also make it easier to attack things like misconfigured containers and insecure APIs, which makes them more flexible and scalable. But there are still problems with putting DevSecOps into practice.

Cyber-attacks not only against the application logic but also against the application infrastructure of organisations are rising both in complexity and frequency. As development teams are being sped up in the amount of time between releases, security teams are being stretched due to alert fatigue, as the current scanning tools overwhelm them with too many warnings generated. Possible outcomes of this strain are the possible presence of critical vulnerabilities, which remain unidentified and unaddressed by the end of the cycle. Furthermore, most of the currently implemented security processes are reactive, time-static, signature-based detection based rules. Such processes are fixed and not dynamic, as the world of modern-day software distribution and dynamic malware is changing on a regular basis.

2.2. Threat Detection in CI/CD Pipelines

DevOps pipelines have enabled software release by establishing Continuous Integration and Continuous Delivery (CI/CD) practices that have contributed to a considerable acceleration of software releases. However, they have also been imposing security weaknesses that should be resolved in real-time. The security tools that are traditionally used and are either obsolete or checked on a regular basis cannot be used in such a manner. Inclusion of threat detection at the task level of CI / CD will give the security checks as a part of the development process, and this will not require manual intervention. The move reduces the time taken to detect and deploy a solution to deal with vulnerabilities and the removal of code in the insecure stage before moving it to production.

New AI-based security tools have been designed in the recent past to contribute greatly to the evolution of CI / CD pipeline development. The tools are used to deliver on-demand scanning of the codebase, libraries, and configuration with AI-enhanced Static Application Security Testing (SAST) and Dynamic Application Security (DAST). In addition, behavioural analytics tools have baselines of usual working behaviour and detect anomalies, e.g., the ability to act outside the baseline, i.e., a violation of the system or an unforeseen update of the code, which can mean an attack. This is further enhanced by autonomous compliance checking that can both maintain regulatory and internal security standards over the pipeline and decrease the possibility of non-compliance with regulations, and make us prepared for audits.

2.3. Role of AI in Cybersecurity

Artificial Intelligence (AI) is soon becoming the bread and butter of the cybersecurity sector of the current era, and AI can potentially augment its capabilities beyond the capabilities allowed by the traditional systems of rules. Artificial intelligence has the capacity to ingest and analyze large quantities of structured and unstructured data on request in order to enable the tracing of small anomalous and trends that would not have been noted otherwise. This is especially useful in detecting zero-day attacks and new patterns of attacks that could not be detected with signature-based detection systems.

One of the best things about AI in cybersecurity is that it can put threats in order of importance. AI models can use information about past attacks to find the vulnerability that is most likely to be exploited and quickly suggest ways to fix it. Being able to predict lets companies go from being watchful to being active. AI also reduces false positives by learning from past incidents and raising its

detection threshold. AI can also help automate many of the repetitive tasks, such as alert triaging, threat model updates, and log correlation. This would free up human analysts to do more complicated work in investigations and other tasks that require a strategic point of view. Machine Learning Security Information and Event Management (SIEM) systems, which have better AI features and more advanced ML algorithms, are now the main part of intelligent, automated, and scalable cyber defense systems.

2.4. Related Research in Adaptive Security Systems

A lot of research has been done on adaptive security systems in the last few years because static-based and rule-based systems have not been able to solve security problems. Safety Adaptive security is a feature of security systems that lets them change their own protection or security property when they see new threats, changes in context, or other feedback on how they work. This ability to adapt is a valuable skill that is already seen to be crucial in the complex and ever-changing world of DevSecOps.

Experiments on things like the security of the Internet of Things (IoT) have shown that adaptive models can be used to find unknown threats. When given a wide range of data sets that are always changing, machine learning algorithms will be able to find an unknown attack pattern. Adaptive systems in DevSecOps can take in telemetry data and use it to improve models and detection of threats in real time and in the future. Such systems assist in proactive defence by detecting security holes before they are exploited and suggesting, or even automatically applying, measures. Furthermore, adaptive security automates tasks and minimises the time spent by security teams on manual activities, allowing them to react more accurately and faster to incidents. The future of the DevSecOps ecosystem is expected to rely increasingly on adaptive security architectures and artificial intelligence-driven security models, which will become a central aspect of protecting the software delivery lifecycle.

3. System Architecture and Methodology

3.1. Adaptive DevSecOps Framework Overview

The Adaptive DevSecOps framework under consideration aims to facilitate the seamless integration of AI-based threat detection into the continuous integration and delivery (CI/CD) pipeline. It aims to transform conventional DevSecOps into smart, reactive, and ongoing security guarantees through intelligent and adaptive strategies. [7-10] The architectural model consists of several layers connected in a certain way, each of which serves a particular purpose, such as a code development layer, a threat detection layer, a policy implementation layer, an incident response layer, and a continuous learning layer. The framework is an automation, machine learning and behavioural analytics-based security framework that delivers real-time, situation-aware security without sacrificing the velocity or responsiveness of contemporary software development.

At the top is the Developer's Level, which contains Infrastructure as Code (e.g., Terraform, Ansible), code repositories (e.g., GitHub, GitLab), and issue tracking systems (e.g., JIRA). These are the elements that trigger the software build process, adding problems and initiating builds when codes are committed. This is followed by passing into the CI/CD Pipeline Layer, where the continuous integration engines (e.g. Jenkins, GitLab CI) compile, test, and build the application. The deployment and storage of artefacts on this layer are also handled with the help of such tools as Docker Hub, Nexus, and Kubernetes.

The Adaptive Framework has the AI-Driven Threat Detection Layer as its core, which is characterized by the CI/CD stack beneath. It uses machine learning and an anomaly detection-powered threat detection engine. This engine collects behaviour perpetually at codebase levels, in the pipeline, and in the access log. It is used in conjunction with external threat intelligence data feeds (e.g., CVEs, STIX, MITRE ATT&CK) and a behavioural analytics engine to detect suspicious activity or emerging threats. The security policy engine should intuitively implement the security policy based on such input, encompassing security-related definite conditions and using adaptive thresholds, in real-time.

The detection knowledge is fed into the Security Operations Layer, where useful alerts and risk scores are displayed in dashboards. If the risk number exceeds a certain threshold, incident response engines (via SOAR tools) are triggered. These tools will automate the process of mitigating threats and send alerts to the SIEM systems, enabling centralised logging of all data and providing compliance reporting. Human analysis can also be provided through incident labelling and the provision of context, which is facilitated within the feedback loop. Lastly, the Adaptive Learning & Feedback Layer enables continuous improvement of the model through retraining and auditing. This layer provides manual review and labelling interfaces, which help minimise false positives and ensure the relevance of models. The retraining module enhances the effectiveness of AI models in addressing emerging and changing threats by

detecting and considering feedback indicators through model drift. A closed-loop system like this is not only possible to implement from a proactive security standpoint but also strengthens long-term resiliency within the dynamic CI/CD environment.

3.2. AI-Driven Threat Detection Layer

These DevSecOps systems are based on the AI-Driven Threat Detection Layer of the analysis, which forms the centre of the adaptive DevSecOps architecture. Machine learning and anomaly detection models are increasingly used in real-time security threat detection. These models learn about system operations and identify changes that may indicate new or larger threats. They analyze various input data, such as access logs, CI/CD pipeline logs, and source code, to set behavioral baselines and identify outliers. The threat detection engine, the most crucial part of this layer, uses advanced AI methods to analyze telemetry data and identify potential threats in the system. This helps to prevent potential attacks and ensures the safety of users. The use of this engine is supported by the use of the Behavioural Analytics Module, which makes it possible to build a correlation between the data from different sources and trace the correlation between the modification of the code, access trace, and executed pipelines.

External threat intelligence feeds, such as CVE databases, MITRE ATT&CK, and STIX repositories, also help enrich this module and enhance the process of detecting previously unknown vulnerabilities and adversary tactics by utilising the most current knowledge of known vulnerabilities and adversarial tactics. The detection results are converted into actionable policies through the Security Policy Engine, which utilises adaptive thresholds and rule-based logic to enforce mitigative measures based on dynamic risk scores.

3.3. Integration with CI/CD Pipelines

The role of AI-driven threat detection in CI/CD pipelines is one of the most critical to delivering secure, high-velocity software delivery. [11-13] This integration ensures that the security checks are not undertaken as individual events but rather incorporated fully in the software development and deployment lifecycle. The AI layer enables communication with any code as it progresses through various stages in the code pipeline, including source control, build, test, and deployment, and performs real-time security scans at each stage, without creating artificial bottlenecks in the pipeline.

At the build and test phases, the framework also uses AI-augmented Static and Dynamic Application Security Testing (SAST, DAST) methods to scan the source code and runtime behavior of applications to identify vulnerabilities, after creating the artifacts and storing them in repositories such as Docker Hub or Nexus, the detection layer proceeds trying to detect configuration weaknesses and dependency vulnerability of the artifacts. Once the system is installed in an environment like Kubernetes or in the cloud, it repeatedly checks the behaviour of systems to find any misconfigurations, unauthorised access and lateral movements that can indicate a breach.

Such an approach to real-time tracking and synchronization of the threat intelligence makes security a continuous process that should be considered as context-aware, but not as a process performed periodically during automated audits. Moreover, the integration of pipelines with tools such as Jenkins or GitLab CI will make it such that when threats are identified, the framework may be used to issue an alert and a remediation automatically. The degree of sharing of security and development pipelines is so close that it helps to detect vulnerabilities at the earliest stage, reduces the speed at which solutions can be achieved and generally improves resilience whilst improving the speed of delivery.

3.4. Feedback Loop and Continuous Learning Mechanism

An important attribute of the suggested structure is adaptive learning, because it is provided by the presence of a loop of continuous feedback. Security in CI/CD environments is dynamic, and consistent models quickly become obsolete because of the changing threats, the development process used and the design of the infrastructure. To this end, the system has had an Adaptive Learning and Feedback Layer that loops the system between detection, input by an analyst, and training the model again.

The Audit and Labelling Interface in the layer allows security analysts to manually review and label flagged incidents, fine-tuning the model to distinguish between real threats and false positives. The labeled data is sent to a Continuous Feedback Module, which generates feedback signals to retrain machine learning models. The Retraining Module detects model drift, ensuring the AI adapts to new threat trends and input. This feedback loop transforms the security structure into a self-learning system, ensuring more accurate detection and reducing analyst overwhelm. The continuous learning process strengthens security and allows it to adapt to the changing threat environment in software delivery pipelines.

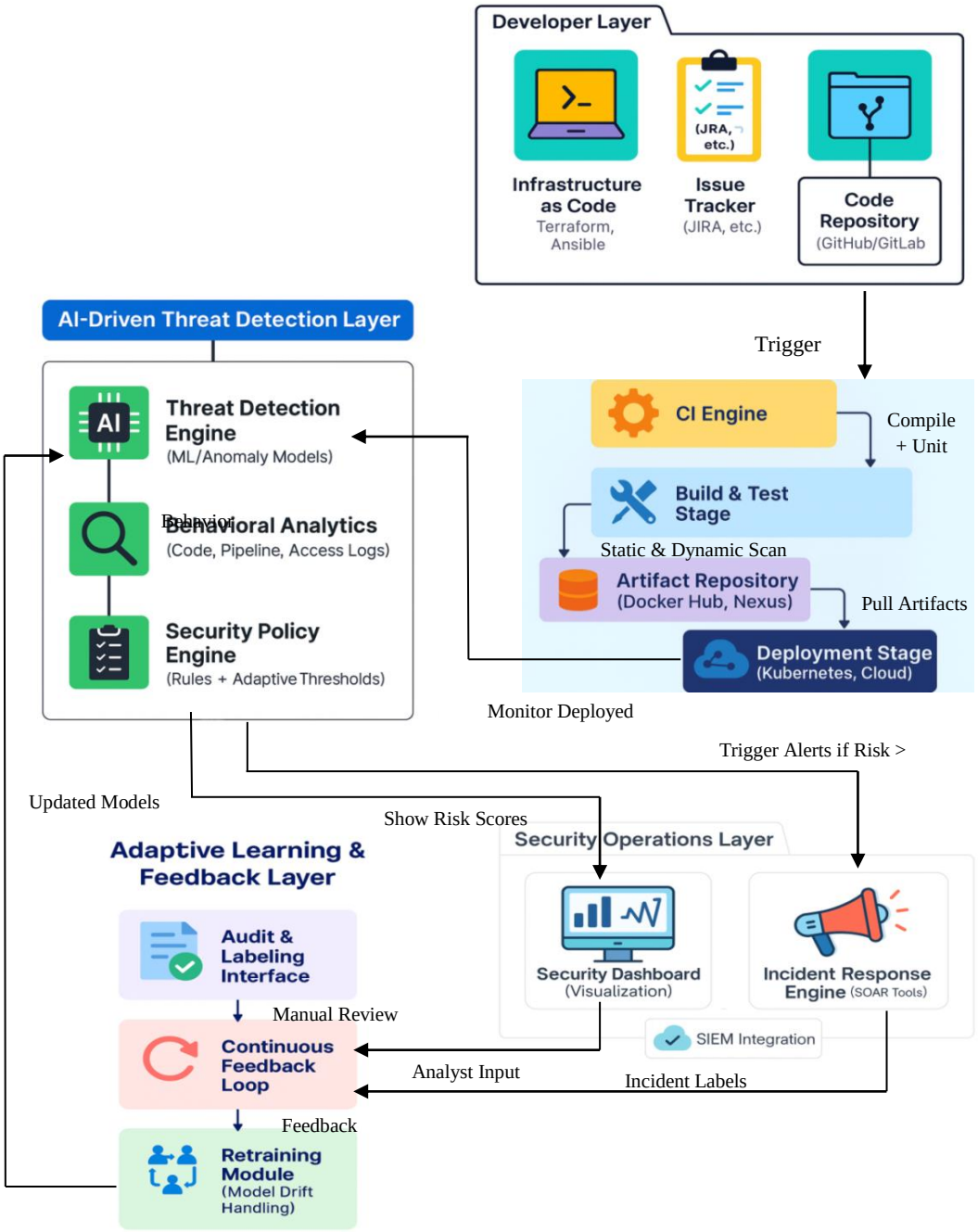

Figure 1. Adaptive Devsecops Framework with AI-Driven Threat Detection In CI/CD Pipelines

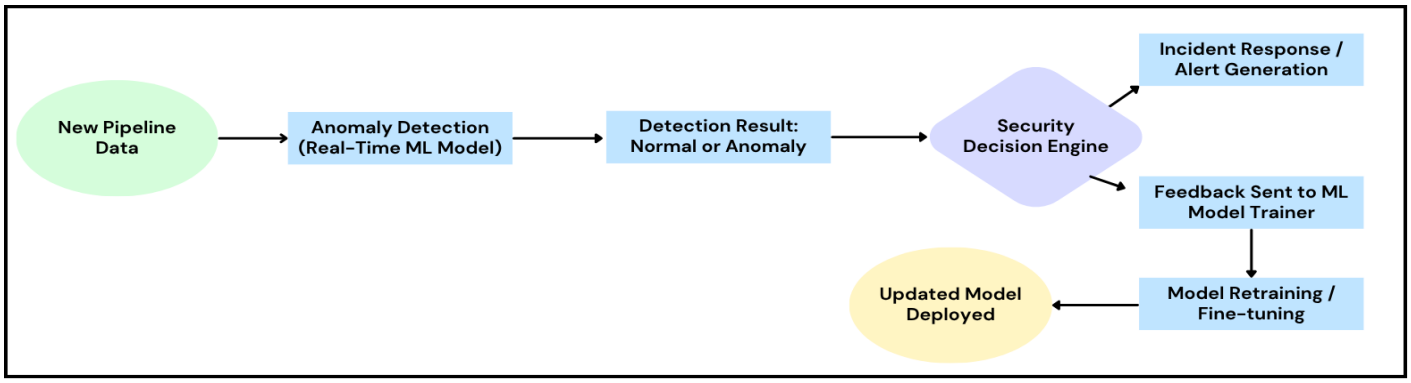

Figure 2. Real-Time Anomaly Detection and Feedback Loop

4. Threat Modelling and Detection Workflow

In the Adaptive DevSecOps model, threat modeling means systematically finding, reviewing, and reducing possible security threats throughout the CI/CD pipeline. [14-17] As software systems become more distributed and dynamic, the attack surface is getting bigger. This includes code repositories, build automation tools, container registries, and deployment environments. The threat modeling process is crucial for identifying vulnerabilities and ensuring robust detection processes. It uses taxonomies, structured data flow analysis, and real-time telemetry to identify threats. AI is then applied to match these models to observable signals, creating a proactive and adaptive detection workflow. Key assets like source code, infrastructure scripting, access credentials, and container images are identified and tracked throughout the CI/CD process. Threats are classified according to a DevSecOps-specific taxonomy, and detection mechanisms are implemented to monitor unusual behavior. The detection layer uses telemetry data and threat models to generate dynamic risk scores. This creates an entire threat response circle beginning with checks before build and continuing with monitoring of deployments.

4.1. Threat Taxonomy in CI/CD

Threat taxonomy, within the context of DevSecOps design, designates a hierarchical categorisation of potential security threats that can occur during software delivery and development. Defined taxonomy is necessary to produce accurate threat identification, effective alert triage, and policy applicability. The automation levels, frequent code changes, and a great number of third-party tools and services make CI/CD environments particularly vulnerable to broad categories of threats. Common categories of threats are code injection (e.g. malicious commits, backdoors), pipeline poisoning (poisoning build scripts or container images), credential leak (exposure of secrets in logs or environment variables), and insider threats (malicious changes of individuals with higher privileges). Other top priorities are supply chain attacks and configurable drift, where malicious dependencies have been added to the infrastructure build and configuration, and the system has been left vulnerable.

The Lack of secure APIs or misconfiguration of infrastructure-as-code (IaC) scripts can also be used by attackers to compromise secure infrastructure or have lateral access. Such a framework contains a taxonomy that conforms to industry standards, including MITRE ATT&CK Enterprise and OWASP Top 10, within DevOps, so that it is diffuse to threats on the one hand and congruent with modern security practices on the other. By applying the mapping of detection models to this taxonomy, this implies that the system can be programmed to actively scan and detect given behaviours and anomalies and the related known and emerging threats. The mapping allows the system to come up with sensible risk scores and prioritise response measures based on estimating the severity of the threat and the relevance of that threat to a situation at hand.

4.2. Data Sources: Logs, Code Repositories, Pipelines

The sensitivity and accuracy of data sources are the determining factors of offering good threat detection through AI. A DevSecOps pipeline produces data continuously using a process of data software delivery by an assortment of tools and platforms. The framework uses this data to detect suspicious behaviour patterns and notice changes in the pattern, and to predict threats with improved ability as data is fed into it. Logs are one of the major sources of essential data. These are access logs, audit trails, system logs, and build logs that allow viewing of activities like user authentication, committing of code, system change, as well as executions. As an example, brute-force logins will be recorded in access logs, and build logs may contain evidence of dependency injection

attempts. These logs are fed into the behavioral analytics engine, which makes a profile of normal pipeline behavior so that it can warn of problems.

GitHub and GitLab are examples of this kind of repository. They both serve as a source and a viewpoint dimension of code-based threats. You can find out what happened, when it happened, and why in changes made by developers, pull requests, branch activity, and other issue-related data. Machine learning models look at commit patterns, changes to code, and metadata to find possible injection of malicious code or changes to code that weren't authorized. The pipeline data itself, which includes workflow definitions, test results, build artifacts, and deployment scripts, gives you a better idea of how software components are built and deployed. This information is very important for finding problems with CI/CD flows, test coverage, or deployment that aren't right. The framework can create a complete, unified model of threat visibility by actively mining and cross-correlating these different data sets. This model can be used for real-time detection and for more in-depth trend analysis.

4.3. ML Model Training and Feature Engineering

The effectiveness of AI-driven threat detection in DevSecOps pipelines is significantly contingent upon the caliber of machine learning (ML) models employed, which are, in turn, critically dependent on a robust model training methodology and thoughtfully crafted features. The proposed adaptive system's ML models learn from a mix of old threat data, pipeline logs, behavioral telemetry, and contextual metadata, such as code repositories and build tools. These models find patterns that are linked to known and new threats, making them a scalable way to find threats.

Feature engineering is the most important step in this process. It cleans up raw data and turns it into useful input for machine learning algorithms. Some of these features are the frequency of coded commits, the entropy of file changes, unexpected API access times, test coverage deviations, changes in the anomaly of infrastructure provisioning patterns, and wrong access behaviors. The framework uses supervised and unsupervised learning strategies to identify threats. Supervised learning methods, like random forests and XGBoost, use labeled datasets to identify known attacks. Unsupervised learning methods, like autoencoders and clustering algorithms, identify attacks without labeling vectors. The system constantly updates the model to address changes in attacker behavior and concept drift. It uses new information from pipeline processing and security operations analysts. The retraining module ensures model accuracy and adaptability to emerging threat patterns. Privacy-preserving methods like data anonymisation and federated learning can be used for training on sensitive enterprise data.

4.4. Real-time Anomaly Detection Flow

The adaptive threat identification framework finds threats and stops breaches by using real-time anomaly detection. It can run at any point in the CI/CD pipeline and uses AI-based analytics to check telemetry streams. The detection engine gets data from different places, like build logs, access logs, deployment events, and code repository activity. It then runs trained models against this data in almost real time.

The first step in finding anomalies is to make a profile of the base. This means figuring out what normal behavior looks like for each user, service, and stage of the pipeline over time. For instance, a baseline could show how often a certain developer commits code (9 AM to 6 PM, weekdays) or how long it takes for one of the integrals that run to build the code to run (3 minutes on average with 90% test coverage). After establishing a baseline, telemetry received by the detection engine is compared to the expected values. Deviations, such as a midnight deployment, unauthorised access to secrets, or an unexpected decline in the test success rate, can detect anomalies.

These anomalies are also rated through a dynamic risk assessment model that considers the severity, frequency, and contextual significance of the event. To illustrate, a low-risk change made to a minor code formatting at an atypical time may create an alarm. In contrast, a change related to a critical infrastructure-as-code change resulting in a score that dictates the need to respond should not be taken lightly. The system may also automatically elicit reactions to pre-configured thresholds, e.g., halting deployment, notifying the security operations centre (SOC), or enforcing policy rollback. Additionally, to minimise noise and false positives, the anomaly detection layer incorporates a feedback mechanism that integrates both the security operations layer and the adaptive learning loop. The correction of analyst input and label errors is returned to the system to improve the model thresholds and feature weights. These are closed-feedback loops, and they make the system more accurate and contextual over time, providing actionable insights while minimising alert fatigue for DevSecOps teams.

5. Implementation and Experimental Setup

5.1. Tooling Stack (e.g., Jenkins, GitLab, AI Models Used)

The DevSecOps adaptive framework was deployed with the help of an integrated set of tooling software that leveraged industry-standard DevOps tools and proprietary AI pieces. [18-20] The pipeline was orchestrated through Jenkins and GitLab CI, as they have a rich collection of available plugins, and they are easily connected to other tools. They were deployed in GitHub code repositories and controlled via Terraform and Ansible infrastructure, which shows the actual DevOps-based features in the world. The build process involved a measure of automated unit testing, artefact storage through Docker Hub and deployment through Kubernetes clusters in a cloud-based environment (AWS and GCP).

The AI-based threat detection layer was developed with a modular detection core that creates deep learning models based on Python and TensorFlow, as well as classical machine learning techniques (Random Forest, XGBoost) based on Scikit-learn. The unsupervised methods, which included Autoencoders, Isolation Forest, and the DBSCAN Clustering algorithm, were used to detect anomalies, whereas the labelled attack datasets were used to train the supervised classifiers. The threat intelligence feed was connected through APIs, such as MITRE ATT&CK, STIX/TAXII, and CVE databases. The system utilised Grafana, Kibana, and a security dashboard that we designed for monitoring operations and visualisation purposes.

5.2. Datasets and Simulation Environment

The experiment was designed to replicate real-life CI/CD procedures with ingrained weaknesses and abnormal functioning. They created synthetic data based on CI logs from open-source projects on GitHub, augmented with access logs, code commits, and infrastructure modifications, to develop an inclusive telemetry picture. Also, the pre-training and benchmarking of AI models were done on publicly available security datasets, including CICIDS2017, UNSW-NB15, and CodeSecurity-AI. These datasets had labelled instances of different types of attacks, including privilege escalation, injection attacks, and code tampering, which were mapped to CI/CD events.

The test environment was designed to simulate a real-life scenario of the attacks, seeding controlled anomalies that included unauthorised deployments, script injections into Docker files, malicious access patterns, and code injections into critical modules. In order to simulate the multi-stage CI/CD workflow, a sandbox using Kubernetes was deployed to track logs and metrics every step of the workflow so that they could be analyzed. Security tools, including OWASP ZAP and Metasploit, were used to perform synthetic attacks in order to analyze the reaction of the models. To check the false positive rates, the simulation was supposed to implement the benign developer activity to improve the robustness of the model put to the test.

5.3. Metrics for Evaluation (Accuracy, Precision, Detection Latency)

The functioning of the proposed adaptive DevSecOps framework has been tested under a set of established criteria so as to measure its performance, as well as effectiveness in detecting threats. This second criterion was tested according to the accuracy and precision of real security threats detection without associated false alarms. The accuracy was of particular relevance in curbing alert fatigue among the security analysts. Recall (sensitivity) was another aspect that was considered so that we would not be losing the actual risks. The F1-score was used to optimise the balance between precision and recall, providing an overall assessment of the model.

One metric that was regularly used in operations was detection latency, or the time it took to detect an anomalous incident and notify the relevant personnel. In a real-time system, low detection latency is critical, as it ensures that threats do not spread. The detection latency in the experiments was less than 2 seconds for most events, making the system applicable in production settings. The false positive rate, model drift rate, and the effectiveness of alert prioritisation (based on threat severity scores) were among other supporting metrics. Lastly, the pipeline was investigated under variable loads to ensure scalability through performance benchmarking. The AI models maintained their performance despite a 5x increase in the scale of telemetry input, demonstrating robustness and versatility in the face of enterprise-scale DevSecOps project deployments.

6. Results and Evaluation

6.1. Detection Accuracy and Comparison with Baselines

The AI-based threat detection system that we developed as part of the adaptive DevSecOps framework demonstrated exceptionally high performance compared to conventional security mechanisms. Empirical analysis revealed that the system had a false positive rate of 2.5% and an overall average system accuracy of 97%. These performance levels exceed those of common rule-

based systems in security, which are often limited in terms of sensitivity and prone to generating numerous false alerts due to their static nature.

In comparison to state-of-the-art solutions, such as threat detection models based on microservices and IoT, the proposed framework demonstrated competitive qualities when benchmarked. Though microservice-oriented AI products demonstrated a detection accuracy of 96.5 - 97.8 percent and a false positive rate of 1.8 - 3.2 percent, the IoT-specific frameworks represented 95 - 98 percent detection accuracy with a slightly higher false positive rate. Conversely, the classical systems exhibited significant limitations in terms of detection, primarily due to their inability to adapt to new threats or situational-based behaviours.

The table below summarizes the comparative results in terms of detection accuracy, false positive rates, latency and scalability:

Table 1. Comparative Evaluation of AI-Driven Threat Detection Solutions across Deployment Environments

Category	Detection Rate (%)	False Positive Rate (%)	Latency	Scalability
Microservice-focused Threat Detection	96.5–97.8	1.8–3.2	Low	High
IoT-Specific Attack Detection	95–98	2–3	High	Low–Medium
Multi-cloud/Cloud-Native Solutions	92–96	3–5.5	Moderate–High	Medium–High
AI-Driven Framework (Case Study)	97	2.5	~5 sec	High

This is a comparative analysis based on all the experiences in use. According to this comparative evaluation, the AI-enhanced adaptive DevSecOps framework not only has a strong detection capability but also minimal false warnings, making it a perfect solution for contemporary high-delivery environment software delivery directives.

6.2. Performance Overhead on Pipelines

A key consideration for introducing a time threat detection mechanism in CI/CD systems is its impact on pipeline performance. In this respect, the proposed framework was used to identify the existing latency or resource limitations imposed by the AI modules. These findings indicate that the developed system imposes a lightweight performance overhead, with an average detection-to-response time of around 5 seconds. This latency is safely within the acceptable bounds of real-time detection, and it does not interfere with the build or deployment processes. Moreover, performance quantities under diverse loads (e.g., 10 times more commits, many parallel pipelines) did not change, which suggests that the framework can support low latency even with high throughput. The AI-powered model, which operates in parallel with other processes in the pipeline, enables product delivery without security or release delays.

6.3. Case Study or Real-World Deployment Example

A real-world deployment scenario involving a cloud-native DevSecOps pipeline provides further validation of the proposed framework. The system, in this case study, was implemented in an enterprise-scale setting, including containerised microservices in Kubernetes, and CI/CD being orchestrated through Jenkins and GitLab. Across a six-week trial, the system successfully identified various types of anomalies, including unknown script modifications to infrastructure, unusual time-based access patterns, and anomalous network traffic patterns during deployment windows.

The most important performance points regarding the case study are:

- Detection Accuracy: 97 percent
- False Positive Rate: 2,5%
- Detection- to -Response Time: 5 seconds
- Scalability: Maintained consistent performance as the number of microservices scaled from 25 to 80

The deployment was most successful due to the dual-mode AI approach, which utilises both categories of learning: supervised (to identify known threats) and unsupervised (for unknown zero-day attacks). Additionally, its ability to integrate with big data processing engines enabled the ingestion and processing of log and telemetry data at scale in real-time, supporting analytics at scale in security.

6.4. Limitations

Although the results are encouraging, the present implementation has shortcomings that require additional research and improvement. The trade-off between the time to detect and the number of IoT devices and multi-cloud environments is one such constraint: the granularity in data variety, volume, and velocity adds complexity to real-time processing. Average detection time can be more than 10 seconds in this setting, which is not a critical fact, but it reduces the possibility of a prompt response to a high-speed attack.

Furthermore, although the rate of false positives is 2.5, which is significantly lower than that of traditional systems, it can also lead to burnout in protection teams in a large-scale setting that processes thousands of incidents every hour. Complex alert correlating and triaging systems are required to enhance the experience of operators. The other issue is that configuration validation and policy enforcement systems, especially those associated with compliance (e.g., GDPR, HIPAA), often lack direct security measures, such as detection accuracy.

These modules are typically rule-oriented and audited or reviewed by an outsider, making them challenging to integrate with analytics AI systems that rely on measurable behaviour metrics. Lastly, there is a concern about model generalizability. It is estimated that a model that is trained on some environments would have poor generalization to a new one unless retrained extensively. As an example, a model that works particularly well on a web framework, full of microservices, might not work in legacy monoliths or industrial control systems, unless it is ported or optimised.

7. Discussion

7.1. Strengths of Adaptive DevSecOps

The flexible DevSecOps strategy has certain obvious advantages that make the delivery pipelines safer and the standard of delivery better. First, adding AI-based threat detection to the CI/CD pipelines changes how security operations work, making them proactive instead of reactive. That mix lets you watch for and respond to threats in real time, which cuts down on the amount of time new threats are out in the open. Also, machine learning (ML) can be used to improve the system as it gets more data and learns from past experiences. This is because it will know about new and changing threat patterns and will be able to tell the difference between normal and harmful activities.

With an increasing number of organisations turning to microservice architectures, the capacity of adaptive DevSecOps systems to support the scale of hundreds of concurrent services and deployments is becoming of primary importance. These types of systems are modular in their design, thus they are able to accommodate a large knowledge base of environments without a sacrifice in their performance and big-data functionality. Also, the implementation of methods of automating tedious security tasks, which include vulnerability scanning and compliance checks, automation removes human practice and the chances of human failures, allowing the security personnel to concentrate more on the difficult decision-making and incident fix practices.

7.2. Security Implications and Risk Reduction

The inclusion of intelligent threat detection systems in DevSecOps pipelines has long-term implications for risk management in organisations. Among these strengths, one has to consider the fact that the vulnerability is detected at the earliest stage of the software construction chain. The system checks infrastructure-as-code, configurations, and application code on demand, and misconfiguration, insecure libraries, and logic errors are identified before they affect production environments. Besides, an adaptive DevSecOps process increases the capacity of an organisation to detect zero-day attacks and emerging threats via anomaly detection and behavioural modelling. The following active nature of the system reduces the mean time to detect (MTTD) and the mean time to respond (MTTR), which are two of the main metrics of cybersecurity resilience. The possibilities to identify anomalies in time lead to fewer data breaches and infringements related to compliance and operational downtimes, which contribute to the reduction of the overall security risk and better business continuity.

7.3. Ethical and Governance Considerations

AI in adaptive DevSecOps practice offers numerous benefits, but there are significant ethical and governance issues to address. Understanding AI models is crucial, as deep learning models often create black boxes, making it difficult to explain anomalies. This can hinder forensic investigations, incident response, and compliance audits. Data privacy is another ethical concern, as user behavior, source codes, and sensitive logs are crucial for training and running AI models. Companies must ensure compliance with data

protection laws like GDPR and CCPA, and add additional steps to protect privacy and control access. Security decisions should be automated but with human oversight to avoid bias and unintended consequences. Governance structures must be changed to consider AI-based security systems, including clear rules for model training, performance monitoring, retraining schedules, and acceptable alerting thresholds. Ethical DevSecOps requires a multidisciplinary approach, incorporating technical, legal, and organizational perspectives to promote responsible and accountable AI utilization in cybersecurity.

7.4. Integration Challenges and Industry Adoption

Adaptive DevSecOps faces integration challenges, particularly in toolchain compatibility, as companies often have legacy systems or diverse CI/CD tools. Integrating AI-driven security components is complex and requires significant pre-processing and tuning effort, including data acquisition, feature construction, and fine-tuning, to train an AI model for specific domain-specific threats.

Organizational culture is another adoption barrier. Adaptive DevSecOps is demanding, and its successful implementation requires the strict cooperation of the development, security, and operations teams. These silos persist in most businesses, frustrating the feedback loops required for continuous learning and threat adaptation. Moreover, some teams may be opposed to the application of AI or the use of automated decision-making tools in critical security situations. In industry terms, an upward trend of adoption is evident, with a focus on cloud-native, fintech, and SaaS solutions. However, it is possible that this will not take off as quickly in more traditional industries, where concerns about regulation, compliance, and reliability may be a bigger issue, such as in healthcare, manufacturing, and government. The adoption must be accelerated by vendors and researchers prioritising the development of interoperable, explainable, and user-friendly AI-fortified security solutions that can easily integrate into existing DevSecOps toolchains and governance regimes.

8. Future Work

As adaptive DevSecOps continues to evolve, several prospective research and development paths can be anticipated. One of the domains is the improvement of model generalization and transferability. Existing threat detection models powered by AI tend to use environment-specific data, which limits their flexibility when applied to other application areas or infrastructure configurations. In the future, these attempts should focus on creating domain-general models and transfer learning methods that can be trained and pre-trained on generic data and then fine-tuned using only a few domain-specific pieces of information. This would not only help minimise the overhead of deployed models but also increase the adoption rate in industries where security experience is low. Inclusion of explainable AI (XAI) in security analytics is another significant direction. With more autonomy being initiated in security decisions as made by AI models, stakeholders, particularly auditors, compliance officers, and SOC teams, will need an explicit understanding of why some anomalies or threats were detected. The inclusion of explainability in AI-based threat detection will enhance trust, usability during debugging and regulatory compliance. Experiments into transparent, simple models or explanatory methods (post-hoc) will prove instrumental in resolving the issue of the disconnect between transparency and performance of models.

Dynamic orchestration of adaptive responses should be incorporated into future systems, in which response actions are dynamically calibrated not only to detect but also mitigate a threat, depending on its severity, system factors, and past success. Existing structures are heavily focused on detection, but with no smart, situation-sensitive mitigation measures that provide low-disruptive measures to operations. Taken together, AI-based detection, automated security playbooks, and reinforcement learning can support closed-loop systems with the ability to protect autonomously in real-time. Finally, federated learning or secure multiparty computation can move threat intelligence sharing to a critical area of exploration. Federated learning can provide a collective learning experience across various attack patterns, with sensitive data remaining on-site, enabling organisations to train shared models on distributed, privacy-preserving datasets. This can significantly enhance the ability to identify emerging threats, including zero-day attacks, while maintaining data sovereignty and compliance with privacy policies.

9. Conclusion

The paper provided in-depth coverage of adaptive DevSecOps models that incorporate AI-assisted threat detection capabilities into continuous delivery chains. These systems can facilitate proactive detection and prevention of vulnerabilities by integrating intelligent option checks at every stage of the software development lifecycle, thus not limiting the rapidity of the development process due to security risks. Machine learning models, including supervised models (such as classifiers in general) and anomaly detectors, can enable organisations to identify known (and even emergent types of) threats in real-time, thereby reducing the risk surface in dynamic and large-scale environments. The results are presented through a nuanced architectural design, implementation plan, and

experimental analysis, which demonstrate that adaptive DevSecOps solutions can deliver high detection accuracy with low false positive rates and minimal performance overhead. Moreover, the system's continuous learning and feedback features enable it to adapt to modern threat environments and complex deployment environments, such as microservices and multi-cloud environments. There are some problems with model generalization and model integration, but the method has a lot of potential to change the way security is implemented in modern DevOps pipelines. In short, Adaptive DevSecOps is a new way of thinking about computer security that focuses on automating and improving the processes for finding threats in the CI/CD environment. Organisations must strike a balance between agility and protection, and AI-driven security architecture helps ensure that software delivery pipelines are resilient, compliant, and secure.

References

[1] Yulianto, S., & Ngo, G. N. C. (2024, September). Enhancing DevSecOps Pipelines with AI-Driven Threat Detection and Response. In 2024 International Conference on ICT for Smart Society (ICISS) (pp. 1-8). IEEE.

[2] Ramaj, X., Sánchez-Gordón, M., Chockalingam, S., & Colomo-Palacios, R. (2023). Unveiling the safety aspects of DevSecOps: evolution, gaps and trends. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 16(3), 61-69.

[3] Kuzlu, M., Fair, C., & Guler, O. (2021). The Role of Artificial Intelligence in Internet of Things (IoT) Cybersecurity. Discover Internet of Things, 1(1), 7.

[4] Anandita Iyer, A., & Umadevi, K. S. (2023). Role of AI and its impact on the development of cybersecurity applications. In Artificial Intelligence and Cyber Security in Industry 4.0 (pp. 23-46). Singapore: Springer Nature Singapore.

[5] Binbeshr, F., & Imam, M. (2025). Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions. arXiv preprint arXiv:2504.19154.

[6] Pekaric, I., Groner, R., Witte, T., Adigun, J. G., Raschke, A., Felderer, M., & Tichy, M. (2023). A systematic review on the security and safety of self-adaptive systems. Journal of Systems and Software, 203, 111716.

[7] Javed, M. A., Hamida, E. B., Al-Fuqaha, A., & Bhargava, B. (2018). Adaptive Security for Intelligent Transportation System Applications. IEEE Intelligent Transportation Systems Magazine, 10(2), 110-120.

[8] Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. Information and software technology, 141, 106700.

[9] Zhao, X., Clear, T., & Lal, R. (2024). Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. Journal of Systems and Software, 112063.

[10] Prabha, M., Hossain, M. A., Samiun, M., Saleh, M. A., Dhar, S. R., & Al Mahmud, M. A. (2024, December). AI-Driven Cyber Threat Detection: Revolutionizing Security Frameworks in Management Information Systems. In 2024 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA) (pp. 357-362). IEEE.

[11] Mousavi, A., Mares, C., & Stonham, T. J. (2015). Continuous feedback loop for adaptive teaching and learning process using student surveys. International Journal of Mechanical Engineering Education, 43(4), 247-264.

[12] Uddin, M. F., Lee, J., Rizvi, S., & Hamada, S. (2018). Proposing enhanced feature engineering and a selection model for machine learning processes. Applied Sciences, 8(4), 646.

[13] Tanikonda, A., Pandey, B. K., Peddinti, S. R., & Katragadda, S. R. (2022). Advanced AI-Driven Cybersecurity Solutions for Proactive Threat Detection and Response in Complex Ecosystems. Journal of Science & Technology, 3(1).

[14] Akbar, M. A., Smolander, K., Mahmood, S., & Alsanad, A. (2022). Toward successful DevSecOps in software development organizations: A decision-making framework. Information and Software Technology, 147, 106894.

[15] Pakalapati, N., Jeyaraman, J., & Sistla, S. M. K. (2023). Building resilient systems: Leveraging AI/ML within DevSecOps frameworks. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 2(2), 213-230.

[16] Zhang, J. Y., & Zhang, Y. (2024). Quantitative DevSecOps Metrics for Cloud-Based Web Microservices. IEEE Access.

[17] Pappula, K. K., & Anasuri, S. (2020). A Domain-Specific Language for Automating Feature-Based Part Creation in Parametric CAD. International Journal of Emerging Research in Engineering and Technology, 1(3), 35-44. https://doi.org/10.63282/3050-922X.IJERET-V1I3P105

[18] Rahul, N. (2020). Optimizing Claims Reserves and Payments with AI: Predictive Models for Financial Accuracy. International Journal of Emerging Trends in Computer Science and Information Technology, 1(3), 46-55. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I3P106

[19] Enjam, G. R. (2020). Ransomware Resilience and Recovery Planning for Insurance Infrastructure. International Journal of AI, BigData, Computational and Management Studies, 1(4), 29-37. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P104

[20] Karri, N. (2021). Self-Driving Databases. International Journal of Emerging Trends in Computer Science and Information Technology, 2(1), 74-83. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I1P10

[21] Pappula, K. K., Anasuri, S., & Rusum, G. P. (2021). Building Observability into Full-Stack Systems: Metrics That Matter. International Journal of Emerging Research in Engineering and Technology, 2(4), 48-58. https://doi.org/10.63282/3050-922X.IJERET-V2I4P106

[22] Pedda Muntala, P. S. R., & Karri, N. (2021). Leveraging Oracle Fusion ERP's Embedded AI for Predictive Financial Forecasting. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 2(3), 74-82. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I3P108

[23] Rahul, N. (2021). Strengthening Fraud Prevention with AI in P&C Insurance: Enhancing Cyber Resilience. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 2(1), 43-53. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I1P106

[24] Enjam, G. R. (2021). Data Privacy & Encryption Practices in Cloud-Based Guidewire Deployments. International Journal of AI, BigData, Computational and Management Studies, 2(3), 64-73. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I3P108
[25] Pappula, K. K. (2022). Architectural Evolution: Transitioning from Monoliths to Service-Oriented Systems. International Journal of Emerging Research in Engineering and Technology, 3(4), 53-62. https://doi.org/10.63282/3050-922X.IJERET-V3I4P107
[26] Jangam, S. K. (2022). Self-Healing Autonomous Software Code Development. International Journal of Emerging Trends in Computer Science and Information Technology, 3(4), 42-52. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P105
[27] Anasuri, S. (2022). Adversarial Attacks and Defenses in Deep Neural Networks. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(4), 77-85. https://doi.org/10.63282/xs971f03
[28] Pedda Muntala, P. S. R. (2022). Anomaly Detection in Expense Management using Oracle AI Services. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(1), 87-94. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P109
[29] Karri, N., & Pedda Muntala, P. S. R. (2022). AI in Capacity Planning. International Journal of AI, BigData, Computational and Management Studies, 3(1), 99-108. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P111
[30] Tekale, K. M., & Rahul, N. (2022). AI and Predictive Analytics in Underwriting, 2022 Advancements in Machine Learning for Loss Prediction and Customer Segmentation. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(1), 95-113. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P111
[31] Rahul, N. (2022). Automating Claims, Policy, and Billing with AI in Guidewire: Streamlining Insurance Operations. International Journal of Emerging Research in Engineering and Technology, 3(4), 75-83. https://doi.org/10.63282/3050-922X.IJERET-V3I4P109
[32] Enjam, G. R. (2022). Energy-Efficient Load Balancing in Distributed Insurance Systems Using AI-Optimized Switching Techniques. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(4), 68-76. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P108
[33] Pappula, K. K. (2023). Reinforcement Learning for Intelligent Batching in Production Pipelines. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(4), 76-86. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I4P109
[34] Jangam, S. K., & Pedda Muntala, P. S. R. (2023). Challenges and Solutions for Managing Errors in Distributed Batch Processing Systems and Data Pipelines. International Journal of Emerging Research in Engineering and Technology, 4(4), 65-79. https://doi.org/10.63282/3050-922X.IJERET-V4I4P107
[35] Anasuri, S. (2023). Secure Software Supply Chains in Open-Source Ecosystems. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 62-74. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P108
[36] Pedda Muntala, P. S. R., & Karri, N. (2023). Leveraging Oracle Digital Assistant (ODA) to Automate ERP Transactions and Improve User Productivity. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 4(4), 97-104. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I4P111
[37] Rahul, N. (2023). Transforming Underwriting with AI: Evolving Risk Assessment and Policy Pricing in P&C Insurance. International Journal of AI, BigData, Computational and Management Studies, 4(3), 92-101. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P110
[38] Enjam, G. R. (2023). Modernizing Legacy Insurance Systems with Microservices on Guidewire Cloud Platform. International Journal of Emerging Research in Engineering and Technology, 4(4), 90-100. https://doi.org/10.63282/3050-922X.IJERET-V4I4P109
[39] Tekale, K. M., Enjam, G. R., & Rahul, N. (2023). AI Risk Coverage: Designing New Products to Cover Liability from AI Model Failures or Biased Algorithmic Decisions. International Journal of AI, BigData, Computational and Management Studies, 4(1), 137-146. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I1P114
[40] Karri, N., Jangam, S. K., & Pedda Muntala, P. S. R. (2023). AI-Driven Indexing Strategies. International Journal of AI, BigData, Computational and Management Studies, 4(2), 111-119. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I2P112
[41] Gowtham Reddy Enjam, Sandeep Channapura Chandragowda, "Decentralized Insured Identity Verification in Cloud Platform using Blockchain-Backed Digital IDs and Biometric Fusion" International Journal of Multidisciplinary on Science and Management, Vol. 1, No. 2, pp. 75-86, 2024.
[42] Karri, N. (2024). Real-Time Performance Monitoring with AI. International Journal of Emerging Trends in Computer Science and Information Technology, 5(1), 102-111. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I1P111
[43] Pappula, K. K., & Anasuri, S. (2024). Deep Learning for Industrial Barcode Recognition at High Throughput. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(1), 79-91. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I1P108
[44] Rahul, N. (2024). Improving Policy Integrity with AI: Detecting Fraud in Policy Issuance and Claims. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(1), 117-129. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I1P111
[45] Reddy Pedda Muntala , P. S. (2024). The Future of Self-Healing ERP Systems: AI-Driven Root Cause Analysis and Remediation. International Journal of AI, BigData, Computational and Management Studies, 5(2), 102-116. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P111
[46] Jangam, S. K., & Karri, N. (2024). Hyper Automation, a Combination of AI, ML, and Robotic Process Automation (RPA), to Achieve End-to-End Automation in Enterprise Workflows. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(1), 92-103. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I1P109
[47] Anasuri, S., & Pappula, K. K. (2024). Human-AI Co-Creation Systems in Design and Art. International Journal of AI, BigData, Computational and Management Studies, 5(1), 102-113. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I1P111.
[48] Pappula, K. K., & Rusum, G. P. (2020). Custom CAD Plugin Architecture for Enforcing Industry-Specific Design Standards. International Journal of AI, BigData, Computational and Management Studies, 1(4), 19-28. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P103
[49] Rahul, N. (2020). Vehicle and Property Loss Assessment with AI: Automating Damage Estimations in Claims. International Journal of Emerging Research in Engineering and Technology, 1(4), 38-46. https://doi.org/10.63282/3050-922X.IJERET-V1I4P105

[50] Enjam, G. R., & Tekale, K. M. (2020). Transitioning from Monolith to Microservices in Policy Administration. International Journal of Emerging Research in Engineering and Technology, 1(3), 45-52. https://doi.org/10.63282/3050-922X.IJERETV1I3P106
[51] Pappula, K. K., & Anasuri, S. (2021). API Composition at Scale: GraphQL Federation vs. REST Aggregation. International Journal of Emerging Trends in Computer Science and Information Technology, 2(2), 54-64. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I2P107
[52] Pedda Muntala, P. S. R., & Jangam, S. K. (2021). Real-time Decision-Making in Fusion ERP Using Streaming Data and AI. International Journal of Emerging Research in Engineering and Technology, 2(2), 55-63. https://doi.org/10.63282/3050-922X.IJERET-V2I2P108
[53] Rahul, N. (2021). AI-Enhanced API Integrations: Advancing Guidewire Ecosystems with Real-Time Data. International Journal of Emerging Research in Engineering and Technology, 2(1), 57-66. https://doi.org/10.63282/3050-922X.IJERET-V2I1P107
[54] Enjam, G. R., & Chandragowda, S. C. (2021). RESTful API Design for Modular Insurance Platforms. International Journal of Emerging Research in Engineering and Technology, 2(3), 71-78. https://doi.org/10.63282/3050-922X.IJERET-V2I3P108
[55] Karri, N., Pedda Muntala, P. S. R., & Jangam, S. K. (2021). Predictive Performance Tuning. International Journal of Emerging Research in Engineering and Technology, 2(1), 67-76. https://doi.org/10.63282/3050-922X.IJERET-V2I1P108
[56] Pappula, K. K. (2022). Containerized Zero-Downtime Deployments in Full-Stack Systems. International Journal of AI, BigData, Computational and Management Studies, 3(4), 60-69. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P107
[57] Jangam, S. K., Karri, N., & Pedda Muntala, P. S. R. (2022). Advanced API Security Techniques and Service Management. International Journal of Emerging Research in Engineering and Technology, 3(4), 63-74. https://doi.org/10.63282/3050-922X.IJERET-V3I4P108
[58] Anasuri, S. (2022). Zero-Trust Architectures for Multi-Cloud Environments. International Journal of Emerging Trends in Computer Science and Information Technology, 3(4), 64-76. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P107
[59] Pedda Muntala, P. S. R., & Karri, N. (2022). Using Oracle Fusion Analytics Warehouse (FAW) and ML to Improve KPI Visibility and Business Outcomes. International Journal of AI, BigData, Computational and Management Studies, 3(1), 79-88. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P109
[60] Rahul, N. (2022). Optimizing Rating Engines through AI and Machine Learning: Revolutionizing Pricing Precision. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(3), 93-101. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I3P110
[61] Enjam, G. R. (2022). Secure Data Masking Strategies for Cloud-Native Insurance Systems. International Journal of Emerging Trends in Computer Science and Information Technology, 3(2), 87-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I2P109
[62] Karri, N. (2022). AI-Powered Anomaly Detection. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 3(2), 122-131. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I2P114
[63] Tekale, K. M. T., & Enjam, G. reddy . (2022). The Evolving Landscape of Cyber Risk Coverage in P&C Policies. International Journal of Emerging Trends in Computer Science and Information Technology, 3(3), 117-126. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P113
[64] Pappula, K. K. (2023). Edge-Deployed Computer Vision for Real-Time Defect Detection. International Journal of AI, BigData, Computational and Management Studies, 4(3), 72-81. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P108
[65] Jangam, S. K. (2023). Importance of Encrypting Data in Transit and at Rest Using TLS and Other Security Protocols and API Security Best Practices. International Journal of AI, BigData, Computational and Management Studies, 4(3), 82-91. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P109
[66] Anasuri, S., & Pappula, K. K. (2023). Green HPC: Carbon-Aware Scheduling in Cloud Data Centers. International Journal of Emerging Research in Engineering and Technology, 4(2), 106-114. https://doi.org/10.63282/3050-922X.IJERET-V4I2P111
[67] Reddy Pedda Muntala , P. S. (2023). Process Automation in Oracle Fusion Cloud Using AI Agents. International Journal of Emerging Research in Engineering and Technology, 4(4), 112-119. https://doi.org/10.63282/3050-922X.IJERET-V4I4P111
[68] Rahul, N. (2023). Personalizing Policies with AI: Improving Customer Experience and Risk Assessment. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 85-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P110
[69] Enjam, G. R. (2023). Optimizing PostgreSQL for High-Volume Insurance Transactions & Secure Backup and Restore Strategies for Databases. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 104-111. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P112
[70] Tekale, K. M., & Rahul, N. (2023). Blockchain and Smart Contracts in Claims Settlement. International Journal of Emerging Trends in Computer Science and Information Technology, 4(2), 121-130. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P112
[71] Karri, N. (2023). Intelligent Indexing Based on Usage Patterns and Query Frequency. International Journal of Emerging Trends in Computer Science and Information Technology, 4(2), 131-138. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P113
[72] Enjam, G. R., & Tekale, K. M. (2024). Self-Healing Microservices for Insurance Platforms: A Fault-Tolerant Architecture Using AWS and PostgreSQL. International Journal of AI, BigData, Computational and Management Studies, 5(1), 127-136. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I1P113
[73] Pappula, K. K., & Rusum, G. P. (2024). AI-Assisted Address Validation Using Hybrid Rule-Based and ML Models. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 5(4), 91-104. https://doi.org/10.63282/3050-9262.IJAIDSML-V5I4P110
[74] Rahul, N. (2024). Revolutionizing Medical Bill Reviews with AI: Enhancing Claims Processing Accuracy and Efficiency. International Journal of AI, BigData, Computational and Management Studies, 5(2), 128-140. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P113
[75] Partha Sarathi Reddy Pedda Muntala, "Enterprise AI Governance in Oracle ERP: Balancing Innovation with Risk" International Journal of Multidisciplinary on Science and Management, Vol. 1, No. 2, pp. 62-74, 2024.

[76] Jangam, S. K. (2024). Research on Firewalls, Intrusion Detection Systems, and Monitoring Solutions Compatible with QUIC's Encryption and Evolving Protocol Features . International Journal of AI, BigData, Computational and Management Studies, 5(2), 90-101. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P110

[77] Anasuri, S., Pappula, K. K., & Rusum, G. P. (2024). Sustainable Inventory Management Algorithms in SAP ERP Systems. International Journal of AI, BigData, Computational and Management Studies, 5(2), 117-127. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P112

[78] Karri, N. (2024). ML Algorithms that Dynamically Allocate CPU, Memory, and I/O Resources. International Journal of AI, BigData, Computational and Management Studies, 5(1), 145-158. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I1P115

[79] Tekale, K. M., & Enjam, G. R. (2024). AI Liability Insurance: Covering Algorithmic Decision-Making Risks. International Journal of AI, BigData, Computational and Management Studies, 5(4), 151-159. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I4P116