ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

Machine Learning–Driven Compliance Intelligence Framework for Automated Vulnerability Mitigation and Continuous Audit Preparedness

Shashikala Valiki
Independent Researcher
shashikala.valiki.researcher@gmail.com

Abstract

Automation of security compliance is limited by the need for continuous monitoring and remediation of vulnerabilities, the foundation of almost all security-related policies. By explicitly linking vulnerability management and policy documentation, a machine-learning model enables not just vulnerability-closure workflows, but also automatic generation of audit-related artifacts—such as evidence of compliance, demonstrable signature of key decision-making, and policy-enforcement operation logs—that together facilitate continuous audit readiness. Orchestration of remediation efforts using risk scores derived from machine-learning-based security analyses allows justification of prioritization decisions based on changes in policy context.

Persistence of security risks and resource constraints often lead organizations to prioritize remediation of vulnerabilities with known exploits over true risk, leaving the most dangerous unpatched. Integrating alerting and remediation capabilities into one artifact provides a crucial foundation to support automated triage and remediation orchestration, ensuring that even vulnerabilities without obvious management precedence receive timely remediation without draining resources. The ability to generate remediation is augmented with a second component: support for the production of evidence and process-mapping artifacts needed for audit readiness. Audit preparation and compliance validation are two of the biggest operational burdens organizations face, yet both can be partly automated through continuous monitoring of infrastructure changes and of knowledge repositories, such as audit logs and change approval records.

Keywords: Compliance Automation, Continuous Monitoring, Vulnerability Management, Audit Readiness, Machine Learning, Security and Compliance Explainability

1. Introduction

Automation serves to accomplish a process without significant human intervention, and capabilities for continuous monitoring make it possible to test environments for compliance when and as required. Such automated checks can be combined with machine learning (ML) methods to enhance compliance automation and, where possible, continuously validate the effectiveness of controls. In these scenarios, not just the remediation of vulnerabilities but also the trails for subsequent audits can benefit from ML. Remediation can be prioritized based on risk-scoring systems that exploit ML instead of traditional risk-indexing approaches. Automated evidence and logical traceability also reduce the burden imposed by continuous audit certification.

Maintaining safety and control in complex environments is a fundamental task, and the effective management of vulnerabilities is key to reducing the available attack surface. The vulnerability lifecycle encompasses the continuous identification of vulnerabilities, the definition of remediations, the execution of such remediations, and the need for audit trails that provide evidence to regulators or controls. Continuous audit readiness can therefore be seen as a by-product of effective vulnerability management. Following a careful evaluation against both remediation and audit-readiness objectives, such models can now be seen as a

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

key enabler of continuous improvement in Cyber Security assurance.

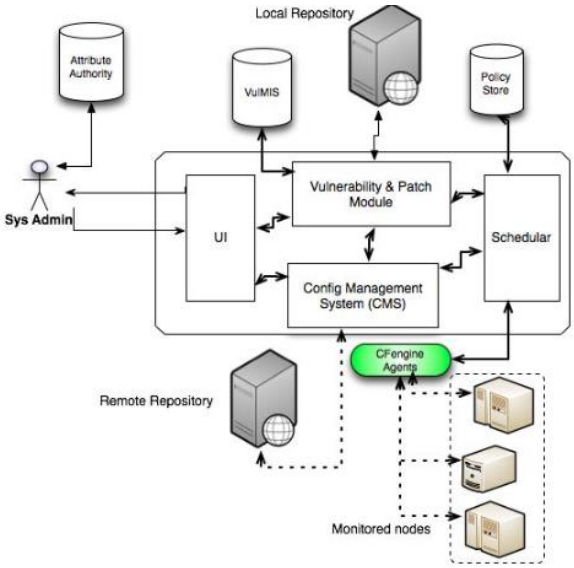


Fig 1: An integrated approach for Vulnerability Management System

1.1. Background and Significance

Continuous monitoring for compliance requires consistent realignment with policy requirements. The focus of security and compliance automation relates more to resource-intensive and time-consuming processes like vulnerability remediation. Real-time validation of compliance status reduces the risk of negative audit outcomes but may also entail identifying, addressing, and providing evidence for all policy violations.

Orchestration of real-time checks of sensitive operations and regular recording of changes against compliance policies helps enable speedy response to audit requests. Machine Learning technologies play an important role in supporting these capabilities but may also influence remediation workloads directly by predicting risk, optimally prioritizing and hurrying up vulnerable changes.

2. Theoretical Foundations

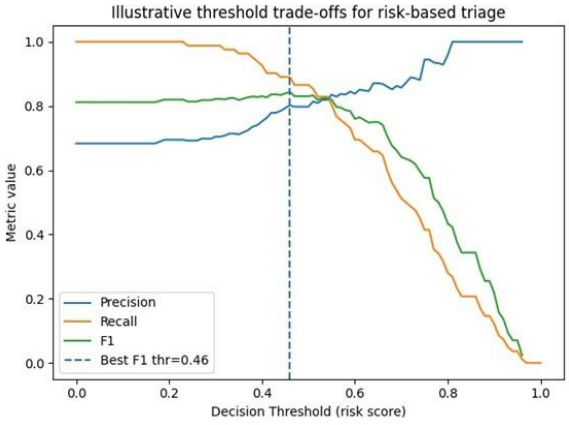
Underlying concepts, models, and paradigms shape technical choices related to security, compliance, and ML governance while guiding the formulation of hypotheses. Compliance Automation and Continuous Monitoring Continuous compliance automation combines policy orchestration, real-time examination, and AI-enabled remediation for surface resistance and preventive vulnerability management. Compliance policies capture corporate intent by defining mandatory configurations or the risks and conditions justifying deviations. Once encoded in operational systems, adherence is continuously validated against end-state checks, proactive examinations, and played-back evidence. Policy enforcement ranges from simple configuration drift detection to complex risk-based controls.

A typical workflow assesses infrastructure security postures, detects deviations in critical defence layers, and orchestrates remediation. Systems scan public-facing components for flaws; signatures identify vulnerable OS and application versions; hosts are assessed for critical patching issues; and misconfigurations are checked. Compliance policies can automate high-confidence corrections with suitable precautions. Orchestrated automation provides the required assurance-level deterrent when simply monitored. Such mechanisms, however, can be overridden during non-production operations or special projects. External regulatory or auditing obligations demand further coverage. Compliance policies thus include continuous-maturity benchmarks for vulnerability lifecycles. Both elements ultimately aim to report security status in real time and contain known threats. However, such maturity automation requires more than real-time checks.

Vulnerability Management and Audit Readiness The vulnerability management life cycle encompasses the entire process, from detection through remediation to the gathering of audit evidentiary markers. Remediation often occurs outside business-as-usual (BAU) operations, under significant time pressure and with resource contention for expert skills, hardware, or downtime; a scalable remediation strategy is therefore needed to satisfy recurring monitoring

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

and governance demands. Such evidence-trail requirements dictate the requisite level of BAU audit-readiness coverage, and thus the depth of automation needed beyond continuous monitoring.



2.1. Compliance Automation and Continuous Monitoring
Compliance Automation encompasses automating policy enforcement and real-time checks against policies, standards, and regulatory requirements. The former typically involves an infrastructure-as-code approach, which enables creating, updating, and decommissioning assets automatically according to the associated security policy definitions. More specifically, security code repositories contain security policies in a machine-readable format, which are orchestrated and executed by relevant tools across the infrastructure and other DevOps phases to ensure conformity to the security policies' intent. The latter involves continuous monitoring mechanisms that detect policy violations, instantly alert the relevant stakeholders, and trigger incident-response workflows.

The automation of the two operations and their orchestration for a complete cycle of compliance automation — that is, an integrated flow of change approvals, actual infra-as-code actions, post-change validation, and policy compliance checks — lead to lower occurrence rates for policy violations. Capital expenditure (capex) on cloud resources is further reduced through proactive mitigation of noncompliance. However, although great progression has been made toward policy definition-as-code and its built-in verification through a set of continuous patches and near-real-time alerting, a major gap still remains with respect to performing the required checks on a continuous basis. Such checks cover several domains, such as identity and access management, network configuration and exposure, encryption, storage security, application-layer checks, changes into the repositories (for configuration files), compliance against DevSecOps cycle security requirements, account-level alerts on anomalous behavior detected, attacks on the applications, and periodic risk assessments from red teams.

Equation 1: Risk score as "predicted likelihood of exploitation" (triage score)

Let features for vulnerability *i* be:

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^\top$$

A common way to output a **probability-like risk score** is logistic regression:

Step A — linear score (logit):

$$z_i = \beta_0 + \sum_{j=1}^{d} \beta_j \, x_{ij} = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i$$

Step B — convert to probability in $[0,1]$**:**

$$r_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}}$$

Interpretation: r_i is the model's estimate of $P(\text{exploit}_i = 1 \mid \mathbf{x}_i)$, matching the paper's "predicted likelihood."

The paper explicitly mentions "designating a risk score above a specific value" as a criterion to trigger alerts.

Define threshold $\tau \in [0,1]$. Predicted exploitability label:

$$\hat{y}_i = \begin{cases} 1 & \text{if } r_i \geq \tau \\ 0 & \text{if } r_i < \tau \end{cases}$$

- $\hat{y}_i = 1$: route to "ASAP work item" / escalated remediation path (paper's triage language).

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

- $\hat{y}_i = 0$: normal queue or deferred (depending on policy appetite).

2.2. Vulnerability Management and Audit Readiness

Continuous vulnerability management encompasses the entire lifecycle of vulnerability assessment, prioritization, and mitigation. It engages diverse sources of vulnerability information, performs periodic assessments, prioritizes software vulnerabilities based on a risk score, and actively manages the remediation progress of identified vulnerabilities. For the purpose of audit readiness, it additionally entails maintaining artifacts and evidence to support preparatory and ongoing audit demands, as well as tropes of changes or remediation undertaken that are relevant to audits.

From a security perspective, audit readiness refers to the organization's ability to provide timely, relevant, and validated evidence that it has adhered to controls and requirements applicable during the audit period. In the context of vulnerability residuum, audit readiness involves accumulating during the vulnerability remediating process the information that will be required to demonstrate its rectification. Various parts of the remediation and vulnerability-tracking processes are being formalized in a manner that supports these requirements and assists in presenting information coherently during an audit.

2.3. Machine-Learning Methods in Security and Compliance

Machine Learning facilitates the automation of time-critical, error-prone, and frequently repeated IT tasks. In the security domain, ML algorithms have been proposed to detect malicious activities and techniques based on labeled datasets created by expert analysts or publicly available repositories. In vulnerability management, an important but expensive action, ML models trained on historical remediation workflows support prioritization and triage. In compliance, a non-existent or poorly maintained evidence trail leads to audit difficulties. Audit logs serve as sources of information for the automatic generation of such evidence, and ML methods can be leveraged to produce complete evidence with little or no human involvement. Other elements directly related to compliance are the alignment of evidence collection with standards and regulations and the risk-aware addition of alternative sources, such as patch status and ownership assignment.

Semi-supervised learning, which operates on partially labeled datasets, is the preferred technique, as it leverages both labeled and unlabeled data without incurring the costs of creating exhaustive labeling. Serialize-Explain-Predict is another suitable paradigm, proposing pipelines that integrate explainable data-driven models within the predictive process to ensure correctness and auditability of decisions.

3. System Architecture and Data Pipeline

The system exhibits a modular structure with contracts defined at the borders of each component. These interfaces promote easy replacement of specific modules or integration of additional ones that interact with the underlying data in a transparent way. Modularity also facilitates scaling of the whole architecture and reduces the chances of requiring a complete stop of data processing for maintenance or upgrades.

The key steps of the data processing pipeline are detailed next. The flow follows a typical pattern comprising data ingestion, normalization, feature extraction, model training, and deployment. Continuous operation requires constant monitoring of data sources, models, and ML system performance. Data source availability checks run in a separate flow and trigger alerts when issues are detected. The model training phase can be initiated automatically when needed, encompassing feature computation and covering model retraining, validation, and deployment. During normal processing, ML models are used to generate outputs that support remediation efforts and prepare the organization for the next regulatory audit.

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

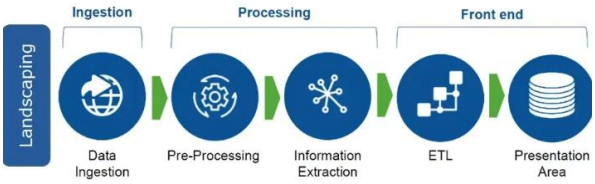


Fig 2: Data pipeline to train AI model

3.1. Data Ingestion and Normalization

Data ingestion comprises the gathering, cleansing, and harmonization of input data for the machine-learning model. Sourced from diverse information repositories, including internal managed platforms and third-party services, ingested data encompass threat feeds, issue trackers, vulnerability databases, public code repositories, and cloud-provider-operations monitoring. Their formats may span JSON, XML, CSV, and other syntactic representations.

Cleansing steps address missing values and outliers. A data-normalization process aligns semantically similar features to consistent semantic typologies, schemas, and coding conventions. Data-lineage information provides traceability and explains the entry of input feature values into the model. Although these steps are initially performed separately, they could be efficiently streamlined through a dedicated ETL (extract–transform–load) system. The complete input data catalogue is listed in Table 1. Note that not all information sources were used for the application of the model to the e-commerce scenario; those specific to that application are marked in the table. Subsequently, the general input matrix of the model is applied with one feature per row and a monitored platform under study as a target column.

3.2. Feature Extraction and Representation

Data models typically contain input and output features that are designed specifically for the problem under consideration. Distinct feature types have been employed to address vulnerabilities and achieve audit readiness, ranging from simple Boolean and categorical representations, to integers quantifying severity scores and remediation complexity, and specialized embeddings such as those based on vulnerability Common Vulnerabilities and Exposures (CVE) or reverse natural language processing techniques. A critical challenge in ML design is the harmonization of diverse feature types into a common representation that optimally supports the learning task.

Feature engineering transforms existing data into a more suitable space for deep learning techniques. For example, the CVE description field can yield rich embeddings of repair complexity through matching against public signatures. A decision-embedding approach converts CVE decision trees into latent representations that preserve semantic information; embeddings can also be obtained from relevant XCCDF remediation templates of security policies or from Cloud Security Posture Management (CSPM) systems. Other features may deserve a temporal perspective, such as the CVSS scores and age of vulnerabilities, the time since the last security compliance check or an assessment of route-to-failure distance. Continuous features may be cast as discrete temporal slices or modeled as hypersurfaces.

3.3. Model Training and Evaluation

Training stages combine a single dataset with cross-validation and prioritization of minority groups; image datasets are adapted for generalization across subjects; configuration settings ensure reproducibility and minimize runtime.

A sequential workflow produces a cyclic-capable model to generate evidence artifacts, capture humans-in-the-loop, and direct monitors for compliance checks. Using real data from diverse sources and synthetic data generation—covering artifacts, strategy risk scores, and operational data schemes—ensures completeness. A refined operational or technological use case guides development and testing of dedicated capabilities.

For remediation and audit-readiness support, lead time plus completeness and explainability indicators inform training and evaluation. Monitoring for drift and failures, continuous validation against governing instruments, and formal capture of human supervisory roles complete critical production

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

functions. Audit scoring focuses on evidence completeness with respect to normative verification activities, change logs, and interpretable rationale behind drive-assistant decisions.

Equation 2: Accuracy, Precision, Recall, F1 (step-by-step)

"Accuracy determines how well the model classifies compromises."

Correct predictions are TP and TN:

$$\text{Accuracy} = \frac{TP + TN}{N}$$

"Precision indicates the proportion of verified compromises among all classified positives."

Predicted positives are TP + FP:

$$\text{Precision} = \frac{TP}{TP + FP}$$

"Recall quantifies the capability for timely detection."

Actual positives are TP + FN:

$$\text{Recall} = \frac{TP}{TP + FN}$$

"F1 balances the two extremes."

Start from harmonic mean:

$$F1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Substitute $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$:

$$F1 = \frac{2}{\frac{TP + FP}{TP} + \frac{TP + FN}{TP}} = \frac{2}{\frac{2TP + FP + FN}{TP}}$$
$$= \frac{2TP}{2TP + FP + FN}$$

That final closed-form is often useful for threshold tuning.

4. Model Components for Remediation and Audit Support

Features extracted from the data pipeline enable support for vulnerability remediation and audit preparations through the following aligned components. First, ML can prioritize and facilitate the triage of detected vulnerabilities, directing attention, time, and resources to the most significant risks before they escalate. Second, the ML system can autonomously synthesize a remediation strategy for addressing the confirmed vulnerabilities, recommending tasks that belong in a single maintenance window and sequencing them to minimize effort and service interruption. Third, the system can produce artifacts that evidence audit readiness, automatically generating trails of change log entries and rationale for decisions affecting sensitive data. These capabilities ease internal and external audit activities, diminish workloads, respond to audit queries, and adapt to shifting standards and regulations.

4.1 Risk Prioritization and Triage

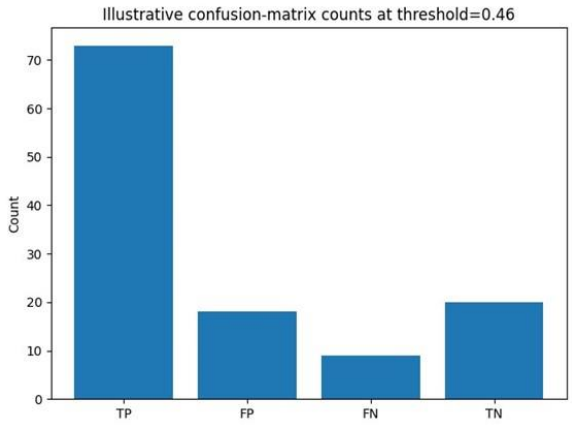
Risk prioritization scores, developed using the training data and aligned with policy provisions, facilitate the effective triage of vulnerabilities discovered during routine scanning. These scores define a distinct optimization dimension, with associated thresholds determining subsequent correction and evidence-collection actions. For each detected vulnerability with a valid risk score, two-way communication with the system's risk-assessment modules enables rapid clarification of any associated risk— through detailed descriptions, technical context, supporting evidence, and the negative impact of successful exploitation—while maintaining a transparent overview of organizational exposure. High-risk issues become priority concerns, serving as ASAP work items attracted to affected platform components, and thus minimizing the chances of critical or extremely critical exposures at time of remediation.

4.2 Remediation Strategy Synthesis

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

When vulnerabilities reach the triage point, the ML system can synthesize a resource-aware remediation strategy that consolidates tasks into a single maintenance window. Such a strategy is advisable even for non-risk-based requests or inquiries, simplifying the scheduling of often-disjointed tasks, mitigating service interruptions, and alleviating operational overhead. The ML model suggests a set of tasks for correction, including any violate-enforcing items— all dedicated to a single affected platform component. Using the classifier-provided temporal representation, these tasks are sequenced to leverage pre-existing downtime, consolidate service disruption, and reduce the workload on personnel. The completed work becomes a series of change-log entries, automatically generated by extracting the primary attributes from input messages.

In addition to these labeled data, other application portfolio statistics have been synthesized in the same style to further assess generalization behavior of temporal models for prediction beyond the training or testing domains. Since applications in the testing domains either had never been vulnerable at the times of model training and validation or were temporally stable snapshots of applications in the training domain, there are grounds to discuss true generalization. Such generalization checks include transfer modeling and synthetically created out-of-range classifiers for convex-hull behavior of the vulnerability classifier transfer pairings. To compensate for commonly applied imbalances in vulnerability evidence distributions, complementary integrity-preserving synthetic instances of all conditions have been produced.



4.1. Risk Prioritization and Triage

Risk scores communicate the predicted likelihood of vulnerabilities being exploited. These scores play a pivotal role in vulnerability management, informing critical decisions concerning remediation that can incur significant costs or operational disruptions. The sequence and exact nature of changes made in the monitored environment are generally determined by human analysts, who prioritize the response to discovered vulnerabilities based on risk scores. Risk scores further underpin triage processes in which the initial process for executing remediation is determined, supporting staff selection from technical resources mapped to processes and policies. Risk-prioritization and triage mechanisms for vulnerability remediation or other processes that must incorporate risk scoring can thus leverage the machine-learning capabilities of the overall solution.

Risk scores must be appropriately grounded in a defined risk appetite for conducting remediation. Criteria for prioritization must be specified, such as designating vulnerabilities with an exploitability status of "proof of concept" as high priority. Decision thresholds may also be defined, such as designating a risk score above a specific value as a criterion for human remediation-triggering alerts.

4.2. Remediation Strategy Synthesis

Based on identified risk priorities and resource constraints,

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

an ML model generates a logically ordered plan for remediating the security gaps.

Determining the best way to address vulnerabilities is a complex and challenging task that requires expert knowledge. Clarifying and sequencing remediation steps can ease the burden on remediation teams, while providing best-remediation advice throughout the organization can lower potential risks. The model helps fill both needs. Using a library of already-documented remediations, it produces detailed plans for recreating known vulnerabilities and formalizes the remediation sequences." Thus, armed with the remediation plans, the organization can remediate vulnerabilities caused by known misconfigurations on a one-off basis. As these remediation actions are completed and the misconfigurations are no longer present, the model will also be able to help with subsequent vulnerabilities that arise from other categories.

Once a superset of the remediation steps has been identified, an optimization model can be built to reduce cost and lead time by reordering the sequence according to the availability of required resources. A lead time can also be imposed naturally (for example, because of a pending audit date) by bounding the completion date of the full set of remediation steps. In addition to providing a prioritized list and a strategy to execute the remaining remediation steps, the natural next question for the organization is, "What is the risk of not remediating these vulnerabilities?" Answering this question typically requires some realistic estimations on the likelihood of exploitation of the vulnerabilities and the associated impact if exploited."

4.3. Audit-Readiness Artifacts and Traceability

Evidence supporting audit readiness is generated automatically, ensuring continuous availability for external scrutiny. Compliance requirements, such as the need for documented processes, decisions, and rationale, are addressed through the creation of change-tracking artifacts alongside remediation controls. Action logs provide detailed accounts of machine-generated changes, while decision rationale is captured in decision trees. The completeness of audit-ready evidence is further augmented by guidance from the training or evaluation corpus.

The automatic generation of audit-ready evidence constitutes a third component. Governance frameworks impose requirements for audit readiness, often considered impractical owing to the continuous operational overhead. Such continuous readiness can be achieved by automatically creating change-tracking artifacts alongside the controls enforcing remediation. For instance, action logs, akin to the Windows Event Logs suppressed by error-level logging, can maintain a record of machine-generated changes. Moreover, decision rationales can be automatically inferred from an interpretable machine-learning model acting as an oracle in cases requiring high-stakes decisions, as in the assignment of service-level priorities. Evidence of remediation can also be made continuous through the application of test-pattern generation on the training or evaluation corpus, complementing the availability of evidence patterns by also supporting completeness checking.

Equation 3: Confusion matrix foundation (needed for Accuracy/Precision/Recall/F1)

Given true labels $y_i \in \{0,1\}$ and predicted labels $\hat{y}_i \in \{0,1\}$:

- **True Positive (TP)**: predicted exploit **and** truly exploit

  $TP = \sum_i \mathbb{1}\,[\hat{y}_i = 1 \wedge y_i = 1]$

- **False Positive (FP)**: predicted exploit but truly not exploit

  $FP = \sum_i \mathbb{1}\,[\hat{y}_i = 1 \wedge y_i = 0]$

- **True Negative (TN)**: predicted not exploit and truly not exploit

  $TN = \sum_i \mathbb{1}\,[\hat{y}_i = 0 \wedge y_i = 0]$

- **False Negative (FN)**: predicted not exploit but truly exploit

  $FN = \sum_i \mathbb{1}\,[\hat{y}_i = 0 \wedge y_i = 1]$

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

Total samples:

$$N = TP + FP + TN + FN$$

5. Continuous Improvement and Feedback Loops

Sustaining model performance and continuous regulatory alignment requires a holistic, feedback-driven approach. Continuous monitoring of ML model drift through established metrics ensures data, feature-input domain, and performance remain stable and act as triggers for retraining whenever necessary. Persistent compliance checks against organisational policies, recognised standards, and applicable regulations apply the brakes on drifts creeping beyond acceptable boundaries. Incorporating an active audit-trail-building component strengthens evidence quality and completeness. Change logs enabling scoping and impact assessments amplify efficiency and reduce lead time during external audits. Human involvement throughout the process guarantees oversight. Decisions that continue to trigger significant model performance drop, reflect a persistent security breach, or generate negative regulatory compliance heat maps are backed by an escalation mechanism for human intervention and override.

To provide reliable output, ML algorithms must be regularly retrained to adapt to dynamic threat landscapes. Within the outlined compliance-automation process, drift of the ML subsystem is continuously monitored, with retraining and full-validation triggers indicated at the ML-model-drift-monitoring stage. Functionality testing checks behaviour against negative-value and risk-mitigation heat maps, focusing on newly introduced vulnerabilities. Flawed remediation suggestions causing changes with risk-mitigation heat-map labels in the opposing direction are categorised as higher-risk change decisions and should be routed for human approval. For business-critical processes triggering unmanageable drift, expert involvement should be sought for remediation guidance. Regular oversight ensures there are no gaps or overlaps in evidence collection for audit-ready preparations.

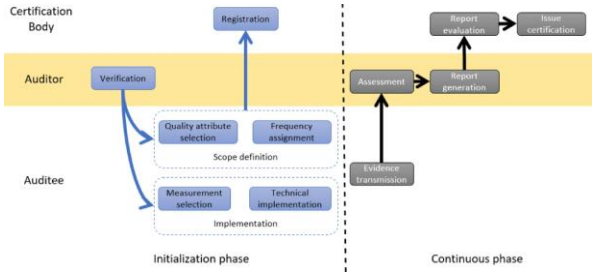


Fig 3: Continuous Improvement and Feedback Loops

5.1. Monitoring Model Drift and Performance

The design incorporates capabilities to flag potential model drift and degradation, enabling timely corrective actions. Drift detection is essential whenever a machine-learning model is deployed in an environment markedly different from its training context. Changes in data sources, representations, or collection methods can affect model performance and generalization. Spanish, for example, pervades much of Mexico but is not strongly represented in US data sets. Drift can occur in the distribution of dependent or independent variables, affecting prediction fidelity. The performance of a machine-learning model should thus be continually monitored and retraining triggered when metrics exceed defined thresholds. Temporal performance tracking can inform normal operating ranges and help detect shifts that make retraining necessary.

Drift detectors assess changes in feature distributions and prediction performance, generating alerts to prompt a review of performance metrics and visualizations. A labelled validation set can facilitate assessments of predictive accuracy and other reliability dimensions. Ongoing operations can trigger periodic retraining, especially when new sources, embeddings, data-processing pipelines, or contexts are introduced.

5.2. Continuous Compliance Validation

Continuous checks against policies, standards, and regulatory requirements, along with systematically generated evidence, play crucial roles in achieving compliance

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

automation. Orchestrating policy enforcement and real-time monitoring of system state and behavior supports the continuous remediation of detected issues. Formal policies clarify expected controls and can express acceptable deviation limits to aid guidance when no control exists or a control is ineffective. Regular testing of controls against policies and standards assesses compliance, identifies deviations, and generates evidence. Combined with supporting evidence for other governance aspects such as security or safety, audit readiness becomes a sustainable objective. The data pipeline thus provides a structured data artifact, aggregating user-defined conditions or expressions for policy verification.

Such a continuous-validation framework is applicable to any domain that creates verified policies with empirical evidence. Examples include Datacatalog.com, which can trigger compliance and privacy checks for permissible data usage and share, or organized security - or risk-committee-controlled change functions, which test process adherence. Efforts are also under way to automate the periodic generation of audit requests for external testing houses.

Real-time policy validation at sufficient depth and coverage is difficult; relying only on control checks significantly affects completeness. Addressing negative security tracks, test-case automation for specific classes of policy violations is a productive method, although some prefer "self-verifying systems" that support formal proofs. Still, completeness is a challenge.

5.3. Human-in-the-Loop Governance

Operationalizing these machine-learning capabilities requires defining the supporting governance structures, especially for the most critical decisions with significant impact on risk posture or compliance. Incorporating a human-in-the-loop (HITL) process delivers direct oversight of the underlying ML models while retaining the speed and efficiency of automation. The transformative potential of AI can be harnessed by establishing a collaborative interaction framework between human experts and the intelligent system, empowering the latter to undertake high-volume, repetitive, or mundane tasks with minimal human intervention while leaving the non-routine, unstructured, and less frequent decisions for human operators.

The HITL governance process spans three dimensions. First, the key decisions, along with their relative criticality and required approval roles, must be specified. Second, the governance interaction must be clearly defined, covering documentation, review cadence, channels, and deciders, to eliminate ambiguity and avoid a bottleneck. Third, the precision, relevance, and potential consequences of the decision must be considered. When the decision holds little risk or the confidence score of the ML model is high, actions can proceed without an approval step; only periodic audits are warranted. Conversely, higher volumes of proposals, lower precision, and elevated risks trigger more frequent reviews.

6. Evaluation and Validation Framework

A rigorous assessment framework demonstrates the efficacy and reliability of the approach, enabling practical deployment and benefits for stakeholders. A comprehensive evaluation plan addresses continuous-active-compliance for both remediating vulnerabilities and preparing for audits. Representative datasets assess the remediation capabilities, while a benchmarking setup tests reliability for various organizations. Supporting audit-readiness—evidence generation, completeness, explainability—has the highest priority, ensuring that automated decisions are auditable and interpretable. Scenario baselines, adversarial tests, and stress tests against policy violations further assess the reliability of supporting processes.

Multiple publicly available datasets capture the remediation process across several domains, enabling an effective evaluation of remediation capabilities. Scalability and reliability across organizations—i.e., domain-generalization or transfer-learning capabilities—are tested using a leave-domain-out approach. Real-data augmentation addresses constraints of using synthetic datasets. Four vital aspects of continuous-active-compliance—risk prioritization, remediation strategy synthesis, proof-collection, audit-

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

readiness requirement—are monitored. Explainability and interpretability support decision validation within a human-in-the-loop governance framework. Completeness evaluates whether the entire remediation-audit-readiness lifecycles remain closed and aligned with governance; a satisfactory lead-time is also desirable.

6.1. Datasets and Benchmarking

Realistic data is essential for reliable performance evaluation of machine-learning models. Media coverage of security vulnerabilities reliably uncovers vulnerable applications and lends credibility to labels for binary classifiers. The dataset of vulnerabilities disclosed in public web applications assembled in the preceding section represents both a broad spectrum of real-world applications over time and data-rich training and testing domains for signature models.

Given the multiple risks and weaknesses reliably detectable throughout disclosed vulnerabilities on public web applications, random examples across each modeling task are also presented to illustrate the modeling strengths and interpretability of the solution space. These representative modeling risks address predictability of the nature of identified threats and confidence in exploitability as successive subsets of non-exploitability per vulnerability condition require cautious interpretation. Reasonable coverage of minority classes is critically evaluated in relation to compliance and auditing support.

Equation 3:

6.2. Evaluation Metrics for Remediation and Audit Readiness

Accuracy, precision, recall, F1, remediation lead time, audit completeness, and explainability constitute the comprehensive metric suite designed to appraise the system's ability to facilitate continuous remediation of vulnerabilities and maintain audit readiness. These metrics evaluate both individual model effectiveness and end-to-end capability to support sustained vulnerability management and regulatory compliance.

Accuracy, precision, recall, and F1 assess performance of the vulnerability-risk model and the remediation-strategy synthesis component. Accuracy determines how well the model classifies compromises, precision indicates the proportion of verified compromises among all classified positives, recall quantifies the capability for timely detection, and F1 balances the two extremes for use in scenarios with low-risk costs. Evaluation hinges on feature sets from different domains and evaluations by separate models trained on single domains. Independent validation against even high-risk domain drift is equally important when testing temporal validity.

Remediation lead time, audit completeness, and explanation quality gauge the overall system’s success in fulfilling requirements and enabling operation. Remediation lead time calculates the elapsed time from compromise classification to final remediation and remediation strategy exposition. Audit completeness, the ratio of generated audit-ready artifacts to needed artifacts, measures traceability of remediation and policy enforcement. Explanation quality rates the solution's ability to elucidate the underlying rationale for each classification.

6.3. Scenario-based Testing and Robustness

Stress testing identifies weaknesses, adversarial scenarios assess resilience to intentional abuses, and simulation of policy violations probes detection capabilities. Thorough evaluation demonstrates model performance, reliability under normal and extreme conditions, and audit-readiness completeness.

Robustness testing ensures that the model maintains accuracy, precision, recall, and F1 scores above the defined thresholds when it receives inputs that differ from those in the training set. Validating error detection against adversarial perturbations strengthens confidence that malicious actors cannot bypass safeguards without being flagged. Finally, confirming detection of simulated violations of compliance policies—such as collaborative misuse by privileged accounts—enhances trust in the solution’s risk prioritization and triage capabilities.

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

Scenario-based testing validates system behavior against anticipated conditions and checks whether any output satisfies compliance policy obligations. Stress tests impose extreme conditions on models for which performance must be maintained at or above a specified level; examples include adding test inputs that respect class distributions but utilize different distributions for the model's features. Such tests use publicly available datasets, such as those in the ALVINN traffic-sign detection dataset and AutoAttack image-classification dataset. Adversarial scenario testing, in which the input is specially crafted to fool an ML model, checks the configuration of detected risks against malicious perturbations of input. Detection of policy violations via supervised scenarios confirms that the outputs arising from model components can detect simulated policy violations with high accuracy.

7. Security, Privacy, and Legal Considerations

Safeguarding data and complying with laws are paramount concerns when developing any system that ingests information from a variety of external environments and stores it for long periods. Privacy and confidentiality of data are ensured through data minimization and limiting the types of information collected from external systems to only what is needed. In addition, sensitive data is encrypted at the storage level, access controls are enforced at the application level, and data de-identification is applied in all development and validation stages.

The learning model has been designed with security in mind, following a systematic framework for threat modeling and security-assured machine-learning system design. Secure development is supported by secure configuration management and secure deployment practices. An incident response plan covers all security incidents, including data breaches. To ensure a high level of confidence in the establishment of security for machine-learning-based systems, continuous assurance practices have been put in place. The continuous compliance validation mechanism allows the assessment of the system against security-related regulations and standards. The review of captured evidence artifacts and change logs ensures compliance with internal governance-related policies.

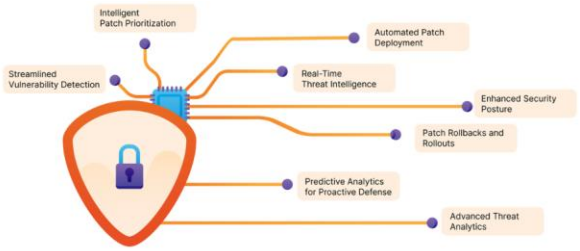


Fig 4: Security, Privacy, and Legal Considerations

7.1. Data Privacy and Confidentiality

Privacy and confidentiality of the data processed is critical to any system, particularly those deploying machine-learning methods. A key principle is data-minimization, that is, only data required to deliver the service should be collected, retained, and processed. Where personal data is unavoidable, data controllers and processors must adhere to applicable legal and regulatory frameworks such as the General Data Protection Regulation (GDPR). Cryptographic techniques mitigate potential privacy breaches resulting from system vulnerabilities, such as disclosure through rogue insiders. Sensitive attributes may also be removed or masked before feeding data into the model, although, completeness and realism of these attributes should be preserved to ensure accurate model predictions. Where this is not possible, the model can be set to ignore sensitive attributes using fairness constraints.

Data encryption, both in transit and at rest, thwarts any attempt to leak sensitive data by compromising the storage infrastructure or exploiting weaknesses in the communication channels used. Access to data must be granted on a least-privilege basis, to reduce the risk of data leakage through internal channels. For instance, users seeking to redeem a group discount offer may be able to view the names of others who used the same offer,

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

potentially leading to embarrassment or harassment; therefore, the system should generate the responses in such a way that the identities of the respondents are not disclosed to each other. Another common approach, based on the assumption that sensitive attributes are mainly correlated with the outputs of certain groups, is to remove or weaken these attributes while generating the output data and re-inject them subsequently, usually as dummy variables. Attribute de-identification methods, which mask the sensitive attribute while preserving as much of the remaining information in the record as possible, can also be employed.

7.2. Security of the ML System

ML applications have been the target of many security incidents and often have undesired behaviors due to malicious influences. As with any attack surface, a threat model helps to understand potential risks and countermeasures. Security must be integrated early on by leveraging concepts from DevSecOps. The inevitable cloud or remote operation introduces the explicit need to harden the system in an as-a-service delivery model.

Security incidents like adversarial training attacks, poisoning, evasion and trojan attacks, model extraction, and other attacks must be addressed as part of deploying machine-learning-based risk automation continuously. Potential attacks must be scoped and analyzed early in the model-building process, and used machine-learning security practices must be supported. A secure-by-design process should encompass data validation, ingredient-creation procedures, artifact validation, and integrity checks, while the deployment phase must take into account secure communications, authentication and authorization of users, and defenses against malicious model use.

7.3. Regulatory and Ethical Compliance

The proposed approach aligns with a wide range of legal, regulatory, and ethical principles that govern the systems, networks, processes, and data it operates upon. Compliance with international and industry-specific standards and laws safeguards users and IT decision-makers by assigning clear responsibilities and duty of care. At the same time, the potential liability exposure can temper the pace of system adoption, and adherence to ethical guidelines helps mitigate shadow side effects.

Though no single law specifically governs the process, its operation affects a multitude of regulations. Financial-sector operations trigger compliance with laws such as the Sarbanes-Oxley Act (SOX) [Addison et al. 2022] and the Payment Card Industry Data Security Standard (PCI DSS) [Kirk 2022]. Health-sector organizations must ensure compliance with laws like the Health Insurance Portability and Accountability Act (HIPAA) [Fitzgerald 2022]. Government bodies may face scrutiny by the Federal Information Security Act (FISMA) [Stewart et al. 2021]. In addition, the model's focus areas are often included in cyber insurance contracts, becoming key for their effectiveness in reducing risk. Its capability of assessing potential violations of compliance-related policies governing the security posture of the monitored environments propels it as an important tool for continuous validation of compliance with the General Data Protection Regulation (GDPR) [Veith 2021].

Emerging guidelines, including the United Nations Guiding Principles for Business and Human Rights [Ruggie 2011] and the Corporation 20 Report [Roe 2020], have introduced new concepts, such as "urging," which places risks generating or amplifying negative impacts on people and the planet at a similar level of priority in enterprise decision-making as risks affecting the corporation itself. Enterprises eager to be seen as upholding their social license must embrace these principles. A failure to do so risks disapproval from individuals and legal authorities, driving responsible organizations to rework their processes, systems, and ML models. The Civic Principles [Civic Tech Field Guide 2021] present a similar imperative for the design and maintenance of systems aimed at serving civil society.

8. Conclusion

Research demonstrates that compliance automation reduces cost by eliminating time-consuming, manual processes. Continuous compliance validation lowers cost by offering

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

near real-time confidence instead of discrete point-in-time checks. AI-supported automation and machine learning have the potential to fulfil both promises simultaneously. Not only can a policy be automatically checked against every change to an environment or system, but a subset of the changes can be risk-ranked and continuously fixed. In addition, these checks—when properly constructed—automatically generate the evidence trail needed to substantiate a compliance audit. Continuous vigilance and assurance become practically achievable.

Even with extensive automation of routine checks, organisations face information security skills shortages that can result in excessive remediation lead times. Organisations need improvements to their incident response and vulnerability management processes and reduced time spent managing vulnerabilities. Monitoring a computing environment for compliance is a classic laboratory problem for machine learning due to the structured nature of networks and systems, the availability of training data, and a relatively stable operating context. Machine-learning models that learn how to automatically remediate detected risks, rarer than monitoring models, offer significant potential to reduce operational costs associated with vigilance by helping to triage incidents and prioritise remediation efforts. To gain the full benefits of machine learning, a risk-driven policy-context-sensitive model is constructive.

8.1. Emerging Trends

The concept of continuous compliance, as tightly integrated automation for policy enforcement, validation, and evidence gathering, is beginning to mature and find adoption. For instance, companies such as Qualys and Synack publicly promote continuous automated security validation capabilities, while AWS and red team assessments validate that security test automation can be continuously run in a more flexible cloud environment.

Emerging tools for machine learning governance are making it possible to attempt a higher level of certification and assurance reasoning than has previously been possible. Such systems may be capable of satisfying internal audit requirements, establishing trust with an AI system's users, or even contributing to compliance with selected external regulations that govern the deployment of AI in the same way as the deployment of traditional embedded software has been governed.

9. References

[1] Ali, S. (2025). Role of automation in hybrid cloud security configAlles, M. G., Kogan, A., & Vasarhelyi, M. A. (2006). Putting continuous auditing theory into practice: Lessons from two pilot implementations. Journal of Information Systems, 20(2), 195–214.

[2] Angermeir, F., Schneider, S., & Pretschner, A. (2024). Towards automated continuous security compliance. arXiv (preprint).

[3] Bhandari, G., Gavric, N., & Shalaginov, A. (2025). Generating vulnerability security fixes with code language models. Information and Software Technology, 185, 107786.

[4] Chan, D. Y., & Vasarhelyi, M. A. (2011). Innovation and practice of continuous auditing. International Journal of Accounting Information Systems, 12(2), 152–160.

[5] Chafjiri, S. B., Legg, P., Hong, J., & Tsompanas, M.-A. (2024). Vulnerability detection through machine learning-based fuzzing: A systematic review. Computers & Security, 143, 103903.

[6] Charmanas, K., Kyriazis, D., & Panagiotopoulos, V. (2023). Exploitation of vulnerabilities: A topic-based machine learning approach. Information, 14(7), 403.

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

[7] Duan, H. K., Vasarhelyi, M. A., & Codesso, M. (2025). Integrating process mining and machine learning for advanced internal control evaluation in auditing. Journal of Information Systems, 39(1), 55–75.

[8] Eulerich, M., Huang, Q., Pawlowski, J., & Vasarhelyi, M. A. (2025). Using process mining as an assurance tool in the three-lines-model. International Journal of Accounting Information Systems, 56, 100731.

[9] Föhr, T. L., Reichelt, V., Marten, K.-U., & Eulerich, M. (2025). A framework for the structured implementation of process mining for audit tasks. International Journal of Accounting Information Systems, 56, 100727.

[10] Hamza, E., & Al-Okaily, A. (2025). Audit and internal control in the era of emerging technologies: Implications for continuous auditing. Proceedings of Atlantis Press.

[11] Herreros-Martínez, A., Rebollo-Monedero, D., & Díaz, I. (2024). Applied machine learning to anomaly detection in enterprise purchase processes. arXiv (preprint).

[12] Hu, Y., Chen, Z., Li, Y., & Dolan-Gavitt, B. (2025). SoK: Automated vulnerability repair: Methods, tools, and evaluation. Proceedings of the USENIX Security Symposium.

[14] Jacobs, J., Romanosky, S., Edwards, B., & Roytman, M. (2023). Enhancing vulnerability prioritization: Data-driven exploit prediction scoring. Proceedings of the Workshop on the Economics of Information Security (WEIS).

[15] Jiang, N., Lutellier, T., & Tan, L. (2021). CURE: Code-aware neural machine translation for automatic program repair. In Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE).

[16] Kalouptsoglou, I., Chatzigeorgiou, A., & Ampatzoglou, A. (2023). Software vulnerability prediction: A systematic mapping study. Information and Software Technology, 160, 107212.

[17] Li, Y., Hu, Y., Chen, Z., & Dolan-Gavitt, B. (2025). SoK: Towards effective automated vulnerability repair. Proceedings of the USENIX Security Symposium.

[18] Mahbub, M., Rahman, M., & Ahmed, S. (2025). A novel vulnerability exploit prediction system using the exploit prediction scoring system. ACM Transactions on Privacy and Security.

[19] Mohammed, K. I., Alsharif, M. H., & Alotaibi, R. (2025). Evolution of DevSecOps and its influence on application security: A systematic literature review. Risks, 13(12), 548.

[20] Nocera, S., Rossi, D., & Di Penta, M. (2025). On the adoption of software bills of materials in open-source software projects. Journal of Systems and Software, 213, 112098.

[21] Nong, Y., Wang, S., & Chen, Z. (2025). APPATCH: Automated adaptive prompting large language models for vulnerability patching. Proceedings of the USENIX Security Symposium.

[22] Pearce, H., Tan, B., Ahmad, B., Karri, R., & Dolan-Gavitt, B. (2021). Examining zero-shot vulnerability repair with large language models. Proceedings of the IEEE Symposium on Security and Privacy Workshops.

ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26 REVISED: MAY 11 ACCEPTED: MAY 24 PUBLISHED: JUNE 09

[23] Port, D., Bui, T., & Boehm, B. (2024). Investigating effectiveness and compliance to DevOps practices: A process improvement perspective. Journal of Systems and Software, 211, 112014.

[24] Prates, L., & Pereira, R. (2025). DevSecOps practices and tools. International Journal of Information Security, 24(1), Article 11.

[25] Prasad, R. D., & Kumar, S. (2024). A deep learning approach to software vulnerability detection. Journal of Theoretical and Applied Information Technology, 102(15), 1–15.

[26] Sadovykh, A., Kotenko, I., & Saenko, I. (2024). Enhancing DevSecOps with continuous security requirements verification in CI/CD pipelines. Computer Research and Modeling, 2024(7), 1–18.

[27] Sayal, A., & Yun, J. J. (2025). Optimizing audit processes through open innovation: An AI-enabled framework. Journal of Open Innovation: Technology, Market, and Complexity, 11(3), 100108.

[28] Sierhieiev, Y., & Olshevska, O. (2023). Detection and prediction of vulnerabilities in software systems using machine learning techniques. In Proceedings of CEUR Workshop Proceedings.

[29] Sinan, M., Happe, L., & Cito, J. (2025). Integrating security controls in DevSecOps: Challenges, practices, and research directions. Journal of Software: Evolution and Process, 37(6), e70029.

[30] Zhang, G., Atasoy, H., & Vasarhelyi, M. A. (2022). Continuous monitoring with machine learning and interactive data visualization: An application to a healthcare payroll process. International Journal of Accounting Information Systems, 46, 100570.

[31] Zhang, Q., Fang, C., Yu, B., Sun, W., Zhang, T., & Chen, Z. (2024). Pre-trained model-based automated software vulnerability repair: How far are we? IEEE Transactions on Dependable and Secure Computing, 21(4), 2507–2525.

[32] Zhao, S., Liu, Y., & Wang, H. (2024). Software vulnerability mining and analysis based on deep learning: An empirical study. Computers, Materials & Continua, 78(2), 1–20.

[33] Zhou, X., Xu, B., Kim, K., Han, D., Nguyen, H. H., Le-Cong, T., He, J., Le, B., & Lo, D. (2024). Leveraging a large language model for automatic patch correctness assessment. IEEE Transactions on Software Engineering, 50(11), 2865–2883.

[34] Zhou, X., Xu, B., Kim, K., Han, D., Nguyen, H. H., Le-Cong, T., He, J., Le, B., & Lo, D. (2025). Large language model for vulnerability detection and repair: A systematic literature review. ACM Computing Surveys.

[35] Zhu, T., Wen, M., & Gao, S. (2024). An empirical study of automatic program repair techniques for injection vulnerabilities. Proceedings of the International Conference on Software Engineering (ICSE) Companion.


ISSN: 3050-9696
E-ISSN: 3050-970X
EUROPEAN ADVANCED JOURNAL FOR SCIENCE & ENGINEERING
Volume: 03 Issue: 02
RECEIVED: APRIL 26
REVISED: MAY 11
ACCEPTED: MAY 24
PUBLISHED: JUNE 09