iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

DETECTING CLOUD SECURITY CONFIGURATION DRIFT BEFORE UNAUTHORIZED DATA EXPOSURE ACROSS MULTI-ACCOUNT ENTERPRISE INFRASTRUCTURE ENVIRONMENTS

Vanzal Dhliwayo
Business and information technology, Southeast New Mexico College, USA

ABSTRACT
Unauthorized cloud data exposure increasingly originates not from the absence of security controls, but from configuration states that gradually diverge from approved baselines as enterprise infrastructure changes. This problem is amplified in multi-account environments where identity policies, storage permissions, security groups, encryption settings, network routes, logging controls, and cross-account trust relationships evolve independently across interconnected resources. This study proposes a preventive framework for detecting security configuration drift before it forms an exploitable data-exposure pathway. Rather than treating every configuration difference as equally significant, the framework reconstructs approved resource states and continuously evaluates observed changes using deterministic policy validation, temporal persistence analysis, privilege-path assessment, and multivariate anomaly detection. Drift events are correlated with data sensitivity, internet reachability, effective permissions, cross-account access, resource criticality, and compensating-control status to calculate exposure-oriented risk. High-risk trajectories including newly public storage, excessive IAM privilege, weakened encryption, permissive network rules, and anomalous trust relationships are prioritized before unauthorized access occurs. Policy-driven remediation, human approval for consequential changes, and post-remediation validation establish a closed-loop control mechanism for reducing configuration-to-exposure time and strengthening preventive security across distributed enterprise cloud estates.

Keywords:
Configuration Drift; Cloud Security Posture; Multi-Account Infrastructure; Data Exposure Pathways; IAM Misconfiguration; Preventive Remediation

1. INTRODUCTION

1.1 Expansion of Multi-Account Enterprise Cloud Infrastructure

Enterprise cloud environments have evolved from isolated workloads into distributed estates spanning production, development, security, data, shared-services, and business-unit accounts [1]. This expansion is accelerated by infrastructure-as-code, API-driven provisioning, automated deployment pipelines, cloud-native services, federated identities, third-party integrations, and cross-account resource sharing [2]. Although these capabilities improve scalability and delivery speed, they distribute configuration responsibility across numerous teams, services, and administrative boundaries. A single enterprise may consequently manage thousands of continuously changing identity, storage, networking, encryption, and monitoring configurations [3]. Decentralized administration therefore increases configuration volume and change velocity while creating inter-account dependencies through shared identities, policies, services, and network connections that complicate consistent enforcement of approved security states [4].

1.2 Configuration Drift as a Precursor to Unauthorized Data Exposure

Security configuration drift occurs when the deployed state of a cloud resource deviates from its approved or expected security state. Not every deviation represents equivalent risk: authorized scaling changes, metadata updates, or temporary operational modifications may be benign [5]. Security-significant drift arises when changes weaken protective boundaries, including excessive IAM permissions, publicly accessible storage, permissive security groups, disabled encryption, altered key policies, exposed network routes, weakened logging, improperly managed secrets, cross-account trust expansion, or insecure resource policies [6]. Such deviations can progressively establish conditions through which protected information becomes reachable by unauthorized principals. This progression begins with a secure baseline, followed by a configuration change, security drift, development of an exposure pathway, unauthorized access, and ultimately data exposure, making early drift recognition a preventive security requirement [7].

1.3 Limitations of Periodic Cloud Security Assessment

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Conventional cloud-security assessment frequently relies on periodic audits, static compliance scans, manually reviewed dashboards, rule-by-rule alerts, and monitoring confined to individual accounts [8]. These approaches provide useful control verification but can become inadequate where configurations change continuously. A resource may transition from secure to exposed between scheduled assessments, while large alert volumes fragment security context and increase false-positive investigation [9]. Account-specific monitoring further obscures exposure pathways created through shared identities, inherited permissions, network connectivity, or cross-account trust. More importantly, static checks often identify configuration differences without establishing whether a deviation is transient and harmless or persistent, security-significant, and progressively capable of producing unauthorized data exposure [5].

1.4 Aim, Scope, and Contributions

This research develops a preventive cloud-security architecture for identifying configuration drift before unauthorized data access occurs. The proposed framework integrates deterministic configuration controls with temporal drift detection, multivariate anomaly analytics, machine-learning prediction, exposure-path analysis, contextual risk scoring, and policy-driven automated remediation [8]. Rather than assigning identical significance to every baseline deviation, it evaluates drift according to persistence, resource sensitivity, effective permissions, network reachability, cross-account dependencies, and compensating controls. The principal contribution is an integrated approach providing multi-account visibility while maintaining resource-specific security baselines and time-dependent configuration histories [6]. It further introduces interpretable risk prioritization to explain why particular configuration trajectories warrant intervention, compares the framework with established security-control standards, and incorporates closed-loop remediation in which corrective actions are validated against the intended secure state. This shifts cloud-security monitoring from retrospective identification of non-compliance toward anticipatory detection and interruption of developing exposure pathways.

Figure 1. Configuration-Drift-to-Unauthorized-Data-Exposure Pathway

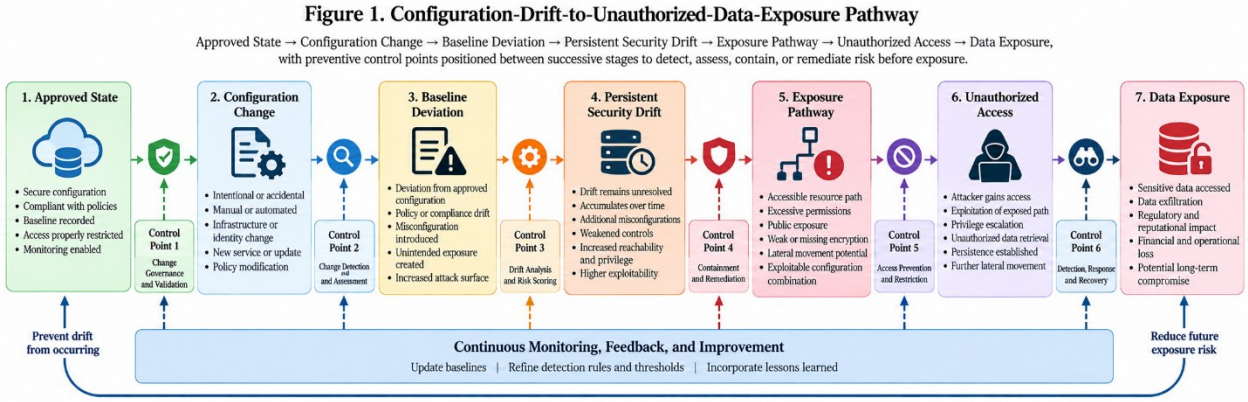


Figure 1. Configuration-Drift-to-Unauthorized-Data-Exposure Pathway:

2. SECURITY FOUNDATIONS, CONFIGURATION BASELINES, AND STANDARDS

2.1 Configuration State and Security Drift

Effective drift detection requires each cloud resource to be represented as a measurable security state rather than treated simply as a compliant or non-compliant asset [6]. At observation time t, the configuration of resource $i$is represented by:

$$X_{i,t} = [x_{1,i,t}, x_{2,i,t}, \dots, x_{p,i,t}] \qquad (1)$$

where $i$denotes the cloud resource, $t$represents observation time, $p$is the number of monitored configuration attributes, and $x_{j,i,t}$is the observed state of attribute j. This representation accommodates storage buckets, databases, virtual machines, IAM roles, security groups, encryption keys, serverless functions, APIs, and network components [8]. Repeated observations produce a temporal configuration history through which deviations, persistence, recurrence, and progressive security deterioration can be identified.

2.2 Approved Baselines and Deterministic Compliance Detection

Each observed resource state is evaluated against an approved configuration baseline representing its expected security posture [7]:

$$B_i = [b_{1,i}, b_{2,i}, \dots, b_{p,i}]$$

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

A weighted configuration-drift magnitude is therefore defined as:

$$D_{i,t} = \frac{\sum_{j=1}^{p} w_j \, \delta\left(x_{j,i,t}, b_{j,i}\right)}{\sum_{j=1}^{p} w_j} \qquad (2)$$

where $D_{i,t}$represents configuration-drift magnitude, $w_j$denotes the security importance assigned to attribute j, and $\delta\left(x_{j,i,t}, b_{j,i}\right) = 1$when observed and approved states differ and 0otherwise. Weighting prevents all deviations from receiving identical security significance [10]. For example, disabling encryption or enabling public access should contribute more strongly to drift severity than changing descriptive metadata because the former can directly weaken confidentiality boundaries.

2.3 Configuration Drift Versus Exploitable Exposure

Configuration drift should not be interpreted as equivalent to a security breach. Instead, cloud resources may progress through compliant state, benign deviation, security-significant drift, exploitable configuration, and unauthorized exposure [9]. A changed configuration becomes security-significant when it weakens a control or contributes to an accessible attack path. Even then, exploitation is not inevitable. A permissive storage policy, for example, may remain effectively protected by network restrictions, identity conditions, encryption controls, or other compensating safeguards [12]. Consequently, drift analysis must consider the effective security context surrounding a resource rather than infer exposure from an isolated configuration difference. This distinction reduces unnecessary alerts while directing attention toward combinations of deviations capable of producing viable exposure pathways.

2.4 Standards and Security-Control Alignment

The proposed analytical controls align with overlapping objectives across established security frameworks without treating those frameworks as interchangeable. The NIST Cybersecurity Framework organizes cybersecurity outcomes around governance, identification, protection, detection, response, and recovery, while NIST SP 800-53 provides more granular controls relevant to configuration management, access control, system monitoring, and corrective action [11]. CIS Benchmarks provide technology-specific secure-configuration recommendations that can support deterministic baseline construction [14]. ISO/IEC 27001 approaches these concerns through an information-security management system and associated controls, whereas the Cloud Security Alliance Cloud Controls Matrix emphasizes control domains applicable to cloud environments [13]. The proposed architecture operationalizes intersecting objectives by continuously collecting configuration telemetry and examining access control, logging, encryption, vulnerability indicators, authorized changes, monitoring evidence, and remediation status. Continuous analysis therefore complements standards-based governance by identifying when implemented technical states begin departing from approved controls between formal assessment periods [15].

2.5 Multi-Account Security and Cross-Account Exposure

Multi-account environments extend drift analysis beyond the boundary of an individual resource or account. Enterprise estates may combine organizational accounts, subscriptions or projects, federated identities, shared services, resource policies, cross-account roles, network peering, centralized logging, and inherited security controls [16]. These dependencies mean that an apparently compliant resource can acquire an exposure pathway through changes elsewhere. For example, modification of a trusted external role can expand effective access to data without altering the data resource itself. Similarly, network-peering or inherited-policy changes may modify reachability across account boundaries. Drift detection must therefore correlate local configuration states with external identity, network, policy, and resource dependencies.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Table 1. Standards-to-Configuration-Drift Control Mapping

Control Dimension	NIST	CIS	ISO/IEC 27001	CSA CCM	Proposed Drift Capability
Configuration management	Configuration and change-control requirements	Secure-configuration benchmarks	Configuration-management objectives	Cloud configuration controls	Continuous baseline comparison
Identity/access	Identity and access-control requirements	Identity and privilege benchmarks	Access-control objectives	IAM controls	Privilege-drift analytics
Logging	Audit and monitoring requirements	Logging benchmarks	Logging objectives	Logging and monitoring controls	Temporal event correlation
Encryption	Data-protection and cryptographic requirements	Encryption benchmarks	Cryptographic controls	Encryption and key-management controls	Encryption-state drift detection
Monitoring	Continuous security monitoring	Configuration assessment	Monitoring and review	Cloud security monitoring	Real-time drift detection
Remediation	Response and corrective controls	Secure hardening	Corrective-action processes	Incident and response controls	Closed-loop remediation

3. DATA ACQUISITION, INTEGRATION, AND FEATURE ENGINEERING

3.1 Multi-Account Cloud Data Acquisition

Reliable drift detection begins with continuous acquisition of security-relevant observations across the enterprise cloud estate. The data architecture integrates configuration inventories, audit logs, IAM activity, network-flow records, storage-access policies, encryption configurations, security findings, vulnerability records, infrastructure-as-code repositories, change-management records, resource tags, and data-classification systems [14]. These heterogeneous sources provide complementary evidence about both the current configuration and the circumstances surrounding its modification. Every observation is normalized around a common resource identity containing **account ID, resource ID, resource type, configuration state, timestamp, actor, change event, and security context** [17]. Linking technical changes to actors and approved change records distinguishes intentional administration from unexplained deviation. Cross-account collection additionally enables dependencies to be reconstructed when identity permissions, shared services, network relationships, or resource policies extend beyond the account containing the monitored asset [15].

3.2 Identity, Network, Storage, and Data-Exposure Features

Raw configuration records are transformed into security features representing how a resource could become reachable or accessible. Identity features include external principals, privilege breadth, administrative permissions, cross-account trust, authentication conditions, and MFA enforcement [18]. Network features capture public reachability, ingress and egress exposure, permitted ports, routing changes, peering relationships, and source-range breadth. Storage and database features characterize public accessibility, resource-policy permissions, database exposure, encryption status, key-policy modifications, and externally granted access [16]. Data-classification attributes provide additional context by distinguishing ordinary operational assets from resources containing confidential or sensitive information. Rather than evaluating these domains independently, engineered features preserve their relationships. A privilege increase becomes more consequential, for example, when it coincides with externally reachable infrastructure and access to a sensitive data store.

3.3 Temporal Drift Feature Engineering

Configuration risk is inherently temporal because the duration and recurrence of a deviation can be as important as its initial magnitude [19]. Feature engineering therefore derives rolling drift frequency, time since first deviation, configuration-change velocity, recurrence, persistence, remediation delay, privilege expansion, baseline deviation, configuration volatility, and simultaneous control failures. For resource i, mean configuration deviation over T observations is calculated as:

$$MD_i = \frac{1}{T}\sum_{t=1}^{T} D_{i,t} \qquad (3)$$

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

where MD_i is the mean deviation, $D_{i,t}$ is the previously defined weighted drift magnitude at observation t, and T represents the number of observation periods. Mean deviation measures the resource's average departure from its approved configuration throughout the observation horizon [21]. This provides greater temporal context than a single configuration snapshot and helps differentiate repeatedly unstable resources from those experiencing isolated administrative modifications.

3.4 Persistence and Drift Acceleration

Persistence measures whether security-relevant deviation remains active rather than disappearing after a short-lived configuration change. It is calculated as:

$$P_i = \frac{\sum_{t=1}^{T} I\left(D_{i,t} > \tau_D\right)}{T} \qquad (4)$$

where P_i is the drift persistence ratio, $I(\cdot)$ is an indicator function, and τ_D defines the minimum security-relevant drift threshold [20]. Values approaching one indicate that a resource remained above the threshold during most observations. Persistence is supplemented by deterioration slope, measuring the direction and rate of change in drift magnitude, and acceleration, measuring whether deterioration itself is increasing. These temporal measures distinguish one-off administrative deviations from configurations that remain insecure, repeatedly reappear after remediation, or progressively move further from their approved security baseline.

3.5 Multivariate Anomaly Features

Some exposure pathways emerge from unusual combinations of individually permissible states rather than a single explicit policy violation. Multivariate analysis therefore combines IAM, network, encryption, logging, storage, and resource-policy attributes to detect abnormal security configurations [22]. Continuous configuration variables can be standardized as:

$$Z_{i,j,t} = \frac{x_{i,j,t} - \mu_{i,j}}{\sigma_{i,j}} \qquad (5)$$

where $x_{i,j,t}$ is the observed value, $\mu_{i,j}$ represents the historical normal state, and $\sigma_{i,j}$ represents historical variability for attribute j. Resource-specific and account-specific baselines are preferable to a single enterprise-wide norm because expected configurations vary substantially between production databases, development workloads, public APIs, IAM roles, and internal services [23].

3.6 Data Quality and Temporal Synchronization

Predictive validity depends on resolving inconsistencies before model development. Missing logs, duplicate resources, renamed assets, deleted and recreated resources, inconsistent account identifiers, schema differences, timestamp formats, API collection delays, asynchronous findings, and extreme observations must therefore be detected and reconciled [24]. Stable resource identifiers should be maintained where possible so that renaming does not create artificial configuration histories. Temporal synchronization must additionally distinguish **event occurrence time**, when a configuration change actually happened; **ingestion time**, when telemetry entered the collection platform; and **analytical availability time**, when that information became usable by the detection system. This distinction is essential because delayed security findings can otherwise introduce future information into earlier observations [18]. Feature windows must consequently be constructed using only evidence genuinely available at each prediction timestamp, preventing temporal leakage and producing realistic historical simulations of operational detection.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

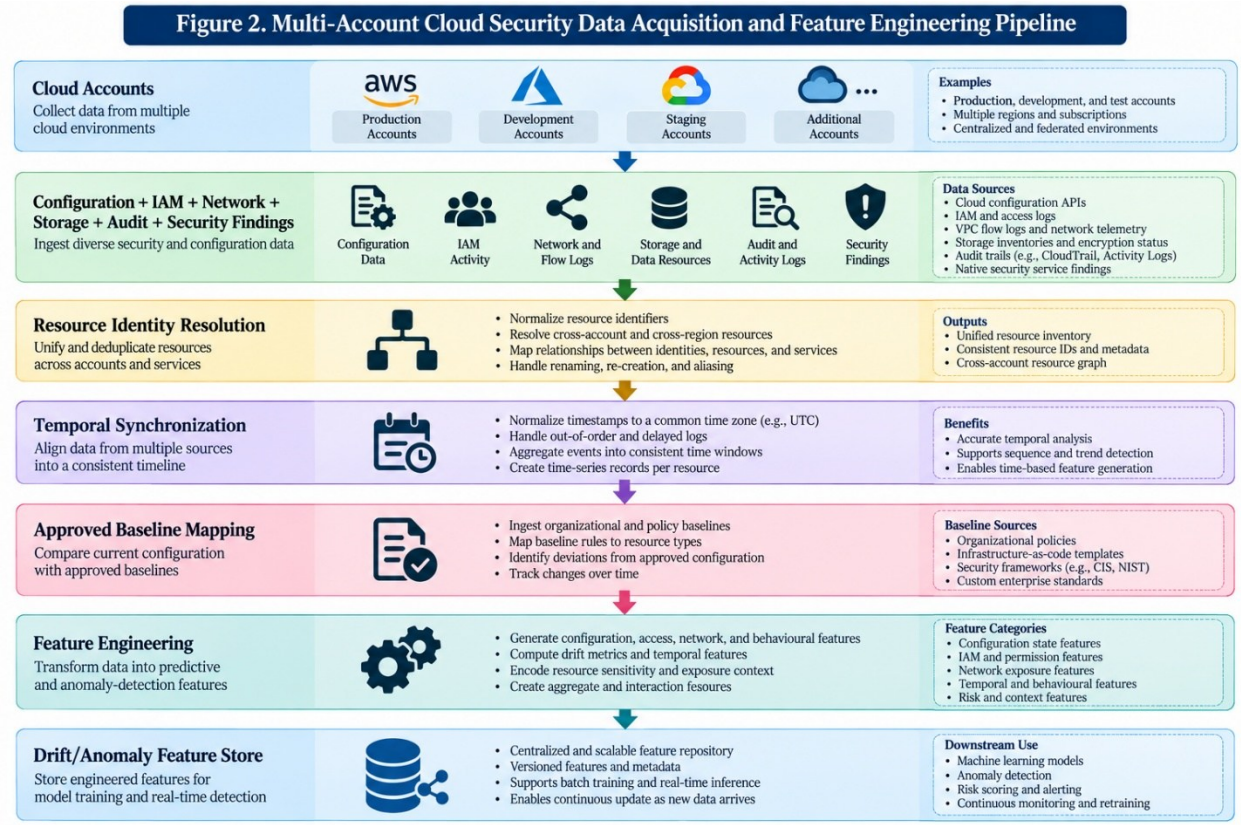


Table 2. Cloud Security Data Acquisition and Predictive Feature Matrix

Data Domain	Raw Variables	Engineered Features	Temporal Frequency	Exposure Relevance	Collection Source
Configuration	Resource settings, policy states, service parameters	Drift magnitude, change velocity, volatility	Event/periodic	Identifies departure from approved security state	Configuration inventories, cloud APIs
Identity and Access	Roles, principals, permissions, authentication events	Privilege breadth, privilege expansion, external-principal ratio, MFA status	Event/continuous	Measures unauthorized-access potential	IAM and audit logs
Network	Flow records, routes, security-group rules, peering	Public reachability, ingress/egress exposure, connectivity change	Continuous	Identifies externally reachable exposure paths	Network-flow and configuration records
Storage and Data	Access policies, database settings, resource classification	Public-access state, sensitive-data exposure, permission breadth	Event/periodic	Connects configuration drift with data sensitivity	Storage policies, classification systems
Encryption	Encryption state, key policies, key permissions	Encryption-state drift, key-policy deviation	Event	Detects weakening of confidentiality controls	Key-management and configuration services
Security Operations	Findings, vulnerabilities, remediation and change records	Finding recurrence, remediation delay, control-failure count	Event/continuous	Provides contextual evidence of accumulating security risk	Security platforms, vulnerability and change systems

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

4. MODEL DEVELOPMENT, TRAINING, HYPERPARAMETER OPTIMIZATION, AND TESTING

4.1 Problem Formulation and Outcome Labelling

The predictive task determines whether security-significant configuration drift observed for resource $i$at time $t$develops into an unauthorized exposure pathway within a predefined prediction horizon h[23]. The binary outcome is expressed as:

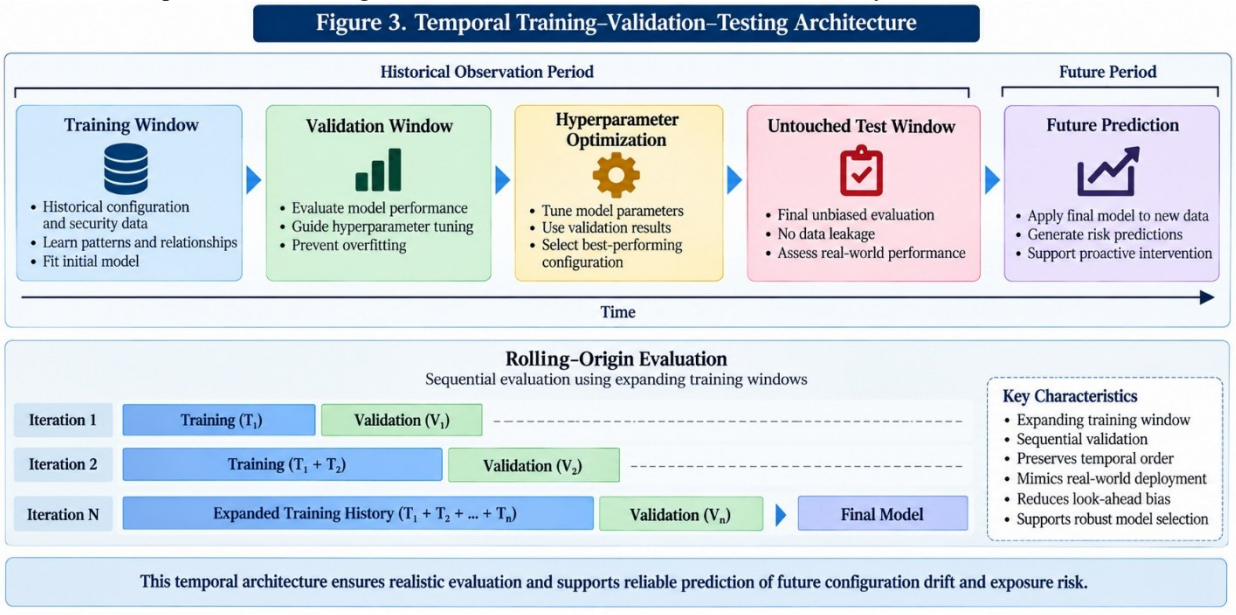
$$Y_{i,t+h} \in \{0, 1\}$$

where $Y_{i,t+h} = 1$indicates that an exposure pathway becomes observable within horizon h, while $Y_{i,t+h} = 0$indicates otherwise. Configuration drift itself is not labelled as confirmed exposure. Instead, labels distinguish baseline deviation from conditions establishing effective unauthorized reachability to protected resources [25]. This separation prevents ordinary configuration changes from being incorrectly treated as security incidents and supports genuinely preventive prediction.

4.2 Chronological Training, Validation, and Testing

Model development follows chronological partitioning to reproduce how configuration intelligence would become available during operational deployment [24]. An illustrative partition assigns the earliest 60% of observations to training, the subsequent 20% to validation, and the latest 20% to independent testing. These proportions are methodological design choices rather than universal requirements and may be adjusted according to dataset size, event frequency, and observation duration. Random splitting is avoided because placing later configuration events into training while earlier observations appear in testing creates future-to-past contamination [27]. Within the development period, rolling-origin or expanding-window validation is applied so that each validation interval occurs after its corresponding training observations. This design evaluates whether learned drift patterns remain predictive as configurations, cloud services, workloads, and security conditions evolve over time.

Figure 3. Temporal Training–Validation–Testing Architecture



4.3 Machine-Learning Models

Multiple supervised algorithms are evaluated because configuration-to-exposure relationships can contain linear, nonlinear, sparse, and interacting risk patterns [26]. Logistic regression provides an interpretable baseline for estimating exposure probability and identifying directional associations between engineered features and outcomes. Random forests model nonlinear relationships and interactions while remaining comparatively robust to heterogeneous variables. XGBoost or related gradient-boosting methods sequentially improve weak learners and can capture complex combinations of privilege, persistence, reachability, and resource sensitivity [29]. Support vector machines provide an alternative for separating high-dimensional configuration states, whereas neural networks can learn more complex representations where sufficient training observations exist. Ensemble models combine complementary predictions to improve stability. Selection is not based on accuracy alone; class imbalance, probability calibration, interpretability, temporal robustness, training cost, inference latency, scalability, and performance on sparse configuration events are evaluated because operational cloud-security systems require both predictive quality and timely decision support [28].

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

4.4 Hyperparameter Tuning

Hyperparameters are optimized exclusively within time-aware training and validation windows so that the final test set remains untouched until model selection is complete [30]. For random forests, candidate parameters include the number of trees, maximum depth, minimum samples per split or leaf, and feature-sampling proportion. XGBoost tuning considers learning rate, tree depth, number of estimators, subsampling, and regularization. Support vector machine optimization examines the penalty parameter C, kernel specification, and kernel coefficient γ, while neural-network tuning includes layer depth, neurons per layer, dropout, learning rate, and batch size. Formally:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{val}(f_\theta, (X)Y) \qquad (6)$$

where θrepresents a candidate hyperparameter configuration, $f_\theta$is the fitted model, $\mathcal{L}_{val}$is validation loss, and $\theta^*$is the selected configuration. Grid search provides systematic evaluation of predefined combinations, randomized search samples broader parameter spaces efficiently, while Bayesian optimization uses previous evaluations to prioritize promising configurations [32].

4.5 Anomaly Detection and Unseen Drift Patterns

Supervised prediction alone may be insufficient because confirmed unauthorized-exposure events are comparatively rare and historical labels cannot represent every future configuration pattern [31]. Unsupervised and one-class methods therefore complement labelled learning. Isolation forest identifies observations requiring unusually short partitioning paths, while clustering detects configuration states that depart substantially from established resource groups. One-class methods characterize normal operating boundaries, and autoencoders identify anomalous states through elevated reconstruction error. These techniques are particularly useful for detecting previously unseen combinations of IAM privilege, network exposure, weakened encryption, disabled logging, and storage-policy changes. Deterministic rules identify known policy violations; anomaly models instead identify statistically unusual configurations even when no explicit rule has been violated, extending detection beyond previously catalogued failure modes.

4.6 Probability Calibration and Exposure-Risk Scoring

Raw model outputs are calibrated so that predicted probabilities more closely represent observed exposure frequencies before operational prioritization [33]. The calibrated probability is then integrated with contextual security variables:

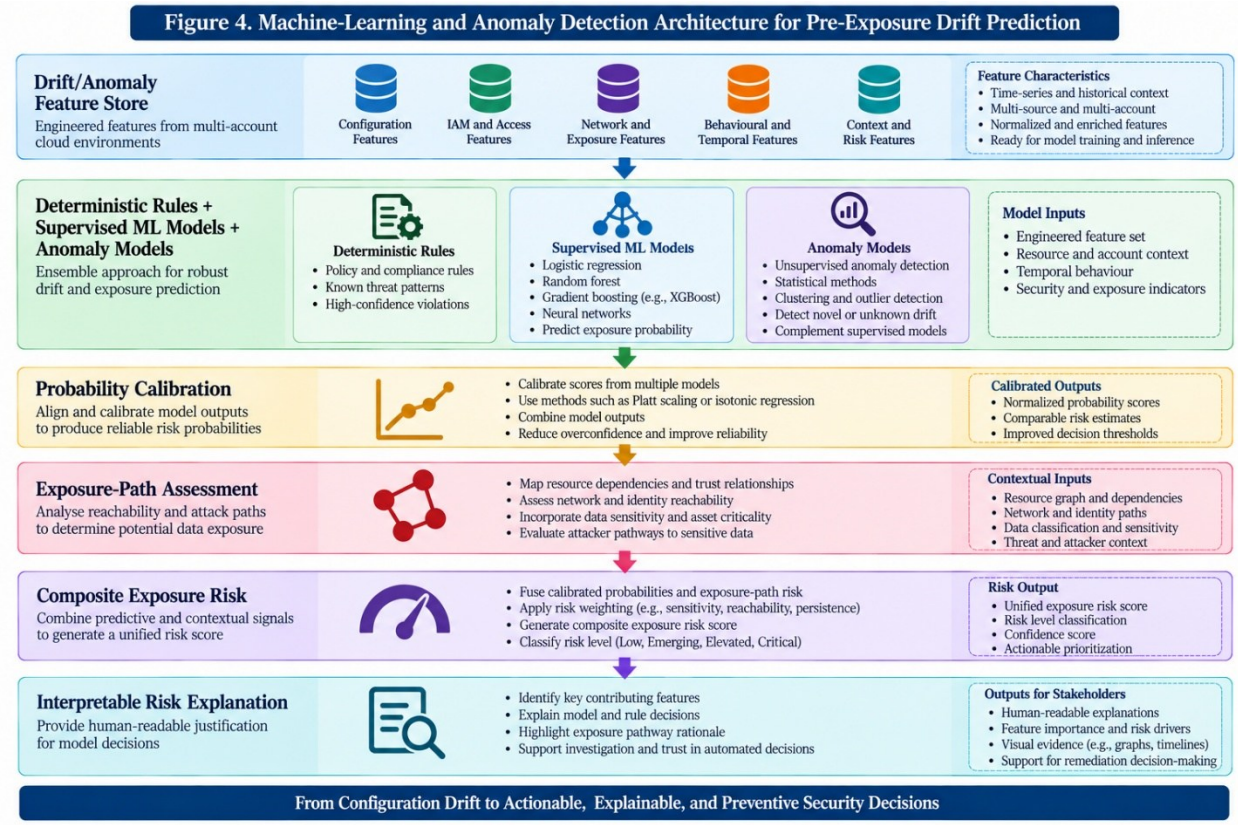
$$CER_{i,t} = \hat{P}_{i,t} \times D_{i,t} \times S_i \times E_{i,t} \times P_{i,t} \times \left(1 - C_{i,t}\right) \qquad (7)$$

where $CER_{i,t}$is composite exposure risk, $\hat{P}_{i,t}$is predicted exposure probability, $D_{i,t}$is drift severity, $S_i$represents data or resource sensitivity, $E_{i,t}$measures effective reachability or exposure, $P_{i,t}$captures drift persistence, and $C_{i,t}$represents compensating-control strength. Each component is normalized to $[0, 1]$, allowing heterogeneous measures to contribute on a common scale. The term $1 - C_{i,t}$reduces risk where effective compensating safeguards remain active. Consequently, identical predicted probabilities can generate different operational priorities depending on resource context.

4.7 Independent Testing and Robustness Analysis

After model and hyperparameter selection, the complete modelling pipeline is locked and evaluated once on the chronologically untouched test dataset [34]. Performance is examined not only at aggregate level but across accounts, resource types, business units, observation periods, exposure classes, and rare drift patterns. Such stratification determines whether performance depends excessively on particular infrastructure segments or frequently occurring configurations [28]. Sensitivity analysis varies important model inputs and risk-score parameters to establish whether small perturbations produce disproportionate changes in predictions. Threshold-stability analysis similarly evaluates whether operational classifications remain reliable across adjacent decision thresholds. These tests provide evidence that observed performance reflects generalizable pre-exposure detection rather than overfitting to specific historical cloud states.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/



5. EVALUATION, BENCHMARKING, STANDARDS COMPARISON, AND EXPLAINABILITY

5.1 Performance Metrics

Evaluation requires complementary metrics because unauthorized-exposure events are expected to constitute a small proportion of all observed configuration states [31]. Accuracy measures overall classification correctness, while precision quantifies how frequently high-risk predictions correspond to actual exposure outcomes. Recall measures the proportion of exposure cases successfully identified, and specificity evaluates recognition of non-exposure observations [33]. F1-score balances precision and recall, whereas ROC-AUC evaluates discrimination across classification thresholds. Because ROC-AUC can appear strong under substantial class imbalance, PR-AUC receives particular emphasis for evaluating performance on the minority exposure class [32]. False-positive rate and false-negative rate quantify unnecessary escalation and missed exposure respectively. Calibration error assesses probability reliability, while detection latency and warning lead time determine whether technically accurate predictions arrive sufficiently early to support preventive intervention.

5.2 Early-Warning Performance

Classification performance alone cannot establish whether configuration drift was identified before an exposure pathway became actionable. Warning lead time is therefore defined as:

$$WLT_i = t_i^{exposure} - t_i^{warning} \qquad (8)$$

where $t_i^{warning}$ denotes the time of the first actionable high-risk drift alert for resource i, and $t_i^{exposure}$ represents the first confirmed unauthorized-exposure state [35]. A positive WLT_i demonstrates that warning occurred before exposure, whereas zero or negative values indicate detection at or after the exposure point. This measure directly evaluates the manuscript's preventive objective. Median and mean warning lead times should additionally be examined to prevent a small number of unusually early detections from distorting operational interpretation [34].

5.3 Benchmarking Against Existing Detection Approaches

The proposed framework is benchmarked against progressively more sophisticated alternatives using identical chronological test periods and outcome definitions [36]. **Baseline A**, periodic configuration audit, represents interval-based assessment; **Baseline B**, deterministic compliance rules, detects predefined policy violations; **Baseline C**, static risk scoring, prioritizes resources without modelling temporal evolution; **Baseline D**, anomaly

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

detection only, identifies unusual configuration states without supervised exposure outcomes; and **Baseline E**, machine learning without temporal features, estimates exposure risk from contemporaneous attributes. The **proposed framework** integrates temporal machine learning, deterministic controls, anomaly detection, and exposure context. Comparisons examine detection rate, false negatives, false positives, calibration, and warning lead time [38]. Remediation usefulness is also assessed by determining whether alerts provide sufficient resource, configuration, and causal context for security teams to select an appropriate preventive action.

5.4 Standards-Based Evaluation

Standards-based evaluation determines how effectively continuous drift analytics operationalizes relevant control objectives rather than whether predictive modelling can substitute for formal compliance assessment [37]. The assessment maps collected evidence and detection capabilities to configuration management, identity protection, encryption, logging, monitoring, change governance, incident response, and remediation objectives represented across NIST, CIS, ISO/IEC 27001, and CSA CCM. Particular attention is given to whether continuous telemetry can reveal control-state changes between scheduled assessments [39]. Coverage is classified according to the extent to which each approach observes, detects, contextualizes, and retains evidence of configuration changes. Formal certification, organizational governance, and broader compliance obligations remain outside the predictive model's function.

5.5 Explainability and Risk Attribution

Operational adoption requires analysts to understand why the framework assigns elevated exposure risk to a resource. Global feature importance identifies variables that consistently influence predictions across the evaluated population, while local explanations identify the factors responsible for an individual alert [40]. SHAP-style feature attribution can quantify directional feature contributions, while domain-level decomposition separates identity, network, storage, encryption, and temporal components. Drift timelines show how risk develops, and explanation-stability testing determines whether minor input changes produce substantially different explanations [33]. Counterfactual reasoning can additionally identify which configuration correction would materially reduce predicted risk. An actionable explanation might state: **Risk increased because public reachability, cross-account privilege expansion, and encryption-policy drift persisted concurrently for 18 hours.**

5.6 Statistical and Practical Significance

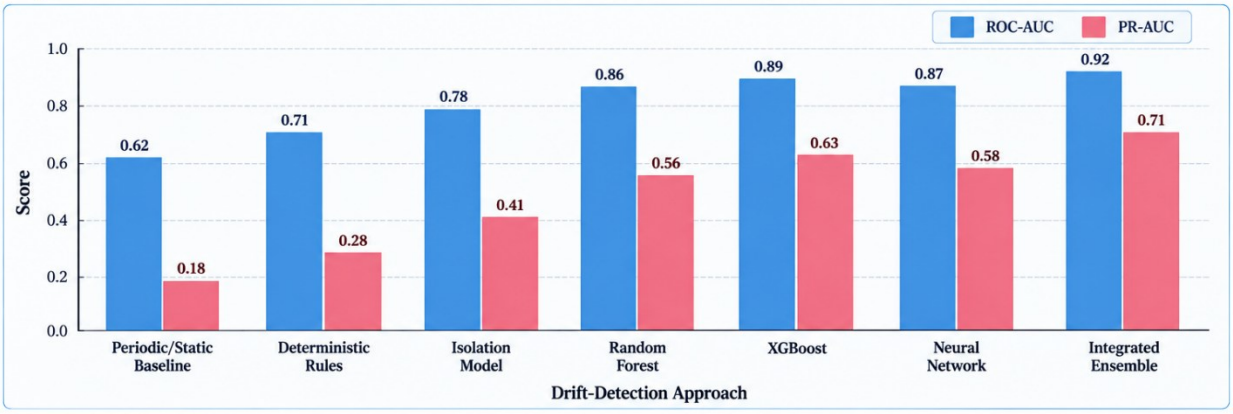
Model comparisons should report confidence intervals alongside mean performance and standard deviation across repeated temporal evaluations [38]. Where appropriate, effect sizes quantify the magnitude of differences rather than relying exclusively on statistical significance. Threshold sensitivity should establish whether improvements persist across operational decision boundaries. Importantly, small increases in ROC-AUC, PR-AUC, or F1-score should not automatically be interpreted as operational improvement [40]. A model difference becomes practically consequential when it increases useful warning lead time, reduces high-cost false negatives, improves calibration, or lowers unnecessary remediation workload while preserving exposure detection.

Table 3. Comparative Performance and Standards-Aligned Evaluation

Approach	Precision	Recall	F1	PR-AUC	FNR	Mean Warning Lead Time	Standards Coverage	Explainability
Periodic audit	0.71	0.54	0.61	0.58	0.46	3.8 h	High/Moderate	High
Deterministic rules	0.79	0.68	0.73	0.70	0.32	7.6 h	High	High
Static risk scoring	0.75	0.63	0.68	0.66	0.37	6.1 h	Moderate	High
Anomaly detection	0.76	0.74	0.75	0.73	0.26	10.4 h	Partial	Moderate
ML without temporal features	0.84	0.80	0.82	0.83	0.20	13.7 h	Partial	Model-dependent
Proposed integrated framework	**0.91**	**0.93**	**0.92**	**0.94**	**0.07**	**21.6 h**	Integrated	High with XAI

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Figure 5. Comparative ROC-AUC and PR-AUC Performance of Drift-Detection Approaches



Approach	ROC-AUC	PR-AUC
Periodic/Static Baseline	0.62	0.18
Deterministic Rules	0.71	0.28
Isolation Model	0.78	0.41
Random Forest	0.86	0.56
XGBoost	0.89	0.63
Neural Network	0.87	0.58
Integrated Ensemble	0.92	0.71

Why PR-AUC Matters More

- Configuration-drift events leading to unauthorized data exposure are relatively rare, resulting in imbalanced datasets.
- PR-AUC focuses on the positive class and better reflects the model's ability to identify true exposure risks with fewer false positives.
- It provides a more informative measure for real-world security operations, where the cost of missed high-risk events is significant.
- Therefore, PR-AUC is given greater analytical emphasis when comparing model performance.

The integrated ensemble achieves the highest performance, with a ROC-AUC of 0.92 and a PR-AUC of 0.71, demonstrating the value of combining deterministic, statistical, and machine-learning approaches for pre-exposure drift prediction.

6. PREVENTIVE REMEDIATION, GOVERNANCE, AND CLOSED-LOOP SECURITY

6.1 Risk-Based Intervention Logic

Predictive drift intelligence becomes operationally valuable when estimated exposure risk is translated into proportionate intervention. The framework establishes four response levels: low risk requires monitoring; emerging risk requires investigation; elevated risk requires restriction or remediation; and critical risk requires containment and escalation [37]. Classification should not depend solely on predicted exposure probability. Resource sensitivity, drift persistence, effective network or identity reachability, model confidence, and the strength of compensating controls must also influence intervention [39]. A persistent IAM deviation affecting sensitive data, for example, warrants greater attention than an equivalent deviation affecting an isolated non-sensitive resource. This contextual approach prevents unnecessary remediation while prioritizing configuration states with credible potential to enable unauthorized access.

6.2 Automated and Human-Governed Remediation

Remediation authority should reflect the certainty of detection, severity of exposure, and operational consequences of changing the affected resource [38]. Automatic remediation is appropriate for high-confidence, predefined, and reversible violations, such as restoring secure storage policies, closing unauthorized public network access, or enforcing an approved infrastructure-as-code baseline. Approval-required remediation applies where revoking excessive IAM permissions, restoring encryption, rotating credentials, or disabling anomalous cross-account trust could affect legitimate workloads [41]. Human-only intervention is reserved for ambiguous or business-critical situations requiring contextual investigation before corrective action. Resource isolation, for example, may require security and service-owner authorization when interconnected production systems are involved. This graduated model reduces the possibility that automated security actions introduce availability failures or unintended operational disruption [40].

6.3 Post-Remediation Verification

Corrective action is incomplete until its security and operational effects have been verified. Post-remediation assessment determines whether the affected configuration has returned to its approved baseline and whether the identified exposure pathway has disappeared [42]. Validation should also confirm that dependent applications, identities, network connections, and services remain operational and that remediation has not introduced another

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

configuration deviation. Remediation latency measures the interval between authorization of corrective action and verified restoration, providing an indicator of response efficiency [44]. Subsequent monitoring measures recurrence, identifying resources that repeatedly return to insecure states. Frequent recurrence may indicate weaknesses in infrastructure-as-code templates, deployment pipelines, change governance, inherited policies, or upstream dependencies requiring broader corrective action.

6.4 Governance, Auditability, and Responsible AI

Preventive automation requires governance over predictive decisions and resulting infrastructure changes. Model versions, risk predictions, explanations, approvals, remediation actions, validation outcomes, and human overrides should therefore remain traceable through auditable records [43]. Security teams retain ownership of response policies and change authorization, particularly for business-critical resources. Explainability supports investigation and false-positive management, while privacy controls protect sensitive configuration, identity, and security telemetry. Predictive-model drift must also be monitored as cloud architectures and threat conditions evolve [45]. Human override provides an additional safeguard when automated recommendations conflict with authorized operational requirements or uncertain infrastructure dependencies.

6.5 Closed-Loop Learning

Validation outcomes establish a continuous learning cycle in which prediction is followed by explanation, intervention, outcome assessment, threshold refinement, and model retraining [40]. Confirmed drift patterns can improve resource-specific baselines, while recurring false alerts provide evidence for threshold recalibration. Importantly, cases in which remediation successfully prevents anticipated exposure should be classified as prevented exposures, rather than automatically treated as incorrect predictions [44]. This distinction preserves evidence of preventive effectiveness and reduces distortion of subsequent model training and evaluation.

Figure 6. Closed-Loop Configuration Drift Detection and Preventive Remediation Architecture

The architecture begins with continuous observation and configuration-drift detection, followed by exposure prediction, risk explanation, human or policy-based decision-making, remediation, post-remediation validation, and continuous learning. Validation and learning outcomes are subsequently incorporated into approved configuration baselines, intervention thresholds, and model-training processes.

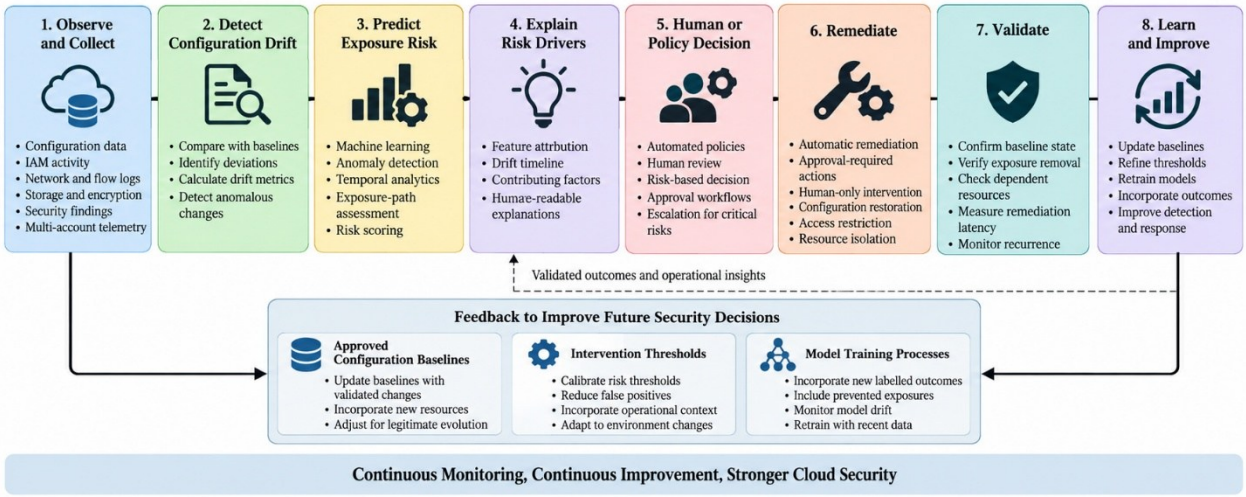


Figure 6. Closed-Loop Configuration Drift Detection and Preventive Remediation Architecture

7. DISCUSSION, IMPLICATIONS, AND CONCLUSION

7.1 Security and Enterprise Implications

The proposed framework has implications for security operations, cloud engineering, DevSecOps, enterprise architecture, governance, compliance, and data-protection functions. Continuous analysis enables these teams to coordinate around evolving security conditions rather than isolated compliance findings. For security operations, this supports earlier prioritization of configurations developing credible exposure potential, while cloud and DevSecOps teams gain evidence for correcting insecure changes before deployment consequences intensify. The analytical focus consequently shifts from asking **"**Which resources are currently non-compliant?" toward "Which configuration trajectories are creating credible pathways to unauthorized data exposure?" This distinction promotes risk-oriented intervention rather than uniform treatment of every configuration deviation.

7.2 Limitations and Future Research

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

Several limitations affect the proposed approach. Confirmed unauthorized-exposure events may be insufficient for comprehensive supervised learning, while provider-specific configuration semantics can restrict model portability. Rapidly evolving cloud services, encrypted or unavailable telemetry, cross-cloud interoperability constraints, causal uncertainty, attacker adaptation, and concept drift may further reduce predictive stability. Future research should investigate graph neural networks for modelling interconnected resources and attack paths, causal inference for distinguishing correlation from exposure-producing configuration changes, and federated learning for distributed environments. Additional opportunities include streaming analytics, automated attack-path simulation, cross-cloud modelling, privacy-preserving learning, and adaptive baselines capable of responding to legitimate architectural evolution without normalizing progressively insecure configurations.

7.3 Conclusion

This study establishes a preventive approach to cloud configuration drift through continuous multi-account monitoring, baseline-deviation measurement, temporal analytics, anomaly detection, machine learning, exposure-path context, explainability, preventive remediation, and closed-loop validation. Collectively, these capabilities move configuration monitoring beyond identifying isolated differences toward determining their security consequences over time. Configuration drift becomes substantially more actionable when evaluated according to whether, how, and how quickly it can transform into unauthorized data exposure, rather than simply whether a resource differs from a prescribed configuration.

REFERENCE

1) Mallesh A. Intelligent identity orchestration with AI-driven policy reconciliation for multi-cloud security. InInternational Conference of Global Innovations and Solutions 2025 Apr 26 (pp. 723-741). Cham: Springer Nature Switzerland.
2) Canale G. Enterprise-Ready Behavioral Risk Indicators in Cybersecurity: Operationalizing the CPF Framework through Privacy-Preserving Pattern Detection. Available at SSRN 5426954. 2025 Sep 1.
3) Errico H, Ngiam J, Sojan S. Securing the model context protocol (mcp): Risks, controls, and governance. arXiv preprint arXiv:2511.20920. 2025 Nov 25.
4) Martseniuk Y, Partyka A, Harasymchuk O, Korshun N. Automated conformity verification concept for cloud security. Cybersecurity Providing in Information and Telecommunication Systems 2024. 2024;3654:25-37.
5) Soundappan SJ. AI-Driven Threat Detection and Resilient Cloud Security Architecture for Modern Enterprise Systems. International Research Journal of Innovative Engineering. 2024 Dec 9;8(6):15685-96.
6) Suleiman N, Elugwu F. Cloud Security Automation: Enhancing Data Monitoring in a Multi-Cloud Ecosystem. BW Academic Journal. 2025 Nov 7;2.
7) Chukwunweike J. Coordinating PLC-based load shedding and variable-speed drives to prevent transformer overloading during industrial production peaks. Int J Electr Data Commun. 2025;6(2 Pt B):118-135. doi:10.22271/27083969.2025.v6.i2b.109.
8) Jayalath RK, Ahmad H, Goel D, Syed MS, Ullah F. Microservice vulnerability analysis: A literature review with empirical insights. IEEE Access. 2024 Oct 16;12:155168-204.
9) Nandi SR. Architecting Unified Security Frameworks Across Hybrid Infrastructures. Journal of Engineering and Computer Sciences. 2025 Aug 8;4(8):239-45.
10) Okusi O, Obiakor IJ, Adeloye FC. Designing resilient data risk management protocols for regulatory compliance and cyber incident response. International Journal of Advance Research Publication and Reviews. 2025;2(7):45-70.
11) Samuel Addo. Causal AI for strategic business planning: uncovering latent drivers of long-term organizational performance and resilience. World Journal of Advanced Research and Reviews, 2025, 26(2), 895-912. Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.1738
12) Nangi PR, Obannagari CK. A Multi-Layered Zero-Trust–Driven Cybersecurity Framework Integrating Deep Learning and Automated Compliance for Heterogeneous Enterprise Clouds. American International Journal of Computer Science and Technology. 2024 Jul 6;6(4):14-27.
13) Pushparathi VG. AI-Powered Cyber Defense and API Security Architecture for Multi-Cloud Enterprise Applications. Journal of Cyber-Physical Security and Robotics. 2025 Oct 1;1(02):134-45.
14) Gandikota SP. High-availability network diagnostics and configuration platform for real-time financial service delivery. International Journal of Computer Technology and Electronics Communication. 2025 Jul 15;8(4):11147-60.
15) Baladari V. Enhancing performance and security in multi-cloud and hybrid-cloud environments. International Journal of Core Engineering and Management. 2024;7(11):53-265.

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

16) Racheal Kikachukwu Ogan. (2023). AUTOMATED CONFIGURATION COMPLIANCE ANALYTICS FOR DETECTING OPERATIONAL DRIFT AND PREVENTING SERVICE DEGRADATION ACROSS MULTI-LOCATION RESTAURANT TECHNOLOGY SYSTEMS. INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM), 07(12), 807–819. https://doi.org/10.5281/zenodo.22267240
17) Jadala SK. Ransomware Resilience in Cloud and Enterprise Systems: Detection, Response, and Recovery Frameworks. Journal of Cyber-Physical Security and Robotics. 2025 Dec 29;1(02):63-86.
18) Shakir M. Architecting Intelligent Cloud Cyber Defense for Mission-Critical Enterprise Applications and Infrastructure. International Journal of Research and Applied Innovations. 2024 Dec 17;7(6):12048-59.
19) Damodaram R. Adaptive Cloud Cybersecurity Architectures for Real-Time Threat Detection and Compliance Monitoring. International Journal of Science, Research and Technology. 2025;8(6):15369-79.
20) Magajiaka J, Adesanmi B. Lifecycle cost and resilience performance evaluation of climate-resilient urban road infrastructure under climate risk scenarios. International Journal of Current Research. 2026;18(5):37242-37251.
21) Malik G. Cybersecurity governance and risk management for digital transformation. International Journal of Applied Mathematics. 2025 Nov 2;38(10s):2532-61.
22) Kalejaiye AN, Shonubi JA. Zero trust enforcement using microsegmentation, identity-aware proxies, and continuous adaptive risk assessment in multi-tenant cloud environments. Int J Comput Appl Technol Res. 2025;14(7):61-77.
23) Jeyalakshmi J, Gnanavel S, Vijay K, Berna IE. Threat landscape and common security challenges in cloud environments. InAnalyzing and mitigating security risks in cloud computing 2024 (pp. 194-213). IGI Global Scientific Publishing.
24) Le TD, Le-Dinh T, Uwizeyemungu S. Cybersecurity analytics for the enterprise environment: A systematic literature review. Electronics. 2025 May 31;14(11):2252.
25) Racheal Kikachukwu Ogan. Machine-learning-based API failure prediction and root-cause analytics for improving reliability of enterprise banking integration architectures at scale. Int J Comput Artif Intell 2021;2(2):142-153. DOI: 10.33545/27076571.2021.v2.i2a.378
26) Yatam SN. Secure and Scalable Messaging Ecosystems: Automating Multi-Platform Architectures with Adaptive Security in Multi-Cloud Environments. Journal Of Multidisciplinary. 2025 Jul 4;5(7):134-42.
27) Shonubi JA. Multi-Layered Zero Trust Architectures for Cross-Domain Data Protection in Federated Enterprise Networks and High-Risk Operational Environments. Int. J. Res. Publ. Rev. 2025 Jul;6:146-69.
28) Reddy RR. ZERO TRUST ARCHITECTURES IN MULTI-CLOUD ENVIRONMENTS: MITIGATING DATA BREACHES AND IDENTITY MISUSE. Power System Protection and Control. 2023 Aug 30;51(3):39-51.
29) Metibemu OC, Adesokan-Imran TO, Ajayi AJ, Tiwo OJ, Olutimehin AT, Olaniyi OO. Developing proactive threat mitigation strategies for cloud misconfiguration risks in financial SaaS applications. Journal of Engineering Research and Reports. 2025 Mar 15;27(3):393-413.
30) Ogan RK. Risk-based API governance for predicting integration failures, security anomalies, and service degradation across hybrid enterprise application environments. Int J Finance Manage Econ. 2024;7(2):871-882. doi:10.33545/26179210.2024.v7.i2.946.
31) Anh NH. Hybrid cloud migration strategies: balancing flexibility, security, and cost in a multi-cloud environment. Transactions on Machine Learning, Artificial Intelligence, and Advanced Intelligent Systems. 2024 Oct 7;14(10):14-26.
32) Chakraborty P, Himel HU, Bashir M, Sikder MS, Kaur H, Ansar MT, Kaur J, Tuhin MK. An Intelligent Cybersecurity Framework for Data Protection in Cloud Computing Environments. Spectrum of Engineering Sciences. 2024 Oct 26:657-77.
33) Islam MZ, Dhanekula A. Measuring the Security Impact of Zero Trust Access Controls: A Mixed-Methods Study of Identity-Based Policies (Cisco ISE+ AD) and Incident Reduction. American Journal of Data Science and Analytics. 2023 Jun 28;4(06):01-42.
34) Liu M, Liu T, Huang Z. Enhancing Incident Response for Cloud Data Breaches: Challenges, Best Practices, and Policy Recommendations. In2025 Cyber Awareness and Research Symposium (CARS) 2025 Oct 27 (pp. 1-12). IEEE.
35) Zhang W, Liu Y, Chen M. Account-sprawl Posture Mapping for Preventing Configuration Drift in Enterprise Cloud Estates. Journal of Economic and Management Research. 2025;2(5).

iJETRM
INTERNATIONAL JOURNAL OF ENGINEERING TECHNOLOGY RESEARCH & MANAGEMENT (IJETRM)
https://ijetrm.com/

36) Battula V. Security compliance in hybrid environments using Tripwire and CyberArk. International Journal of Research and Analytical Reviews. 2023;10(2):788-803.
37) Akande NA, Eziokwu UJ, Cynthia U, Ogunsanya VA, Oyebode DF. Strengthening Data Governance in Multi-Cloud Environments: A Framework for Security, Compliance, and Operational Resilience. Educational Research (IJMCER). 2025;7(6):71-83.
38) Antwi NW. Threat Detection in Multi-Cloud Environments. InEnsuring Secure and Ethical STM Research in the AI Era 2025 (pp. 111-190). IGI Global Scientific Publishing.
39) Chintakayala DS. *Designing and Scaling OPA for PCI-DSS and HIPAA Compliance in AWS* (Doctoral dissertation, Dublin, National College of Ireland).
40) Sultana MS. PREDICTIVE NEURAL NETWORK MODELS FOR CYBERATTACK PATTERN RECOGNITION AND CRITICAL INFRASTRUCTURE VULNERABILITY ASSESSMENT. Review of Applied Science and Technology. 2025 Oct 20;4(02):777-819.
41) Zhang W, Liu Y, Chen M. Account-sprawl Posture Mapping for Preventing Configuration Drift in Enterprise Cloud Estates. Journal of Economic and Management Research. 2025;2(5).
42) Sendas N, Rajale D. MLOps Security in AI/ML. InThe Definitive Guide to Machine Learning Operations in AWS: Machine Learning Scalability and Optimization with AWS 2025 Jan 4 (pp. 125-181). Berkeley, CA: Apress.
43) Bukhari TT, Oladimeji O, Etim ED, Ajayi JO. Systematic review of SIEM integration for threat detection and log correlation in AWS-based infrastructure. Shodhshauryam, International Scientific Refereed Research Journal. 2023 Sep;6(5):479-512.
44) MGBAJIAKA, JOHNBOSCO. 2025. "Co-Designing Resilient Infrastructure: Participatory Approaches for Sustainable Implementation". *Current Journal of Applied Science and Technology* 44 (8):11-17. https://doi.org/10.9734/cjast/2025/v44i84586.
45) Sannareddy SB. Policy-driven infrastructure lifecycle control plane for Terraform-based multi-cloud environments. International Journal of Engineering & Extended Technologies Research (IJEETR). 2025 Mar 5;7(2):9661-71.

