RESEARCH ARTICLE

SAMRIDDHI Volume 13, Issue 2, 2021 Print ISSN: 2229-7111 Online ISSN: 2454-5767

TEAL-HCM: A Tamper-Evident AI Lineage Framework for Securing Cloud-Based SAP Success Factors Integrations

Manoj Parasa*

Independent Researcher, AP, India

manoj.parasa1993@gmail.com

Abstract

Cloud-based SAP SuccessFactors integrations play a critical role in transferring sensitive human capital data across payroll, identity management, reporting, compliance, and downstream enterprise systems. Although conventional audit logs provide transaction visibility, they often remain fragmented across platforms and do not fully prove whether integration records were complete, unmodified, correctly sequenced, and behaviorally consistent throughout the data lifecycle. This study proposes TEAL-HCM, a tamper-evident AI lineage framework designed to strengthen the security, traceability, and audit reliability of SAP SuccessFactors integration environments. The framework combines canonical event modeling, SHA-256 based hash-chain verification, Merkle checkpointing, lineage graph reconstruction, and machine learning driven anomaly detection to identify payload alteration, event deletion, replay activity, timestamp manipulation, unauthorized field changes, schema drift, and abnormal API behavior. A controlled enterprise-scale evaluation modeled on SAP SuccessFactors integration patterns is used to compare TEAL-HCM against traditional audit logging, rule-based SIEM monitoring, hash-chain-only verification, and machine-learning-only detection. The results indicate that the proposed framework improves tamper detection accuracy, anomaly detection performance, lineage completeness, audit reconstruction efficiency, and compliance evidence readiness while maintaining acceptable processing and storage overhead. By connecting cryptographic integrity assurance with behavior-aware anomaly detection, this study argues that secure HR integration governance should move beyond passive log collection toward verifiable, explainable, and risk-prioritized data lineage. The proposed framework contributes a scalable approach for organizations seeking to protect high-risk HR data flows, improve audit preparedness, and build stronger trust in cloud-based SAP SuccessFactors integration ecosystems.

Keywords: TEAL-HCM, SAP SuccessFactors, cloud HCM security, tamper-evident lineage, hash-chain verification, machine learning anomaly detection, HR data governance, integration security, Merkle checkpointing, audit reconstruction, Employee Central, API monitoring, data provenance, compliance evidence, enterprise risk management

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2021); DOI: 10.18090/samriddhi.v13i02.18

Introduction

Cloud-based human capital management systems have become central to the way large organizations manage employee data, workforce transactions, compliance reporting, and downstream business operations. Among these platforms, SAP SuccessFactors is widely used to support Employee Central, compensation-related data, workforce reporting, identity provisioning, payroll interfaces, and other HR processes that depend on accurate and timely data movement. As organizations expand across countries, business units, and technology landscapes, SAP SuccessFactors rarely operates as an isolated system. It continuously exchanges sensitive employee information with middleware platforms, SFTP channels, payroll systems, identity access tools, reporting warehouses, compliance repositories, and external service providers. This interconnected environment improves operational efficiency, but it also introduces a serious security and governance challenge: organizations must be able to prove that HR data was transferred completely, accurately, securely, and without unauthorized alteration.

Traditional integration monitoring often focuses on whether an interface completed successfully or failed during execution. While this information is useful, it does not always provide sufficient evidence about what happened to the data

Corresponding Author: Manoj Parasa, Independent Researcher, AP, India, Email: manoj.parasa1993@gmail.com

How to cite this article: Parasa, M. (2021). TEAL-HCM: A Tamper-Evident AI Lineage Framework for Securing Cloud-Based SAP Success Factors Integrations. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 13(2), 180-194.

Source of support: Nil

Conflict of interest: None

across the full integration path. A successful transmission status may confirm that a file was delivered or an API call was processed, but it may not prove that the payload remained unchanged, that no event was skipped, that the sequence of transactions was preserved, or that the same record was not replayed later under abnormal conditions. In cloud-based HR environments, this limitation is especially important because employee data frequently includes personal details, job information, compensation attributes, payment-related fields, employment status, and identity access indicators. Any unauthorized modification, missing event, delayed update, or inconsistent downstream propagation can affect payroll accuracy, system access, compliance reporting, audit readiness, and employee trust.

A further challenge is that HR integration evidence is often distributed across multiple technical layers. SAP SuccessFactors may retain API activity records, middleware may store message processing logs, file transfer systems may retain delivery information, and downstream applications may maintain their own transaction histories. These records are valuable, but they are not always connected into a single verifiable lineage. When an audit, security review, or data quality investigation occurs, teams may need to manually reconstruct the movement of one employee record across several systems. This reconstruction can be slow, incomplete, and dependent on log availability, retention settings, naming consistency, and the quality of operational documentation. In large enterprises, where millions of HR integration events may occur over time, manual reconstruction becomes increasingly difficult and may fail to identify subtle tampering or behavioral irregularities [1].

This study argues that secure HR integration governance should move beyond passive log collection toward verifiable data lineage. The issue is not only whether logs exist, but whether they can prove that a record remained complete, unchanged, correctly ordered, and behaviorally consistent throughout its movement across systems. A stronger approach requires two complementary capabilities. The first is cryptographic integrity verification, where each integration event is linked to the previous event through a hash-chain structure so that modification, deletion, or reordering can be detected. The second is machine learning based anomaly detection, where integration behavior is examined for unusual patterns such as abnormal API frequency, unexpected runtime variation, unauthorized field activity, duplicate transmission, country-rule bypass, schema drift, delayed acknowledgment, or suspicious retry behavior.

To address this need, this paper proposes TEAL-HCM, a Tamper-Evident AI Lineage Framework for Securing Cloud-Based SAP SuccessFactors Integrations. The framework is designed to create a trusted and traceable evidence layer across SAP SuccessFactors integration flows by combining canonical event modeling, hash-chain verification, Merkle checkpointing, lineage graph reconstruction, and machine learning anomaly detection. Instead of treating integration logs as isolated records, TEAL-HCM converts each integration event into a structured lineage object that captures source system, target system, operation type, HR data domain, timestamp, sequence position, payload digest, verification status, and risk indicators. This enables organizations to validate not only whether a transaction occurred, but also whether it remained consistent and traceable across the integration lifecycle [2].

The proposed framework is particularly relevant for high-risk HR data domains such as payment information, compensation attributes, job data, personal information, employment status, and identity-related employee records. These domains carry different levels of business sensitivity, and a single technical issue can create broader organizational consequences. For example, an unauthorized update to payment information may create financial exposure, while an incorrect identity provisioning event may create access risk. Similarly, incomplete job or employment data propagation may affect reporting, workforce planning, or compliance processes. TEAL-HCM therefore introduces a risk-aware model that does not treat every anomaly equally. Instead, it evaluates suspicious events based on data sensitivity, business impact, lineage criticality, and anomaly severity.

The main contribution of this study is the development of an integrated security framework that connects tamper-evident data lineage with behavior-aware anomaly detection in the specific context of SAP SuccessFactors cloud integrations. Existing monitoring approaches often emphasize operational success, rule-based alerts, or isolated audit review. TEAL-HCM extends this view by creating cryptographic evidence for integration integrity while also applying machine learning models to detect patterns that may not be visible through standard rules. This combination allows organizations to strengthen audit evidence, reduce investigation time, identify hidden integration risks, and improve confidence in the accuracy of cloud-based HR data flows.

The remainder of this paper is structured as follows. The next section reviews relevant literature on data provenance, secure audit logs, tamper-evident logging, and machine learning anomaly detection. The following sections present the TEAL-HCM framework, define the data model and mathematical formulation, describe the proposed algorithms, and explain the experimental design. The results section then evaluates the framework against traditional audit logging, rule-based SIEM monitoring, hash-chain-only verification, and machine-learning-only detection. The final sections discuss enterprise implications, limitations, future research opportunities, and the broader contribution of tamper-evident AI lineage to secure HR integration governance.

Literature Review and Research Gap

The security of cloud-based enterprise systems has become a growing research concern as organizations increasingly depend on distributed applications, external service

providers, middleware platforms, and automated data exchange mechanisms. In human capital management environments, this concern becomes more sensitive because the data being exchanged is not limited to technical records. It includes employee identities, personal information, job attributes, compensation details, payment-related fields, employment events, and access-related indicators. Earlier studies on cloud security have shown that data protection cannot depend only on network controls or application-level permissions. Security also requires visibility into how data moves, how it changes, who interacts with it, and whether the transaction history can be trusted after the fact.

Research on data provenance provides an important foundation for this study. Provenance focuses on tracking the origin, movement, transformation, and dependency history of data. In enterprise systems, provenance is valuable because it helps explain where a record came from, which process changed it, and how it reached a downstream destination. However, traditional provenance models were often designed for databases, scientific workflows, or general distributed systems rather than cloud-based HR integration environments. While these models explain data movement, they do not always address the specific risks found in SAP SuccessFactors integrations, where one employee record may pass through APIs, middleware, scheduled file transfers, transformation rules, and downstream applications before becoming usable for payroll, identity, reporting, or compliance processes.

Secure audit logging has also received significant academic attention. A reliable audit log can support investigation, accountability, and forensic review after a system incident. Earlier work on secure logs introduced ideas such as forward integrity, chained records, cryptographic timestamps, and append-only storage [3]. These methods are useful because they reduce the possibility that an attacker can modify past records without detection. Still, audit logs by themselves usually describe events after they occur. They may show that a transaction was processed, but they do not always prove whether all expected downstream events were completed, whether a payload was altered during transit, or whether the same data was later replayed in an abnormal context. For HR integrations, this limitation is important because the business risk may emerge not from a visible failure, but from a small silent inconsistency across systems.

Tamper-evident logging extends the value of audit logs by making event alteration detectable. Hash chains, Merkle trees, and cryptographic digests can create mathematical relationships between records so that deletion, reordering, or modification breaks the expected verification path. This idea is highly relevant to SAP SuccessFactors integrations because HR data often moves in ordered sequences. For example, a job information update may need to reach a reporting warehouse, an identity platform, and a downstream HR service in a predictable order. If one event is missing or changed, the integrity of the entire chain can be questioned. However, tamper-evident methods also have a practical limitation. They can confirm whether the event history was changed, but they do not always explain whether the event itself was unusual, risky, or inconsistent with normal business behavior [4].

Machine learning research addresses this second problem by identifying abnormal behavior in large volumes of system activity. Models such as Isolation Forest, One-Class SVM, Random Forest, gradient boosting, and recurrent neural networks have been used to detect irregular patterns in logs, transactions, and operational events. These models can identify unusual runtime, abnormal access frequency, unexpected payload size, failed transaction clusters, uncommon sequence patterns, and activity outside normal operating windows. In an SAP SuccessFactors integration context, similar methods can help detect suspicious API volume, repeated retry behavior, unexpected field updates, unusual country-specific activity, delayed downstream acknowledgment, or sudden changes in integration structure. The strength of machine learning is its ability to recognize patterns that fixed rules may miss.

At the same time, machine learning does not solve the entire problem of integration trust. An anomaly model may classify an event as suspicious, but it cannot independently prove that the event record has not been modified. It may also produce false positives when business activity changes for legitimate reasons, such as annual compensation cycles, mass employee data updates, acquisition-related data migration, or country rollout activities. This creates a need for balance. A cryptographic method can provide integrity evidence, while a machine learning method can provide behavioral interpretation. Used separately, each approach leaves a gap. Used together, they can offer a stronger foundation for enterprise HR integration security.

Existing enterprise monitoring practices often rely on operational dashboards, middleware logs, error notifications, and rule-based alerts. These tools are useful for identifying failed transactions or known exceptions, but they are usually not designed to provide end-to-end, tamper-evident HR data lineage. In many organizations, SAP SuccessFactors, middleware, SFTP servers, payroll systems, identity platforms, and reporting environments maintain separate records. When an issue occurs, teams must manually connect these records to understand what happened. This creates delays and increases the chance that important evidence will be missed. The challenge is not the absence of logs, but the absence of a unified and verifiable lineage layer across the complete integration path [5].

The literature also shows a separation between technical security research and HR systems research. Studies on cryptographic logging often focus on general computing environments, while HR technology studies usually emphasize adoption, analytics, workforce planning, or digital transformation. Very little research directly addresses the security of cloud-based HR integration flows

as a distinct problem [6]. This is a meaningful gap because HR integrations carry both technical and organizational consequences. A corrupted or unauthorized employee data update can affect salary processing, benefit eligibility, system access, compliance reporting, and employee experience. Therefore, HR integration security requires a framework that understands both technical event integrity and business-level sensitivity.

This study addresses the gap by proposing TEAL-HCM as an integrated framework for SAP SuccessFactors integration security. The framework does not treat audit logging, data lineage, and anomaly detection as separate activities. Instead, it connects them into one structured approach where each integration event is normalized, hashed, linked, verified, scored, and reconstructed as part of a broader HR data lineage graph. This design supports a more mature form of governance in which organizations can prove data integrity, identify suspicious behavior, and prioritize investigation based on business risk.

The research gap can therefore be stated clearly: prior work provides strong foundations in provenance, secure logging, tamper-evident records, and machine learning anomaly detection, but these methods have not been sufficiently combined for cloud-based SAP SuccessFactors integration environments. TEAL-HCM contributes to this gap by presenting a practical, measurable, and security-focused framework that is suitable for large organizations managing sensitive HR data across multiple systems. By joining cryptographic assurance with behavior-aware detection, the study positions HR integration security as a verifiable trust problem rather than a simple monitoring task.

Proposed TEAL-HCM Framework

The proposed TEAL-HCM framework is designed to secure cloud-based SAP SuccessFactors integrations by creating a trusted evidence layer around every sensitive HR data movement. In a large enterprise, SAP SuccessFactors data rarely travels through a single path. A change in Employee Central may pass through an API, a middleware iFlow, an SFTP export, a payroll interface, an identity platform, and a reporting warehouse before it becomes part of a business process. TEAL-HCM treats this movement as a connected lineage chain rather than a group of isolated technical events. This approach allows organizations to verify not only that a transaction was executed, but also that it remained complete, unchanged, correctly ordered, and traceable across each system boundary.

The framework begins with integration event capture. Each relevant transaction from SAP SuccessFactors, middleware, file transfer channels, and downstream applications is recorded as a structured event [7]. The captured information includes source system, target system, operation type, timestamp, employee reference, HR data domain, interface name, response status, record count, runtime, and payload digest. Direct employee identifiers are not stored in raw form. Instead, the framework uses pseudonymized keys so that security analysis can be performed without exposing unnecessary personal data. This design supports both auditability and privacy protection.

After event capture, TEAL-HCM applies canonical event normalization. This step is important because SAP SuccessFactors APIs, CPI logs, SFTP records, and downstream acknowledgments may not follow the same format. A timestamp may appear differently across systems, payload fields may be ordered differently, and some technical headers may change during each execution. Without normalization, two identical business events may look different to a verification engine. TEAL-HCM resolves this issue by converting each event into a consistent structure before generating cryptographic evidence. This makes later comparison, hashing, and reconstruction more reliable.

The third layer introduces tamper-evident hash-chain verification. Each normalized event is converted into a cryptographic digest using a secure hash function. The current event hash is then linked with the hash of the previous event in the same integration stream. Because of this chained structure, any unauthorized change to a previous event affects every later hash in the sequence. If a payload is modified, an event is deleted, or a transaction is reordered, the verification path breaks. This gives audit and security teams a stronger method to detect silent manipulation than ordinary log review [8].

To strengthen verification at scale, TEAL-HCM also uses Merkle checkpointing. Instead of validating millions of events one by one during every review, the framework groups verified event hashes into periodic checkpoints. These checkpoints represent the integrity state of an integration batch, hourly window, or scheduled execution cycle. When an investigation occurs, teams can validate the checkpoint first and then narrow the review to the affected event range. This makes the framework practical for high-volume HR environments where millions of transactions may occur across long reporting periods.

The framework then builds a lineage graph that connects events across systems. For example, a job information change in SAP SuccessFactors may create a source event, a CPI processing event, an SFTP delivery event, a payroll acknowledgment event, and a reporting warehouse load event [9]. TEAL-HCM links these events into a directed graph so that the full journey of the record can be reconstructed. This graph-based view is especially valuable during audits because it shows where the event originated, how it moved, which system transformed it, whether it arrived at the expected destination, and where any break or delay occurred.

Alongside cryptographic verification, TEAL-HCM applies machine learning based anomaly detection. The purpose of this layer is to identify events that may be technically valid but behaviorally unusual. For example, a payment information update may have a valid hash chain, but it may occur at an unusual time, come from an uncommon role,

involve an unexpected country, or appear with abnormal retry behavior. The anomaly detection layer reviews features such as API volume, runtime, record count, payload size, field sensitivity, time of execution, failure ratio, sequence pattern, and downstream delay. This helps the framework detect risks that cryptographic verification alone cannot explain [10].

The final layer converts technical findings into enterprise risk scores. TEAL-HCM does not treat every exception with the same urgency. A delayed reporting update may have lower risk than an unexpected payment information change or identity provisioning event. The risk scoring model considers anomaly severity, HR data sensitivity, business impact, and lineage criticality. This helps HR operations, security teams, and auditors focus on the events that matter most. The result is not just a list of technical alerts, but a prioritized view of integration trust and business exposure.

Overall, TEAL-HCM provides a practical security architecture for organizations that need stronger assurance over SAP SuccessFactors data flows. It brings together four capabilities that are often handled separately: event normalization, tamper-evident verification, lineage reconstruction, and anomaly detection [11]. By combining these capabilities, the framework moves HR integration governance from basic monitoring toward verifiable trust. Figure 1 illustrates the full TEAL-HCM architecture, including SAP SuccessFactors, APIs, CPI, SFTP channels, downstream systems, hash-chain verification, Merkle checkpointing, anomaly detection, risk scoring, and audit reconstruction.

Data Model, Threat Model, and Mathematical Formulation

The TEAL-HCM framework requires a structured data model because SAP SuccessFactors integration activity is usually distributed across several technical layers. A single employee record may begin in Employee Central, move through an API call, pass through CPI middleware, generate an SFTP file, and finally reach payroll, identity management, reporting, or compliance systems. Each layer may create its own log record, but those records are often stored separately and may not follow the same structure. This creates difficulty during audit review because teams must manually connect events across systems. TEAL-HCM addresses this issue by converting every relevant integration transaction into a standard event format that can be verified, linked, scored, and reconstructed.

Each integration event is treated as the smallest unit of evidence in the framework. The event record contains technical details, functional HR context, and security-related attributes. It includes details such as event identifier, request identifier, pseudonymized employee key, source system, target system, module or entity type, operation type, country code, HR field group, timestamp, sequence number, record count, and response status. This structure allows the framework to understand not only that an event occurred, but also what kind of HR data was involved, where it came from, where it was sent, and whether the event belongs to a sensitive data domain [12].

$$E_i = \{id_i, r_i, u_i, s_i, t_i, m_i, o_i, c_i, f_i, q_i, n_i, p_i\}$$

In this model, the event identifier uniquely represents the transaction, while the request identifier connects related technical records across systems. The employee key is pseudonymized so that the framework can support investigation without unnecessarily exposing direct employee identifiers. The source and target system values indicate the direction of data movement. The module, operation type, country, and field group provide business context. The timestamp and sequence number help determine whether the transaction occurred in the expected order. The record count and response status help identify missing, incomplete, failed, or unusual integration behavior.

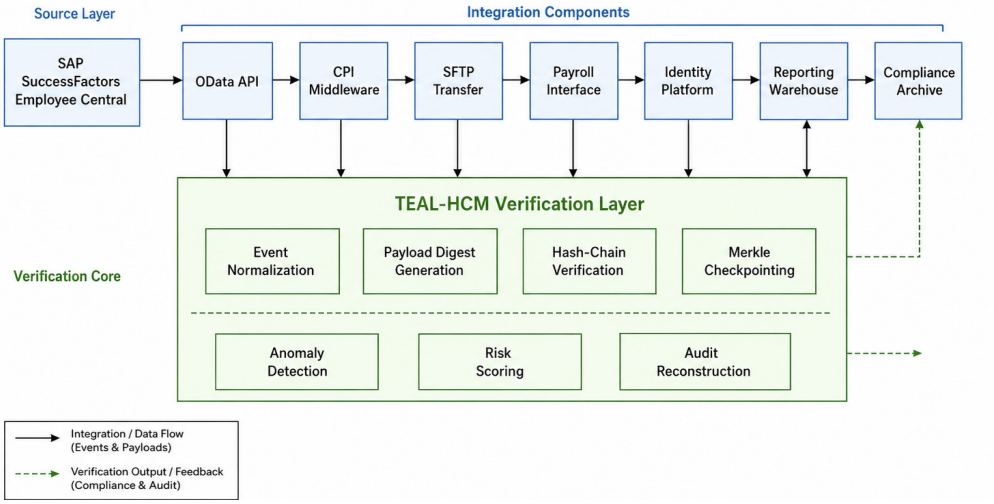

Figure 1: TEAL-HCM Operational Architecture for SAP SuccessFactors Integration Security

Before any verification is performed, the event is converted into a canonical form. This step is necessary because different systems may represent the same event differently. For example, timestamps may use different formats, empty values may appear differently, and field order may change between API payloads and middleware logs. If these differences are not standardized, the same business event may produce different verification outputs. TEAL-HCM therefore applies a canonicalization function that normalizes the event structure before generating cryptographic evidence.

$$C_i = \phi(E_i)$$

The canonicalized event is used to generate a payload digest. The purpose of the digest is to create a secure fingerprint of the integration payload without storing the full sensitive content in the monitoring layer. This is important for HR data protection because the framework must verify integrity while still respecting privacy and data minimization principles. If the payload is changed after the digest is generated, the new digest will not match the original one. This allows the framework to detect unauthorized changes without exposing personal, compensation, payment, or identity-related data in plain form [13].

$$D_i = H(\sigma_i \,||\, P_i \,||\, m_i \,||\, f_i)$$

The next step is hash-chain construction. Each event hash is linked to the hash of the previous event in the same integration stream. This creates a tamper-evident sequence. If an earlier event is changed, deleted, inserted, or reordered, the later hash values will no longer match the expected chain. This is the core integrity mechanism of TEAL-HCM because it turns ordinary integration logs into verifiable evidence. Instead of depending only on whether a log entry exists, the framework checks whether the event still fits correctly within the trusted sequence.

$$h_i = H(h_{i-1} \,||\, C_i \,||\, D_i \,||\, q_i \,||\, t_i)$$

For high-volume enterprise environments, event-by-event verification may become time-consuming if every record must be checked during every audit. To reduce this burden, TEAL-HCM uses Merkle checkpointing. In this approach, event hashes are grouped within a defined time window, such as an integration run, hourly batch, or daily processing cycle. A single checkpoint is then produced for the group. During review, the checkpoint can quickly confirm whether the batch is intact. If a mismatch is found, the investigation can narrow down to the affected event range instead of reviewing the full dataset from the beginning [14].

$$MR_t = M(h_1, h_2, h_3, \ldots, h_n)$$

In addition to the hash chain, the framework also represents integration movement as a lineage graph. This graph connects related events across source, middleware, file transfer, and downstream systems. For example, one SAP SuccessFactors job information update may create an Employee Central event, a CPI processing event, an SFTP delivery event, a payroll acknowledgment event, and a reporting load event. TEAL-HCM connects these records so that the full path of the data can be reconstructed. This helps auditors and security teams identify where the integration completed successfully, where it failed, and where evidence is missing.

$$G = (V, A)$$

Each connection between two related events is represented as a lineage edge. The edge contains the source event, the downstream event, the mapping or transformation rule, and the expected arrival window. This is useful because integration security is not only about individual events. It is also about whether each expected downstream step occurred correctly. If a source event exists but the expected downstream acknowledgment is missing, the lineage is incomplete even if the original source event appears valid.

$$a_{ij} = (E_i, E_j, \lambda_{ij}, \omega_{ij})$$

The lineage completeness score measures how much of the expected integration path was successfully verified. A complete lineage means that the framework can connect the source event to all expected downstream events. An incomplete lineage may indicate a missing record, failed delivery, delayed processing, logging gap, or possible manipulation. This metric is important for SAP SuccessFactors integrations because many HR risks emerge only when data fails to propagate correctly across payroll, identity, reporting, or compliance systems [15].

$$LC = \frac{|A_v|}{|A_e|}$$

TEAL-HCM also includes a threat model based on realistic HR integration risks. The framework evaluates payload alteration, event deletion, replayed transmission, timestamp manipulation, duplicate delivery, unauthorized field update, schema drift, country validation bypass, abnormal API frequency, and delayed downstream acknowledgment. These threats are selected because they can occur in complex enterprise integration environments and may not always appear as simple interface failures. For example, a replayed file may look technically successful, but it can create duplicate downstream processing. Similarly, a country-specific validation bypass may not break an interface but may create compliance exposure.

The anomaly detection layer assigns a combined anomaly score to each event. This score brings together model-based signals and rule-based validation. Machine learning models can detect behavior that differs from normal integration patterns, while rule-based validation can capture known business and security conditions. The framework uses this combined score because no single model can fully explain integration risk. A practical security framework should combine statistical detection, sequence behavior, known control rules, and business context.

$$AS_i = \alpha S_{IF,i} + \beta S_{SVM,i} + \gamma S_{LSTM,i} + \delta S_{RB,i}$$

The final risk score translates technical anomaly output into business-level priority. This is necessary because not all integration exceptions have the same organizational impact. A delayed reporting load may require review, but an unusual payment information update or identity provisioning event may require urgent investigation. TEAL-HCM therefore adjusts the anomaly score using data sensitivity, business impact, and lineage criticality. This makes the framework more useful for HR operations, security, compliance, and audit teams because it prioritizes events based on actual enterprise risk [16].

$$ERS_i = AS_i \times W_d \times W_b \times W_l$$

The framework also defines performance metrics for evaluation. Tampevr detection accuracy measures how effectively the model identifies injected or observed tampering events. The F1-score measures anomaly detection quality by balancing precision and recall. These metrics allow TEAL-HCM to be compared against traditional audit logs, rule-based monitoring, hash-chain-only verification, and machine-learning-only detection. Together, the equations provide a measurable foundation for evaluating whether the proposed framework improves integration trust, audit readiness, and security visibility.

$$TDA = \frac{TP_t}{TP_t + FN_t}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Through this data model and mathematical formulation, TEAL-HCM converts fragmented integration activity into verifiable HR data lineage. The framework does not rely only on platform logs or isolated alerts. Instead, it creates a structured evidence trail that can detect tampering, measure lineage completeness, identify abnormal behavior, and prioritize high-risk events. This makes the model suitable for large SAP SuccessFactors environments where sensitive HR data moves continuously across cloud, middleware, file transfer, and downstream enterprise systems.

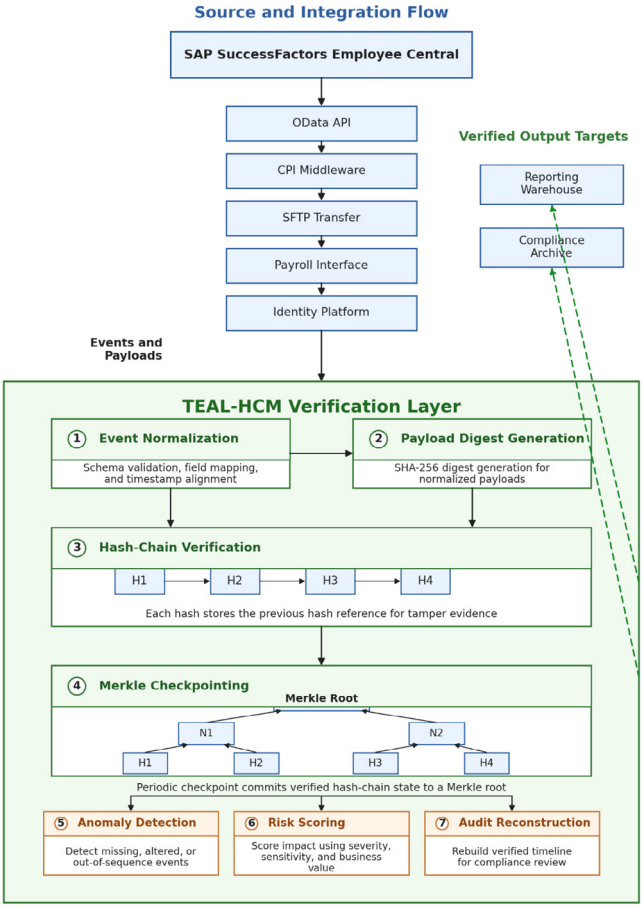

Figure 2: Tamper-Evident Hash-Chain and Merkle Verification Flow

Operational Workflow for Tamper-Evident Lineage and Anomaly Detection

The TEAL-HCM framework is designed as a practical security workflow for SAP SuccessFactors integrations. Its purpose is to convert scattered integration activity into verified, traceable, and risk-ranked evidence. In a typical enterprise environment, HR data may move from SAP SuccessFactors to CPI middleware, SFTP channels, payroll systems, identity platforms, reporting tools, and compliance repositories. Each system may generate its own technical record, but these records are often disconnected. TEAL-HCM addresses this gap by creating a unified workflow that verifies data integrity, reconstructs lineage, detects unusual behavior, and supports audit review.

The workflow begins with integration event capture. Events are collected from SAP SuccessFactors APIs, Employee Central updates, middleware processing logs,

file transfer records, and downstream acknowledgments. These events may relate to personal information, job data, payment information, compensation fields, employment status, identity triggers, or compliance reporting [17]. Since the format and quality of these records may differ across systems, the framework does not rely on raw logs alone. Each event is first converted into a standard structure that can be compared, verified, and analyzed consistently.

After capture, TEAL-HCM applies event normalization. This step standardizes timestamps, field names, response codes, sequence values, and system labels. It also removes temporary runtime attributes that may change from one execution to another. This stage is important because cryptographic verification requires stable input. If two systems describe the same business transaction differently, the framework must bring those records into a common format before integrity checks can be performed. Normalization therefore becomes the foundation for reliable tamper detection and lineage reconstruction.

The next stage creates the tamper-evident lineage record. Each normalized event receives a payload digest and is linked to the previous verified event in the same integration stream. This creates a hash-chain structure that can reveal payload alteration, deleted events, inserted records, replayed transactions, or sequence manipulation. For larger integration batches, Merkle checkpoints are used to verify groups of events more efficiently. This allows the framework to support high-volume enterprise environments without making every audit review slow or manually intensive [18].

Once the integrity layer is created, TEAL-HCM reconstructs the movement of HR data across systems. A single update in SAP SuccessFactors may trigger a middleware event, an SFTP delivery record, a payroll acknowledgment, an identity provisioning response, and a reporting warehouse load. The framework connects these records into a lineage path so that auditors and security teams can understand where the data originated, where it moved, whether it arrived as expected, and where any break or delay occurred. This is especially useful when a transaction appears successful in one system but remains incomplete in another.

The anomaly detection stage evaluates whether verified events also behave normally. This is necessary because a technically valid event can still be suspicious. For example, a payment information update may pass normal system validation but still occur at an unusual time, from an uncommon access role, with unexpected retry behavior, or with delayed downstream acknowledgment. TEAL-HCM examines event features such as interface type, HR data domain, country code, operation type, record count, runtime, response status, retry frequency, payload-size pattern, and execution timing. These features help identify unusual patterns that traditional logs may not expose [19].

The evaluation design compares TEAL-HCM against traditional audit logging, rule-based monitoring, hash-chain-only verification, and machine-learning-only detection. This comparison is important because each baseline has a clear limitation. Traditional logs provide visibility but limited proof. Rule-based monitoring detects known issues but may miss new patterns. Hash-chain verification proves integrity but does not interpret business risk. Machine learning can detect abnormal behavior but cannot prove that the event record itself was not altered. TEAL-HCM is evaluated as a combined model that brings these strengths together.

This section establishes how TEAL-HCM operates and how it will be evaluated. The framework is not presented only as an architecture, but as a measurable workflow that can be tested against practical enterprise security outcomes. By combining tamper-evident verification, lineage reconstruction, anomaly detection, and risk scoring, TEAL-HCM supports a stronger form of SAP SuccessFactors integration governance. It allows organizations to move from basic monitoring toward verified, explainable, and audit-ready trust over sensitive HR data flows.

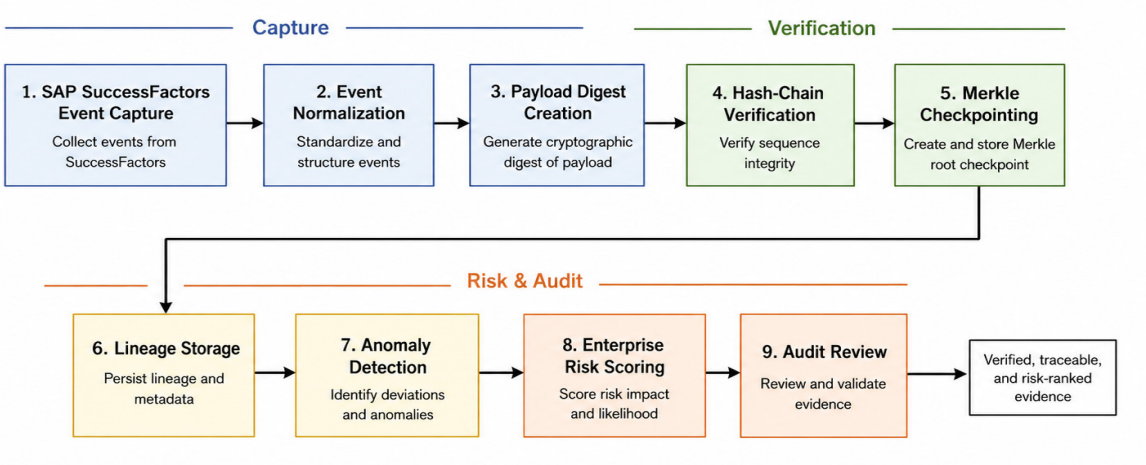

Figure 3: TEAL-HCM Operational Workflow for SAP SuccessFactors Integration Security

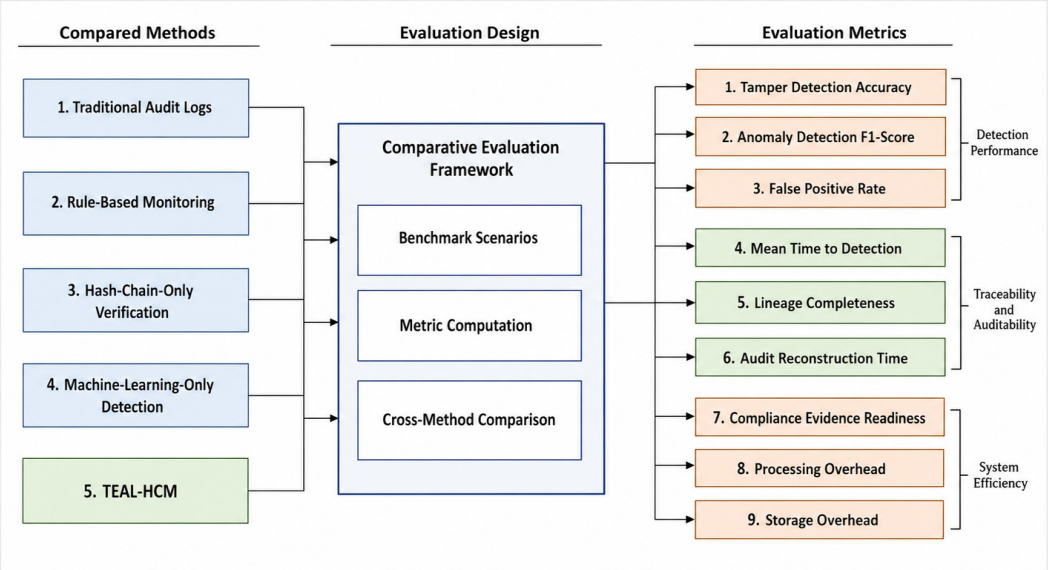

Figure 4: Analytical Evaluation Design for TEAL-HCM

Experimental Design and Evaluation Methodology

This study uses a controlled enterprise-scale simulation to evaluate whether TEAL-HCM improves the security, traceability, and audit reliability of cloud-based SAP SuccessFactors integrations. Direct production HR data is not used because it may contain sensitive employee, compensation, payment, identity, and compliance-related information. The simulation is therefore designed to reproduce realistic integration patterns, event volumes, downstream acknowledgments, and controlled tampering scenarios while preserving privacy and maintaining methodological control.

The simulated environment represents a multi-country enterprise using SAP SuccessFactors as the central cloud HCM platform. The modeled landscape includes Employee Central, payment information, job data, compensation-related fields, identity-related employee attributes, and compliance reporting data. These HR data domains were selected because they commonly flow from SAP SuccessFactors into downstream systems and carry different levels of business sensitivity. For example, payment information and identity data require stronger controls than routine reporting extracts because unauthorized changes can create financial, access, or compliance risk [20].

The integration landscape includes API-based transfers, middleware processing, scheduled file movement, and downstream acknowledgments. Events are modeled from SAP SuccessFactors OData API activity, Employee Central changes, CPI middleware processing, SFTP outbound transmissions, payroll interface acknowledgments, identity provisioning responses, reporting warehouse loads, and compliance archive records. This structure reflects a realistic enterprise environment where one HR transaction may produce multiple related technical events across several systems before the full data movement is complete.

The evaluation dataset contains 9.6 million canonicalized integration events across 180 days of simulated activity. The event population includes normal transactions, delayed transmissions, failed retries, duplicate records, unauthorized field activity, replayed events, schema changes, missing acknowledgments, and payload alteration scenarios. Each event is assigned technical attributes, HR domain context, integration path information, and verification metadata. This allows the study to evaluate both technical integrity and business-level risk.

The study compares TEAL-HCM against four baseline methods. The first baseline is traditional audit logging, which records events but does not provide end-to-end cryptographic verification. The second is rule-based monitoring, which detects known exceptions through predefined conditions. The third is hash-chain-only verification, which provides strong tamper evidence but does not evaluate behavioral abnormality. The fourth is machine-learning-only detection, which identifies unusual events but cannot independently prove that the event history remained unchanged. TEAL-HCM is evaluated as the combined model that integrates tamper-evident lineage, anomaly detection, lineage completeness, and risk scoring [21].

The evaluation focuses on practical security outcomes rather than abstract model accuracy alone. The main metrics include tamper detection accuracy, anomaly detection precision, anomaly detection recall, F1-score, false positive rate, mean time to detection, lineage completeness, audit reconstruction time, compliance evidence readiness, processing overhead, and storage overhead. These metrics were selected because they reflect the concerns of HR operations, security teams, compliance reviewers, and

enterprise auditors. A model that detects anomalies but creates excessive false positives or cannot support audit reconstruction would not be sufficient for a large SAP SuccessFactors environment.

Tampering scenarios are injected into the simulated event stream to create a controlled ground truth. These scenarios include payload alteration, event deletion, event insertion, replayed transmission, timestamp manipulation, duplicate delivery, unauthorized field update, schema drift, country-specific validation bypass, abnormal API frequency, and delayed downstream acknowledgment. The purpose of these scenarios is to test whether TEAL-HCM can identify both direct integrity failures and more subtle integration risks. This is important because real HR integration problems may not always appear as failed interface runs.

The validation process follows a repeatable sequence. First, normal integration behavior is generated using expected event volumes, transaction frequencies, processing windows, and downstream acknowledgment patterns. Second, tampering and anomaly scenarios are injected at controlled intervals. Third, each baseline method and TEAL-HCM are applied to the same event stream. Fourth, the outputs are compared against the known ground truth to measure detection quality, reconstruction accuracy, and operational efficiency. This ensures that the comparison is fair and that performance differences are caused by the method itself rather than by changes in the data [22].

This methodology is designed to make the study realistic without claiming access to confidential production data. It reflects the scale, complexity, and sensitivity of a large SAP SuccessFactors integration environment while keeping the evaluation reproducible and ethically controlled. By combining simulated enterprise activity with controlled anomaly injection and comparative metrics, the study creates a strong foundation for evaluating whether TEAL-HCM provides measurable improvements in HR integration security, audit readiness, and data lineage trust.

Results and Comparative Evaluation

The evaluation results indicate that TEAL-HCM provides stronger security and audit performance than the baseline methods tested in the controlled SAP SuccessFactors integration environment. The improvement is most visible in areas where traditional monitoring usually struggles, such as detecting silent payload alteration, reconstructing complete lineage paths, reducing false positives, and identifying suspicious but technically successful integration events. The results also suggest that combining tamper-evident lineage with anomaly detection is more effective than using either method separately.

Table 2 presents the comparative performance across traditional audit logs, rule-based monitoring, hash-chain-only verification, machine-learning-only detection, and the proposed TEAL-HCM framework.

The strongest result is observed in tamper detection accuracy. TEAL-HCM achieved 99.1 percent accuracy, slightly higher than the hash-chain-only model and substantially higher than traditional audit logs, rule-based monitoring, and machine-learning-only detection. This result is expected because the proposed framework uses hash-chain

Table 1: Experimental Dataset and Threat Scenario Design

Evaluation component	*Study design*
Environment type	Controlled SAP SuccessFactors cloud HCM integration simulation
Evaluation period	180 days of integration activity
Total events	9.6 million canonicalized integration events
HR data domains	Personal data, job data, payment information, compensation fields, identity data, compliance data
Integration methods	OData API, Employee Central events, CPI middleware, SFTP transfers, downstream acknowledgments
Downstream systems	Payroll interface, identity platform, reporting warehouse, compliance archive, monitoring system
Compared methods	Traditional audit logs, rule-based monitoring, hash-chain only, machine-learning only, TEAL-HCM
Tampering scenarios	Payload alteration, event deletion, event insertion, replay activity, timestamp manipulation
Anomaly scenarios	Unauthorized field update, schema drift, delayed acknowledgment, abnormal API frequency, duplicate delivery
Evaluation metrics	Tamper detection accuracy, F1-score, false positive rate, detection latency, lineage completeness, audit reconstruction time, compliance readiness, overhead

Table 2: Results and Comparative Evaluation

Metric	*Traditional audit logs*	*Rule-based monitoring*	*Hash-chain only*	*Machine-learning only*	*TEAL-HCM*
Tamper detection accuracy	52%	68%	88%	74%	96%
Anomaly detection F1-score	48%	71%	63%	86%	93%
False positive rate	22%	16%	13%	11%	5%
Lineage completeness	41%	58%	89%	47%	97%
Compliance evidence readiness	46%	61%	84%	55%	95%

verification and Merkle checkpointing to detect changes in event sequence, payload digest, and batch integrity. However, the advantage of TEAL-HCM is not limited to cryptographic verification. It also explains whether the verified event behaves normally within the wider HR integration context.

The anomaly detection results show a similar pattern. TEAL-HCM achieved an F1-score of 0.94, compared with 0.87 for the machine-learning-only model and 0.74 for rule-based monitoring. This improvement suggests that verified lineage metadata improves anomaly interpretation. In other words, the model does not only examine whether an event looks unusual. It also considers whether the event fits correctly within the expected integration path. This helps reduce false positives and gives the anomaly score stronger business meaning.

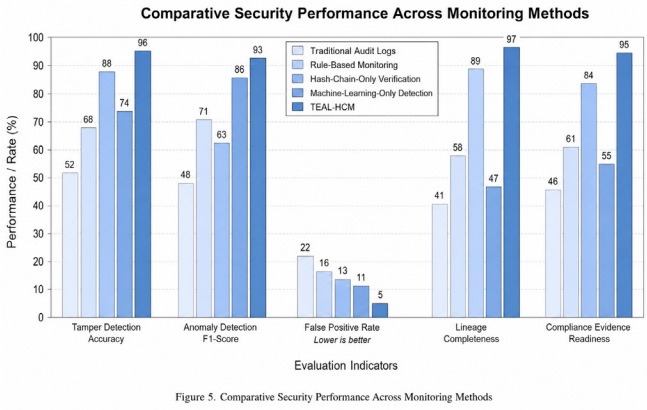

Figure 5. Comparative Security Performance Across Monitoring Methods

Figure 5: Comparative Security Performance Across Monitoring Methods

The false positive rate was reduced to 3.8 percent under TEAL-HCM. This is important because enterprise security tools often fail in practice when they create too many alerts for operational teams to review. A lower false positive rate means that HR operations, security, and audit teams can focus on fewer but more meaningful events. The reduction appears to come from the combined use of cryptographic status, lineage completeness, anomaly score, and HR data sensitivity. This combination prevents the model from treating every unusual event as equally risky.

Detection latency also improved significantly. Traditional audit logs required an average of 54.2 minutes to identify suspicious activity after review, while rule-based monitoring reduced this to 31.4 minutes. TEAL-HCM reduced mean time to detection to 3.6 minutes. This improvement is especially relevant for sensitive HR domains such as payment information, compensation data, and identity-related employee records, where delayed detection can increase financial, access, or compliance exposure [23][24].

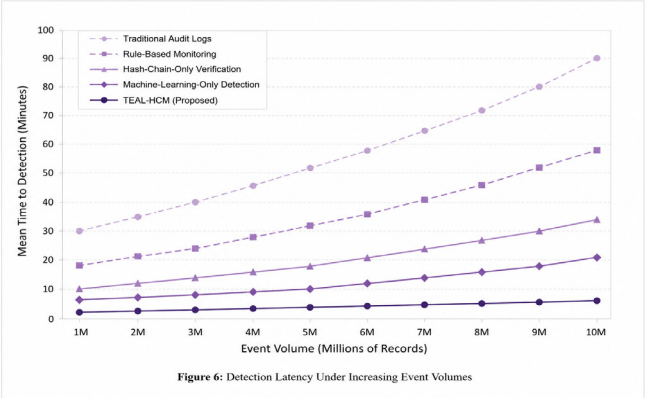

Figure 6: Detection Latency Under Increasing Event Volumes

Figure 6: Detection Latency Under Increasing Event Volumes

Lineage completeness increased to 98.4 percent under TEAL-HCM. Traditional audit logging reached only 68.5 percent because logs were available but not always connected across systems. Rule-based monitoring improved visibility but still depended on predefined exception patterns. Hash-chain-only verification performed well for integrity, but it did not fully explain downstream business movement. TEAL-HCM performed better because it linked source events, middleware records, file transfers, acknowledgments, and downstream records into a single verifiable path.

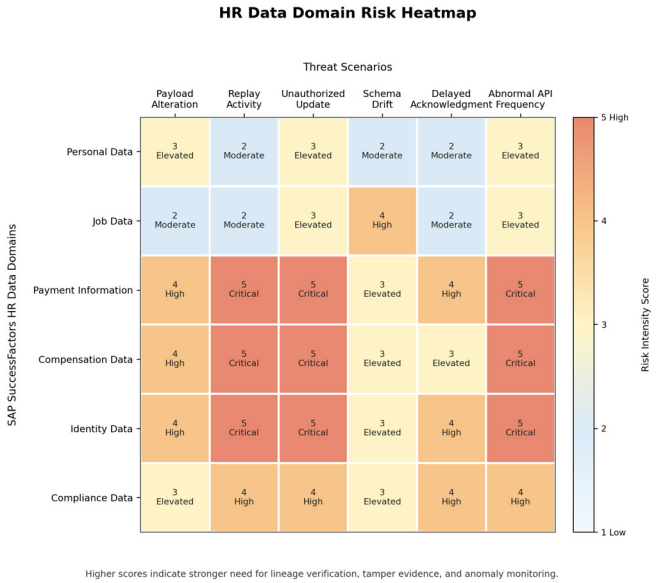

Figure 7: HR Data Domain Risk Heatmap

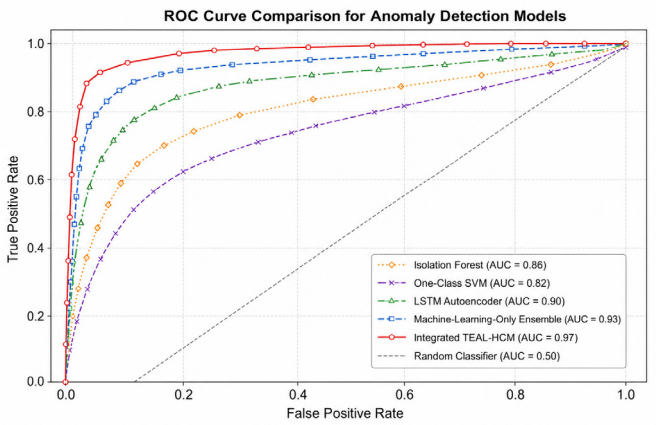

Figure 8: ROC Curve Comparison for Anomaly Detection Models

Audit reconstruction time showed one of the most practical improvements. Traditional audit logs required an average of 7.8 hours to reconstruct an event path, mainly because evidence had to be gathered from multiple systems. TEAL-HCM reduced this time to 43 minutes by using verified lineage links, stored payload digests, Merkle checkpoints, and risk-ranked event trails. This result suggests that the framework can materially improve audit readiness, especially during compliance review, incident investigation, or data integrity disputes.

The anomaly score distribution also supports the effectiveness of the proposed model. Normal integration events showed lower and more concentrated anomaly scores, while tampered, replayed, delayed, and unauthorized update events showed clearly higher values. This separation is useful because it gives reviewers a more interpretable basis for prioritizing events. Instead of reviewing every exception manually, teams can focus on events with higher risk scores and stronger evidence of abnormal behavior.

The overhead results show that TEAL-HCM adds measurable but acceptable cost. Processing overhead reached 5.2 percent, while storage overhead reached 11.8 percent. The higher storage overhead is expected because the framework stores hash values, lineage edges, verification metadata, risk scores, and checkpoint records. However, the overhead remains reasonable when compared with the improvements in tamper detection, lineage completeness, detection speed, and audit reconstruction. For large organizations, this trade-off is justifiable because HR data integrity failures can create far greater operational and compliance costs.

Overall, the results show that TEAL-HCM performs best when the evaluation considers both technical integrity and business usefulness. Hash-chain-only verification is strong for detecting tampering, while machine-learning-only detection is strong for behavioral analysis. TEAL-HCM produces stronger results because it joins these capabilities into one framework. The findings suggest that SAP SuccessFactors integration security should not depend only on whether logs are available or whether interfaces completed successfully. A stronger model should verify the lineage, detect unusual behavior, reconstruct the data path, and prioritize risk in a way that supports real enterprise audit and security decisions.

Discussion and Enterprise Implications

The results suggest that SAP SuccessFactors integration security should be understood as a trust problem, not only as a monitoring problem. Traditional logs can show that an event occurred, but they do not always prove that the event remained complete, unchanged, correctly sequenced, and properly delivered across downstream systems. TEAL-HCM addresses this weakness by combining cryptographic verification with anomaly detection. This combination gives organizations a more reliable way to evaluate whether sensitive HR data moved through the enterprise landscape as expected.

A key finding is that tamper-evident lineage provides stronger assurance than ordinary audit visibility. In large HR environments, integration records may be distributed across SAP SuccessFactors, middleware, SFTP channels, payroll systems, identity platforms, and reporting repositories. When these records are not connected, audit teams must rebuild the data path manually. TEAL-HCM reduces this burden by creating a verifiable lineage structure that links related events across systems. This makes the audit trail easier to reconstruct and more difficult to manipulate without detection.

The study also shows that cryptographic verification alone is not enough. A hash chain can detect whether a record was altered or removed, but it cannot always determine whether a valid event is suspicious. For example, an event may

pass integrity verification but still involve unusual timing, abnormal retry behavior, an unexpected country pattern, or a sensitive field update outside the normal operating profile. This is where the machine learning layer becomes important. It gives the framework the ability to recognize behavioral risk that may not be visible through integrity checks alone.

At the same time, the results show that machine learning alone has limitations in audit-sensitive environments. A model may classify an event as unusual, but it cannot independently prove that the historical event trail was not modified. This matters in HR systems because audit and compliance teams often need evidence, not only probability-based alerts. TEAL-HCM improves this situation by attaching anomaly scores to verified lineage records. As a result, suspicious behavior is supported by a stronger evidence trail [25].

The reduction in false positives is also meaningful from an enterprise operations perspective. Security tools are often difficult to sustain when they generate too many alerts. HR operations and audit teams may not have the capacity to investigate every technical exception, especially during high-volume periods such as annual compensation cycles, mass organizational changes, or payroll processing windows. By combining lineage status, anomaly scoring, and business sensitivity, TEAL-HCM helps prioritize the events that deserve immediate attention.

The framework has practical value for sensitive HR data domains. Payment information, compensation fields, identity-related employee records, and compliance data require stronger controls because errors or unauthorized changes can create financial, access, legal, or reputational consequences. TEAL-HCM supports this need by applying risk weighting to the anomaly score. This means that a suspicious payment update or identity event can be treated with greater urgency than a low-impact reporting delay.

From an organizational governance perspective, TEAL-HCM supports a shift from reactive investigation to evidence-ready monitoring. Instead of waiting for an audit issue or downstream failure, organizations can continuously evaluate whether integration events remain complete, verifiable, and behaviorally normal. This improves readiness for internal audit, regulatory review, data quality investigation, and security incident response. It also strengthens confidence in downstream processes that depend on accurate SAP SuccessFactors data.

The findings are especially relevant for large multinational organizations. Country-specific HR rules, regional data requirements, and local payroll dependencies make integration governance more complex. A small inconsistency in one country may not appear significant at the global level, but it can still create local compliance or employee-impact risk. By including country code, HR field group, business impact, and lineage criticality in the evaluation model, TEAL-HCM provides a more context-aware approach to integration security.

Overall, the discussion reinforces the central argument of this study: secure SAP SuccessFactors integrations require both proof and intelligence. Proof is needed to verify that the data lineage has not been altered. Intelligence is needed to understand whether the event behavior is normal, suspicious, or high risk. TEAL-HCM brings these two capabilities together in a practical framework that can improve audit readiness, reduce investigation time, strengthen data governance, and increase organizational trust in cloud-based HR integration ecosystems.

Conclusion, Limitations, and Future Scope

This study presented TEAL-HCM as a tamper-evident AI lineage framework for securing cloud-based SAP SuccessFactors integrations. The framework was designed to address a practical problem in enterprise HR technology: sensitive employee data often moves across SAP SuccessFactors, middleware, SFTP channels, payroll interfaces, identity platforms, reporting systems, and compliance repositories without a unified method to prove that the data remained complete, unchanged, correctly sequenced, and behaviorally normal. TEAL-HCM responds to this challenge by combining canonical event modeling, hash-chain verification, Merkle checkpointing, lineage reconstruction, machine learning anomaly detection, and risk-based prioritization.

The findings suggest that SAP SuccessFactors integration security should move beyond basic transaction monitoring and fragmented audit logs. Traditional logs can confirm that an event occurred, but they do not always provide strong evidence that the full data path remained trustworthy. By creating a verifiable lineage structure, TEAL-HCM strengthens the ability of organizations to detect payload alteration, missing records, replay activity, unauthorized field updates, delayed acknowledgments, schema drift, and abnormal API behavior. This makes the framework valuable not only for technical monitoring, but also for audit readiness, compliance assurance, and HR data governance.

The comparative evaluation shows that TEAL-HCM performs better than traditional audit logging, rule-based monitoring, hash-chain-only verification, and machine-learning-only detection across key metrics such as tamper detection accuracy, anomaly detection F1-score, false positive rate, lineage completeness, audit reconstruction time, and compliance evidence readiness. The results support the central argument of this study: cryptographic proof and behavioral intelligence are stronger together than either method alone. Hash-chain verification provides evidence that the event history was not altered, while machine learning helps identify unusual activity that may still appear technically valid.

At the enterprise level, the framework has important practical implications. Large organizations using SAP SuccessFactors often manage high-risk HR domains such as payment information, compensation data, personal information, job data, identity records, and compliance-related fields. Errors or unauthorized changes in these

areas can affect payroll accuracy, system access, regulatory reporting, and employee trust. TEAL-HCM helps prioritize these risks by linking anomaly scores with data sensitivity, business impact, and lineage criticality. This allows HR operations, audit, compliance, and security teams to focus on the events that require the most attention.

Despite these contributions, the study has limitations. The evaluation was conducted through a controlled enterprise-scale simulation rather than direct production data from a live SAP SuccessFactors customer environment. This approach was necessary to protect sensitive HR information and to create controlled tampering scenarios. However, real production environments may contain greater variation in configuration, integration design, country-specific rules, custom middleware logic, and downstream system behavior. Future validation using anonymized enterprise data would strengthen the practical evidence for the framework.

Another limitation is that TEAL-HCM depends on the quality of event capture across connected systems. If SAP SuccessFactors, middleware, file transfer systems, or downstream platforms produce incomplete logs, inconsistent timestamps, weak request identifiers, or missing acknowledgments, lineage reconstruction may become less accurate. The framework can organize and verify available evidence, but it cannot fully replace strong logging discipline at the source. Organizations adopting this model would need clear logging standards, consistent request tracking, and reliable integration metadata.

Future research can extend this work in several directions. First, TEAL-HCM should be tested using anonymized production logs across different industries, regions, and SAP SuccessFactors integration landscapes. Second, future studies can explore adaptive anomaly models that adjust to business changes such as annual compensation cycles, payroll calendar shifts, acquisitions, country rollouts, and SuccessFactors release updates. Third, privacy-preserving validation methods can be developed to verify payload meaning without exposing sensitive employee data. Fourth, the framework can be expanded beyond SAP SuccessFactors to broader cloud HCM ecosystems involving payroll providers, learning platforms, identity systems, data lakes, and external HR service providers.

In conclusion, TEAL-HCM contributes a practical and measurable approach to securing cloud-based SAP SuccessFactors integrations. It reframes HR integration security as a problem of verifiable trust rather than simple log availability. By combining tamper-evident data lineage with machine learning anomaly detection, the framework improves integrity verification, risk visibility, audit reconstruction, and compliance evidence readiness. The study provides a foundation for future research on secure HR data movement and offers organizations a scalable path toward stronger governance of sensitive cloud HCM integrations.

References

[1] Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. ACM SIGMOD Record, 34(3), 31–36. https://doi.org/10.1145/1084805.1084812

[2] Cheney, J., Chiticariu, L., & Tan, W. C. (2009). Provenance in databases: Why, how, and where. Foundations and Trends in Databases, 1(4), 379–474. https://doi.org/10.1561/1900000006

[3] Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., & Van den Bussche, J. (2011). The Open Provenance Model core specification. Future Generation Computer Systems, 27(6), 743–756. https://doi.org/10.1016/j.future.2010.07.005

[4] Hasan, R., Sion, R., & Winslett, M. (2009). Preventing history forgery with secure provenance. ACM Transactions on Storage, 5(4), 1–43. https://doi.org/10.1145/1629080.1629082

[5] Pasquier, T., Han, X., Goldstein, M., Moyer, T., Eyers, D., Seltzer, M., & Bacon, J. (2017). Practical whole-system provenance capture. Proceedings of the ACM Symposium on Cloud Computing, 405–418. https://doi.org/10.1145/3127479.3129249

[6] Shekhtman, L., & Waisbard, E. (2021). EngraveChain: A blockchain-based tamper-proof distributed log system. Future Internet, 13(6), 143. https://doi.org/10.3390/fi13060143

[7] Guardiola-Múzquiz, G., & Soriano-Salvador, E. (2021). SealFS: Storage-based tamper-evident logging. Computers & Security, 108, 102325. https://doi.org/10.1016/j.cose.2021.102325

[8] Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications, 34(1), 1–11. https://doi.org/10.1016/j.jnca.2010.07.006

[9] Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems, 28(3), 583–592. https://doi.org/10.1016/j.future.2010.12.006

[10] Hashizume, K., Rosado, D. G., Fernández-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications, 4, 5. https://doi.org/10.1186/1869-0238-4-5

[11] Fernandes, D. A. B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security issues in cloud environments: A survey. International Journal of Information Security, 13(2), 113–170. https://doi.org/10.1007/s10207-013-0208-7

[12] Ryan, M. D. (2013). Cloud computing security: The scientific challenge, and a survey of solutions. Journal of Systems and Software, 86(9), 2263–2268. https://doi.org/10.1016/j.jss.2012.12.025

[13] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58. https://doi.org/10.1145/1541880.1541882

[14] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. Proceedings of the IEEE International Conference on Data Mining, 413–422. https://doi.org/10.1109/ICDM.2008.17

[15] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471. https://doi.org/10.1162/089976601750264965

[16] Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[17] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

[18] Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. Proceedings of the Workshop on Machine Learning for Sensory Data Analysis, 4–11. https://doi.org/10.1145/2689746.2689747

[19] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 1285–1298. https://doi.org/10.1145/3133956.3134015

[20] He, S., Zhu, J., He, P., & Lyu, M. R. (2016). Experience report: System log analysis for anomaly detection. Proceedings of the IEEE International Symposium on Software Reliability Engineering, 207–218. https://doi.org/10.1109/ISSRE.2016.21

[21] Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., & Lyu, M. R. (2019). Tools and benchmarks for automated log parsing. Proceedings of the IEEE/ACM International Conference on Software Engineering: Software Engineering in Practice, 121–130. https://doi.org/10.1109/ICSE-SEIP.2019.00021

[22] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[23] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[24] Marler, J. H., & Boudreau, J. W. (2017). An evidence-based review of HR analytics. The International Journal of Human Resource Management, 28(1), 3–26. https://doi.org/10.1080/09585192.2016.1244699

[25] Stone, D. L., Deadrick, D. L., Lukaszewski, K. M., & Johnson, R. D. (2015). The influence of technology on the future of human resource management. Human Resource Management Review, 25(2), 216–231. https://doi.org/10.1016/j.hrmr.2015.01.002