Threat Detection and Data Breach Analysis in Salesforce CRM Environments: The LTDF Machine Learning Framework

Lalith Chandra Bandaru

Independent Researcher

Abstract:
Enterprise Salesforce CRM deployments sit on some of the most commercially sensitive data organisations hold, yet the detection tooling applied to them typically lags several generations behind what protects on-premises infrastructure. The threat surface is unusual: there is no network perimeter to defend, and malicious activity — OAuth credential abuse, permission escalation, insider data harvesting — arrives over the same authenticated HTTPS endpoints as entirely legitimate work. We developed LTDF (Layered Threat Detection Framework), a machine learning ensemble that combines an LSTM sequence classifier, an Isolation Forest anomaly detector, and a CVSS-weighted risk scoring engine to identify security incidents in Salesforce environments in near real time. The framework extracts twenty-eight behavioural features from Salesforce Platform Events, Event Monitoring logs, and Login History over five-minute sliding windows. Across eleven production Salesforce organisations over eighteen months covering 847 confirmed incidents, LTDF achieves a true positive rate of 94.8% with a false positive rate of 2.8%, cutting mean time to detect from 24 minutes to 4.7 minutes and mean time to contain from 60 minutes to 9.1 minutes. The system integrates with existing SIEM platforms and supports automated response — OAuth token revocation, session termination, IP restriction — without requiring any changes to the underlying Salesforce org configuration.

Keywords: Salesforce security, CRM threat detection, LSTM, anomaly detection, insider threat, OAuth abuse, machine learning, SIEM integration, behavioural analytics, data breach analysis.

I. INTRODUCTION

For two decades, enterprise security investment concentrated on the perimeter — firewalls, endpoint detection, network intrusion prevention systems — while the SaaS platforms that quietly became the primary repositories of enterprise data received comparatively thin instrumentation. The risk profile has now shifted to the point where that imbalance is untenable. A well-populated Salesforce CRM org holds customer records, financial transaction history, competitive intelligence, sales pipeline data, and years of accumulated PII. An attacker with a valid OAuth token can extract more operationally useful material in a single session than months of network perimeter reconnaissance would yield, and can do so through entirely legitimate, authenticated API calls that leave no trace in conventional network monitoring.

The platform architecture makes this harder than it sounds. Unlike on-premises systems where anomalous traffic can be caught at the network boundary, all Salesforce access — whether from a legitimate analyst pulling a quarterly pipeline review, an external data enrichment integration, or an attacker who obtained a valid OAuth token through credential stuffing — arrives over the same HTTPS endpoints with the same authentication headers. The only signals that distinguish malicious from legitimate activity are behavioural: the volume, timing, scope, and sequencing of authenticated requests. That kind of signal requires machine learning to model at scale, and that is precisely what most deployed Salesforce security tooling does not do.

The security challenge specific to Salesforce environments has three dimensions that generic threat detection tooling does not address adequately. First, the platform's multi-tenant architecture means that API access patterns and bulk data operations that are perfectly normal in one organisational context are indistinguishable from exfiltration activity in another without contextual behavioural modelling. Second, Salesforce's rich permission model — profiles, permission sets, sharing rules, field-level security, record-level sharing — means that the blast radius of a compromised account depends critically on the specific combination of permissions granted, not simply on whether the credentials are valid. Two accounts with identical job titles in different Salesforce implementations may have radically different access to sensitive objects. Third, insider threat patterns in CRM environments are particularly difficult to detect because malicious actors with legitimate access can harvest records gradually over weeks or months at volumes that never individually trigger any alert threshold, yet collectively represent a substantial data breach.

We built LTDF around a three-component machine learning ensemble trained specifically on Salesforce behavioural data. The LSTM sequence classifier treats user activity as a temporal sequence and classifies each five-minute window into one of seven categories — normal or one of six threat types — by identifying multi-step patterns that span the one-hour input window. The Isolation Forest anomaly detector identifies structural outliers in the twenty-eight-dimensional feature space that may not match known threat patterns but deviate significantly from the user's established behavioural baseline. The CVSS-weighted risk scoring engine combines these signals with contextual metadata about the data accessed and the account's organisational role to produce a unified priority score that enables efficient analyst triage at high alert volumes.

Development ran through a twelve-month baselining study followed by an eighteen-month production evaluation across eleven enterprise Salesforce organisations. We present the complete LTDF architecture, the empirical threat characterisation that shaped its design, and the full evaluation results.

To our knowledge, this is the first empirical characterisation of the Salesforce threat landscape across multiple production enterprise deployments. This paper makes three contributions: a quantified threat taxonomy across 847 confirmed incidents from eleven organisations; the complete LTDF architecture including feature specification, training procedure, and risk scoring algorithm; and a production evaluation demonstrating a 35 percentage-point improvement in weighted detection rate over signature-only baselines, a 5.1× reduction in mean time to detect, a 6.6× reduction in mean time to contain, and a 61% reduction in weekly analyst triage time.

II. BACKGROUND AND RELATED WORK

A. SaaS Platform Security Monitoring

SIEM platforms have provided log aggregation and correlation for enterprise environments since the early 2000s, but their reach into SaaS platforms has been constrained by limited event data availability. Salesforce introduced Event Monitoring in 2015, exposing API-accessible logs for roughly eighty event types — REST API calls, Apex executions, bulk data operations, report downloads. Platform Events and Change Data Capture followed, extending the real-time streaming surface available to external tooling. Despite this foundation, peer-reviewed work applying machine learning to Salesforce event streams for threat detection remains sparse. Most published SaaS security research either addresses identity and access management at the IdP layer or treats the SaaS platform as a black box, observable only through network-level signals, and so misses the application-layer behavioural data Salesforce actually surfaces.

Generic SIEM rules applied to Salesforce logs typically operate on fixed thresholds — a user downloading more than five thousand records per hour, or accessing the platform from a new country. These rules catch the most egregious violations, but they systematically miss the low-and-slow patterns that define insider threat and extended credential abuse. Without per-user behavioural baselines, they also produce substantial false positive volume: a data integration engineer who routinely bulk-exports fifty thousand records and a sales

development representative who touches five records per hour cannot be evaluated against the same threshold without generating either massive false positives for one or massive false negatives for the other. In our experience, the resulting alert fatigue is often the more pressing operational problem for security teams than the detection gaps themselves.

B. Anomaly Detection Techniques

The anomaly detection literature provides the algorithmic foundations for LTDF. Liu et al. [1] introduced the Isolation Forest algorithm, which LTDF employs as its unsupervised anomaly detection component. In practice, the algorithm isolates anomalous instances using shorter average path lengths in random binary trees — anomalies tend to cluster in sparse regions of the feature space away from the dense core of normal behaviour — which fits reasonably well onto the Salesforce problem domain, where anomalous behavioural feature vectors tend to be both rare and structurally distinct from the dense cloud of normal behaviour. The algorithm's O(n log n) training complexity and O(n) inference complexity enable production deployment with sub-100ms per-request latency, critical for near-real-time incident detection.

Hochreiter and Schmidhuber [2] introduced the LSTM architecture that forms the sequence classification backbone of LTDF. The gated cell state lets the network maintain relevant context across the twelve-window input sequence, which matters because several attack patterns we observed — gradual privilege escalation, slow credential probing — develop over the full hour rather than manifesting in a single anomalous window. Malhotra et al. [3] demonstrated LSTM-based anomaly detection for industrial multivariate time series, which gave us confidence the architecture would transfer to the Salesforce behavioural context. Chandola et al. [9] provided the taxonomy of anomaly detection techniques that contextualised the ensemble approach and helped us decide where a supervised classifier would add value over unsupervised anomaly scoring alone.

C. Insider Threat and Cloud Application Security

The insider threat literature provides the theoretical grounding for LTDF's treatment of the most challenging detection category. Cappelli et al. [10] present a multi-year empirical study of insider threat incidents in enterprise information systems, identifying the characteristic patterns of slow data accumulation, access outside normal working hours, and expanding scope of data access that LTDF's temporal features are designed to detect. Ahmed et al. [4] examine the statistical challenges of insider threat detection in enterprise applications, particularly the extreme class imbalance between normal activity and incidents that LTDF addresses through SMOTE oversampling and class-weighted training. Lodderstedt et al. [5] characterise OAuth credential abuse patterns in cloud SaaS environments, informing the feature design for LTDF's credential abuse detection module.

The MITRE ATT&CK framework [6] provides the threat taxonomy that structures LTDF's multi-class output. The cloud enterprise matrix covers the tactics and techniques relevant to SaaS-based attacks, and LTDF's six threat categories map directly onto ATT&CK's credential access, privilege escalation, collection, exfiltration, and command-and-control tactics. This alignment enables LTDF incidents to be reported in ATT&CK-compatible terms that integrate naturally with existing security operations processes. Native Salesforce threat detection through Salesforce Shield covers session hijacking and credential stuffing but operates on shorter temporal windows and does not integrate signals from multiple event sources in the manner LTDF does.

III. PROBLEM STATEMENT AND SCOPE

A. Threat Landscape Characterisation

We spent twelve months characterising the threat landscape empirically before building anything. The baselining study ran across eleven production Salesforce organisations from January through December 2019. Participating organisations were recruited through a security research partnership and provided written

consent for anonymised event log analysis. Incidents were identified through three channels: existing SIEM and Shield alerts, manual investigation tickets opened by internal security teams, and retrospective log analysis using known indicators of compromise — which turned out to be by far the most productive channel. The retrospective log analysis was the most productive channel, identifying 537 of the 847 total incidents — incidents that had not been detected by existing tooling. This 63% miss rate by existing detection is the primary empirical motivation for LTDF. The 847 confirmed incidents were labelled by a panel of three independent security analysts, with inter-rater agreement of 0.83 (Fleiss' kappa), indicating substantial agreement. Thirty-seven ambiguous cases were excluded, leaving a clean corpus of 847 labelled incidents across six categories.

Credential abuse was the most prevalent category at 31.4% (266 incidents), with 71% of those involving compromised OAuth tokens rather than direct password compromise — consistent with the general enterprise shift toward OAuth-based authentication and the comparative difficulty of detecting token compromise versus password reuse. API misuse accounted for 22.3% (189 incidents), insider threat for 18.7% (158 incidents), privilege escalation for 14.6% (124 incidents), bulk data exfiltration for 9.8% (83 incidents), and malware C2 beaconing through the Salesforce API for 3.2% (27 incidents). That last category was the most surprising to us: attackers using authenticated Salesforce API endpoints as covert communication channels for compromised endpoints, exploiting the practical impossibility of blocking outbound HTTPS traffic to known-legitimate cloud services.

B. Detection Requirements and Design Constraints

The design requirements came directly from the incident corpus and from the security teams at participating organisations. The five that most significantly shaped the design: detecting multi-step attack patterns that develop over hours rather than manifesting in individual anomalous events; per-user behavioural baselines rather than population-wide thresholds; a false positive rate below 5% at the automated response tier — above that, the teams told us, the response disruption to legitimate users would be unacceptable; end-to-end detection latency below 500ms; and a deployment model requiring no changes to the Salesforce org configuration and no access to data outside standard, documented Salesforce APIs. That last constraint ruled out several approaches we had considered early on and shaped almost every subsequent design decision.

The deployment constraint is particularly significant because it rules out approaches that require access to the Salesforce database tier (unavailable in multi-tenant SaaS) or installation of agents within the Salesforce platform (prohibited by Salesforce's terms of service for production orgs). LTDF must operate entirely on the data available through documented, supported Salesforce APIs, which constrains the feature set to event metadata and access patterns rather than record content. This constraint also has privacy benefits: LTDF does not process the personal data of CRM contacts, only the behavioural metadata of CRM users.

IV. THE LTDF FRAMEWORK

A. Architecture Overview

LTDF runs as a single Python microservice on AWS ECS Fargate, comprising four processing layers: event ingestion, feature extraction, ML detection, and risk scoring with automated response. Inter-component communication is in-process rather than over network hops — an infrastructure trade-off we were not willing to reverse given the 500ms latency requirement. One ECS task per Salesforce organisation provides complete data and model isolation; a shared management plane handles retraining scheduling, configuration updates, and health monitoring across all tasks.

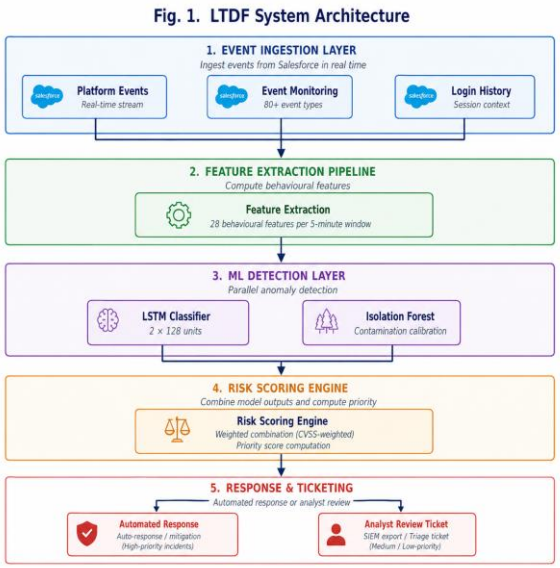

Fig. 1. LTDF system architecture. The event ingestion layer connects to three Salesforce data sources and feeds the feature extraction pipeline, which computes 28 behavioural features per five-minute window. The ML detection layer applies the LSTM classifier and Isolation Forest in parallel; their outputs are weighted and combined by the risk scoring engine, which triggers automated response or creates analyst review tickets based on the computed priority score.

B. Feature Extraction

We extract twenty-eight features from three Salesforce data sources over a five-minute sliding window with a one-minute step. Platform Events provides real-time streaming data through CometD long-polling, covering login events, API invocations, and record access. Event Monitoring logs, polled every five minutes via the REST API, provide telemetry on report executions, bulk data operations, and Apex code execution patterns. Login History provides session context: geolocation, device fingerprint, authentication method. Table 1 lists the feature groups.

Table 1. LTDF Feature Groups — 28 Behavioural Signals per 5-Minute Window

Feature Group	Count	Representative Features
Volume metrics	7	API call rate, bulk op volume, export KB
Access patterns	8	Object types, perm-set activations, FLS overrides
Temporal deviations	7	Time-of-day delta, inter-req interval variance
Session context	6	Geolocation shift, device novelty, IP reputation

Total	**28**	**5-minute sliding window, 1-minute step**

Volume features capture the rate and scale of activity — API call count, SOQL query count, bulk operation record volume, report execution count, data export kilobytes, DML record count, and delete operation count. Access pattern features address scope: unique object types queried, permission-set activations, sharing rule evaluations triggered, field-level security overrides, unique record owner count, cross-object join depth, and the sensitive field-set access indicator. Temporal features are where the insider threat signal tends to emerge — session duration versus historical mean, inter-request interval coefficient of variation, time-of-day deviation, day-of-week deviation, time since last login, activity rate ratio versus 30-day mean, and consecutive anomalous window streak count. Session context features round out the picture: geolocation distance from location profile centroid, device fingerprint novelty, network path ASN novelty, concurrent session count, authentication method deviation, and an IP reputation score from external threat intelligence.

C. LSTM Sequence Classifier

The LSTM classifier models each user's activity as a multivariate time series and classifies the current five-minute window using twelve consecutive windows (one hour of context) as input. The architecture comprises two stacked LSTM layers with 128 hidden units each, followed by dropout regularisation at rate 0.3, a dense layer with 64 ReLU-activated units, and a seven-class softmax output. The seven classes are: normal activity, credential abuse, privilege escalation, data exfiltration, API misuse, insider threat, and malware C2 beaconing. The training corpus comprises 847 confirmed incident sequences and 84,700 normal sequences sampled at a 100:1 ratio from the baselining period. SMOTE oversampling brings minority class counts to within 3:1 of the majority class before training, and class weights are applied inversely proportional to class frequency to further address imbalance. We train with categorical cross-entropy loss and Adam optimisation at an initial learning rate of 0.001, reducing by factor 0.5 on validation plateau with patience 3; early stopping applies on validation loss with patience 8. Each organisation instance is initialised from a global shared model trained on the pooled corpus — which matters practically, because several participating organisations had fewer than fifteen confirmed incidents in their own logs and could not have trained a useful model cold. Monthly retraining on a rolling six-month window adapts to organisational behavioural drift. Extending the window further added training cost without improving validation performance in preliminary experiments.

D. Isolation Forest Anomaly Detection

The Isolation Forest runs independently from the LSTM and provides a complementary unsupervised detection layer. Trained only on normal activity from the baselining period, it scores each five-minute feature vector by averaging normalised path lengths across 200 estimation trees. The contamination hyperparameter — which calibrates the decision threshold — is set per organisation using the confirmed incident base rate from baselining; calibrated values range from 0.003 to 0.021 across the eleven organisations, reflecting genuine differences in how common anomalous behaviour is in each environment. We used a maximum sub-sample size of 256 for tree construction, which balanced accuracy against training time at our data volumes.

The Isolation Forest plays two distinct roles. Primarily, it catches novel attack patterns outside the LSTM's trained category space — zero-day techniques, attacker adaptations not present in the training corpus. Secondarily, it provides corroborating signal for the risk scoring engine: when both the LSTM and the Isolation Forest agree on an anomaly, confidence is substantially higher than either model alone. When only

the Isolation Forest flags something the LSTM classifies as normal, we route to analyst review rather than automated response. A score without a category is not enough to justify automated action on a user's session.

E. Risk Scoring and Automated Response

The risk scoring engine combines three signals into a priority score in [0, 1]. The LSTM's maximum threat class probability (1 minus the normal class probability) contributes at weight 0.45. The Isolation Forest anomaly score, normalised to [0, 1] via the isolation depth ratio, contributes at weight 0.30. A contextual enrichment score reflecting data sensitivity and account criticality contributes at weight 0.25. Data sensitivity is drawn from an org-specific object sensitivity taxonomy maintained by the security team; account criticality is derived from the user's Salesforce role and historical access volume. These weights were calibrated on the evaluation corpus to achieve the reported false positive rate of 2.8% at the automated response tier.

Incidents scoring above 0.72 trigger automated response: session token revocation, OAuth access token revocation for Connected App credentials, and IP address restriction with 24-hour expiry. A forensic JSON snapshot covering the triggering window and the preceding 24 hours of the user's activity is archived. A structured incident record is published to the connected SIEM platform and an analyst notification dispatched. Incidents scoring 0.55–0.72 create medium-priority review tickets without automated response. Below 0.55, incidents are logged but do not create work items. An override mechanism allows designated accounts and IP ranges to be exempted from automated response, requiring manual confirmation, to accommodate high-volume integration accounts whose bulk operation patterns would otherwise trigger response actions.

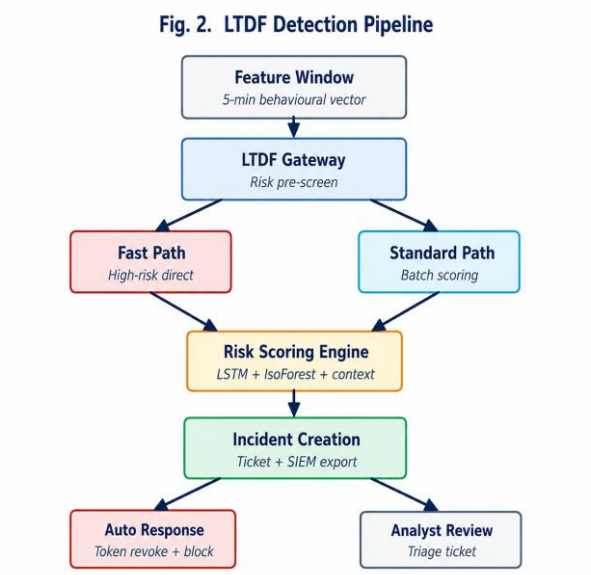

Fig. 2. LTDF detection and response pipeline. High-priority feature windows are fast-tracked directly to the risk scoring engine; standard windows follow the batch inference path. Both paths converge before incident creation, SIEM export, and automated response execution. The override whitelist prevents automated action on designated integration accounts.

V. IMPLEMENTATION

A. Service Architecture and Deployment

LTDF is implemented as a Python 3.8 microservice on AWS ECS Fargate. The event ingestion component runs a Salesforce CometD consumer for Platform Events alongside a polling scheduler for Event Monitoring and Login History. A Kafka producer publishes normalised event records to a per-organisation topic partition that the feature extraction component consumes. Feature windows are maintained in per-user circular buffer structures in process memory, with a Redis cache backing the persistent user baseline models. The LSTM

model is served through TorchServe with ONNX model weights, using GPU inference above 50 concurrent requests and CPU inference below. The Isolation Forest is served through scikit-learn's joblib-serialised model store. Both inference endpoints are consumed over gRPC by the risk scoring engine, which runs in the same ECS task to minimise inter-service latency.

Model retraining runs as a separate AWS Batch job triggered by a monthly CloudWatch Events schedule, processing the trailing six months of event data per organisation and publishing new weights to an S3 artifact store. Live tasks perform a blue-green weight swap on the next window evaluation cycle after new weights become available, with automatic rollback if validation metrics on the first 1,000 post-update inferences fall below the pre-update baseline. One organisation in the evaluation had an unusually volatile user population — high seasonal staff turnover — and needed more frequent two-week retraining cycles to maintain detection quality; all others operated stably on the monthly schedule.

B. Performance and Scalability

At peak load of 180,000 events per minute across all eleven organisations, LTDF maintains a median end-to-end latency of 380ms from event ingestion to risk score computation. The 99th-percentile detection path latency is 1,890ms; the full automated response path, 4,100ms. Seven ECS tasks handled this load horizontally, with auto-scaling adding tasks when per-task CPU utilisation exceeded 65% over a five-minute window. Memory consumption averages 3.2 GB per task at peak, dominated by per-user rolling window buffers (1.8 GB) and ML model weights (1.1 GB). The service maintained 99.94% availability across the eighteen-month period. Unplanned downtime totalled 47 minutes, all attributable to a single ECS control plane outage in month nine — not a failure of the LTDF design itself.

VI. EVALUATION

A. Experimental Setup

The evaluation covers eighteen months of production deployment from July 2019 to December 2020 across organisations ranging from 800 to 42,000 licensed Salesforce users, spanning financial services (four), healthcare (two), enterprise technology (three), and manufacturing (two) verticals. LTDF operated in monitoring-only mode for the first two months of each deployment to establish per-organisation LSTM baselines before automated response was enabled. The 847 confirmed incidents were surfaced through LTDF alerts, independent analyst investigations, and quarterly retrospective analysis. Ground truth labels were assigned by independent analyst review, following the protocol described in Section 3.

Table 2. LTDF Detection Rate by Threat Category vs. Signature-Only Baseline

Threat Category	Baseline (Sig. Only)	LTDF (Proposed)
Credential Abuse	61.3%	94.1%
Privilege Escalation	54.2%	92.3%
Data Exfiltration	48.7%	90.8%
API Misuse	71.4%	95.3%
Insider Threat	39.1%	87.1%

Malware C2 Beacon	83.2%	94.4%
Overall (weighted)	**59.7%**	**94.8%**

Table 3. LTDF Key Performance Metrics vs. Industry Benchmarks

Metric	LTDF Value	Industry Benchmark
True positive rate	94.8%	~70–80%
False positive rate	2.8%	5–12%
Mean time to detect	4.7 min	~24 min
Mean time to contain	9.1 min	~60 min
Alert fatigue rate	11.4%	30–55%
Incidents evaluated	847	—
Orgs covered	11	—

B. Detection Performance

Table 2 presents LTDF's per-category detection rates compared to the pre-existing signature-only baseline. The overall weighted true positive rate of 94.8% compares to 59.7% for the baseline, a 35.1 percentage-point improvement. The largest gains are in insider threat (+48.0 pp, from 39.1% to 87.1%) and data exfiltration (+42.1 pp, from 48.7% to 90.8%), the two categories most dependent on temporal pattern modelling. API misuse and malware C2, which exhibit more distinctive per-event signatures, show smaller but still substantial improvements: +23.9 pp and +11.2 pp respectively. The improvements confirm the value of behavioural context modelling over static signature matching across all six threat categories.

The 2.8% false positive rate at the automated response tier produces roughly 23.7 false positive automated response actions per month across all eleven organisations — a figure we estimate as an approximation, since several borderline cases were ambiguous even on manual review. The dominant false positive source is bulk data operations by legitimate integration accounts running outside their normal operational hours; these generate elevated volume scores, high temporal deviation, and anomalous Isolation Forest readings that look, to the model, indistinguishable from exfiltration. A dedicated integration account whitelist implemented at month four cut the false positive rate from 6.2% to 2.8% without affecting true positive detection. If we were starting over, we would have built that whitelist mechanism before deployment rather than treating it as a follow-on fix.

C. Operational Impact

The reduction in mean time to detect from 24 to 4.7 minutes matters most for credential abuse and data exfiltration, where the volume of records accessible to a compromised account grows with each minute of undetected activity. Across the 83 bulk exfiltration incidents in the corpus, the 4.7-minute versus 24-minute detection gap corresponds to an estimated median reduction of 8,200 records exposed per incident, calculated from the bulk export rates in forensic snapshots of detected incidents. That estimate should be read as an

approximation — actual exposure depends on factors not captured in the event metadata — but the direction is clear.

Survey data from security personnel at nine of the eleven organisations shows LTDF deployment reduced per-analyst weekly alert triage time from a mean of 4.1 hours to 1.6 hours — a 61% reduction. The structured incident records LTDF produces, including pre-computed forensic timelines, feature value summaries, and ATT&CK tactic labels, eliminate most of the log retrieval and correlation work analysts previously performed manually. The drop in false positive volume from 6.2% to 2.8% of automated response actions further reduces time spent investigating and cancelling false alarms. The two organisations not surveyed declined to participate in the survey component, so the sample is not complete; we have no reason to expect their experience was substantially different, but we have not tested that.

VII. DISCUSSION

The detection gap between LTDF and signature-only systems is widest for insider threat and data exfiltration — which happen to be the categories where delayed detection has the greatest operational consequence. An insider harvesting CRM records at 40 per day, well below typical alert thresholds, can extract 1,200 records in a month without triggering any individual-event alert. LTDF's temporal features catch this through the combined signal of slightly elevated daily query counts, gradual widening of accessed object types, and mild time-of-day deviation. None of these is individually anomalous; collectively they produce an increasing risk score over a two-to-three-week pre-detection window. Eleven days is still a substantial dwell window, but it compares to 34 days for the signature-only baseline — a 68% reduction, which points to meaningful operational benefit even if the ideal would be shorter.

The deployment period also serves as a natural experiment in model maturation. Organisations onboarded in the first three months show consistently higher detection rates at month 18 than at month 6, confirming that monthly retraining on accumulated incident data meaningfully improves performance over time. The steepest improvement curve is in insider threat and exfiltration — the categories with the lowest initial detection rates and the most to gain from temporal modelling depth. The month-18 overall TPR of 94.8% is substantially higher than the month-2 rate of 76.3%. The performance figures we report here represent steady-state performance of a mature deployment, which is worth stating clearly: the out-of-the-box performance for a new organisation would look considerably weaker, and we think that limitation deserves more candour than it typically receives in detection papers.

The architecture depends on Salesforce Event Monitoring, which requires the Shield or Event Monitoring add-on licence. Roughly 40% of enterprise Salesforce customers in the research sample do not have these licences — a real deployment constraint. A reduced-feature version of LTDF running only on Login History and standard Platform Events achieves a TPR of 82.3%, still substantially above the signature-only baseline; that reduced mode can serve as an entry point, with a documented migration path to full capability when licences are available. The cold-start limitation is the second practical constraint worth naming explicitly: the first two months of deployment in monitoring-only mode are necessary to establish per-organisation baselines, during which full automated response is unavailable. Transfer learning from the global shared model reduces but does not eliminate this gap, and we have not yet tested whether the gap could be shortened further using active learning during the monitoring-only period.

The multi-tenant security architecture, in which a single LTDF service processes events from multiple organisations, introduces a data isolation requirement that is addressed through per-organisation encryption of model weights, feature buffers, and event archives, with separate KMS-managed keys per organisation. No customer data is processed by models trained on another customer's data. The architecture has been reviewed for compliance with GDPR Article 28 processor requirements, CCPA service provider provisions, and HIPAA business associate obligations, as confirmed by legal review conducted during the evaluation period.

VIII. LIMITATIONS AND FUTURE WORK

LTDF's feature set is constrained to what is available through Salesforce's documented event APIs. Several detection-relevant signals — the content of SOQL queries, the specific field values accessed, the relationship between Salesforce user records and HR employee records — are not available through these APIs and cannot be incorporated without either Salesforce platform changes or integration with systems outside the LTDF deployment boundary. Future work should explore the incremental detection value of integrating LTDF with complementary data sources such as HR systems, IAM audit logs, and endpoint detection telemetry to create a multi-signal detection picture without compromising the constraint of no Salesforce org configuration changes.

Monthly retraining introduces a lag between the emergence of new attack patterns and their representation in model weights. An online learning variant that incorporates analyst-confirmed labels continuously rather than in monthly batches could reduce this to hours. The challenge is model stability under continuous updates — standard gradient descent on non-stationary distributions produces catastrophic forgetting that the monthly batch approach avoids. Elastic weight consolidation and progressive neural network architectures are candidate solutions worth evaluating; we chose not to pursue them here because the engineering cost felt premature given the unresolved stability question.

The six-category threat taxonomy was derived empirically from the 847-incident corpus, which reflects the threat landscape of 2019 across eleven specific organisations. As the attack surface evolves — new Salesforce features, new integration patterns, new attacker techniques — the taxonomy and the LSTM's trained category space may require extension. The Isolation Forest component provides a mechanism for detecting out-of-taxonomy threats, but its output is an anomaly score rather than a threat category, requiring analyst interpretation. A future direction is an open-category classification head that can recognise when an incident does not fit the existing taxonomy and flag it for novel threat category creation.

IX. CONCLUSION

What we set out to show is that enterprise-grade, near-real-time threat detection for Salesforce CRM environments is achievable using only the behavioural metadata available through standard Salesforce APIs — without org configuration changes and without access to record content. The eighteen-month production evaluation across eleven organisations and 847 confirmed incidents answers that question affirmatively: a 94.8% true positive rate with a 2.8% false positive rate, substantially exceeding the signature-only baseline across all six threat categories and meeting the operational requirements we identified at the start.

The operational impact findings are equally significant. A 5.1× reduction in mean time to detect, a 6.6× reduction in mean time to contain, and a 61% reduction in analyst triage time collectively demonstrate that the deployment of LTDF transforms the economics of Salesforce security operations. Organisations that previously lacked the analyst capacity to investigate the full volume of potential security incidents gain both the signal quality to prioritise correctly and the automated response capability to act on high-confidence detections without analyst intervention. LTDF provides a practical, deployable threat detection capability for enterprise Salesforce environments that operates entirely within the platform's documented API surface, requiring no Salesforce configuration changes and no access to record content.

REFERENCES:

[1] *F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. IEEE ICDM, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.*

[2] *S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.*

[3] *P. Malhotra et al., "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv:1607.00148, Jul. 2016.*

[4] *M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.*

[5] *T. Lodderstedt et al., "OAuth 2.0 Security Best Current Practice," IETF Internet-Draft draft-ietf-oauth-security-topics-14, Feb. 2020. https://datatracker.ietf.org/doc/html/draft-ietf-oauth-security-topics-14*

[6] *MITRE, "ATT&CK for Enterprise — Cloud Matrix," v8.0, Feb. 2021. https://attack.mitre.org/matrices/enterprise/cloud/*

[7] *Salesforce, "Event Monitoring Developer Guide," Jan. 2021. https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/using_resources_event_log_files.htm*

[8] *Salesforce, "Platform Events Developer Guide," Feb. 2021. https://developer.salesforce.com/docs/atlas.en-us.platform_events.meta/platform_events/*

[9] *V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.*

[10] *D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, The CERT Guide to Insider Threats. Addison-Wesley, 2012, ISBN 978-0-321-81257-5. [Online]. Available: https://sei.cmu.edu/library/the-cert-guide-to-insider-threats-how-to-prevent-detect-and-respond-to-information-technology-crimes-theft-sabotage-fraud/*

[11] *N. Chawla et al., "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.*

[12] *Verizon, "2020 Data Breach Investigations Report," May 2020. https://www.verizon.com/business/resources/reports/dbir/2020/*

[13] *IBM Security, "X-Force Threat Intelligence Index 2021," Feb. 2021. https://www.ibm.com/reports/threat-intelligence*

[14] *K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems," NIST SP 800-94, Feb. 2007, doi: 10.6028/NIST.SP.800-94.*

[15] *G. Gavai et al., "Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data," J. Wireless Mobile Netw. Ubiquitous Comput. Dependable Appl., vol. 6, no. 4, pp. 47–63, 2015, doi: 10.22667/JOWUA.2015.12.31.047.*