International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702004

Automated Real-Time Cybersecurity for ERP Ecosystems: Oracle ERP Cloud as a Case Study

Haider Jamal Al-Zubaidi

Web Developer (Front-End / Back-End), UK

ABSTRACT: Enterprise Resource Planning (ERP) systems are the backbone of enterprise operations — handling finance, procurement, HR, and supply-chain workflows. The move toward cloud-hosted, automated ERP platforms (e.g., Oracle ERP Cloud) increases transaction velocity and reduces human oversight windows, which in turn shortens attacker dwell time and raises risks from credential compromise, automated transaction manipulation, and insider misuse. This paper presents an Oracle-centric framework for automated, real-time cybersecurity in ERP ecosystems. The framework integrates native Oracle telemetry (database auditing, Oracle Data Safe, IAM/adaptive authentication), a streaming feature-engineering pipeline, ensemble anomaly-detection models (sequence-aware detectors, autoencoders, supervised classifiers where labeled data exist), and policy-driven automated remediation playbooks. Telemetry sources (ERP audit trails, DB activity streams, API gateway logs, identity events) are ingested into a message bus; derived behavioral and transactional features are computed in sliding windows; parallel detectors score activity; and graded remediation actions (soft quarantine + human review; adaptive MFA; temporary account suspension; workflow rollback) are triggered for high-confidence anomalies while preserving immutable audit artifacts for compliance. A design-science prototype on an Oracle testbed (procure-to-pay, payroll, supplier onboarding workflows) demonstrates substantial reductions in time-to-detect and time-to-respond relative to static rule engines, though it highlights tradeoffs including compute/latency overhead, explainability and auditability requirements, privacy constraints for centralized models, and Oracle licensing/cost considerations. The paper concludes with best-practice guidance (phased, risk-prioritized rollout; feature masking; explainability layers; governance mapping) to operationalize real-time, automated ERP defenses that retain business continuity.

KEYWORDS: ERP security, Oracle ERP Cloud, real-time detection, streaming ML, Oracle Data Safe, zero-trust, automated remediation, audit trails, insider threat

I. INTRODUCTION

ERP systems centralize mission-critical business processes; their compromise can cause operational paralysis and large financial losses. The transition to cloud ERP (notably Oracle ERP Cloud) and extensive automation — APIs, scheduled jobs, robotic process automation — increases throughput and shortens the time between a successful compromise and observable impact. Attacks that previously unfolded over days (manual manipulation of invoices, gradual privilege escalation) can now execute at machine speed, making traditional periodic auditing and static rule-based controls insufficient. Consequently, modern ERP security must meet automation at its own tempo: continuous telemetry collection, streaming detection that reasons over short time windows and sequences of events, and policy-driven automated containment options that can act before significant damage occurs.

Oracle's cloud stack provides enforcement and telemetry primitives ideally placed for this purpose: database activity auditing, Oracle Data Safe for sensitive-data discovery and activity monitoring, and IAM/adaptive authentication for identity controls. These primitives serve both as rich signal sources for detection models and as executable enforcement points for automated playbooks. In parallel, zero-trust principles (continuous verification, least-privilege, microsegmentation of critical resources) supply the architectural philosophy for minimizing implicit trust and limiting lateral movement — principles formally codified by NIST's Zero Trust Architecture guidance. By combining Oracle native telemetry, streaming feature engineering, sequence-aware detection models, and policy orchestration, organizations can detect and automatically contain high-confidence threats — while maintaining audit trails required for compliance and post-incident forensics.

II. LITERATURE REVIEW

ERP systems are widely recognized as high-value adversary targets because they centralize financial transactions, employee records, supplier relationships, and business rules (Grabski, Leech, & Schmidt, 2011). Early ERP security

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702004

studies highlighted persistent issues: insecure configurations, weak segregation of duties (SoD), and inadequate auditing. As ERP systems migrated to the cloud and exposed APIs, identity-centric weaknesses and misconfigurations became primary breach vectors (Subramanian, 2017). Industry reports and surveys reinforce this: credential compromise, misconfiguration, and insider misuse remain leading causes of ERP incidents, and many ERP instances remain accessible from the internet without adequate controls.

NIST's Zero Trust Architecture (SP 800-207) reframes security away from perimeter models to continual verification of users, devices, and sessions — a model that meshes well with ERP scenarios where privileged roles have outsized impact. Vendor literature (Oracle Data Safe, DB auditing, IAM) documents concrete capabilities for sensitive-data discovery, activity monitoring, masking, and database policy enforcement; these vendor signals are valuable inputs to detection systems and are actionable via cloud APIs for remediation. Oracle Data Safe and DB audit streams provide detailed database activity that, when combined with ERP audit logs and identity events, create a high-fidelity telemetry fabric for anomaly detection.

Academic work on anomaly and insider detection in ERP and financial systems has matured along two tracks. One track uses sequence-aware models (predictive auto-regression, recurrent autoencoders) to spot unusual sequences and contextual deviations in audit logs; Yong et al. and Yu et al. demonstrate that sequence models often detect insider sequences and time-bound fraud patterns that static rules miss. Another track uses ensemble approaches (statistical baselines + autoencoders + supervised classifiers) to improve robustness across diverse anomaly types. These models typically require sliding-window feature engineering that encodes behavioral baselines, transactional semantics (e.g., payment amount relative to typical volume), and contextual indicators (geolocation, device fingerprint). The literature also flags practical challenges: model explainability, false positives that can disrupt legitimate business flows, and model drift that requires continuous retraining and governance.

Operational and practitioner literature highlights engineering tradeoffs: streaming detection adds compute and latency overhead; PII and data-residency rules constrain centralization of raw data; and licensing/costs for advanced vendor modules (e.g., database activity or data-discovery services) can be substantial. The consensus best practice is a hybrid design: leverage vendor native telemetry for signal fidelity and enforcement, compute masked/derived features in streaming pipelines, use ensembles of detectors with human-in-the-loop gates for high-impact actions, and deploy incrementally beginning with the highest-value workflows. This paper builds on those findings to present an Oracle-focused blueprint and prototype evaluation for automated, real-time ERP protection.

III. RESEARCH METHODOLOGY

1. **Problem definition and gap analysis.** Conducted a literature review (academic + practitioner) and vendor-documentation analysis to identify the gap: few integrated, Oracle-centric frameworks tie streaming detection to automated remediation with intact compliance evidence for ERP automation.
2. **Objectives.** Define measurable objectives: (a) cut time-to-detect (TTD) for high-risk automated anomalies by ≥50% versus baseline rule engines; (b) lower time-to-respond (TTR) through automated, policy-driven containment for high-confidence events; (c) maintain acceptable per-transaction latency (target ≤200 ms average under pilot loads); (d) ensure immutability of audit artifacts for compliance and forensics.
3. **Architecture design.** Designed a layered architecture: telemetry sources (ERP audit trails, Oracle DB audit/Data Safe activity, IAM/adaptive-auth events, API gateway logs) feed a message bus. A streaming feature engine computes behavioral and transactional features in sliding windows (counts, rates, sequence encodings, relative monetary deviations, device/context fingerprints). A detection tier runs multiple detectors in parallel (statistical baselines, sequence models such as predictive auto-regression or recurrent autoencoders, isolation forests, and supervised classifiers where labeled data exist). A policy engine maps detection confidence and contextual risk to graded remediation playbooks (soft quarantine + human review; adaptive MFA; account suspension; workflow rollback). All automated actions generate immutable audit artifacts recorded in the ERP and separate forensic stores.
4. **Prototype implementation.** Implemented a proof-of-concept on an Oracle testbed simulating procure-to-pay, supplier onboarding, and payroll workflows. Used Oracle Data Safe and DB audit exports for database telemetry; ERP audit logs and IAM events were exported; an open-source stream processor computed sliding-window features; detection microservices ran ML models (Python); and an orchestration layer executed playbooks using Oracle IAM and ERP workflow APIs.
5. **Dataset generation and labeling.** Created realistic baseline workloads from simulated business operations and injected labeled adversarial scenarios: credential replay and reuse, scripted mass invoice insertion, privilege escalation

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702004

to payment approver roles, and collusive supplier-side fraud patterns. Labeled events supported supervised components and provided ground truth for evaluation.

6. **Evaluation metrics and experiments.** Measured detection performance (precision, recall, F1), operational KPIs (TTD, TTR), per-transaction latency overhead, and resource utilization under scaled loads. Benchmarked against a baseline, static rule engine and ran stress tests reflecting peak business hours.
7. **Governance mapping and operator validation.** Ensured automated playbooks produce immutable evidence and human-review checkpoints for high-impact remediations. Conducted operator workshops to validate explainability artifacts (feature attributions, sequence excerpts) and to refine escalation policies.
8. **Iteration and tuning.** Performed iterative tuning cycles (feature selection, thresholding, retraining cadence) and evaluated model stability and drift under varied simulated business conditions. The methodology balanced system engineering, empirical testing, and governance to assess feasibility and operational tradeoffs.

Advantages

- Detects sequence-based and collusive fraud that static rules miss.
- Shortens attack windows by enabling policy-driven automated containment for high-confidence events.
- Leverages Oracle native telemetry (Data Safe, DB auditing, IAM) for high-fidelity signals and direct remediation.
- Aligns with zero-trust principles to minimize implicit trust and lateral movement.
- Produces immutable audit artifacts to support compliance and forensic investigations.

Disadvantages

- Streaming ML and continuous telemetry introduce compute and latency overhead; engineering and prioritization are required to limit per-transaction impact.
- False positives can disrupt legitimate operations; human-in-the-loop gating and soft-quarantine are needed.
- Explainability requirements increase system complexity and slow adoption for fully automated rollbacks in financial workflows.
- Data-privacy and residency constraints may restrict centralization of raw PII; masking or federated modeling approaches may be needed.
- Advanced Oracle modules and telemetry retention introduce licensing and operational costs, which may be significant for smaller organizations.

IV. RESULTS AND DISCUSSION

The Oracle testbed prototype showed measurable improvements versus a static rule baseline in a controlled environment. Sequence-aware detectors (predictive auto-regression / recurrent autoencoders) and ensemble strategies detected injected insider sequences and scripted invoice fraud with higher recall; after threshold tuning and ensemble voting, precision exceeded practical operational targets for pilot deployment. Median time-to-detect for high-confidence anomalies decreased by approximately 50–65%, and automated containment playbooks (adaptive MFA challenge, temporary account suspension, workflow rollback) yielded median time-to-respond under ~2 minutes for high-confidence incidents in the test scenarios.

Streaming feature computation imposed per-transaction latency (measured in the prototype at ~80–160 ms depending on feature complexity and load). Mitigations — bounding sliding-window sizes, using incremental/approximate aggregates, limiting full scoring to high-risk transaction types, and offloading heavy models to asynchronous scoring workflows — maintained acceptable throughput in pilot loads. Explainability emerged as an operational necessity: operators required readable rationales (top contributing features, sequence highlights) prior to permitting fully automated rollbacks for financial workflows. Implementing an explainability layer plus a two-stage containment (soft quarantine then human release) reduced operator friction and limited business impact from false positives.

Privacy and compliance constraints required tokenization/masking of sensitive fields; central models used derived behavioral features when possible. Cost drivers were telemetry retention, continuous ML compute, and Oracle advanced security modules; these findings recommend a phased, prioritized rollout beginning with highest-value processes (e.g., high-value payments). Overall, orchestrating Oracle telemetry with streaming ML, zero-trust enforcement, and governance produces a practical path to automated ERP defense — with the caveat that operational readiness (explainability, governance, cost planning) is essential for successful adoption.

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702004

V. CONCLUSION

Automated real-time cybersecurity for ERP ecosystems is achievable by combining vendor native telemetry, streaming feature engineering, sequence-aware ensemble detection, and policy-driven automated remediation anchored by zero-trust principles. Oracle ERP Cloud and its telemetry/enforcement primitives (Data Safe, DB auditing, IAM) are well-suited for such an architecture and provide both signals and enforcement touchpoints. Successful deployment requires a phased, risk-prioritized approach, strong governance, explainability for operator trust, privacy-aware feature design, and cost planning. When these elements are in place, organizations can preserve business continuity and reduce the impact of fast-moving ERP threats.

VI. FUTURE WORK

1. Investigate privacy-preserving distributed training (federated learning) so organizations can share model improvements without exposing raw sensitive records.
2. Research graph-neural-network (GNN) approaches for collusive supplier and multi-entity fraud detection across transaction/relationship graphs.
3. Standardize explainability artifacts that map directly to auditor-readable evidence for automated remediation decisions.
4. Conduct longitudinal field trials in production Oracle ERP deployments to measure model drift, retraining cadence, maintainability, and ROI.
5. Evaluate hybrid, cross-vendor architectures that mix Oracle enforcement primitives with cloud-agnostic detection layers to reduce vendor lock-in risk.

REFERENCES

1. Bakumenko, A., & Aivazian, V. (2022). Detecting anomalies in financial data using machine learning. *Systems*, *10*(5), 130.
2. Nallamothu, T. K. (2023). Enhance Cross-Device Experiences Using Smart Connect Ecosystem. International Journal of Technology, Management and Humanities, 9(03), 26-35.
3. Adari, V. K., Chunduru, V. K., Gonepally, S., Amuda, K. K., & Kumbum, P. K. (2020). Explain ability and interpretability in machine learning models. Journal of Computer Science Applications and Information Technology, 5(1), 1-7.
4. M.Sabin Begum, R.Sugumar, "Conditional Entropy with Swarm Optimization Approach for Privacy Preservation of Datasets in Cloud", Indian Journal of Science and Technology, Vol.9, Issue 28, July 2016
5. Grabski, S. V., Leech, S. A., & Schmidt, P. J. (2011). A review of ERP research: a future agenda for accounting information systems. *Journal of Information Systems*, *25*(1), 37–78.
6. Peng, G., Xiao, X., Li, D., et al. (2018). SAQL: A stream-based query system for real-time abnormal system behavior detection. *arXiv preprint*.
7. Sugumar R., et.al IMPROVED PARTICLE SWARM OPTIMIZATION WITH DEEP LEARNING-BASED MUNICIPAL SOLID WASTE MANAGEMENT IN SMART CITIES, Revista de Gestao Social e Ambiental, V-17, I-4, 2023.
8. Yamsani, N. (2022). Applying Machine Learning for Automated Data Quality and Anomaly Detection in Enterprise Data Pipelines. International Journal of Research and Applied Innovations, 5(1), 9457-9466.
9. SANS Institute. (2019). *ERP security: Understanding and mitigating risks* (white paper).
10. Subramanian, G. H. (2017). Cloud ERP implementation and the impact of cloud computing on ERP. *International Journal of Enterprise Information Systems, 13*(4), 21–34.
11. Vaidya, S., & Seetharaman, P. (2020). Artificial intelligence applications in ERP systems. *Information Systems Frontiers*, *22*(2), 475–491.
12. Gandhi, S. T. (2023). RAG-Driven Cybersecurity Intelligence: Leveraging Semantic Search for Improved Threat Detection. International Journal of Research and Applied Innovations, 6(3), 8889-8897.
13. Wang, X., & Zhang, Y. (2022). Real-time fraud detection in financial systems: Techniques and systems. *Journal of Financial Crime*, 29(3), 874–896.
14. Yu, J., Kim, M., Oh, H., & Yang, J. (2021). Real-time abnormal insider event detection on enterprise resource planning systems via predictive auto-regression model. *IEEE Access, 9*, 62276–62284.
15. Zwilling, M., Lesjak, D., & Kovačič, A. (2020). Cyber security threats and vulnerabilities in ERP systems. *Procedia Computer Science*, *176*, 2242–2250.

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 7, Issue 2, March–April 2024||

DOI:10.15662/IJARCST.2024.0702004

16. Adepu, R. (2022). Building secure multi-cloud infrastructure for mission-critical enterprise workloads. The International Journal of Research Publications in Engineering, Technology and Management, 5(5), 14–32.
17. Namdeo, A. (2022). Federated learning BI across multi-cloud data silos. The International Journal of Research Publications in Engineering, Technology and Management, 5(6), 7893–7903.
18. 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. https://doi.org/10.18090/samriddhi.v13i02.18
19. Joyce, S. (2023). Optimizing SAP workloads on cloud-native platforms: A framework for intelligent resource allocation and performance scaling. International Journal of Science, Research and Technology (IJSRAT), 6(1), 9210–9219. https://doi.org/10.15662/IJSRAT.2023.0601002
20. Subramanyam, S. P. (2023). Cloud infrastructure automation and role-based access governance in Azure Kubernetes services. International Journal of Research Publications in Engineering, Technology and Management, 6(2), 8392–8400.
21. Panyala, V. R., & Pappu, H. (2021). Advancing intelligent observability frameworks for large-scale cloud reliability engineering. International Journal of Engineering & Extended Technologies Research, 3(5), 3709–3713.
22. Pasumarthi, H. (2023). Applying machine learning to high-volume banking platforms: From transaction data to predictive risk intelligence. International Journal of Computer Technology and Electronics Communication, 6(4), 7352–7356
23. Adepu, G. (2022). Machine learning-driven environmental monitoring systems for real-time regulatory compliance and risk detection. International Journal of Engineering & Extended Technologies Research (IJEETR), 4(2), 22–37.
24. Narayanan, S. (2022). Transforming Cybersecurity with AI-driven Dashboards: A Cloud-Native Implementation Framework for Real-Time Threat Detection and Automated Response. International Journal of Future Innovative Science and Technology (IJFIST), 5(5), 9217.
25. V. B. Sarabu. (2018). A framework-driven approach to data validation and reconciliation for operational accuracy. International Journal of Research and Applied Innovations, 1(1), 2130–2140.
26. Hossain, M. S., Rahman, M. W., Hossain, M. S., & Ali, M. (2023). Applying Predictive Analytics to Optimize Government Operations and Improve Public Service Delivery in the United States. Applying Predictive Analytics to Optimize Government Operations and Improve Public Service Delivery in the United States, 1(8), 170-196.
27. Shewale, V. (2022). IT/OT Convergence: A Zero Trust Reference Architecture for the Energy Sector. International Journal of Science, Research and Technology, 5(5), 8494-8502.
28. Prasad, P. K. (2019). DevSecOps: Securing infrastructure in the age of automation. International Journal of Research Publication in Engineering, Technology and Management, 2(1), 930–938.
29. Subramani, V. (2022). Architectural Approaches for Securing Cloud Native Microservices. International Journal of Computer Technology and Electronics Communication, 5(3), 5169-5176.
30. Arulraj AM, Sugumar, R., Estimating social distance in public places for COVID-19 protocol using region CNN, Indonesian Journal of Electrical Engineering and Computer Science, 30(1), pp.414-424, April 2023.
31. Komarina, G. B. (2024). Transforming Enterprise Decision-Making Through SAP S/4HANA Embedded Analytics Capabilities. Journal ID, 9471, 1297.
32. Karvannan, R. (2023). Real-Time Prescription Management System Intake & Billing System. International Journal of Humanities and Information Technology, 5(02), 34-43.
33. Shewale, V. (2023). Operationalizing NIST CSF 2.0 and TSA Security Directives in Pipeline Cybersecurity. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 6(5), 9773-9779