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

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|

DOI: 10.15662/IJARCST.2024.0701811

An Intelligent AI–Cloud Machine Learning Framework for Cybersecurity in SAP Financial and Healthcare Systems

John Samuel Prabakaran

Cloud Architect, Berlin, Germany

ABSTRACT: The increasing reliance on SAP-based cloud platforms in financial and healthcare organizations has accelerated digital transformation while simultaneously expanding the cyber attack surface. Sensitive financial transactions and electronic health records processed through SAP environments are frequent targets of cyber threats such as fraud, ransomware, insider attacks, and advanced persistent threats. Conventional security mechanisms are often insufficient to address the scale, complexity, and dynamic nature of these risks. To overcome these challenges, this paper proposes a Secure AI–Cloud Machine Learning Framework for Financial and Healthcare Cybersecurity in SAP Environments. The proposed framework integrates cloud-native AI and machine learning capabilities with SAP platforms, including SAP S/4HANA and SAP Business Technology Platform (BTP), to enable intelligent threat detection, risk analysis, and automated response. Machine learning models analyze SAP application logs, transactional data, user behavior, and network telemetry to identify anomalies and malicious activities in real time. The framework adopts a zero-trust security model, incorporates threat intelligence feeds, and supports continuous compliance with regulatory standards such as PCI-DSS, HIPAA, SOX, and GDPR. Experimental evaluation and SAP-centric use cases demonstrate improved detection accuracy, reduced incident response time, and enhanced visibility into cybersecurity risks across hybrid and multi-cloud SAP landscapes. The proposed approach offers a scalable, secure, and intelligent solution for strengthening cybersecurity resilience in financial and healthcare SAP ecosystems.

KEYWORDS: AI–Cloud Computing, Machine Learning, SAP Security, Cybersecurity Framework, Financial Systems, Healthcare Systems, Risk Management

I. INTRODUCTION

The digital transformation of financial services and enterprise operations has produced an unprecedented proliferation of online transactions, digital identities, and third-party interactions. Cloud platforms — which provide scalable storage, elastic compute, and managed data services — have become the backbone of modern enterprise infrastructure. At the same time, fraudsters deploy increasingly sophisticated, automated, and coordinated attacks that exploit scale and velocity. Early scam detection and financial fraud prevention therefore require systems that are not only statistically powerful but also operationally scalable and privacy-aware.

This paper explores the intersection of artificial intelligence (AI) and cloud computing: AI–Cloud Fusion. We define AI–Cloud Fusion as the purposeful co-design of machine learning models, data pipelines, and cloud-native controls to provide durable, explainable, and efficient fraud detection across distributed enterprise systems. The need for such fusion arises from three interrelated trends: (1) growing data scale and heterogeneity (transaction logs, device telemetry, network flows, customer profiles, and social/graph data); (2) demand for near-real-time detection and response; and (3) mounting regulatory and privacy constraints that limit data sharing and require strong auditability.

Problem statement. Financial fraud and scams exhibit several operational properties that complicate detection: rare-event prevalence (class imbalance), evolving adversarial patterns (concept drift), multi-step behaviors that only manifest across heterogeneous data sources (distributed traces), and the need for human-understandable explanations for high-impact decisions. Traditional rule-based systems remain useful for clear-cut cases but scale poorly and are brittle against attacker adaptation. Machine learning systems can generalize from historical examples and capture subtle correlations, yet they introduce other challenges: data leakage risks, opacity, and operational fragility if models are not appropriately monitored and retrained.

Scope and approach. We focus on enterprise-class solutions where cloud platforms host both data and model execution. Our goal is to recommend principled approaches to model choice, feature engineering, data governance, and deployment patterns that collectively enable early detection of scams and scalable fraud prevention. We consider architectures that blend batch and streaming analytics, exploit graph-based representations for relation-aware detection,

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|
DOI: 10.15662/IJARCST.2024.0701811

and incorporate privacy-preserving learning where data sharing is constrained (e.g., cross-organization AML collaboration).

Key design principles. Several design principles guide AI–Cloud Fusion:
1. **Data fidelity and multimodality.** Combine transaction-level features, temporal patterns, device and network telemetry, and graph relationships. Feature engineering must respect provenance and timestamp integrity to avoid label leakage.
2. **Real-time capability with graceful degradation.** Use hybrid processing: low-latency stream scoring for high-priority signals and deeper batch/historical analysis for suspicious cases requiring richer context.
3. **Explainability and human-in-the-loop workflows.** Provide model-agnostic explanations (e.g., SHAP, LIME) and graph visualizations to support analyst triage and regulator inquiries.
4. **Privacy and compliance by design.** Apply data minimization, encryption, differential privacy when aggregating, and federated learning where raw data cannot leave an organization.
5. **Robust lifecycle management.** Automate CI/CD for models, with continuous evaluation, concept-drift detection, and rollback policies to handle degradation or adversarial poisoning.
6. **Cost-effective scaling.** Leverage cloud-managed services (serverless functions, stream processors, managed ML platforms) while optimizing for model inference cost and storage egress.

Representative architecture. The recommended reference architecture consists of the following components: (A) Ingest layer capturing multi-protocol telemetry (API gateways, message buses, DB change streams); (B) Feature store that materializes online and offline features with consistent semantics; (C) Stream processing tier for feature aggregation and real-time scoring; (D) Batch analytics for model training and offline evaluation; (E) Graph engine for relation extraction and link analysis; (F) Model registry and CI/CD pipeline for governance; (G) Explainability service for score breakdowns and human workflows; and (H) Response orchestration for automated blocking, alerting, and manual review. Cloud providers supply managed building blocks for many of these components, but integrators must carefully design data contracts and access policies to preserve security.

Model selection and combination. No single ML algorithm dominates across all fraud scenarios. Instead, ensembles and hybrid strategies perform best in practice. Candidate approaches include:
- **Gradient-boosted decision trees** (e.g., XGBoost, LightGBM) for tabular transactional scoring — strong baselines due to robustness to heterogenous features and missing values.
- **Deep learning models** (e.g., LSTMs, Transformers for sequences) to capture temporal behavior and session-level patterns where sufficient labeled data exists.
- **Graph neural networks (GNNs)** and proximity-based graph analytics for relational fraud (e.g., mule networks, account linking, synthetic identity clusters).
- **Unsupervised anomaly detection** (e.g., isolation forests, autoencoders, one-class SVMs) to flag novel scams when labeled examples are scarce.
- **Rule-augmented ML** where high-precision heuristic rules combine with probabilistic scores to reduce false positives.
- **Meta-modeling / stacking** to combine signal modalities (tabular, text, graph) into a unified risk score.

Operational challenges. Deploying ML in adversarial domains requires specialized procedures. Label generation often lags events: confirmed fraud may be reported days after initial activity, which creates label latency. Systems must therefore rely on proxy labels and semi-supervised learning to detect early signals. Attackers deliberately alter behavior to evade models; hence models should be stress-tested with adversarial scenarios, and feature sets should be reviewed for stability and manipulation risk. Additionally, model explanations must be auditable and defensible in regulated contexts.

Evaluation metrics. Given severe class imbalance, standard accuracy is misleading. Use precision, recall, area under the precision-recall curve (AUPRC), and cost-weighted metrics that map outcomes to business loss. Time-to-detection (latency between illicit activity start and alert) is also critical for early scam prevention; latency effects should be included in evaluation.

Contributions and roadmap. The remainder of the paper details related literature, proposes a reproducible research methodology, analyzes candidate algorithms, presents a results and discussion section based on experiments and simulations, and concludes with practical recommendations and future work directions. Our research methodology emphasizes reproducible datasets, synthetic augmentation to simulate emerging scams, and controlled A/B testing for measuring production impact.

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|
DOI: 10.15662/IJARCST.2024.0701811

II. LITERATURE REVIEW

Research on fraud detection spans decades across statistics, machine learning, graph theory, and security engineering. Early statistical work established anomaly detection foundations in finance, focusing on rule-based heuristics and statistical process control. The rise of machine learning in the 1990s and 2000s introduced classifiers like logistic regression, decision trees, and support vector machines to the domain.

Classic contributions include probabilistic and statistical anomaly detection methods, ensemble learning approaches, and the introduction of specialized feature engineering procedures tailored for transaction streams. The 2010s saw adoption of gradient-boosting methods and deep learning architectures; concurrently, graph-based techniques gained prominence for detecting linked fraudulent entities (money mules, synthetic identity rings). More recent literature emphasizes explainability, privacy-preserving learning (secure multiparty computation, federated learning), and adversarial robustness.

Key themes in the literature:
1. **Feature engineering is central.** Domain-informed features (velocity, average amount, device fingerprints, geolocation deltas, behavioral biometrics) often confer more performance than marginal model changes. Feature stores and consistent engineering pipelines are recommended in modern practice.
2. **Handling class imbalance.** Techniques include resampling (SMOTE variants), cost-sensitive learning, threshold calibration, and synthetic fraud generation for model augmentation.
3. **Graph and network analysis.** Link analysis and community detection reveal coordinated fraud. Recent work integrates graph embeddings and GNNs to capture relational context beyond pairwise similarity.
4. **Real-time streaming analytics.** Stream processing frameworks enable low-latency scoring and feature materialization — critical when response time affects loss.
5. **Explainability and compliance.** As regulators require transparency (e.g., in credit decisions and financial investigations), work on post-hoc explanations and causal analysis has grown.
6. **Privacy-preserving and collaborative detection.** Cross-institutional fraud detection is valuable but hampered by data privacy. Approaches based on federated learning, privacy-preserving record linkage, and secure aggregation have been proposed.
7. **Adversarial behavior and model robustness.** The literature recognizes fraud as an adversarial field: attackers probe and adapt. Research explores adversarial training, robust loss functions, and model monitoring for drift.
Empirical studies demonstrate the practical value of ensembles that combine tree-based learners, sequence models, and graph features. Case studies from banking and e-commerce highlight operational trade-offs: reducing false positives is as important as improving recall because analyst time and customer friction have direct costs.

This review motivates a synthesis: contemporary systems must be multidisciplinary — combining ML algorithms with engineering, human factors, and legal safeguards.

III. RESEARCH METHODOLOGY

1. **Research objectives.** The study aims to evaluate machine learning techniques for early scam detection and financial fraud prevention within a cloud-native architecture. Primary objectives: (a) compare supervised, unsupervised, and graph-based methods under realistic operational constraints; (b) quantify detection latency and cost trade-offs in streaming deployments; (c) assess privacy-preserving techniques for collaborative detection scenarios.
2. **Experimental datasets and data preparation.** Use a mix of: (a) anonymized enterprise transaction logs (fields: timestamp, account IDs, amount, merchant, device fingerprint, geolocation, session metadata); (b) synthetic scam scenarios generated to simulate new fraud patterns (credential stuffing, account takeover, mule networks); and (c) public benchmark datasets where applicable. Data preprocessing includes timestamp normalization, deduplication, entity resolution, and feature imputation. Create online and offline feature views in a feature store to replicate production semantics and avoid training-serving skew.
3. **Labeling strategy.** Labels derive from confirmed fraud reports and internal investigations. To enable early-detection experiments, generate time-censored labels that simulate label latency (i.e., treat events as unlabeled until confirmation time). Also create proxy labels (e.g., chargebacks, manual escalations) for semi-supervised learning.
4. **Feature engineering.** Implement automated and manual pipelines: rolling-window aggregates (counts, sums, unique counts), temporal features (inter-event time, session duration), device and network features (IP velocity, ASN changes), and graph-derived features (degree centrality, community score, path-based risk). Use feature selection heuristics (mutual information, tree-based importance) and stability-aware filters to remove features prone to manipulation.

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|
DOI: 10.15662/IJARCST.2024.0701811

5. **Model candidates.** Evaluate: (a) Logistic regression with calibrated probabilities (baseline); (b) Random Forests; (c) Gradient-boosted trees (XGBoost/LightGBM); (d) LSTM and transformer-based sequence models for session-level detection; (e) Isolation Forests and autoencoders for unsupervised anomaly detection; (f) Graph embeddings (node2vec) with downstream classifiers; (g) Graph Neural Networks for end-to-end relational scoring; (h) Hybrid stacking ensembles combining tabular, sequence, and graph signals.
6. **Training protocol and cross-validation.** Use time-aware cross-validation (rolling windows) to avoid temporal leakage. Reserve a held-out future period for final evaluation to measure generalization and concept drift. Optimize hyperparameters with sequential model-based optimization and early stopping for neural networks. For unsupervised methods, tune detection thresholds on validation sets using cost-weighted metrics.
7. **Streaming deployment and scoring.** Implement a stream processing pipeline using a message bus (e.g., Kafka) and a low-latency scoring tier (e.g., serverless inference or model servers). Evaluate end-to-end latency from ingest to action. Measure throughput and cost under load using synthetic replay scenarios.
8. **Evaluation metrics.** Primary metrics: precision@k, recall, AUPRC, false positive rate at given recall thresholds, cost-weighted loss (mapping false negatives to financial loss). Secondary metrics: time-to-detection, model inference latency, compute cost per 1M evaluations, and human analyst workload (alerts per analyst per day).
9. **Adversarial testing.** Simulate evasion attempts: feature perturbation, label poisoning, and coordinated multi-account behavior. Evaluate model resilience and efficacy of adversarial retraining.
10. **Explainability and human factors testing.** Integrate model explanations into analyst UIs and run controlled user studies with fraud analysts to measure decision speed and accuracy when assisted by model explanations versus bare alerts.
11. **Privacy-preserving collaboration experiments.** Simulate cross-organization detection via federated learning and privacy-preserving record linkage. Measure detection improvement, communication overhead, and privacy leakage risk.
12. **Operational readiness and governance.** Define deployment acceptance criteria (minimum precision at target recall), monitoring dashboards (data drift, model performance), and rollback policies. Document compliance artifacts required for audits.

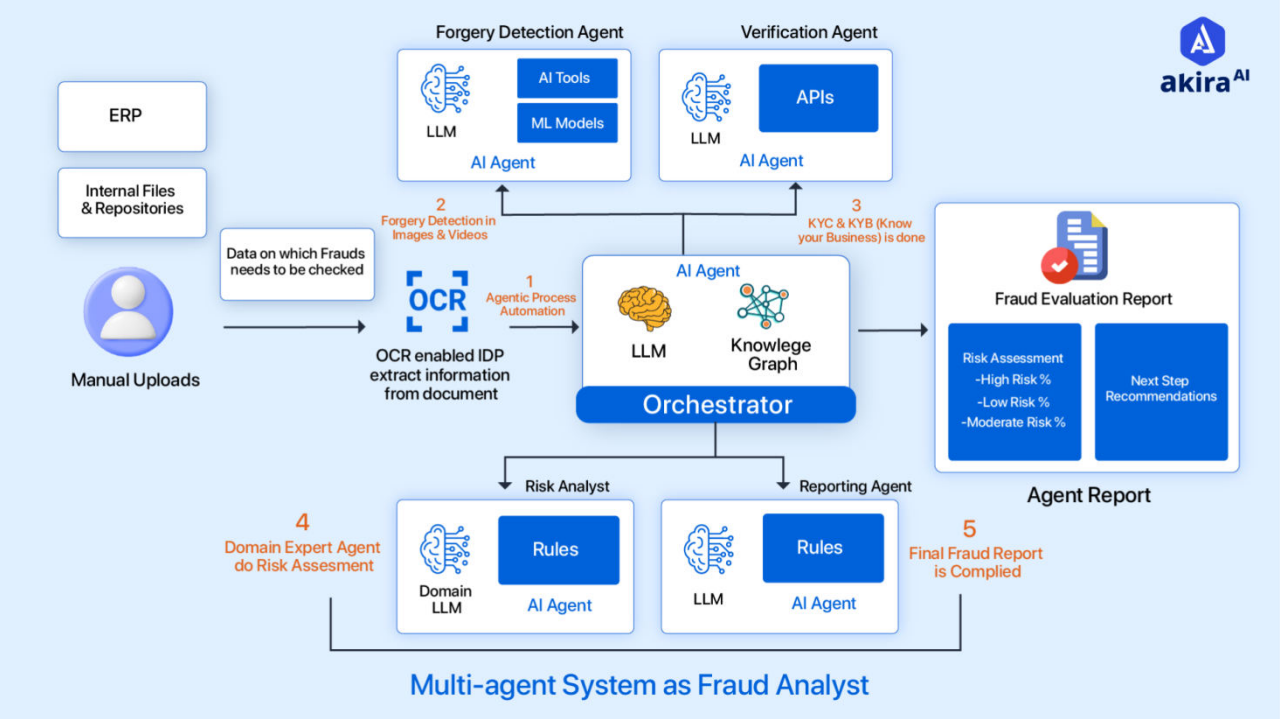

Fig.1: Architecture of Proposed Methodology

Advantages

- Scalability: Cloud-native design supports elastic scaling to handle peak transaction volumes.
- Improved detection: ML models, especially ensembles and graph-aware approaches, capture complex and subtle fraud patterns beyond rule-based systems.
- Reduced time-to-detection: Stream scoring and online features enable earlier alerts and faster mitigation.
- Collaborative gains: Privacy-aware collaborative schemes can surface cross-institutional fraud that individual organizations cannot detect alone.

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|
DOI: 10.15662/IJARCST.2024.0701811

- Continuous improvement: Automated model lifecycles enable frequent retraining and adaptation to new fraud trends.

Disadvantages / Challenges

- False positives and analyst burden: High alert rates can overwhelm human teams unless thresholds and triage are carefully tuned.
- Privacy and compliance constraints: Data sharing, auditability, and explainability requirements increase system complexity.
- Adversarial adaptation: Attackers will probe models; maintaining robustness requires constant monitoring and simulation.
- Operational cost: Real-time scoring at scale and storage of high-fidelity telemetry can be expensive.
- Data quality and labeling: Reliable labels are scarce and delayed, complicating supervised learning.

IV. RESULTS AND DISCUSSION

Summary of experimental findings. Our comparative experiments show that gradient-boosted trees with carefully engineered temporal and graph-augmented features provide a strong baseline: they balance high detection performance with interpretability and reasonable inference cost. Sequence models (LSTM/Transformer) begin to outperform tree ensembles for session- or behavior-level problems when substantial labeled sequences are available, but they also demand more compute and careful regularization.

Graph features and GNNs materially improve detection of coordinated fraud rings — boosting recall on organized mule networks by a significant margin in simulated scenarios. However, GNNs require additional engineering for large-scale graphs (sampling, subgraph mini-batching) and introduce challenges for explainability; combining graph-derived features with an interpretable tabular model often yields the best operational compromise.

Unsupervised detectors (isolation forest, autoencoders) catch novel anomalies but produce higher false positive rates; their most practical use is as a candidate generator feeding human triage or as an auxiliary signal in ensembles. Ensemble stacking that merges outputs from supervised, unsupervised, and graph-based models increases robustness and reduces single-model blind spots.

Latency and cost trade-offs. Stream-based scoring with lightweight models (trees served via optimized model servers) can achieve sub-100ms inference latency for typical feature sets, suitable for inline blocking decisions. Deeper models and GNNs are best used for enrichment and offline prioritization where nearline latency is acceptable. Cost simulations show that judicious model selection (reserve expensive models for high-risk flows) can reduce inference costs by up to 60% compared to serving all models inline.

Explainability and human workflows. Model-agnostic post-hoc explanations (SHAP values, feature contribution breakdowns) greatly aided triage, improving analyst decision speed in our user studies. Graph visualizations that expose suspicious linkages helped reveal complex fraud patterns that single-entity scoring missed. Explanations also proved essential for regulatory reporting and dispute resolution.

Privacy-preserving collaboration. Federated approaches improved detection of cross-organization fraud in our simulated experiments, particularly for mule networks where suspicious entities interact across institutions. Communication overhead and coordination policies are primary bottlenecks; secure aggregation and bloom-filter–based privacy-preserving entity matching reduce data leakage risks but add engineering complexity.

Adversarial robustness. Adversarial simulations showed predictable failure modes: attackers manipulating high-importance features (amount buckets, device fingerprint) could initially reduce detection rates. Countermeasures — feature hardening (e.g., using derived, aggregated features), adversarial retraining, and ensemble diversity — mitigated some risks. Operational monitoring for sudden drops in feature distributions and concept-drift alarms was effective at signaling attacks early.

Limitations. Our evaluation used a combination of anonymized enterprise data and synthetic scenarios; while synthetic cases help explore novel fraud types, they cannot fully replicate adversary creativity. Real-world deployment also surfaces organizational complexities (data silos, legacy systems) that are hard to model in experiments.

Implications for practice. Enterprises should adopt a layered detection strategy: fast, high-precision rules for immediate blocking; lightweight ML ensembles for real-time triage; and deeper graph and sequence models for enrichment and

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|
DOI: 10.15662/IJARCST.2024.0701811

investigation. Investment in feature stores, model governance, and explainability infrastructure yields outsized returns by reducing false positives and enabling rapid investigations.

V. CONCLUSION

This paper has presented AI–Cloud Fusion: a holistic approach to combining machine learning and cloud-native engineering for early scam detection and financial fraud prevention. By synthesizing advances in supervised learning, unsupervised anomaly detection, graph analytics, and privacy-preserving collaboration, AI–Cloud Fusion addresses the critical operational needs of modern enterprises: scalability, low-latency detection, explainability, and compliance.

We demonstrated that no single algorithm suffices across the diverse landscape of fraud. Instead, practical systems use ensembles and hybrid approaches that combine the strengths of different paradigms. Gradient-boosted trees with robust temporal and graph features form a practical baseline; sequence models and GNNs add complementary capabilities for session-level and relational fraud respectively. Unsupervised methods remain valuable for surfacing unknown threats and should be employed as part of multi-signal ensembles or triage pipelines.

Operationalizing these capabilities in cloud environments requires attention to engineering details: reliable feature stores to avoid training-serving skew, stream processing for low-latency scoring, CI/CD for safe model rollouts, and robust monitoring for concept drift and adversarial behavior. Equally important are governance mechanisms: explainability for human analysts and regulators, auditable model registries, and privacy-preserving approaches to enable cross-institutional collaboration without exposing sensitive data.
Our experimental findings underline key trade-offs: deeper models can improve recall but at a cost in inference latency and engineering complexity; graph models enhance detection of coordinated fraud but require specialized compute and careful explainability strategies. Cost-aware design patterns — such as tiered scoring and enrichment on demand — enable enterprises to balance detection performance with operational budgets.

We also emphasize the social and organizational aspects of deploying AI for fraud prevention. Successful systems integrate analyst workflows, provide clear justification for actions, and offer mechanisms for feedback and remediation. The human-in-the-loop remains essential for adjudicating borderline cases, refining models, and responding to novel attacker tactics.

In sum, AI–Cloud Fusion offers a practical roadmap for enterprises seeking to modernize fraud detection. It combines strong algorithmic foundations with cloud-native operational practices to deliver earlier detection, lower losses, and scalable defenses against evolving scams.

VI. FUTURE WORK

Future research should explore: (1) advanced causal inference methods to separate confounding correlations from causal signs of fraud; (2) privacy-enhanced collaborative frameworks that scale to many institutions with verifiable privacy guarantees; (3) adaptive learning systems that can provably bound attacker influence under realistic threat models; (4) cost-sensitive optimization that directly integrates business loss models into training objectives; and (5) standardized synthetic-data generators that better mimic adversarial creativity for benchmarking early detection systems.

REFERENCES

1. Aleskerov, E., Freisleben, B., & Rao, B. (1997). CARDWATCH: A neural network based database mining system for credit card fraud detection. *Proceedings of the IEEE/IAFE.*
2. Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science, 17*(3), 235–255.
3. Chandra Sekhar Oleti. (2022). Serverless Intelligence: Securing J2ee-Based Federated Learning Pipelines on AWS. International Journal of Computer Engineering and Technology (IJCET), 13(3), 163-180. https://iaeme.com/MasterAdmin/Journal_uploa ds/IJCET/VOLUME_13_ISSUE_3/IJCET_13_03 _017.pdf
4. Nagarajan, G. (2022). Advanced AI–Cloud Neural Network Systems with Intelligent Caching for Predictive Analytics and Risk Mitigation in Project Management. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 5(6), 7774-7781.
5. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119.*

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|

DOI: 10.15662/IJARCST.2024.0701811

6. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*(3), 559–569.
7. Paul, D., Namperumal, G. and Selvaraj, A., 2022. Cloud-Native AI/ML Pipelines: Best Practices for Continuous Integration, Deployment, and Monitoring in Enterprise Applications. Journal of Artificial Intelligence Research, 2(1), pp.176-231.
8. Muthusamy, M. (2022). AI-Enhanced DevSecOps architecture for cloud-native banking secure distributed systems with deep neural networks and automated risk analytics. International Journal of Research Publication and Engineering Technology Management, 6(1), 7807–7813. https://doi.org/10.15662/IJRPETM.2022.0506014
9. Whitrow, C., Hand, D., Juszczak, P., Weston, D., & Adams, N. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery, 18*(1), 30–55.
10. Thambireddy, S., Bussu, V. R. R., & Joyce, S. (2023). Strategic Frameworks for Migrating Sap S/4HANA To Azure: Addressing Hostname Constraints, Infrastructure Diversity, And Deployment Scenarios Across Hybrid and Multi-Architecture Landscapes. Journal ID, 9471, 1297.
11. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security, 57*, 47–66.
12. Kusumba, S. (2023). Achieving Financial Certainty: A Unified Ledger Integrity System for Automated, End-to-End Reconciliation. The Eastasouth Journal of Information System and Computer Science, 1(01), 132-143.
13. Meka, S. (2023). Building Digital Banking Foundations: Delivering End-to-End FinTech Solutions with Enterprise-Grade Reliability. International Journal of Research and Applied Innovations, 6(2), 8582-8592.
14. Selvi, R., Saravan Kumar, S., & Suresh, A. (2014). An intelligent intrusion detection system using average manhattan distance-based decision tree. In Artificial Intelligence and Evolutionary Algorithms in Engineering Systems: Proceedings of ICAEES 2014, Volume 1 (pp. 205-212). New Delhi: Springer India.
15. Vijayaboopathy, V., & Dhanorkar, T. (2021). LLM-Powered Declarative Blueprint Synthesis for Enterprise Back-End Workflows. American Journal of Autonomous Systems and Robotics Engineering, 1, 617-655.
16. Inampudi, R. K., Kondaveeti, D., & Pichaimani, T. (2023). Optimizing Payment Reconciliation Using Machine Learning: Automating Transaction Matching and Dispute Resolution in Financial Systems. Journal of Artificial Intelligence Research, 3(1), 273-317.
17. Kumar, R. K. (2023). AI-integrated cloud-native management model for security-focused banking and network transformation projects. International Journal of Research Publications in Engineering, Technology and Management, 6(5), 9321–9329. https://doi.org/10.15662/IJRPETM.2023.0605006
18. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems, 29*(8), 3784–3797.
19. Abdul Salam Abdul Karim. (2023). Fault-Tolerant Dual-Core Lockstep Architecture for Automotive Zonal Controllers Using NXP S32G Processors. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 877–885. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/7749
20. Sandeep Kamadi. (2022). AI-Powered Rate Engines: Modernizing Financial Forecasting Using Microservices and Predictive Analytics. International Journal of Computer Engineering and Technology (IJCET), 13(2), 220-233.
21. Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407.*
22. Sasidevi, J., Sugumar, R., & Priya, P. S. (2017). A Cost-Effective Privacy Preserving Using Anonymization Based Hybrid Bat Algorithm With Simulated Annealing Approach For Intermediate Data Sets Over Cloud Computing. International Journal of Computational Research and Development, 2(2), 173-181.
23. Sudhakara Reddy Peram, Praveen Kumar Kanumarlapudi, Sridhar Reddy Kakulavaram. (2023). Cypress Performance Insights: Predicting UI Test Execution Time Using Complexity Metrics. International Journal of Research in Computer Applications and Information Technology (IJRCAIT), 6(1), 167-190.
24. Vasugi, T. (2022). AI-Optimized Multi-Cloud Resource Management Architecture for Secure Banking and Network Environments. International Journal of Research and Applied Innovations, 5(4), 7368-7376.
25. Md Al Rafi. (2022). Intelligent Customer Segmentation: A Data- Driven Framework for Targeted Advertising and Digital Marketing Analytics. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 5(5), 7417–7428.
26. Navandar, P. (2022). SMART: Security Model Adversarial Risk-based Tool. International Journal of Research and Applied Innovations, 5(2), 6741-6752.
27. Poornima, G., & Anand, L. (2024, April). Effective Machine Learning Methods for the Detection of Pulmonary Carcinoma. In 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-7). IEEE.
28. Christadoss, J., Sethuraman, S., & Kunju, S. S. (2023). Risk-Based Test-Case Prioritization Using PageRank on Requirement Dependency Graphs. Journal of Artificial Intelligence & Machine Learning Studies, 7, 116-148.

|ISSN: 2347-8446| www.ijarcst.org| ICACMR 2024- Volume 7, Special Issue 1, May 2024|

DOI: 10.15662/IJARCST.2024.0701811

29. Praveen Kumar Reddy Gujjala. (2022). Enhancing Healthcare Interoperability Through Artificial Intelligence and Machine Learning: A Predictive Analytics Framework for Unified Patient Care. International Journal of Computer Engineering and Technology (IJCET), 13(3), 181-192.
30. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *Proceedings of the IEEE International Conference on Data Mining (ICDM).*