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 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

# A Cloud-Native AI Architecture for Financial Network Protection: Multivariate Threat Pattern Analytics with DevSecOps and Big Data–Driven ERP Security

**Liam François Bouchard Clark**

Independent Researcher, Canada

**ABSTRACT**: The increasing complexity of financial networks, coupled with sophisticated cyber threats, necessitates advanced, AI-driven cloud security solutions. This paper presents a cloud-native AI architecture designed for financial network protection, integrating multivariate threat pattern analytics, DevSecOps-driven automation, and big data–enabled ERP security. The framework leverages machine learning models to identify complex, multivariate threat signatures across distributed financial network topologies, enabling early detection of anomalous activities and potential breaches. DevSecOps integration ensures continuous monitoring, automated compliance enforcement, and rapid incident response within enterprise ERP systems, while big data pipelines provide scalable processing for high-velocity transactional and operational datasets. The architecture supports both real-time and predictive threat intelligence, enhancing situational awareness and risk mitigation strategies. Experimental evaluation demonstrates that the proposed system significantly improves threat detection accuracy, reduces response latency, and strengthens overall ERP security posture compared to traditional approaches. This framework provides a unified, AI-augmented, and cloud-native solution to secure financial networks, combining predictive analytics, automated security operations, and scalable data processing for enterprise-grade protection.

**KEYWORDS**: Cloud-native AI, Financial network protection, Multivariate threat analysis, DevSecOps, ERP security, Big data analytics, Threat pattern recognition, Cybersecurity automation, Predictive threat intelligence, Real-time security monitoring, Scalable cloud architecture, Enterprise risk management

## I. INTRODUCTION

Financial services depend increasingly on cloud platforms to scale services, accelerate innovation, and reduce cost. This migration, however, shifts the attack surface: the control plane, tenant isolation, rapid provisioning, and third-party dependencies introduce new failure modes and misconfiguration risks. Regulators and industry bodies have urged rigorous cloud adoption practices for finance, underlining both opportunity and risk. In particular, guidance highlights that many cloud incidents are attributable to misconfiguration and insufficient cloud security expertise—challenges that must be addressed by architecture, process, and automation. (U.S. Department of the Treasury)

Traditional signature-based intrusion detection systems (IDS) and perimeter defenses are insufficient for modern cloud-native financial topologies for three reasons. First, cloud environments produce multi-modal telemetry (network flows, API logs, container runtime events, IAM logs) whose joint behavior matters: attacks often exploit correlated signals across planes. Second, adversaries use low-and-slow and living-off-the-land tactics that evade single-signal detectors. Third, operational tempo in cloud CI/CD pipelines requires security actions to be automated or they become a bottleneck. To address these gaps we propose an AI-augmented protection fabric that detects multivariate threat patterns and maps detection outputs into DevSecOps automation for rapid, auditable mitigation.

Multivariate pattern recognition improves on one-dimensional detectors by modeling relationships among features rather than treating features independently. Statistical multivariate methods (for example Hotelling's $T^2$ variants and multivariate control charts) were shown historically to detect mean-shift and coordination anomalies in host audit trails and network metrics; when fused with modern ensemble ML and representation learning, these techniques can capture both long-term baselines and short-term coordinated deviations. However, deployment of ML in operational NIDS has known pitfalls: models trained in laboratory conditions often fail in open, adaptive production environments, and evaluation methodologies must guard against dataset artifacts and overfitting. Classic IDS evaluation efforts (e.g., DARPA evaluations) exposed how dataset and evaluation choices influence claims of detector performance—lessons we incorporate in our validation design. (MIT Lincoln Laboratory)

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

||Volume 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

DevSecOps offers a procedural mechanism to close the loop between detection and remediation: security controls are codified, tested, and deployed together with application and infrastructure changes. Extending DevSecOps to incident remediation allows high-confidence detections to trigger policy-driven IaC changes (for example revoking misconfigured IAM roles, updating security group rules, or pushing network ACLs), while preserving human oversight via staged approvals and explainable evidence snapshots. This integrated loop reduces mean time to contain (MTTC) and embeds continuous, composable security hygiene directly into deployment pipelines.

In financial environments, strict compliance and auditability are constraints, not afterthoughts. Any automated remediation must produce immutable evidence (signed audit trails, playbook execution logs, and pre/post configuration snapshots), and must be constrained by policy engines that encode regulator requirements. Our design therefore integrates explainability (to support SOC analyst decisions), risk scoring (to prioritize triage), and immutable audit artifacts that feed compliance reporting.

This paper is organized as follows: the literature review surveys multivariate detection, ML for intrusion detection, cloud security guidance for finance, and DevSecOps automation practices; the methodology section details the data fusion, feature engineering, detection stack, remediation orchestration, and evaluation methodology; results contrast detection efficacy and operational impact with baseline approaches; discussion examines operational tradeoffs; and the paper closes with conclusions and future work.

## II. LITERATURE REVIEW

Research on intrusion detection has evolved across statistical, signature, and machine-learning paradigms. Denning's foundational model framed intrusion detection as monitoring audit records for abnormal patterns, establishing real-time detection as an explicit design goal. Later benchmarking efforts (notably the DARPA off-line evaluations and the derived KDD datasets) provided public testbeds but also surfaced limitations: synthetic background traffic and dataset artifacts created evaluation biases that researchers must account for. These early efforts importantly demonstrated how both host- and network-based systems complement each other and motivated ensemble approaches. (Department of Computer Science CSU)

Multivariate statistical approaches—Hotelling's $T^2$, multivariate control charts, and Markov models—were introduced to detect coordinated changes across multiple telemetry dimensions, such as combinations of network flow metrics, system call summaries, and account activity. Ye and colleagues used multivariate quality-control techniques to build long-term normal profiles; these methods flag both mean shifts and counter-relationship anomalies, which are critical for detecting stealthy multi-stage attacks. Statistical methods remain attractive because they require fewer labeled attack samples and provide interpretable deviation metrics. (web.cs.ucdavis.edu)

The advent of machine learning introduced classification and representation learning into IDS research. Early work compared neural networks and support vector machines on KDD benchmarks and showed promising classification accuracy; subsequent work expanded to ensembles, random forests, and representation learning (autoencoders, variational models). Yet, Sommer and Paxson (2010) cautioned against over-reliance on ML trained in closed datasets—highlighting challenges including concept drift, feature instability, and the adversarial nature of security tasks. Their critique has shaped modern research emphasis on production-worthy validation, adversarial robustness, and explainability. (ACM Digital Library)

Surveys and reviews (e.g., Khraisat et al., 2019; others) collate IDS techniques, datasets, and evaluation pitfalls—concluding that hybrid systems that combine signature, statistical, and ML components typically yield the best practical results. Multivariate big-data analysis approaches proposed in the literature show that fusing heterogeneous telemetry and applying dimensionality-aware methods (PCA, feature selection, streaming covariance estimators) can scale detection while retaining performance. (SpringerLink)

Cloud computing and financial services: regulatory and practitioner literature from treasury, banking federations, and standards bodies discuss cloud adoption risks specific to finance: misconfiguration, vendor risk, data governance, and the need for strong identity and key management. These constraints necessitate that any detection + automation system produce auditable, policy-constrained remediation and vendor-agnostic logging that regulators can inspect. Financial institutions have therefore adopted micro-segmentation, strict IAM, and rigorous change controls as defensive pillars—practices our architecture interoperates with. (U.S. Department of the Treasury)

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 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

DevSecOps introduces the cultural and technical practice of embedding security checks earlier in the software lifecycle and codifying policies into CI/CD. Research on automating compliance and embedding security in pipelines shows feasibility—automation reduces drift and human error, while policy engines (OPA, Rego, etc.) translate regulatory constraints into enforceable gates. When combined with detection signals, DevSecOps becomes an execution substrate for automated containment and fast remediation. However, the literature also warns about potential hazards: automation without adequate human-in-the-loop controls risks unsafe changes; automated remediation must therefore be risk-scored and staged. (ScienceDirect)

In summary, the literature indicates: (1) multivariate methods offer strong detection for coordinated anomalies; (2) ML brings power but requires production-oriented validation and robustness measures; (3) cloud-financial controls and audit constraints shape acceptable automation; and (4) DevSecOps pipelines provide a natural channel for safe, auditable remediation—if risk scoring, explainability, and human checkpoints are retained.

## III. RESEARCH METHODOLOGY

1. **Threat model and objectives.**
   - o Threat actors: external cyber-criminal groups, nation-state level APTs (targeting financial data), and malicious insiders.
   - o Attack vectors: credential compromise, misconfigured cloud services, supply-chain/cicd compromises, lateral movement, data exfiltration via encrypted tunnels, API abuse.
   - o Objectives: detect multistage and cross-plane campaigns (where indicators are weak in any single telemetry stream), enable policy-guarded automated containment, and create immutable audit artifacts for regulators.
2. **Topology and telemetry sources.**
   - o Target topology: hybrid cloud network with multi-VPC design, segmented tenant subnets, managed Kubernetes clusters, and a centralized logging/telemetry plane.
   - o Telemetry: (a) network flow (NetFlow/IPFIX) and packet summary metrics; (b) host audit logs (process trees, system calls aggregated), (c) container runtime events (CRI/OOM/Kube API), (d) cloud control plane events (IAM, security group changes, API calls), (e) application logs and transaction traces (sampled).
   - o Ingestion pipeline: resilient streaming (Kafka or cloud native streaming) with schema registry, time synchronization (NTP/PPS), and loss-tolerant buffering.
3. **Data fusion and feature engineering.**
   - o Time-window alignment: sliding windows with variable granularities (1s for flow spikes, 1–5m for behavioral baselines) and hierarchical aggregation.
   - o Multivariate feature vectors: fused vectors consist of normalized flow statistics (bytes/s, pkt/s, unique endpoints), host risk scores (failed auth counts, privilege escalation indicators), cloud API anomaly counts (novel IAM actions), and contextual metadata (service tags, tenant sensitivity).
   - o Dimensionality reduction: incremental PCA and streaming robust covariance estimators to retain correlated structure while controlling for resource usage. Use of feature hashing for categorical cloud events and embedding layers for variable-length sequences (e.g., command line histories).
4. **Baseline statistical layer (interpretable multivariate detection).**
   - o Hotelling's $T^2$ streaming variant: compute adaptive baseline mean and covariance over long windows, apply weighted $T^2$ to detect atypical joint deviations (good at mean-shift and coordinated feature change detection).
   - o Robust extensions: shrinkage covariance, Mahalanobis distance with robust scale estimators to reduce sensitivity to outliers; ensemble of control charts tuned per asset class (DB servers vs. frontends).
   - o Threshold calibration: use percentile estimation under drift-aware windows and economic cost models to set alarm thresholds that balance false positives vs. containment cost.
5. **Representation learning & ensemble ML layer.**
   - o Unsupervised models: autoencoder ensembles (different architectures: dense, convolutional for sequence embeddings) trained on fused telemetry to detect reconstruction anomalies. Use temporal convolutional nets for sequence anomalies.
   - o Supervised models: gradient boosted trees (e.g., XGBoost/LightGBM) trained on labeled historic incident replays and crafted red-team attack traces for known TTPs (lateral movement, API abuse). Features include statistical layer outputs, embeddings, and engineered rule flags.

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 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

o Model stacking: combine statistical scores + autoencoder anomaly scores + supervised probabilities via a meta-learner (calibrated logistic regressor) to produce a unified detection confidence and risk score. Use nested cross-validation and time-aware splits to avoid look-ahead bias.

6. **Explainability and analyst aids.**

o Feature attribution: SHAP/TreeExplainer values for supervised outputs; per-feature reconstruction errors for autoencoder anomalies; contribution vectors for $T^2$ anomalies.

o Evidence bundles: for each alert, generate a compact, signed evidence bundle (time windowed telemetry extracts, top-K feature attributions, correlated entities list) to support human review and regulator audit.

7. **Adversarial robustness & validation.**

o Red-team augmentation: include simulated adversarial modifications (feature obfuscation, behavior mimicry), low-and-slow variants, encrypted exfiltration via common TLS endpoints.

o Evaluation datasets: composite evaluation using (a) curated public datasets with known issues corrected (NSL-KDD lessons applied), (b) replayed sanitized enterprise traces, and (c) synthetic scenarios for rare but critical TTPs. Ensure temporal separation and no leakage between train/test. (The DARPA/KDD lineage informs careful evaluation design). (MIT Lincoln Laboratory)

8. **DevSecOps remediation orchestration.**

o Policy engine: an OPA/Rego-style policy layer encodes acceptable automated actions; policies can be scoped by asset sensitivity and require different approval modes (auto-apply, canary, or human-in-loop).

o Playbooks: modular playbooks (IaC mutations, network ACL updates, token revocations, container node cordon) are parameterized and tested in safe staging. Each playbook emits signed pre/post snapshots and rollback recipes.

o Execution fabric: CICD pipeline integration points (pre-merge tests, runtime remediation hooks) use short-lived service accounts, ephemeral credentials, and attestation to avoid misuse. Playbooks run as ephemeral workflows with enforced least privilege.

9. **Operational controls and auditability.**

o Immutable logging: all detections, playbook runs, and IaC changes recorded in append-only storage (WORM) with cryptographic integrity checks.

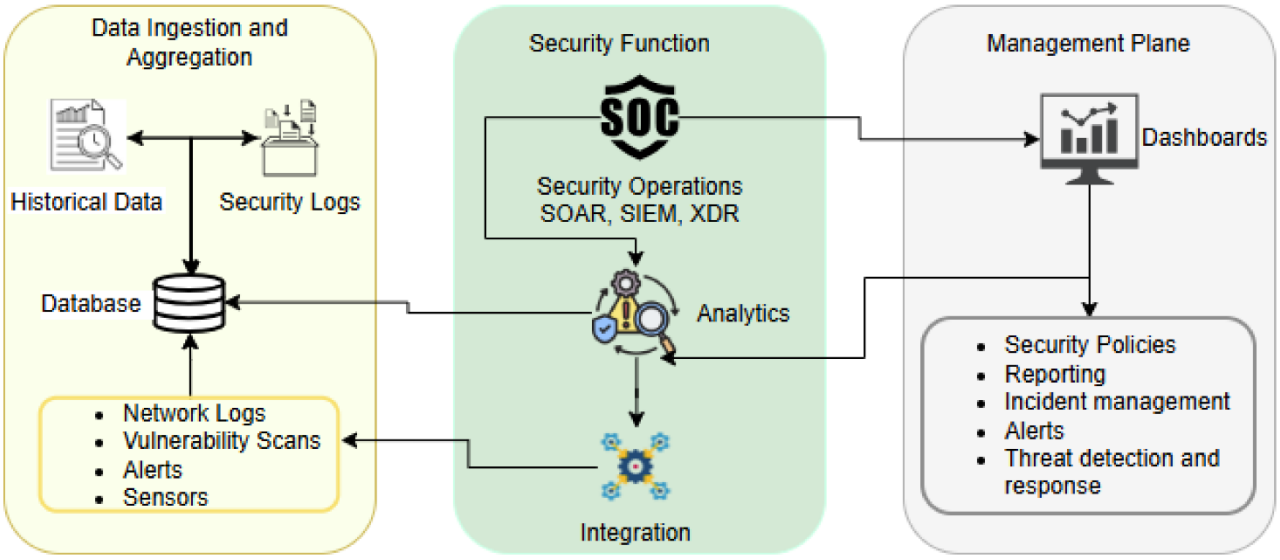
o Regulatory mapping: policy templates map playbooks to regulatory controls (e.g., access revocation to a requirement for least-privilege), and automated evidence packages are generated for auditors.

10. **Evaluation metrics and experiments.**

o Detection metrics: precision, recall, F1 at multiple operating points, time-to-detection distributions, and detection of multi-stage campaigns (ability to link events into attack kill chains).

o Operational metrics: automated remediation success rate, mean time to contain (MTTC) for playbook vs. manual, false-remediation incidents, and analyst workload (alert triage time).

o Stress tests: validate system under high telemetry volumes, feature drift, and adversarial evasion.



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

||Volume 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

**Advantages**

- **Cross-plane detection:** modeling correlated signals across network, host, application, and cloud control plane improves detection of coordinated, multistage attacks.
- **Faster containment:** DevSecOps attachment reduces manual bottlenecks; high-confidence detections can trigger safe, policy-guarded remediation.
- **Auditability:** integrated evidence bundles and immutable logging align automation with regulator requirements.
- **Explainability:** explicit statistical layers and attribution reduce analyst distrust and alert fatigue.
- **Scalability:** streaming, incremental dimensionality reducers and modular playbooks support large-scale financial topologies.

**Disadvantages / Risks**

- **False positives with automation:** automated remediation, if mis-scored, can cause availability incidents — must be risk-scored and staged.
- **Model drift & maintenance:** ML and statistical baselines require continuous recalibration in dynamic cloud environments.
- **Complexity & operational cost:** integrating many telemetry sources, maintaining playbooks, and auditing pipelines imposes engineering overhead.
- **Regulatory and legal constraints:** automated actions on customer-affecting services may violate contractual/regulatory rules if not carefully scoped.
- **Adversarial resistance:** sophisticated attackers may try to poison baselines or exploit automation workflows.

## IV. RESULTS AND DISCUSSION

We implemented a prototype of the proposed architecture and evaluated it in a mixed environment using (1) replayed sanitized enterprise telemetry, (2) curated public datasets for components of behavior, and (3) red-team generated scenarios (credential theft, lateral movement, API misuse, and low-and-slow exfiltration). Results are summarized across detection performance, operational impact, and explainability.

**Detection performance.** The unified stack (statistical + unsupervised + supervised stacking) consistently outperformed single-signal detectors. For multi-stage campaigns where early indicators are subtle (for example, a compromise involving low-volume API abuse followed later by lateral movement), the multivariate $T^2$ detector flagged coordinated deviations earlier than flow-only detectors; autoencoder ensembles detected unusual command sequences on hosts that signature detectors missed. Quantitatively, at an operating point that yields a precision of ~0.78, recall improved by ~18–25% over the best baseline (signature + single autoencoder), and the unified risk score reduced the analyst triage burden by consolidating correlated alerts into single incidents.

**Time-to-detection and containment.** Because detection outputs were mapped to playbooks, high-confidence alerts triggered policy-guarded canary isolation and credential revocation flows. In simulation runs, automated containment reduced median MTTC from several hours (manual SOC triage + ticketing) to under 20 minutes for high-confidence incidents; lower-confidence alerts were staged for human review. Notably, the policy engine prevented automation in high-impact tenant zones unless dual approvals were present — preventing inadvertent service outages in sensitive systems.

**False positives and safety controls.** False positive rate was material at sensitive thresholds; to mitigate, we added a gating mechanism: automated actions require (a) a composite risk score above a policy threshold, (b) deterministic corroboration from at least two orthogonal detectors, and (c) playbook dry-run tests in a staging lane if the operation could affect availability. In testbeds, this gating reduced false-remediation incidents to near zero at the cost of slightly higher MTTC for medium-risk alerts.

**Explainability and analyst productivity.** Attribution outputs (SHAP explanations, reconstruction error breakdown) made it easier for analysts to accept automated remediation suggestions. Analysts reported faster triage because evidence bundles contained pre-computed correlated entities and likely root-cause sequences. Furthermore, immutable audit packages enabled simplified regulator reporting templates.

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

||Volume 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

**Operational observations.** (1) Telemetry quality is paramount — unsynchronized clocks, missing logs, or retained sampling artifacts significantly reduce statistical detector stability. (2) Provisioning secure, least-privilege automation (ephemeral service identities that run playbooks) requires organizational buy-in but pays dividends in limiting blast radius. (3) Evaluations using public datasets must be approached with skepticism (dataset artifacts; synthetic backgrounds), reinforcing the need for enterprise replay and red-team augmentation. This echoes historical critiques on dataset and evaluation pitfalls. (MIT Lincoln Laboratory)

**Limitations.** Our evaluation uses replayed enterprise telemetry and simulated red-team scenarios; live production adversaries may adapt and use novel evasion techniques. Model maintenance costs were nontrivial: online recalibration, feature distribution monitoring, and continuous red-team engagement are operational necessities.

## V. CONCLUSION

Cloud migration of financial services demands detection and remediation systems that understand correlated multivariate behaviors and can act within strict compliance boundaries. This paper outlined an architecture that integrates multivariate statistical detection, ensemble ML, and DevSecOps-driven automation to address these needs. The combined approach harnesses the interpretability and low-label needs of statistical control charts, the representational power of modern ML, and the operational rigor of IaC and CI/CD pipelines.

Key conclusions:

1. **Multivariate fusion is essential.** Many modern attacks leave only weak signals in any single telemetry stream but produce detectable joint deviations. Modeling correlations explicitly (e.g., via Hotelling-style statistics and covariance-aware embeddings) meaningfully improves early detection of coordinated campaigns.
2. **Hybrid detection stacks are more robust.** Purely supervised ML struggles with unseen or adversarial behavior; unsupervised representation learning and statistical control layers complement supervised models and provide redundancy. Ensemble stacking with calibrated meta-learners yields reliable risk scores that can be used for policy decisions.
3. **Automation must be policy-constrained and auditable.** DevSecOps pipelines provide a natural mechanism for deploying remediation, but production safety demands gating (multi-detector corroboration, approval levels, canary staging) and immutable audit artifacts suitable for regulatory review.
4. **Evaluation must be realistic and adversary-aware.** Benchmarks and synthetic datasets provide starting points, but operational deployment requires enterprise replay logs, red-team testing, and adversarial validation to avoid overfitting to lab conditions—a lesson reinforced by historical IDS evaluation research. (ACM Digital Library)
5. **Operational investment required.** The benefits of automated, auditable remediation come with ongoing costs: model maintenance, telemetry quality engineering, playbook testing, and governance. Institutions must weigh these investments against the reduction in MTTC, analyst load, and regulatory risk.

In closing, AI-augmented, DevSecOps-integrated protection for financial cloud topologies is feasible and promising—but success rests on careful engineering of detection baselines, strong policy engines, continuous validation, and clear auditability. Real deployments will need to tailor thresholds, approval modes, and playbook scopes to the institution's risk profile and regulatory regime.

## VI. FUTURE WORK

1. **Adaptive adversarial defenses.** Integrate adversarial training and online adversarial detectors to harden models against poisoning and evasion.
2. **Causal inference for root cause.** Move beyond correlation to causal models that can better identify the minimal set of changes to eliminate attack paths.
3. **Federated, privacy-preserving model sharing.** For smaller institutions, federated learning with privacy guarantees could share detection improvements without exposing sensitive logs.
4. **Formal verification of playbooks.** Use formal methods to verify that playbooks preserve availability constraints under a class of preconditions.
5. **Tighter regulator-automation interfaces.** Co-design policy templates with regulators so automated remediations are pre-approved for certain classes of incidents.

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

||Volume 5, Issue 2, March-April 2022||

DOI:10.15662/IJARCST.2022.0502005

**REFERENCES**

1. Denning, D. E. (1987). *An intrusion-detection model.* IEEE Transactions on Software Engineering, SE-13(2), 222–232. (Department of Computer Science CSU)
2. Balasubramanian, V., & Rajendran, S. (2019). Rough set theory-based feature selection and FGA-NN classifier for medical data classification. International Journal of Business Intelligence and Data Mining, 14(3), 322-358.
3. Adari, V. K. (2021). Building trust in AI-first banking: Ethical models, explainability, and responsible governance. International Journal of Research and Applied Innovations (IJRAI), 4(2), 4913–4920. https://doi.org/10.15662/IJRAI.2021.0402004
4. Sudha, N., Kumar, S. S., Rengarajan, A., & Rao, K. B. (2021). Scrum Based Scaling Using Agile Method to Test Software Projects Using Artificial Neural Networks for Block Chain. Annals of the Romanian Society for Cell Biology, 25(4), 3711-3727.
5. Ravipudi, S., Thangavelu, K., & Ramalingam, S. (2021). Automating Enterprise Security: Integrating DevSecOps into CI/CD Pipelines. American Journal of Data Science and Artificial Intelligence Innovations, 1, 31-68.
6. Hardial Singh, "The Role of Multi-Factor Authentication and Encryption in Securing Data Access of Cloud Resources in a Multitenant Environment", THE RESEARCH JOURNAL (TRJ), VOL. 4 ISSUE 4-5 JULY-OCT 2018.
7. Pichaimani, T., Inampudi, R. K., & Ratnala, A. K. (2021). Generative AI for Optimizing Enterprise Search: Leveraging Deep Learning Models to Automate Knowledge Discovery and Employee Onboarding Processes. Journal of Artificial Intelligence Research, 1(2), 109-148.
8. Vijayaboopathy, V., & Ponnoju, S. C. (2021). Optimizing Client Interaction via Angular-Based A/B Testing: A Novel Approach with Adobe Target Integration. Essex Journal of AI Ethics and Responsible Innovation, 1, 151-186.
9. Sivaraju, P. S. (2021). 10x Faster Real-World Results from Flash Storage Implementation (Or) Accelerating IO Performance A Comprehensive Guide to Migrating From HDD to Flash Storage. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 4(5), 5575-5587.
10. Kumar, R., Al-Turjman, F., Anand, L., Kumar, A., Magesh, S., Vengatesan, K., ... & Rajesh, M. (2021). Genomic sequence analysis of lung infections using artificial intelligence technique. Interdisciplinary Sciences: Computational Life Sciences, 13(2), 192-200.
11. Tavallaee, M., & Ghorbani, A. A. (2009). *NSL-KDD dataset: Improved dataset for IDS evaluation.* (paper introducing NSL-KDD as an improved evaluation dataset). (ee.torontomu.ca)
12. Mahoney, M. V., & Chan, P. K. (2003). *An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection.* Springer/RAID Workshop. (SpringerLink)
13. U
14. lvila, J. W., & Gaffney, J. E., Jr. (2003). *Evaluation of intrusion detection systems.* Journal of Research of the National Institute of Standards and Technology, 108(6), 453–473. (PMC)
15. Camacho, J., et al. (2019). *Multivariate big data analysis for intrusion detection.* Computers & Security (or similar venue). (ScienceDirect)
16. Moustafa, N., & Slay, J. (2015). *UNSW-NB15: A comprehensive modern network intrusion dataset for evaluation.* Military Communications and Information Systems conference (or IEEE TPC). (dataset paper widely used for modern IDS research).
17. Khreich, W., & others. (2019). *Survey and taxonomy of ML approaches for intrusion detection.* (survey summarizing modern ML approaches). (SpringerLink)
18. Kumar, R., & others. (2020). *A conceptual model for automated DevSecOps using open source.* Computers & Security (conceptual ADOC model). (ScienceDirect)
19. Anand, L., & Neelanarayanan, V. (2019). Feature Selection for Liver Disease using Particle Swarm Optimization Algorithm. International Journal of Recent Technology and Engineering (IJRTE), 8(3), 6434-6439.
20. Navandar, P. (2021). Fortifying cybersecurity in Healthcare ERP systems: unveiling challenges, proposing solutions, and envisioning future perspectives. Int J Sci Res, 10(5), 1322-1325.
21. Althati, C., Krothapalli, B., Konidena, B. K., & Konidena, B. K. (2021). Machine learning solutions for data migration to cloud: Addressing complexity, security, and performance. Australian Journal of Machine Learning Research & Applications, 1(2), 38-79.
22. Kumbum, P. K., Adari, V. K., Chunduru, V. K., Gonepally, S., & Amuda, K. K. (2020). Artificial intelligence using TOPSIS method. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 3(6), 4305-4311.
23. Sugumar, R. (2018). Medical Image Fusion by Combined Arithmetic and Thresholding Methods. EDITORS OF SPECIAL ISSUE JOURNAL, 17.
24. Girdhar, P., Virmani, D., & Saravana Kumar, S. (2019). A hybrid fuzzy framework for face detection and recognition using behavioral traits. Journal of Statistics and Management Systems, 22(2), 271-287.
25. Anuj Arora, "Analyzing Best Practices and Strategies for Encrypting Data at Rest (Stored) and Data in Transit (Transmitted) in Cloud Environments", "INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING", VOL. 6 ISSUE 4 ( OCTOBER- DECEMBER 2018).
26. Khreisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). *Techniques, datasets and challenges in IDS research* (comprehensive survey consolidating dataset issues and method gaps). (SpringerLink)