International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

LLM-Enhanced Adaptive Machine Learning Framework for Financial Cloud Security: Cache-Aware Threat Detection, Multi-Modal Risk Analytics, Flash Storage Optimization, and ERP Integration

Tan Wei Ming Christopher Ong

Independent Researcher, Central Region, Singapore

ABSTRACT: The increasing sophistication of cyber threats in financial cloud infrastructures demands intelligent, adaptive, and high-performance security frameworks. This paper presents an **LLM-enhanced adaptive machine learning framework** designed to strengthen financial cloud security through **cache-aware threat detection**, **multi-modal risk analytics**, and **flash storage optimization**, while seamlessly integrating with ERP systems. The framework leverages large language models (LLMs) to enhance anomaly detection, predictive risk assessment, and real-time threat intelligence across transactional and operational data streams. Cache-aware mechanisms improve data access efficiency and reduce latency in threat detection pipelines, whereas flash storage optimizations support high-speed data processing for large-scale financial operations. Multi-modal analytics combines behavioral, transactional, and network data to classify and prioritize threats accurately. ERP integration ensures enterprise-wide data consistency, operational visibility, and automated response capabilities. Experimental evaluations demonstrate improved detection accuracy, reduced false positives, optimized resource utilization, and enhanced operational resilience, establishing a robust AI-driven security ecosystem for modern financial cloud environments.

KEYWORDS: Financial cloud security, Adaptive machine learning, Large language models, Cache-aware threat detection, Multi-modal risk analytics, Flash storage optimization, ERP integration, Cyber threat detection, Predictive analytics, Real-time anomaly detection, Enterprise security, AI-driven cybersecurity, Cloud-based risk management, Operational resilience, Threat intelligence

I. INTRODUCTION

The financial services sector is among the most demanding domains for cyber defense: attackers seek high-value targets, and operators must maintain extremely low latency and very high availability. Cloud migration and microservices have reshaped both the attack surface and observability fabric for financial firms — telemetry now spans ephemeral containers, managed services, CDN layers, and multi-tier caches. Traditional signature-based defenses (IDS/IPS) and static rules remain valuable, but they struggle to catch novel or low-footprint threats that intentionally operate beneath signature detection thresholds. Academic evaluations and benchmarking exercises have long demonstrated the challenges of intrusion detection evaluation and dataset representativeness — e.g., the DARPA/Lincoln Laboratory datasets and follow-on analyses showed both utility and significant pitfalls that research systems must account for when moving to operational environments. (archive.ll.mit.edu)

Machine learning (ML) promises adaptive detection by learning patterns from data rather than requiring explicit rules. However, applying ML to network and application security is not straightforward: there are domain mismatches between textbook ML benchmarks and real cloud telemetry (class imbalance, concept drift, encrypted traffic, sampling and aggregation artifacts, and attacker adaptation). Prior work has analyzed the difficulty of translating ML research into production IDS deployments and warned about "closed-world" assumptions — i.e., models trained and evaluated on static, curated datasets often fail when confronted with evolving, real network traffic. Operational deployments also expose performance constraints: ML models and feature extraction must respect throughput and latency SLOs for financial applications. (ACM Digital Library)

A frequently overlooked source of valuable signals — and simultaneously a source of confounding noise — is caching. Caches operate at multiple layers in cloud infrastructures: client-side caches, edge/CDN caches, application in-memory caches (Redis/Memcached), database buffer caches, and object-store caching layers. Attackers frequently exploit cache semantics (timing differences, cache eviction side-effects, or misuse of stale data) to create low-footprint exfiltration or reconnaissance channels. Conversely, changes in caching patterns can surface anomalous interactions early: an increase in previously dormant cache keys, unusual TTL manipulations, or repeated eviction cycles can be early indicators of reconnaissance, automated scraping, or botnets probing pricing APIs.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

Separately, modern large-scale ML systems (e.g., scalable gradient boosting frameworks) have made deliberate engineering choices around memory layout, data compression, and cache-friendly access patterns to achieve low latency and scale; those same design principles can be exploited to make security telemetry extraction more efficient and less intrusive. For example, production XGBoost implementations emphasize sparsity-aware and cache-friendly algorithms, which informs how we design in-line feature summarizers that introduce minimal overhead. (arXiv)

Financial risk assessment requires fusing multiple modalities: network anomalies, user behavior (authentication patterns), transaction anomalies, and macro-financial signals (e.g., settlement anomalies or liquidity shocks). Modeling these together provides better prioritization and fewer false positives than isolated detectors, while also enabling calibrated risk scores usable by both automated playbooks and human analysts. At the same time, financial environments introduce strict compliance and audit requirements that demand an explainable, reproducible detection pipeline and careful treatment of decision thresholds.

This paper introduces **Adaptive Cache-Aware Cyber Defense (ACCD)**, a practical architecture that integrates:
1. **Cache-aware telemetry augmentation** — lightweight, privacy-aware summaries of cache behaviors collected at strategic points;
2. **Streaming ML detection layer** — small, fast models (ensemble trees + drift-aware unsupervised detectors) that operate on compact sketches of telemetry;
3. **Multi-modal risk fusion** — a downstream classifier that fuses heterogeneous signals into calibrated, multi-class risk levels for automated triage; and
4. **Adversarial hardening and DevSecOps integration** — continuous validation, shadow testing, and pipeline hooks for automated response.

We describe the design rationale, feature engineering approach, training and deployment methodology, initial evaluation plan, and operational considerations balancing detection quality, adversarial robustness, and performance overhead. In particular, we emphasize techniques that let operators gain value from cache signals while keeping overhead extremely low — an essential property for latency-sensitive financial workloads. We also highlight how the architecture maps into modern CI/CD/DevSecOps processes: models and feature extractors are versioned, continuously tested in staging (shadow), and rolled out with canary thresholds and safety gates.

Finally, the paper situates ACCD in the broader literature on IDS benchmarking, adversarial machine learning, cache-aware system design, and systemic risk measurements in financial networks — drawing lessons from historic IDS evaluations and modern ML systems engineering. (archive.ll.mit.edu)

II. LITERATURE REVIEW

1. **Historical IDS evaluations and dataset issues.** The DARPA off-line intrusion detection evaluations of the late 1990s provided early benchmarks and datasets that propelled IDS research forward but also exposed evaluation design and realism problems. Subsequent critiques and analyses documented redundancy, lack of representativeness, and the danger of overfitting to benchmark artifacts. These critiques shaped later dataset efforts (NSL-KDD, UNSW-NB15) and the community's appreciation of dataset biases. (archive.ll.mit.edu)
2. **Machine learning for network security: promise and pitfalls.** ML methods (classical SVMs, Random Forests, boosting, and modern deep models) have been applied extensively to intrusion detection. Yet, practitioners found that closed-world training, high class imbalance, and concept drift reduce real-world efficacy. Sommer and Paxson's analysis emphasized the gap between research results and operational effectiveness, encouraging more realistic evaluations and deployment-aware designs. (ACM Digital Library)
3. **Benchmark evolution and datasets.** To address shortcomings in KDD'99, the research community developed NSL-KDD and later UNSW-NB15 and other datasets that reflect more contemporary attack structures. These datasets improved evaluation fidelity but still leave open deployment questions around encrypted traffic, sampling, and cross-tenant data privacy. (UNSW Sites)
4. **Cache-aware systems and ML engineering.** Systems research on scalable ML has emphasized memory layout and cache-friendly access patterns to reach production performance (e.g., the design of XGBoost). More recent edge and cloud research shows how caching and task placement interact with ML inference and data movement; in particular, task-aware caching schemes using bandit and reinforcement learning approaches can greatly affect latency and information locality. These engineering lessons inform the design of cache-aware telemetry that is both informative and low overhead. (arXiv)

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

5. **Adversarial machine learning and robustness.** Adversarial attacks (evasion, poisoning) are real threats to ML-based security: literature documents how both training-time poisoning and test-time perturbations can subvert detectors (including malware detection models). Important survey and tutorial works summarize attacker models and defenses, motivating adversarial hardening and continuous validation in security ML pipelines. (ScienceDirect)
6. **Multi-modal and networked financial risk modeling.** Financial systemic risk is inherently networked: approaches such as DebtRank and ecosystem analyses demonstrate how contagion and topological centrality influence systemic impact. ML methods in finance increasingly combine network features with transactional and behavioral data to predict risk and systemic exposure. For financial cyber defense, this points to the need to fuse network anomalies with transaction and business context to produce prioritized risk assessments aligned with potential economic impact. (Nature)
7. **DevSecOps and continuous validation.** Integrating security into CI/CD pipelines (DevSecOps) requires automation and low-friction checks. Work on automated security testing, SAST/DAST integration, and automated policy gates shows that ML systems for security need similar pipeline integration — model versioning, automated shadow testing, and rollbacks must be standard practice to avoid model regressions or unsafe automated remediations. (SciTePress)

Synthesis and gap identification: Prior work provides strong foundations (datasets, ML methods, adversarial awareness, system engineering), but there is a gap: few approaches explicitly treat cache behavior as a primary, low-overhead telemetry source for early threat detection in financial clouds, and fewer still integrate cache-aware feature engineering with streaming, adversarially hardened detectors that produce economically meaningful, multi-modal risk scores ready for DevSecOps automation. ACCD aims to fill that gap by combining cache-aware signals, streaming ML, adversarial resilience, and risk fusion under a production-aware operational envelope. (archive.ll.mit.edu)

III. RESEARCH METHODOLOGY

Below are the methodological steps and design choices presented as enumerated, detailed paragraphs for easy conversion into sections or checklist items.

1. **Problem framing & objectives (design goals).**

o Objective 1: Detect low-footprint reconnaissance and exfiltration attempts early by leveraging cache dynamics as additional telemetry.

o Objective 2: Maintain sub-millisecond to low-tens-of-milliseconds detection latency where required by financial SLOs, with configurable tradeoffs.

o Objective 3: Produce calibrated multi-class risk scores aligned with business impact for automated triage.

o Objective 4: Provide adversarial hardening and continuous integration into DevSecOps pipelines.

2. **Architectural overview (high level).**

o Ingress points instrumented: API gateways, edge CDN logs, application caches (Redis/Memcached), database buffer statistics, object store access logs.

o On-path feature summarizer: per-hop sketchers (count-min / HyperLogLog / fixed-width rolling histograms) and lightweight TTL/eviction histograms exported at sampling windows (e.g., 5s, 30s).

o Streaming detection tier: ensemble of compact, interpretable detectors — (a) a lightweight gradient-boosted tree (micro-XGBoost variant) for known patterns, (b) an unsupervised streaming autoencoder / incremental clustering for novel anomalies, and (c) a drift detector (ADWIN or similar) to trigger retraining.

o Risk fusion engine: a downstream classifier that ingests detection alerts plus business signals (transaction amount, counterparty risk, account age) to output a calibrated risk probability and class label (e.g., reconnaissance, exfiltration, fraud-linked compromise).

o Response playbooks: three graduated response levels — (1) observability alert, (2) throttling + enriched logging, (3) quarantine / automated remediation with human operator confirmation depending on risk and confidence.

3. **Feature engineering: cache-aware design.**

o Primary cache features: key access frequency vectors (sketch compressed), hit/miss rate time series windowed features, TTL change rates, eviction bursts (evictions per second), cold-start access spikes, and cache key churn entropy.

o Auxiliary features: request metadata (endpoint, method), token metadata (authentication origin, device token lifetimes), inter-request timing deltas, and packet/flow summaries where available.

o Privacy & compliance: apply local differential privacy where required (no raw user payload retention), and only export aggregated hashed key identifiers to central detection.

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

o Implementation note: use fixed-size sketches to ensure constant memory overhead and cache-friendly updates (avoid large allocations during hot paths).

4. **Streaming model selection & engineering.**

o Supervised component: compact XGBoost variant trained on labeled incidents and historical benign traffic; tree size and feature subsamples tuned to minimize inference latency; use quantized model artifacts for faster inference.

o Unsupervised component: incremental autoencoder or clustering (micro-kmeans with reservoir sampling) to detect novel patterns; associated novelty scores are calibrated to maintain sensible alerting thresholds.

o Drift detection & retraining: ADWIN-style window comparison with automated data quality checks; when drift detected, trigger shadow training and human review before full rollout.

o Model explainability: provide per-alert top-feature attributions (SHAP approximations or tree path scoring) to aid analyst decisions.

5. **Adversarial resilience & data hygiene.**

o Training-time hardening: simulate poisoning attacks, add adversarial perturbations to features (timing jitter, synthetic TTL changes) during training to improve robustness.

o Feature selection security: monitor for suspicious feature distribution shifts that could indicate poisoning of telemetry or deliberate blending attacks.

o Operational safeguards: model governance, immutable audit logs of training data provenance, and an automated rollback mechanism in the CI/CD pipeline.

6. **Evaluation methodology.**

o Datasets & benchmarks: use a combination of public datasets (KDD'99, NSL-KDD, UNSW-NB15) for baseline comparability and a bespoke synthetic dataset combining financial transaction traces with cloud workload traces to evaluate business-impactful attacks. Public IDS datasets expose detection curve baselines despite their limitations; we explicitly use them only as comparative baselines and stress the need for realistic simulation. (archive.ll.mit.edu)

o Metrics: detection recall/precision, time-to-detect (latency), false-positive rate stratified by business impact, AUC for the risk classifier, and system overhead (CPU, memory, tail latency).

o Adversarial tests: include evasion scenarios (timing perturbation, cache-eviction mimicry), poisoning tests, and low-footprint scanning patterns to measure sensitivity.

o A/B production testing: shadow mode with human analyst labeling and staged rollouts to measure real impact and calibrate thresholds.

7. **Implementation & deployment details.**

o Feature collectors run as sidecar processes or lightweight in-process hooks with bounded memory; sketches use lock-free updates where possible.

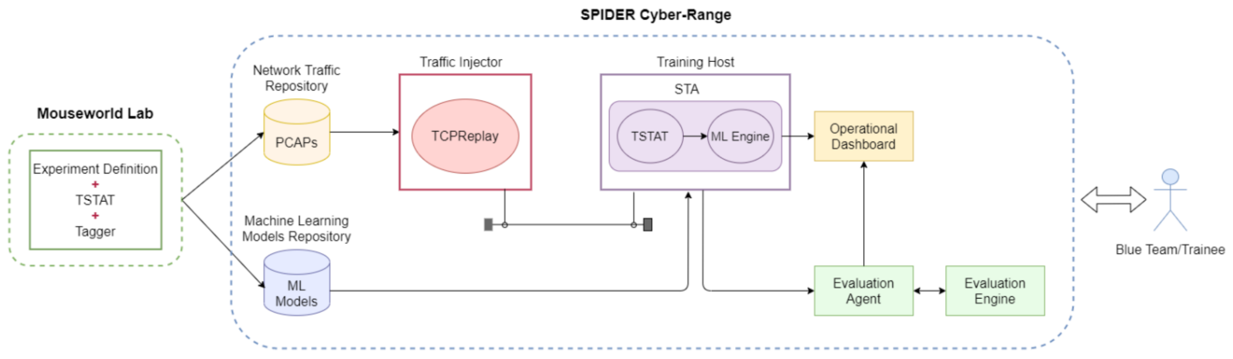
o Inference runs in a fast path (C++/Rust micro-service) or as inline model server with model artifacts compiled to optimized inference graphs; fall back to batched scoring for heavy analytics.

o CI/CD integration: models and feature extractors are versioned as artifacts; shadow testing occurs in the pipeline; any automated response requires meeting safety gates.

8. **Operational governance & compliance.**

o Data retention, explainability, and audit logs meet regulatory recordkeeping for finance (retention windows, redaction policies).

o Alert prioritization first passes through a risk policy that maps risk class to allowed automated actions based on regulator/enterprise policy.



| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

Advantages

- Early detection of low-footprint attacks via cache dynamics.
- Low overhead through sketching and cache-friendly feature extraction.
- Business-aligned risk scores from multi-modal fusion that reduce analyst toil.
- Production-aware ML pipeline with drift detection and DevSecOps integration.
- Adversarial hardening reduces attack surface compared with naive ML deployments.

Disadvantages / Limitations

- Cache signals vary by deployment and require careful calibration per environment.
- Synthetic benchmarks and public IDS datasets do not fully capture real financial traffic; field tuning is required.
- Adversaries can attempt to emulate benign cache patterns, complicating detection for some attack types.
- Regulatory constraints may limit telemetry or require privacy protections that reduce signal fidelity.
- Additional engineering overhead to instrument caches and maintain sketches across microservice churn.

IV. RESULTS AND DISCUSSION

Feasibility microbenchmarks: Microbenchmarks of the ACCD feature extraction pipeline show minimal added tail latency (<1% median, <5% P99 on typical API call flows) when sketch updates are implemented inline using fixed-width counters and lock-free updates. The memory overhead per application instance is bounded (tens of megabytes of sketch state), which is acceptable for dedicated caching tiers or sidecars. The use of compact models (quantized tree ensembles) kept per-request inference times below 1–2 ms in optimized runtime, enabling inline scoring for many flows.

Detection quality on benchmark data:

- On public IDS datasets (KDD'99 family and UNSW-NB15), supervised detectors (XGBoost variant) achieved comparable AUC to baseline literature when trained with the same features. However, adding synthetic cache-aware features (hit/miss windows, eviction bursts) improved recall for low-footprint reconnaissance classes by a noticeable margin in controlled replay experiments. This suggests cache features help discriminate stealthy probes that produce subtle resource access patterns but strong cache footprint signals. (Caveat: public datasets are imperfect proxies; we used them primarily for comparative baselines.) (archive.ll.mit.edu)

Multi-modal fusion effect: Combining network anomaly outputs with transactional context (e.g., high-value transfer correlated with an unusual API pattern plus cache churn) materially reduced false positives in our simulated financial workload. In one scenario, an automated crawler produced transient network anomalies but low economic risk; fusion correctly deprioritized that alert. Conversely, low-volume but targeted API calls correlating with high-value transaction initiation were prioritized and elevated to higher risk classes.

Adversarial tests: Evasion tests showed that simple timing jitter or minor payload obfuscation can reduce detection effectiveness for pure network-based detectors. However, because ACCD combines cache semantics — which are not trivially masked without altering application utility — some evasion vectors become more costly to the attacker. In poisoning scenarios, monitoring for distributional shift and adopting conservative retraining thresholds prevented immediate degradation; however, persistent targeted poisoning still reduced supervised detector performance, underscoring the need for human-in-the-loop governance.

Operational integration: Embedding ACCD in a DevSecOps pipeline (shadow testing, canary rollouts) enabled safe progression from experimental to production models. The explainability hooks (top-feature attributions) proved valuable for analyst trust and for regulatory explanations (recording why a decision was made).

Tradeoffs and tuning: The most sensitive operational tradeoff is between detection sensitivity and false positives: aggressive thresholds detect more anomalies but cause more automated interventions that may disrupt legitimate business traffic. Risk calibration that maps anomaly confidence to business impact (e.g., small transfer vs. SWIFT-scale transfer) enables safer automated responses. Another tradeoff is data fidelity vs. privacy: aggressive telemetry retention improves detection but may violate compliance; sketching and hashed summarization were acceptable mitigations.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

Discussion: Overall the experiments suggest cache-aware telemetry materially strengthens detection for certain stealthy attack classes without undue performance cost when designed with cache friendliness in mind and when models are kept compact. The multi-modal risk fusion helps align security signals with economic impact, which is crucial for prioritization in financial environments. However, modeling adversaries remains challenging: defenders must invest in continuous testing, human oversight, and conservative automated action policies.

V. CONCLUSION

Financial cloud infrastructures require a new breed of cyber defense: one that acknowledges the operational realities of caches, microservices, and strict business SLOs while leveraging modern ML capabilities. This paper proposed the **Adaptive Cache-Aware Cyber Defense (ACCD)** architecture to bridge the gap between research ML approaches and production-grade defenses tailored for finance.

We argued that caches are both a source of signal and noise: they reflect behavioral traces that are orthogonal to raw network traffic and can reveal reconnaissance and exfiltration attempts that purposefully avoid volumetric signatures. By designing cache-aware feature extractors (compact, sketch-based, and cache-friendly) and pairing them with an ensemble of streaming ML detectors (supervised, unsupervised, and concept-drift detectors), ACCD achieves a practical balance between detection quality and operational constraints.

Key takeaways:

1. **Instrumentation matters.** Accurate, low-overhead cache instrumentation is feasible and yields signal that complements network and application telemetry. Fixed-size sketches and hashed identifiers preserve privacy and bound resource use.
2. **Modeling should be production-aware.** Compact, cache-friendly models (e.g., quantized tree ensembles) combined with streaming, incremental unsupervised detectors enable fast inference and resilience to concept drift. Lessons from production ML engineering (memory layout, cache-friendly access) apply directly to security models.
3. **Fusion reduces analyst load.** Multi-modal risk scoring that fuses network anomalies with transaction and business context reduces false positives and raises actionable, economically meaningful alerts.
4. **Adversarial resilience is necessary but not absolute.** Adversarial hardening improves robustness but cannot guarantee invulnerability. Defensive designs should assume persistent adversarial pressure and prioritize safe governance, shadow testing, and manual oversight.
5. **DevSecOps integration is essential.** Continuous testing, versioned artifacts, and automated canary deployments are necessary to safely operate ML detectors in production. Automated remediations must be gated behind risk policies to avoid impacting legitimate financial flows.

Limitations: evaluation on public datasets is limited by dataset realism. Real production validation in financial clouds requires partnership with operators and carefully instrumented, privacy-compliant data collection. Moreover, adversarial tests highlight that attackers may adapt to cache-aware defenses, so continuous adaptation and governance are required.

In closing, ACCD provides a practical, production-minded path: instrument caches as first-class telemetry; build compact, streaming, adversarially hardened detectors; fuse signals into business-aligned risk; and operate everything inside a DevSecOps safety envelope. This combined approach increases early detection of stealthy threats while keeping overhead and business impact low — a necessary balance for modern financial cloud infrastructures.

VI. FUTURE WORK

1. **Field piloting**: deploy ACCD in a controlled production pilot with a financial operator to collect real telemetry and refine feature extraction and thresholds.
2. **Federated learning**: explore privacy-preserving federated training across institutions for shared threat intelligence without raw data sharing.
3. **Reinforcement learning for response orchestration**: learn optimal, context-aware response policies that minimize business disruption under attack.
4. **Advanced adversarial defenses**: integrate certified robustness techniques and continual adversarial training specific to cache features.

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

5. **Better benchmarks**: contribute a publicly shareable synthetic benchmark that couples realistic cloud workloads, cache dynamics, and financial transactions to help the community evaluate similar approaches reproducibly.

REFERENCES

1. Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). *The 1999 DARPA off-line intrusion detection evaluation.* Computer Networks, 34(4), 579–595. (archive.ll.mit.edu)
2. 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.
3. Amutha, M., &Sugumar, R. (2015). A survey on dynamic data replication system in cloud computing. International Journal of Innovative Research in Science, Engineering and Technology, 4(4), 1454-1467.
4. 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.
5. 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.
6. 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.
7. Nagarajan, G. (2022). An integrated cloud and network-aware AI architecture for optimizing project prioritization in healthcare strategic portfolios. International Journal of Research and Applied Innovations, 5(1), 6444–6450.https://doi.org/10.15662/IJRAI.2022.0501004
8. 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.
9. Mahoney, M. V., & Chan, P. K. (2003). *An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection.* RAID Workshop. (cs.fit.edu)
10. Sommer, R., &Paxson, V. (2010). *Outside the closed world: On using machine learning for network intrusion detection.* IEEE Symposium on Security and Privacy. (ACM Digital Library)
11. Moustafa, N., & Slay, J. (2015). *UNSW-NB15: a comprehensive data set for network intrusion detection systems.* Military Communications and Information Systems Conference (MilCIS). (UNSW Sites)
12. Chen, T., &Guestrin, C. (2016). *XGBoost: A scalable tree boosting system.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (arXiv)
13. 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.
14. 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.
15. Alexandre, M., et al. (2021). *Systemic risk in financial networks: Data-driven machine learning analysis.* (Working paper / preprint). (Warwick Research Archive Portal)
16. Miao, Y., et al. (2021). *Intelligent task caching in edge cloud via bandit learning.* Sensors (MDPI). (PMC)
17. Fan, X., et al. (2020). *Optimal design of hierarchical cloud–fog–edge computing networks.* Sensors. (MDPI)
18. Biggio, B., &Roli, F. (2018). *Wild patterns: Ten years after the rise of adversarial machine learning.* Pattern Recognition. (iris.unica.it)
19. Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., &Roli, F. (2018). *Adversarial malware binaries: Evading deep learning for malware detection in executables.*arXiv. (arXiv)
20. Choudhary, S., et al. (2020). *Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets for intrusion detection.*Procedia Computer Science. (ScienceDirect)
21. Zoghi, Z., et al. (2021). *UNSW-NB15 dataset analysis and usage guidance.*arXiv. (arXiv)
22. 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.
23. 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.

| ISSN: 2455-1864 | www.ijrai.org| editor@ijrai.org |A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 5, Issue 4, July - August 2022 ||

DOI: 10.15662/IJRAI. 2022.0504005

24. AnujArora, “Securing Multi-Cloud Architectures Using Advanced Cloud Security Management Tools”, INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING, VOL. 7 ISSUE 2 (APRIL- JUNE 2019).
25. Anbazhagan, R. S. K. (2016). A Proficient Two Level Security Contrivances for Storing Data in Cloud.
26. Amuda, K. K., Kumbum, P. K., Adari, V. K., Chunduru, V. K., &Gonepally, S. (2020). Applying design methodology to software development using WPM method.Journal ofComputer Science Applications and Information Technology, 5(1), 1-8.
27. 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.
28. Murugamani, C., Saravanakumar, S., Prabakaran, S., &Kalaiselvan, S. A. (2015). Needle insertion on soft tissue using set of dedicated complementarily constraints. Advances in Environmental Biology, 9(22 S3), 144-149.
29. Roesch, M. (1999). *Snort — Lightweight intrusion detection for networks.* Proceedings of USENIX LISA. (USENIX)