Residual Data Leakage Detection Across Object Storage Lifecycle Transitions

Authors

Lukas Meier[1], Anna Keller[2*]

Affiliations

Department of Computer Science, ETH Zurich, Zurich 8092, Switzerland

***Corresponding author:** anna.keller@gmail.com

Abstract

Cloud object storage is commonly used for application data, archives, logs, backups, analytics outputs, and cross-service data exchange. Although access control is important, data-security risk may also arise from improper lifecycle policies, residual objects, unexpired temporary files, vulnerable processing services, and inconsistent deletion rules. This study proposes a policy-aware vulnerability risk assessment framework for object storage lifecycle and residual data exposure in cloud platforms. The framework evaluates object sensitivity, lifecycle rule completeness, deletion-policy compliance, bucket vulnerability context, temporary URL exposure, processing-service privileges, replication settings, and retention violations. A lifecycle-risk propagation model is used to connect vulnerable processing workloads with generated storage objects and retained data copies. Experiments are conducted on a cloud storage environment containing 38,500 buckets, 126 million object metadata records, 4,760 lifecycle rules, 21,300 temporary access URLs, 7,900 data-processing services, and 31,200 vulnerability records linked to storage-facing workloads. The proposed method identifies 4,180 residual data exposure chains, including expired temporary files, replicated sensitive objects without aligned deletion rules, vulnerable processors writing to public-facing buckets, and long-lived temporary URLs linked to regulated datasets. It consolidates 31,200 vulnerability findings into 4,640 lifecycle-aware remediation units. The storage-policy engine scans 126 million object records in 41.3 minutes and completes daily incremental analysis in 6.8 minutes. After two remediation batches, residual sensitive-object cases decrease by 2.31 million records, and long-lived temporary URLs linked to sensitive data decrease from 3,420 to 870. The median scoring time is 24 ms per bucket-level risk unit. These findings indicate that lifecycle-aware vulnerability assessment can uncover data-security risks that are missed by access-control-only cloud storage reviews.

Keywords: Object storage security; lifecycle policy risk; residual data exposure; cloud vulnerability assessment; policy-aware risk scoring; temporary URL exposure; cloud data governance.

1. Introduction

Cloud object storage is widely used for application files, logs, backups, analytics outputs, and cross-service data exchange. Its scalability, elasticity, and low operational cost make it a core data layer in modern cloud platforms. However, the same features also create data-security risks that go beyond conventional access control. Sensitive data may remain exposed through incomplete lifecycle rules, inconsistent deletion policies, residual temporary objects, long-lived temporary URLs, unsafe replication settings, and vulnerable processing services that continuously write data into storage buckets. Recent research on cloud infrastructure security has shown that vulnerability information, infrastructure context, policy checks, and remediation logic can be integrated into a practical security framework for cloud environments [1]. Related studies have further shown that misconfigured cloud buckets can be discovered at scale and may expose credentials, configuration files, application secrets, and sensitive records [2,3].

Secure object storage management requires more than preventing unauthorized access at a single point in time. Data may be created by applications, transformed by processing services, replicated across regions, shared through temporary URLs, retained for compliance reasons, archived for backup, or deleted according to lifecycle policies [4,5]. During these processes, security exposure may occur when metadata are incomplete, encryption states are inconsistent, retention policies are not enforced, temporary objects are not removed, or deletion verification is absent [6,7]. Data-lifecycle and cloud governance studies have indicated that secure storage should cover data creation, use, sharing, archival, deletion, verification, metadata quality, encryption, and policy enforcement [8,9]. These studies provide useful foundations for data governance, but most of them still treat access control, encryption, deletion, retention, and privacy compliance as separate problems [10,11]. In practical cloud platforms, object storage exposure is often produced by the interaction of multiple factors rather than by an isolated misconfiguration. A bucket with restricted access may still create risk if vulnerable processing services write sensitive temporary files into it, lifecycle rules fail to delete residual objects, temporary URLs remain valid for an excessive period, or replication policies create unmanaged copies in another region [12,13]. Similarly, a deletion policy may appear complete at the bucket level but fail to cover derived files, intermediate results, replicated objects, or backup copies [14,15]. These cases show that storage risk should be evaluated along the object lifecycle and the service chain that produces, transfers, retains, and deletes data. Without such a lifecycle-linked view, security teams may identify exposed buckets or vulnerable services but still fail to understand how residual data exposure is formed and propagated. Vulnerability prioritization methods have also moved beyond severity-only ranking. CVSS provides a technical severity baseline, while EPSS improves exploit-likelihood estimation [16,17]. Context-aware methods further consider asset exposure, business value, environmental conditions, and remediation constraints when ranking vulnerabilities [18,19]. Cloud-native studies also show that vulnerabilities and misconfigurations often appear across containers, deployment files, access policies, service dependencies, and cloud resource links [20,21]. These methods are valuable for improving vulnerability management, but

they are usually designed at the CVE, package, host, image, or asset level. They do not fully include object sensitivity, lifecycle-rule completeness, deletion compliance, temporary URL exposure, replication mismatch, residual copy risk, and the privilege scope of storage-facing processing services. This limitation is important because cloud object storage risk is closely related to data state and lifecycle behavior. A medium-severity vulnerability in a processing workload may become a high-risk issue if the workload has write access to buckets containing sensitive analytics outputs or regulated records [22,23]. A temporary URL may become more serious when it points to objects that should have been deleted under a retention policy. A replication setting may increase exposure when sensitive objects are copied to a region or storage class that lacks equivalent encryption, access review, or deletion controls. Therefore, a practical risk assessment framework should connect vulnerability context with object sensitivity, lifecycle policy, replication behavior, temporary sharing, and deletion integrity. Such a framework can help distinguish isolated technical findings from storage-exposure chains that may cause real data-confidentiality and compliance impacts.

To address this gap, this study develops a policy-aware vulnerability risk assessment framework for object storage lifecycle and residual data exposure in cloud platforms. The framework evaluates object sensitivity, lifecycle-rule completeness, deletion-policy compliance, bucket vulnerability context, temporary URL exposure, processing-service privilege, replication settings, retention violations, and residual copy risk. A lifecycle-risk propagation model is further designed to link vulnerable processing workloads with generated objects, replicated copies, temporary files, and retained data. The purpose is to identify lifecycle-linked exposure chains rather than merely rank vulnerable buckets or storage-facing services. By grouping raw findings into actionable remediation units, the proposed framework helps security teams prioritize storage risks that may affect data confidentiality, deletion integrity, and policy compliance. This study therefore provides a more operationally useful basis for securing object storage in large-scale cloud platforms where data are continuously generated, shared, replicated, archived, and deleted.

2. Materials and Methods

2.1 Sample and Study Environment Description

This study was conducted in a cloud object-storage environment used for application data, logs, backups, analytics outputs, temporary files, and cross-service data exchange. The sample included 38,500 buckets, 126 million object metadata records, 4,760 lifecycle rules, 21,300 temporary access URLs, 7,900 data-processing services, and 31,200 vulnerability records linked to storage-facing workloads. These records were collected from bucket configurations, object metadata tables, lifecycle policy logs, temporary URL records, replication settings, deletion-policy records, and vulnerability reports from processing services. Daily storage-policy snapshots were used to cover routine bucket updates, object refreshes, lifecycle-rule changes, URL renewal events, replication changes, and workload upgrades. The study objects included public and private

buckets, sensitive objects, temporary objects, replicated copies, retained backup objects, processing-service outputs, and objects generated by vulnerable workloads. Each bucket-level risk unit was described by object sensitivity, lifecycle-rule completeness, deletion compliance, temporary URL exposure, replication state, retention violation, processing-service privilege, and related vulnerability context.

2.2 Experimental Design and Control Experiments

The experiment examined whether lifecycle and residual-data context improved vulnerability-risk assessment in cloud object storage. The experimental group used the policy-aware lifecycle-risk assessment method. This method combined object sensitivity, lifecycle rule completeness, deletion-policy compliance, bucket vulnerability context, temporary URL exposure, processing-service privilege, replication setting, and retention violation into one bucket-level risk score. Four control groups were used for comparison: CVSS-only ranking, access-exposure ranking, object-sensitivity ranking, and lifecycle-rule ranking. These baselines were selected because they represent common methods in storage-security review. CVSS reflects technical severity, access-exposure ranking reflects bucket reachability, object-sensitivity ranking reflects data value, and lifecycle-rule ranking reflects policy completeness. All methods were tested on the same bucket inventory, object metadata records, lifecycle-policy table, temporary URL list, processing-service map, replication records, and vulnerability dataset. The comparison focused on residual exposure-chain detection, remediation grouping, sensitive-object reduction, long-lived URL reduction, scanning cost, and score stability.

2.3 Measurement Methods and Quality Control

Bucket and object metadata were extracted from storage catalogs, bucket policies, object tags, lifecycle rules, deletion logs, replication configurations, temporary access URL records, and processing-service access records. Object sensitivity was assigned according to storage labels, including public, internal, personal, financial, operational, and regulated data. Lifecycle rule completeness was measured by checking expiration rules, transition rules, delete-marker handling, non-current version cleanup, and temporary-object removal. Deletion-policy compliance was measured by comparing object age, object type, retention period, and required deletion deadline. Temporary URL exposure was measured by checking URL validity period, access scope, linked object sensitivity, and external reachability. Processing-service privilege was measured by reviewing bucket read, write, delete, list, replicate, and policy-management permissions. Vulnerability records were mapped to storage-facing workloads by service name, container image digest, package version, connector version, and bucket-write relationship. Quality control included duplicate removal, link validation, and manual review. Duplicate records were removed by CVE ID, package version, image digest, service ID, and bucket ID. Object-to-service and replication links were checked against storage access logs and configuration snapshots to remove stale

mappings. In addition, 5% of high-risk residual exposure chains were reviewed by two cloud data-security engineers, and inconsistent labels were corrected by checking deletion logs, lifecycle rules, URL records, and bucket policies.

2.4 Data Processing and Model Formulas

All variables were converted into bucket-level or object-chain indicators before scoring. Binary variables were used for public bucket exposure, long-lived temporary URL, missing encryption, retention violation, incomplete lifecycle rule, and vulnerable processing-service output. Ordered variables were used for object sensitivity, processing-service privilege, replication risk, deletion-policy severity, and vulnerability severity. Continuous variables, including object age, URL lifetime, number of retained objects, number of replicated copies, and number of vulnerability records, were normalized to the range of 0–1. The risk score of bucket-level unit i was calculated as $R_i=10\times \sum_{k=1}^{m} w_k x_{ik}+\alpha L_i+\beta U_i$, where R_i is the final risk score, x_{ik} is the normalized value of factor k,w_k is the factor weight,L_i is the lifecycle-risk propagation term, U_i is the temporary URL exposure term, and α and β control the effects of lifecycle violation and URL exposure. Higher weights were assigned to regulated objects, unexpired temporary files, long-lived temporary URLs, vulnerable processors, public-facing buckets, and replicated objects without aligned deletion rules. The remediation grouping rate was calculated a $RGR=1-\frac{G}{V}$, where G is the number of lifecycle-aware remediation units and V is the number of raw vulnerability findings. A larger RGR indicates a stronger reduction of repeated storage-related findings.

2.5 Evaluation Protocol and Statistical Analysis

The method was evaluated from six aspects: residual exposure-chain detection, remediation grouping, sensitive-object reduction, long-lived URL reduction, scanning cost, and score stability. Residual exposure-chain detection was measured by counting chains that linked vulnerable processing services, retained objects, temporary URLs, replication settings, and sensitive data labels. Remediation grouping was assessed by comparing raw vulnerability findings with lifecycle-aware remediation units after bucket, object, service, and policy records were grouped. Sensitive-object reduction was measured by counting residual sensitive-object records before and after remediation batches. Long-lived URL reduction was measured by tracking temporary access URLs linked to sensitive data. Scanning cost was measured by the total time required to scan 126 million object records and the median scoring time per bucket-level risk unit. Score stability was assessed across daily storage-policy snapshots by calculating score changes after routine bucket updates, lifecycle-policy changes, object refreshes, and workload upgrades. The ranked results were compared with manually reviewed high-risk exposure chains. Precision, recall, and F1-score were used to measure ranking quality, where $F1=\frac{2\times Precision\times Recall}{Precision+Recall}$. The Wilcoxon signed-rank test was used to compare paired ranking results, and $p<0.05$ was considered statistically significant.

3. Results and Discussion

3.1 Residual Exposure-Chain Identification

The policy-aware framework identified 4,180 residual data exposure chains from 31,200 vulnerability records related to storage-facing workloads. Most chains were linked to expired temporary files, replicated sensitive objects with inconsistent deletion rules, vulnerable processors writing to public-facing buckets, and long-lived temporary URLs connected to regulated datasets [24,25]. These results show that object-storage risk cannot be assessed only by bucket permission status or CVE severity. Some high-severity workload findings had limited storage impact because they wrote only to short-retention internal buckets. In contrast, several medium-severity vulnerabilities became high-priority risks when the affected services generated sensitive objects, used long-lived URLs, or wrote data into buckets with incomplete lifecycle controls. This pattern is consistent with data-lifecycle security studies, which suggest that data protection should cover collection, processing, storage, sharing, archiving, and destruction rather than access control alone, as shown in Fig. 1.

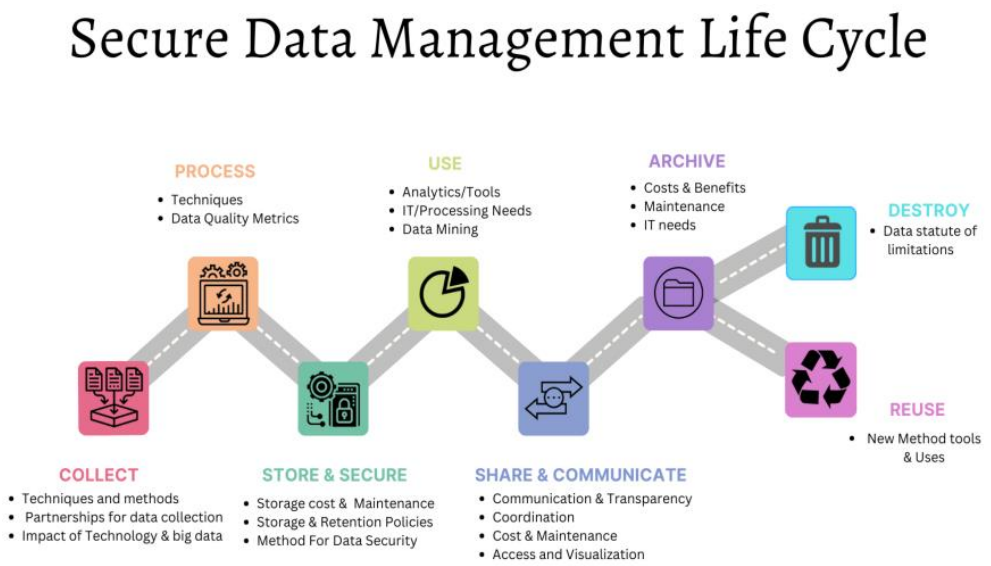

Fig. 1. Secure data life cycle covering collection, processing, storage, sharing, archiving, and destruction.

3.2 Comparison with Baseline Ranking Methods

The proposed method provided a more practical remediation order than CVSS-only ranking, access-exposure ranking, object-sensitivity ranking, and lifecycle-rule ranking. CVSS-only ranking reflected technical severity, but it could not determine whether a vulnerable workload generated retained sensitive objects. Access-exposure ranking detected public-facing buckets, but it missed private buckets with long-lived temporary URLs or inconsistent replication rules. Object-sensitivity ranking captured data value, but it did not include vulnerable processing services or lifecycle-rule gaps. Lifecycle-rule ranking reflected deletion-policy completeness, but it did not determine whether retained objects were sensitive or generated by vulnerable workloads.

In comparison, the proposed framework combined vulnerability evidence with lifecycle and residual-data context. This result extends cloud storage security assessment from isolated permission or encryption checks to storage-object exposure chains [26,27]. The storage-protection logic used for comparison is summarized in Fig. 2.

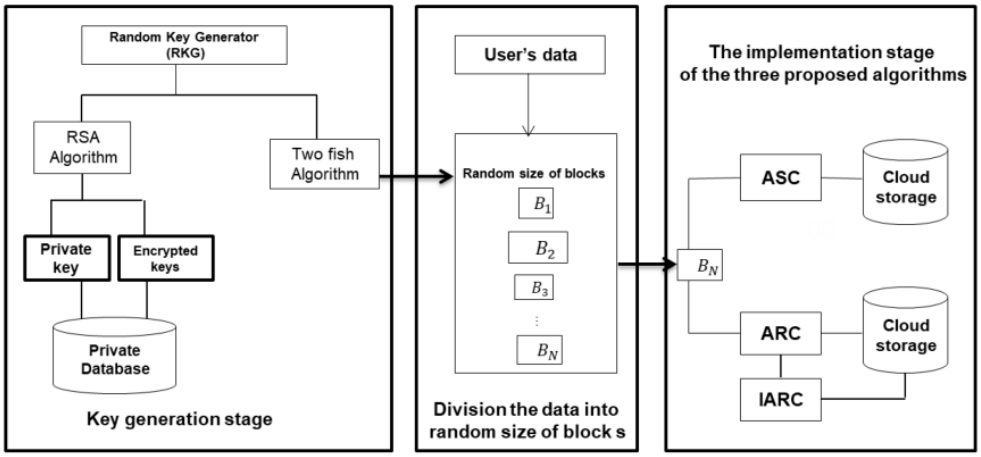

Fig. 2. Automated encryption and decryption framework for cloud data protection.

3.3 Remediation Grouping and Exposure Reduction

The framework consolidated 31,200 vulnerability findings into 4,640 lifecycle-aware remediation units. This grouping reduced repeated findings from storage-facing workloads, connector versions, processing images, and bucket-level policy records while preserving the affected object chain. After two remediation batches, residual sensitive-object cases decreased by 2.31 million records, and long-lived temporary URLs linked to sensitive data decreased from 3,420 to 870. These results indicate that lifecycle-aware grouping can reduce security workload and improve remediation priority. Traditional vulnerability management usually groups findings by CVE, package, image, or host. This approach is useful for patch tracking, but it is less suitable for object storage security. A single vulnerable processor may write to several buckets, and each bucket may contain different object classes, lifecycle rules, replication settings, and temporary URL states [28,29]. The proposed grouping method therefore provides a clearer repair unit by linking each vulnerability to generated objects, lifecycle violations, URL exposure, replication mismatch, and affected data class.

3.4 Scanning Cost, Stability, and Limitations

The storage-policy engine scanned 126 million object records in 41.3 min and completed daily incremental analysis in 6.8 min. The median scoring time was 24 ms per bucket-level risk unit, indicating that the method can support large-scale storage review with acceptable processing cost. These results are important because object storage systems often contain many small files, retained versions, replicated copies, and temporary outputs. Access-control-only reviews may miss this residual exposure because they do not track object age, URL lifetime, deletion deadlines,

or workload-to-object links [30]. However, the framework depends on accurate object metadata, complete lifecycle rules, reliable URL records, current replication settings, and correct workload-to-bucket mappings. If deletion logs are incomplete or object labels are outdated, residual exposure may be underestimated. Future work should include runtime object-access evidence, cross-cloud storage validation, and automated deletion-proof checks to improve lifecycle-aware risk assessment.

4. Conclusion

This study developed a policy-aware vulnerability risk assessment framework for object storage lifecycle and residual data exposure in cloud platforms. The framework combined object sensitivity, lifecycle-rule completeness, deletion-policy compliance, bucket vulnerability context, temporary URL exposure, processing-service privilege, replication settings, and retention violations into one assessment model. The results showed that the method detected 4,180 residual data exposure chains from 31,200 vulnerability records and grouped them into 4,640 lifecycle-aware remediation units. After two remediation batches, residual sensitive-object cases decreased by 2.31 million records, and long-lived temporary URLs linked to sensitive data decreased from 3,420 to 870. The storage-policy engine scanned 126 million object records in 41.3 min and completed daily incremental analysis in 6.8 min, with a median scoring time of 24 ms per bucket-level risk unit. These results indicate that lifecycle-aware assessment can identify storage risks that may be missed by access-control-only reviews. The main contribution of this study is that it links vulnerable processing workloads with generated objects, retained copies, temporary URLs, replication rules, and deletion-policy gaps. This design provides a clearer basis for prioritizing risks that may affect data confidentiality and deletion integrity. The framework can support cloud storage governance, residual-data cleanup, lifecycle-policy review, and vulnerability remediation planning. However, its accuracy depends on complete object metadata, reliable lifecycle rules, current URL records, and correct workload-to-bucket mappings. Future work should include runtime object-access evidence, cross-cloud storage validation, and automated deletion-proof checks to improve the stability and wider use of the method.

References

1. Shao, W. (2026). Policy-Driven Vulnerability Risk Quantification framework for Large-Scale Cloud Infrastructure Data Security. arXiv preprint arXiv:2604.06252.

2. El Yadmani, S., Gadyatskaya, O., & Zhauniarovich, Y. (2025, May). The File That Contained the Keys Has Been Removed: An Empirical Analysis of Secret Leaks in Cloud Buckets and Responsible Disclosure Outcomes. In 2025 IEEE Symposium on Security and Privacy (SP) (pp. 3180-3198). IEEE.

3. Gui, H., Wang, B., Lu, Y., & Fu, Y. (2025). Computational Modeling-Based Estimation of Residual Stress and Fatigue Life of Medical Welded Structures.

4. Qi, C., & Qiao, X. (2026, May). Design Patterns for Multi-Agent Systems in Production. In 2026 International Conference on Intelligent Systems, Automation and Control (ISAC) (pp. 198-202). IEEE.
5. Ige, A. B., Chukwurah, N., Idemudia, C., & Adebayo, V. I. (2024). Managing data lifecycle effectively: Best practices for data retention and archival processes. International Journal of Engineering Research and Development, 20(8), 199-207.
6. Ahanger, A. S., Masoodi, F. S., Khanam, A., & Ashraf, W. (2024). Managing and securing information storage in the Internet of Things. In Internet of Things vulnerabilities and recovery strategies (pp. 102-151). Auerbach Publications.
7. Su, D., & Wu, Y. (2026). Multilevel Growth and Social Network Modeling for Functional Communication Enhancement in Students with Intellectual Disabilities.
8. Ma, Y. (2026). Industrial Financial Risk Prediction Model Based on Graph Neural Network and Knowledge Graph Inference. Journal of Circuits, Systems and Computers, 35(16), 2650103.
9. Gao, G., Gao, R., Lu, C., Gao, R., & Kuang, Y. (2026, March). Security Governance Methods and Quantitative Evaluation for Enterprise SMS and Verification Code Systems. In 2026 International Conference on Generative Artificial Intelligence and Information Security (GAIIS) (pp. 455-458). IEEE.
10. Islam, A. (2024). Data Governance And Compliance In Cloud-Based Big Data Analytics: A Database-Centric Review. Academic Journal on Innovation, Engineering & Emerging Technology, 1(1), 53-71.
11. Li, Y., & Liu, S. (2026, May). A Study on Dynamic Optimization of Alerting Policies and Multi-Agent Decision-Making Mechanisms in Cloud Environments. In 2026 7th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT) (pp. 703-706). IEEE.
12. Ganpathy, S. (2026). Automated Governance and Protection of Non-Human Identities in Cloud IAM. Emerging Innovations Society: Science, Technology, and Social Studies, 08-20.
13. Liang, R., Fan, F., Liang, Y., & Li, S. (2025). Constructing an Adaptive Optimization Model for Ribbon Recommendation and Interface for User Habits.
14. Zhu, W., & Yang, J. (2025). Causal Assessment of Cross-Border Project Risk Governance and Financial Compliance: A Hierarchical Panel and Survival Analysis Approach Based on H Company's Overseas Projects.
15. Scope, N., Rasin, A., Lenard, B., Wagner, J., & Heart, K. (2022). Purging compliance from database backups by encryption. Journal of Data Intelligence, 3(1).
16. Erkalkan, E. (2026). Queueing Analysis by Simulation of Risk Duration Prioritized Patch Scheduling Under Gated Windows. Sakarya University Journal of Computer and Information Sciences, (Advanced Online Publication), 739-753.
17. Liu, H., Xu, D., Ma, Q., Xu, S., & Qiu, D. (2026). Memory Poisoning Propagation and Repair Mechanism in Multi-Agent Collaborative Environments.

18. Yin, J., Rao, H., & Huang, Y. (2026, March). Dynamic Modeling and Heterogeneity Analysis of Platform User Behavior Time Series. In 2026 International Conference on AI in Education Technology and Applications (AIETA) (pp. 30-33). IEEE.
19. Ahmadi Mehri, V., Arlos, P., & Casalicchio, E. (2022). Automated context-aware vulnerability risk management for patch prioritization. Electronics, 11(21), 3580.
20. Yang, J. (2026). Computational Analysis of How Digital Non-Clinical Communication Tools Influence Social Participation Among Older Adults in Community-Based Elderly Care. Available at SSRN 6682838.
21. Deng, X., & Yang, J. (2026). Design of a Quantitative Futures Trading Model Incorporating Multimodal Feature Fusion and Tail Risk Control. Available at SSRN 6702398.
22. Kearns, E. J., Saah, D., Levine, C. R., Lautenberger, C., Doherty, O. M., Porter, J. R., ... & Chishtie, F. (2022). The construction of probabilistic wildfire risk estimates for individual real estate parcels for the contiguous United States. Fire, 5(4), 117.
23. Zhang, Z. (2026). A Study on Multimodal Product Understanding and Assembly Guidance Optimization in e-commerce for Complex Consumer Goods. Available at SSRN 6734559.
24. Innocenti, T. (2025). Identity Is the New Perimeter: Strengthening Authentication Protocols Through Standard Improvement (Doctoral dissertation, Northeastern University).
25. Zhang, Z., Gao, Y., & Tong, Y. (2026). Semantic-Temporal Graph Learning for Demand Forecasting and Resource Allocation in Public Information Systems. Available at SSRN 6792279.
26. Vergadia, P. (2022). Visualizing Google Cloud: 101 Illustrated References for Cloud Engineers and Architects. John Wiley & Sons.
27. Zhao, J., Fan, J., & Li, L. (2026). A Study on an Explainable Causal-Enhanced LLM Agent for Predicting the Forming Quality of Automotive Component Materials.
28. Xu, T., Zhang, J., & Zhu, W. (2026). Fairness-Accuracy Trade-offs and Calibration Mechanisms in Machine Learning-Based Subprime Credit Scoring. Available at SSRN 6893759.
29. Hindi, M., Mahmood, Y., Mohammed, L., Bouktif, S., & Mediani, M. (2026). Coding Agents in the Wild: Failure Modes and Rejection Patterns of AI-Generated Pull Requests. IEEE Access.
30. Fang, X., Yang, Y. H., & Wang, S. (2026). Energy-Efficient Context-Aware Multimodal AI Inference at the Edge.