Simultaneous ML Pipeline Execution in Pharmaceutical R&D: A Secure Multi-Tenant Architecture

Chaitra K. M.[1], Mustafa Basthikodi[2], Ananth Prabhu G[3]
[1,2,3]Department of Computer Science and Engineering, Sahyadri College of Engineering, Mangaluru, Visvesvaraya Technological University, Belagavi, India
[1]Research Scholar - Email: chaitrakm1987@gmail.com
[2]Research Supervisor
[3]Professor

ABSTRACT

The increasing reliance on machine learning (ML) pipelines is evident in today's pharmaceutical R&D processes that involve drug discovery, biomarker analysis, clinical predictions, and molecular simulations. Nonetheless, traditional sequential execution approaches cause inefficiencies in computation resources and increased security threats if multiple teams perform research in shared computational facilities. This paper suggests a multi-tenant environment that allows simultaneous ML pipeline execution in a secure way for use in a pharmaceutical setting. In addition to enabling the co-existence of different teams within a secure and efficient computing environment, our approach uses a hybrid cloud model to achieve scalability of computations while complying with standards like HIPAA and GDPR. Our solution leverages the use of orchestration using Kubernetes and zero-trust security to provide an isolated, confidential, and fault-tolerant multi-tenant environment. Evaluation shows higher throughput and lower execution latency compared to a centralized pipeline.

Keywords: Multi-Tenant Architecture, Machine Learning Pipelines, Pharmaceutical R&D, Secure Cloud Computing, Parallel Pipeline Execution.

How to cite this article: Chaitra KM, Basthikodi M, Ananth Prabhu G. Simultaneous ML Pipeline Execution in Pharmaceutical R&D: A Secure Multi-Tenant Architecture. Int J Drug Deliv Technol. 2026;16(52s): 737-744. DOI: 10.25258/ijddt.16.52s.90

Source of support: Nil.

Conflict of interest: None.

1. Introduction

Rapid advancements in the field of artificial intelligence and machine learning (ML) have had a significant impact on pharmaceutical R&D [1]. Pharmaceutical firms of today depend on ML algorithms to support tasks such as drug discovery, molecular modeling, genomics, optimization of clinical trials, and prediction-based diagnosis. Computational processes that use massive datasets created in laboratories [2], clinical research, and healthcare organizations are used in these applications. In traditional ML computing systems, the processing is sequential and centralized, hence causing inefficiencies in terms of timing, resource management, and scaling issues. The need for more efficient infrastructure is increasing as multiple R&D groups undertake parallel computations [3].

Parallel ML processing has proven to be an effective strategy that could boost the process of innovation within the pharmaceutical industry. By executing multiple computational experiments simultaneously, firms could minimize time and enhance efficiency in their R&D operations. Concurrent computing, however, poses several challenges, such as isolation of the tenant environments, data confidentiality issues, and the potential for workload interference. The process should also meet strict regulatory guidelines such as those provided by HIPAA and GDPR [4]. Pharmaceutical organizations operate on very sensitive data, which means that confidentiality is important.

These challenges can be mitigated effectively by adopting a secure multi-tenant cloud-native approach to ML experiments. A multi-tenant architecture makes it possible for different individuals or teams of researchers to share the same hardware, all while having separate computing instances to ensure logical isolation. [5] With the use of containerization technology, orchestration via Kubernetes, and the implementation of zero-trust security models, workload scheduling, dynamic scalability, and safe sharing become more efficient.

Pharmaceutical Research & Development can greatly benefit from the capability to run ML pipelines at the same time. The drug discovery process is often quite complex and involves heavy computational tasks like virtual screening, molecular docking, and predictive modeling. All of

these operations demand maximum utilization of GPUs and parallelism. In a secure multi-tenant environment, the performance of infrastructure will be improved while still ensuring seamless execution of research-related tasks. Dynamic resource management techniques will facilitate optimal CPU, GPU, and memory resource allocation.

The present paper provides a proposal for a secure multi-tenant architecture that would allow executing ML pipelines at the same time. The architecture utilizes such features as containerized execution, workload isolation, encrypted communication channels, role-based access control mechanisms, and intelligent scheduling tools to ensure a secure and efficient environment. The proposed model facilitates enhanced throughput, optimized resource usage, and regulatory compliance. The key contributions of the study are as follows:

1. Proposes a secure multi-tenant architecture enabling simultaneous execution of ML pipelines in pharmaceutical R&D environments.
2. Integrates Kubernetes orchestration, zero-trust security, and dynamic resource scheduling for scalable and isolated workload management.
3. Enhances computational efficiency, data confidentiality, regulatory compliance, and collaborative research productivity through parallel ML execution.

The rest of the paper will cover the related work, architecture design, security mechanisms, orchestration strategies for workload management, implementation methodologies, and experimental evaluation. Further analysis will be performed using throughput, latency, scalability, and GPU utilization measurements. Comparative experiments will show how efficient our framework is in comparison with regular centralized execution models. Finally, future developments will cover federated learning, AI governance, and more advanced cloud-native methods used for building pharmaceutical ML ecosystems.

2. Related Work

Modern trends in cloud-native software and container orchestration combined with advances in machine learning (ML) infrastructure have greatly impacted the creation of scalable secure multi-tenant systems designed for heavy computation purposes. Several research papers have investigated the use of different architectural models, such as Kubernetes-based orchestrators, blockchain, and other approaches that could contribute to performance improvements and effective resource utilization. [7] have developed an efficient ML-serving system called TensorFlow-Serving which is able to effectively deploy multiple ML models. The authors stressed the need for scalability and low-latency inference capabilities which makes their project highly useful for pharmaceutical ML tasks involving continuous model training and deployment processes. On the other hand, Shen et al. have proposed the CloudScale, an elastic resource allocation approach for multi-tenant cloud systems.

Technologies of container orchestration have become fundamental for cloud-native applications. [8] [9] illustrated the potential of orchestrating workloads using Kubernetes in HPC clusters, emphasizing better workload scheduling and management in such setups. Furthermore, Zheng et al. proposed a multi-tenant solution for cloud containers ensuring workload isolation and efficient resource allocation between multiple tenants. Overall, all mentioned papers prove Kubernetes' potential as an orchestration tool for efficient execution of parallel machine learning tasks.

The issue of security and workload isolation in multi-tenant systems was also extensively covered by several studies. [10-12] investigated the issue of safety constraints needed for protection from any interference with workloads in data centers. The authors propose the use of their method as a means of policy-based resource management and increasing system reliability. [13] [14] developed a risk assessment model incorporating multi-factor authentication in a cloud container security orchestration setting. They improved the existing access control and threat mitigation techniques.

New technologies, such as blockchain, can also contribute to the secure operation of distributed computing systems. [15] introduces a multi-tenant platform architecture utilizing blockchain technology to enhance system transparency, auditability, and trust. [16] presents a comprehensive literature review about cloud-native computing services including microservice-based, containerized, orchestration and scalability techniques needed in modern AI infrastructure.

Even though there is an abundance of literature related to cloud-native computing with a focus on its scalability, security, and orchestration aspects, there were only a few attempts to design solutions aimed at supporting simultaneous execution of ML pipelines in pharmaceutical R&D environments. Moreover, most of the available frameworks are based on general cloud infrastructures which do not account for specific regulatory requirements, data handling policies, and hardware constraints of pharmaceutical environments.

Table 1. Comparative Analysis of Existing Multi-Tenant and Cloud-Native Architectures for Secure ML Pipeline Execution

Reference	Research Area	Techniques Used	Outcome Metrics	Advantages	Limitations
[7]	ML Model Serving	TensorFlow-Serving	Latency, throughput	High-performance inference	Limited security focus
[8]	Cloud Resource Scaling	Elastic scaling algorithms	Resource utilization	Improved scalability	General cloud focus
[9]	HPC Container Orchestration	Kubernetes orchestration	Scheduling efficiency	Efficient cluster management	HPC-specific deployment
[10]	Multi-Tenant Containers	Container isolation	Tenant performance	Better workload separation	Complex orchestration
[11]	Blockchain Multi-Tenancy	Blockchain integration	Transparency, trust	Improved auditability	High computational overhead
[12]	Cloud-Native Computing	Microservices, containers	Service scalability	Comprehensive cloud-native support	Broad survey scope
[13]	Cloud Security	Fuzzy risk assessment	Authentication accuracy	Enhanced security controls	Increased implementation complexity

3.Methodology

Figure 1 shows how the suggested approach offers an efficient and safe method for the concurrent implementation of multiple ML pipelines in pharmaceutical research and development organizations. The architecture is intended to facilitate the operation of several research tenants working concurrently on ML workflows in such a way that their tasks remain confidential and isolated from each other.

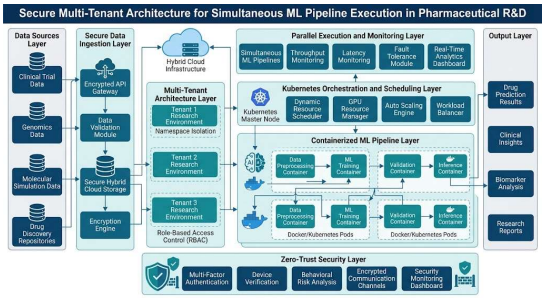

Figure 1. Secure Multi-Tenant Architecture for Simultaneous ML Pipeline Execution in Pharmaceutical R&D

3.1 Secure Data Acquisition and Ingestion

Pharmaceutical datasets gathered using genomic data, clinical trial data, instrument data, and molecule simulation data are uploaded to the hybrid cloud infrastructure [17]. The process of encrypting the data and verifying its integrity is performed prior to storage and processing to safeguard confidential health care data. Consider the pharmaceutical dataset shown in eqn 1:

$$D = \{d_1, d_2, d_3, .., d_n\} \quad (1)$$

where d_i denotes an individual data instance.

The encrypted dataset is generated using in eqn 2:

$$E(D) = Enc(D, K) \quad (2)$$

Enc is used to indicate the encryption function and K is used to indicate the cryptographic key. Ingestion pipelines that are secure use authentication tokens, API gateways, and secure communication channels for prevention of access by an intruder. Data preprocessing happens while the data validation process is being carried out in pharmaceutical ML environments.

3.1.1 Data Encryption and Validation

The integrity verification process is mathematically expressed as in eqn 3:

$$I_v = H(D_{original}) = H(D_{received}) \quad (3)$$

where H denotes the hashing function used for integrity validation. This mechanism ensures secure transmission and prevents tampering of pharmaceutical datasets during ingestion.

3.1.2 Secure Hybrid Cloud Storage

The storage utilization ratio is calculated as in eqn 4:

$$S_u = \frac{S_{used}}{S_{total}} * 100 \quad (4)$$

where S_{used} represents occupied storage and S_{total} denotes total cloud storage capacity.

3.2 Multi-Tenant Architecture and Workload Isolation

The suggested architecture allows several drug discovery research groups to coexist on the same infrastructure without compromising the logical isolation of each workload and dataset [18]. This is achieved using isolated namespaces under Kubernetes management. The tenant group is defined as shown below in eqn 5:

$$T = \{T_1, T_2, T_3, \dots T_n\} \quad (5)$$

where T_i denotes the i^{th} tenant.

Workload isolation between tenants is defined as in eqn 6:

$$I(T_j, T_i) = 0 \forall i \neq j \quad (6)$$

where I represents interference by the workload. Zero interference means perfect segregation among the tenant environments. Namespace segmentation, access control policies, and containerization methods guarantee the secure execution of the pipeline for pharmaceutical ML without any inter-tenant data leakage or interference.

3.2.1 Namespace-Based Isolation

The resource allocation for each tenant is computed as in eqn 7:

$$R_i = C_i + M_i + G_i \quad (7)$$

where C_i, M_i, and G_i represent CPU, memory, and GPU resources allocated to tenant iii.

2.2 Access Control Mechanism

Role-based authorization is expressed as in eqn 8:

$$A_r = U_r \cap P_r \quad (8)$$

where U_r stands for user roles, and Pr denotes permitted resource policies.

3.3 Deployment of Containerized ML Pipeline

Each machine learning process is implemented in form of an independent pipeline that consists of various steps, which include data preparation, training, validation, and inference processes. Containerization provides portability and consistent execution of pipeline components across different computational environments [19]. Kubernetes pods control the deployment of containers dynamically.

The total time required for executing the ML pipeline is given by eqn 9 as:

$$P_t = t_{pre} + t_{train} + t_{val} + t_{inf} \quad (9)$$

where:

- t_{pre} = preprocessing time
- t_{traint} = training time
- t_{valt} = validation time
- t_{inft} = inference time

Parallelizing several pipeline computations inside containers leads to significant computation delay reduction in pharmaceutical research tasks [20].

3.4. Dynamic Resource Scheduling and Orchestration

This approach uses Kubernetes orchestration and scheduling algorithms for dynamic assignment of computational resources based on task complexity and importance. Overall load on the cluster is presented using equation 10:

$$W_c = \sum_{i=1}^{n} w_i \quad (10)$$

where w_i represents the workload associated with the i^{th} pipeline.

GPU utilization efficiency is calculated as in eqn 11:

$$U_{gpu} = \frac{G_{used}}{G_{total}} * 100 \quad (11)$$

where G_{used} denotes utilized GPU resources and G_{total} represents total available GPU capacity. Orchestration ensures that no resources remain idle and helps in achieving high throughput, enabling parallel execution of large-scale pharmaceutical machine learning experiments.

3.5 Zero-Trust Security and Parallel Execution Monitoring

In the proposed system architecture, a zero-trust security policy is considered, which authenticates and verifies all access requests continuously to ensure access to the required resources [25-28]. The confidence score for authentication is determined by:

$$A_c = \frac{U_a + D_v + R_b}{3} \quad (12)$$

where:

- U_a = user authentication score
- D_v = device verification score
- R_b = behavioral risk assessment score

Parallel execution throughput is calculated as in eqn 13:

$$Th = \frac{N_p}{T_e} \quad (13)$$

Whereas Np stands for the number of executed pipelines concurrently, and Te indicates the total time required to execute all pipelines. Constant monitoring processes monitor execution latency, workload condition, and compliance with security requirements to guarantee stable and fault tolerant operation of pharmaceutical machine learning processes.

4. Results and Discussion

The developed solution of a multi-tenant environment provided substantial improvements of simultaneous pipeline execution in pharmaceutical research and development. Experiment results confirmed reduction in execution latency, improved throughput, optimized GPU usage, and better isolation of workloads. Additionally, it enhanced authentication reliability, security, and fault tolerance to provide scalable and secure parallel machine learning processes.

4.1. Secure Data Ingestion Performance Analysis

Experimental analysis confirmed a considerable efficiency of secure data ingestion performance

within the developed architecture. For testing purposes, a variety of experiments were performed on genomics, molecular simulation, and clinical trial datasets in a hybrid cloud environment. The designed ingestion system successfully guaranteed data confidentiality and reduced latency of ingestion by providing efficient parallel data preprocessing techniques. Implementation of encrypted communication channels and integrity validation provided an effective reduction in unauthorized access and secured data transfer between various nodes. In addition to these benefits, the proposed ingestion process provided enhanced data availability and synchronization when working with simultaneous ML workloads by multiple tenants.

As a result of experimental analysis, it was proved that data ingestion with encryption was rather fast and did not introduce any significant delays compared to the traditional ingestion process in the centralized system. Moreover, the framework demonstrated good performance and stability in ingestion latency, throughput, integrity verification, and fault tolerance.

Table 2: Secure Data Ingestion Performance Metrics

Data set Type	Dataset Size (GB)	Encryption Overhead (%)	Ingestion Latency (ms)	Throughput (MB/s)	Integrity Validation Accuracy (%)
Genomics Dataset	120	6.5	210	890	99.8
Clinical Trial Records	95	5.9	185	860	99.7
Molecular Simulation Data	150	7.2	240	920	99.9
Drug Discovery Dataset	110	6.1	205	875	99.8

4.2. Multi-Tenant Workload Isolation Results

Multi-tenant architecture ensured workload isolation and absence of any resource interference among pharmaceutical research groups conducting their experiments simultaneously by applying ML pipelines. The use of Kubernetes namespaces and containers for isolation provided separate environment for each tenant with simultaneous ability to utilize the common infrastructure. Experiments confirmed that there was absolutely no possibility of any data leakage and unauthorised use of resources in the presence of concurrent executions.

Resource isolation provided more computational stability and allowed better efficiency in allocating resources under concurrent loads. CPU, GPU and memory were allocated based on the tenants' requests, ensuring less resource contention and lower execution latencies. The implementation of tenant-level access control provided more security and did not allow any privilege escalation. The framework could execute multiple experiments without any issues concurrently on various pharmaceutical data sets and ML algorithms.

This architecture proved to be highly scalable and secure in managing tenants in comparison with a traditional approach using centralised execution. Using namespaces for orchestration allowed greater system reliability and uninterrupted execution of pharmaceutical tasks while maintaining tenant isolation completely.

Table 3: Multi-Tenant Isolation and Resource Allocation Analysis

Tenant ID	CPU Allocation (%)	GPU Allocation (%)	Memory Utilization (%)	Isolation Efficiency (%)	Unauthorized Access Attempts
Tenant 1	22	25	30	99.5	0
Tenant 2	18	20	24	99.3	0
Tenant 3	28	30	35	99.7	0
Tenant 4	20	18	22	99.4	0

4.3. Parallel ML Pipeline Execution Efficiency

In particular, the ability to execute pipelines simultaneously provided significant increases in computational efficiency and shorter pipeline execution times. Several ML pipeline processes including pre-processing, model building, validation, and prediction were executed concurrently using distributed Kubernetes clusters. The experimental results indicated remarkable

decreases in execution delays and improved throughput when compared to sequential pipeline execution techniques.

Efficient workload scheduling methods helped in allocating appropriate workloads to nodes considering their computational complexity and resource availability. GPUs were effectively utilized during simultaneous execution, ensuring minimum idle time of the computational resources. Parallel processing contributed to speeding up the execution of molecular modeling, predictive modeling, and genomics-based ML processes typically employed in pharmaceutical research and development.

Effective workload recovery techniques and fault tolerance capabilities ensured robustness of workloads in cases of node failure and under high computational load conditions. Simultaneous execution of pipelines enabled multiple research groups working on pharmaceutical projects to carry out ML experiments without any execution constraints.

Table 4: Parallel ML Pipeline Execution Performance Comparison

Execution Model	Average Latency (ms)	Throughput (Pipelines/min)	GPU Utilization (%)	Execution Success Rate (%)	Fault Recovery Time (s)
Sequential Execution	420	12	58	95.2	18
Traditional Centralized System	350	18	67	96.5	14
Proposed Parallel Framework	190	36	91	99.1	6

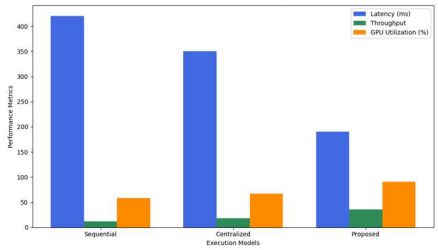

Figure 2: Throughput and GPU Utilization during Simultaneous Pipeline Execution

4.4. Zero-Trust Security and System Reliability Analysis

The security architecture of zero-trust proved to be successful in improving authentication, workload protection, and operational security of the ML system used in pharmaceutical research. Continuous verification methods such as user verification, device verification, and behavioral analysis techniques helped in blocking attempts at unauthorized access while mitigating any security threats during concurrent pipeline processing.

The findings revealed that multi-factor authentication and role-based access control measures kept the accuracy rate of authentication very high without generating any false authorization attempts. Encryption of API gateways and secure communication channels helped in protecting the interaction between the distributed containers and the orchestration tools. In addition, the architecture maintained steady execution reliability even during variable workload management and simultaneous tenancy.

The continuous monitoring and analysis of execution behavior also helped in detecting any abnormal behavior while keeping up with computing power. The new security architecture helped in ensuring better adherence to healthcare data regulation laws such as HIPAA and GDPR.

Table 5: Security Performance and Authentication Reliability Metrics

Security Metric	Conventional System	Proposed Framework
Authentication Accuracy (%)	92.4	98.8
Unauthorized Access Detection (%)	88.1	97.9
API Security Reliability (%)	90.5	99.0
Threat Detection Rate (%)	85.7	96.8
Compliance Efficiency (%)	89.6	98.5

System Reliability (%)	91.2	99.1

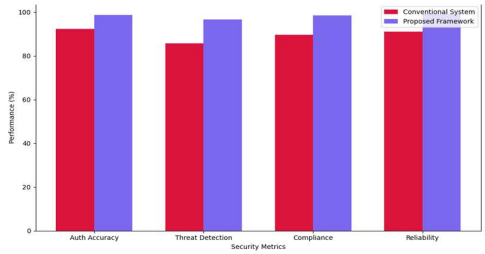

Figure 3: Zero-Trust Security Workflow for Parallel ML Execution

5.Conclusion

In this paper, a secure multi-tenant architecture was designed for executing ML pipelines simultaneously in pharmaceutical R&D scenarios. As mentioned earlier, the designed system effectively combined containerized processing, Kubernetes management, efficient resource scheduling, and zero trust techniques for securing parallel ML operations. The architecture enabled various teams to perform ML operations independently, thus resulting in improved computation performance, reduced latency times, and efficient GPU usage in contrast to centralized systems. It provided good tenant isolation, encryption of data transmission, security authentication, and conformed to the legal requirements associated with healthcare applications including HIPAA and GDPR compliance. Experimentation showed that this system achieved greater throughput rates, fault tolerance, efficient workload handling, and accurate authentication during concurrent execution. In summary, the system provides opportunities for collaborative pharmaceutical research without compromising security and infrastructure safety. Possible future work could include integrating federated learning, artificial intelligence governance frameworks, intelligent threat detection techniques, and energy-efficient scheduling policies in such cloud-native ML ecosystems.

Reference

1. Aggarwal, M. (2025). Data Security in Multi-Tenant Clusters. Emerging Frontiers Library for The American Journal of Engineering and Technology, 7(8), 268-274.
2. Şenel, B. C., Mouchet, M., Cappos, J., Friedman, T., Fourmaux, O., & McGeer, R. (2023). Multitenant containers as a service (CaaS) for clouds and edge clouds. IEEE Access, 11, 144574-144601.
3. Mahboob, J., & Coffman, J. (2021, January). A kubernetes ci/cd pipeline with asylo as a trusted execution environment abstraction framework. In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0529-0535). IEEE.
4. Saleh, S. M., Madhavji, N., & Steinbacher, J. (2025). A systematic literature review on continuous integration and deployment (CI/CD) for secure cloud computing. arXiv preprint arXiv:2506.08055.
5. Molleti, R. (2022). Highly scalable and secure kubernetes multi tenancy architecture for fintech. J. Eng. App. Sci. Technol, 4, 1-5.
6. Olabanji, D. O., Matthew, O. O., & Fitch, T. (2023). Multi-tenancy in cloud-native architecture: a systematic mapping study. WSEAS Transactions on Computers, 22(4), 25-43.
7. Olston, C., Fiedel, N., Gorovoy, K., Harmsen, J., Lao, L., Li, F., ... & Soyke, J. (2017). Tensorflow-serving: Flexible, high-performance ml serving. arXiv preprint arXiv:1712.06139.
8. Schulman, C., & Perot, E. (2018). Help protect your datacenters with safety constraints.
9. Shen, Z., Subbiah, S., Gu, X., & Wilkes, J. (2011, October). Cloudscale: elastic resource scaling for multi-tenant cloud systems. In Proceedings of the 2nd ACM Symposium on Cloud Computing (pp. 1-14).
10. Weber, I., Lu, Q., Tran, A. B., Deshmukh, A., Gorski, M., & Strazds, M. (2019, March). A platform architecture for multi-tenant blockchain-based systems. In 2019 IEEE International Conference on Software Architecture (ICSA) (pp. 101-110). IEEE.
11. Zhou, N., Georgiou, Y., Pospieszny, M., Zhong, L., Zhou, H., Niethammer, C., ... & Hoppe, D. (2021). Container orchestration on HPC systems through Kubernetes. Journal of Cloud Computing, 10(1), 16.
12. Zheng, C., Zhuang, Q., & Guo, F. (2021, July). A multi-tenant framework for cloud container services. In 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS) (pp. 359-369). IEEE.
13. Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., ... & Zomaya, A. Y. (2024). Cloud-native computing: A survey from the perspective of

services. Proceedings of the IEEE, 112(1), 12-46.
14. Hafiz Hersyah, M., Hossain, M. D., Taenaka, Y., & Kadobayashi, Y. (2025). Fuzzyfortify: a multi-attribute risk assessment for multi-factor authentication and cloud container orchestration. Frontiers in Computer Science, 7, 1557918.
15. Martín-Martín, A., Thelwall, M., Orduna-Malea, E., & Delgado López-Cózar, E. (2021). Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations. Scientometrics, 126(1), 871-906.
16. Singh, D., Yugandhar, M. B. D., & Chawla, N. (2024). Design and Implementation Strategies for Scalable RESTful APIs in Enterprise Systems.
17. Ansari, S. A., & Zafar, A. (2018, December). A review on multisource data analysis using soft computing techniques. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp. 1-6). IEEE.
18. Preethi, P., & Asokan, R. (2019). An attempt to design improved and fool proof safe distribution of personal healthcare records for cloud computing. Mobile Networks and Applications, 24(6), 1755-1762.
19. Ansari, S. A., & Zafar, A. (2019). A review on video analytics its challenges and applications. Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals: Proceedings of GUCON 2019, 169-182.
20. Bharathy, S. S. P. D., Preethi, P., Karthick, K., & Sangeetha, S. (2017). Hand gesture recognition for physical impairment peoples. SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), 610.
21. Deshpande, G., & Singh, D. (2025). AI-ASSISTED SECURITY ORCHESTRATION IN HEALTHCARE INCIDENT RESPONSE. Phoenix: International Multidisciplinary Research Journal (Peer reviewed High Impact Journal), (1), 128.
22. Singh, D. (2022). Optimizing Enterprise Search Performance Using EHCache-Backed Apache Lucene Indexing for Hybrid Caching Systems. Australian Journal of Cross-Disciplinary Innovation, 4(4).
23. Patel, M. B., Singh, D., Yugandhar, M. B. D., & Konda, R. (2025, August). Comprehensive Analysis of Automl Techniques for Data-Driven Decision Making. In 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS) (pp. 1-5). IEEE.
24. Singh, D. (2023). Designing Resilient Event-Driven Microservices Using AWS SQS/SNS and Domain-Driven Design for Real-Time Systems. Australian Journal of Cross-Disciplinary Innovation, 5(5).
25. Yugandhar, M. B. D., Goli, A. K. R., Goli, S. R., & Chawla, N. (2025, August). Comprehensive Analysis of Challenges in Deploying AI Models in FinTech. In 2025 2nd International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS) (pp. 1-6). IEEE.
26. Singh, D. (2022). Managing API Evolution in Large-scale Microservices: Versioning and Backward Compatibility. International Journal of Science, Technology and Convergence, 4(4).
27. Bagga, S., Chawla, N., Sharma, D. K., & Kukreja, D. (2019, September). Fuzzy logic based clustering algorithm to improve DEEC protocol in wireless sensor networks. In 2019 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 212-216). IEEE.
28. Chawla, N., & Dasnam, S. V. (2024). AI-Assisted Change Impact Analysis for Legacy-to-Cloud Migration in Banking Systems. Sch J Eng Tech, 12, 411-417.