



Cloud-Native ETL Transformation in Healthcare: Performance Optimization and Cost Reduction Through AWS Infrastructure Modernization

Surya Naga Naresh Babu Juttuga*

Independent Researcher, USA

* Corresponding Author Email: suryannbjuttuga@gmail.com - ORCID: 0000-0002-1007-0050

Article Info:

DOI: 10.22399/ijcesen.4514
Received : 21 October 2025
Revised : 10 December 2025
Accepted : 15 December 2025

Keywords

Cloud Computing,
Healthcare Data Infrastructure,
Etl Modernization,
Gitops Automation,
Aws Containerization

Abstract:

Abstract should be about 100-250 words. It should be written times new roman and 10 punto. Healthcare enterprises are under constant pressure to deliver faster analytics on growing clinical and claims datasets while staying compliant and cost-efficient. This paper presents a cloud-native ETL modernization that I led for a regulated U.S. healthcare organization, migrating a legacy on-prem batch ETL stack into AWS using containerization and GitOps automation. The modern platform runs Informatica workloads inside Docker containers on Amazon ECS, orchestrated with AWS Step Functions, and lands curated outputs in Snowflake. All infrastructure and runtime configuration are defined as code using AWS CDK and version-controlled in Git with promotion pipelines in GitHub Actions. The program replaced a tightly coupled, manually operated batch system that required ~34 hours per major run and incurred fixed licensing/compute costs, with an elastic, auditable platform completing equivalent runs in ~2.5 hours (~90% faster) and reducing per-run cost from roughly \$183 to \$18 (~85% lower). Performance gains were validated via Step Functions execution histories, ECS/CloudWatch metrics, and Snowflake query logs; cost reductions were verified through AWS Cost Explorer and Snowflake billing usage. Beyond speed and cost, the modernization introduced repeatable deployment patterns, safer multi-environment promotions, and measurable improvements in reliability and compliance traceability. The resulting approach offers a practical, evidence-backed blueprint for healthcare data organizations modernizing ETL under strict regulatory constraints.

1. Opening Context and Foundational Overview

1.1 Critical Need for Medical Information System Advancement

Healthcare organizations now generate and consume data at a scale that legacy ETL environments were never designed to handle. Electronic health records, claims and billing feeds, device telemetry, and enterprise analytics workloads produce continuous, high-volume pipelines that must be processed reliably and quickly. Traditional on-prem ETL platforms—often built around monolithic batch jobs and fixed compute—struggle to meet modern expectations for timely insight, seamless interoperability, and rapid iteration. This modernization challenge is not limited to infrastructure upgrades. It changes how hospitals convert raw operational data into usable

clinical and administrative intelligence. The platform must support diverse workload types (batch, event-driven, and mixed modes) while meeting strict regulatory requirements for protected health information (PHI) and maintaining high availability. In short, healthcare needs data platforms that scale elastically, deploy safely, and deliver trustworthy outcomes under compliance constraints.

1.2 Obstacles Presented by Outdated Infrastructure

Obsolete technology stacks within medical organizations impose substantial limitations that undermine institutional performance. Tightly integrated system architectures resist modification efforts, creating elevated risk profiles and extended implementation timelines. Insufficient documentation practices surrounding configuration modifications eliminate reliable audit trails,

comprehensive testing frameworks, and dependable rollback capabilities when errors occur. Capacity constraints manifest as data volumes surpass original architectural specifications, with rigid designs unable to accommodate fluctuating demand patterns typical of contemporary healthcare analytics [2]. Fixed licensing commitments mandate continuous payments regardless of actual utilization levels, generating wasteful expenditures compared to flexible consumption models. Connectivity barriers stem from vendor-specific protocols, bespoke interface mechanisms, and deprecated exchange standards that prevent smooth integration with modern distributed computing platforms and third-party applications.

1.3 Investigation Goals and Examination Boundaries

This documentation captures a comprehensive platform transformation executed at a major American healthcare facility, migrating from traditional premises-based data operations to distributed cloud computing environments. Primary goals encompass recording design decisions, implementation tactics, and quantifiable results stemming from infrastructure renewal efforts. Coverage spans entire project lifecycles from preliminary evaluation through operational assessment, highlighting containerized software deployment, automated orchestration mechanisms, and code-defined infrastructure provisioning. Technical elements examined include isolated runtime environments, distributed coordination services, event-triggered workflow automation, and scalable warehouse architectures. Organizational factors receive equivalent consideration, investigating financial restructuring, operational procedure modifications, and transferable architectural patterns applicable beyond individual institutions. Documentation furnishes detailed accounts of challenge resolution approaches, performance enhancement techniques, and conformance strategies addressing sector-specific requirements for data protection, regulatory compliance, and governance structures.

Primary contributions of this case study.

This work contributes a regulated-healthcare ETL modernization pattern that goes beyond lift-and-shift. The approach combines (1) container-native batch execution on Amazon ECS Fargate, (2) stateful orchestration with AWS Step Functions to replace legacy schedulers and scripts, and (3) GitOps-audited IaC promotion across Dev/Test/UAT/Prod. Together, these elements enable elastic scaling, deterministic recovery, and

compliance-ready deployment traceability within HIPAA-aligned controls.

1.4 Correspondence with Broader Healthcare Policy Goals

Platform renewal efforts directly support established policy priorities emphasizing expanded treatment access, expenditure management, and enhanced system dependability. Improved processing capabilities accelerate clinical decision support by furnishing practitioners with immediate access to complete patient histories and synthesized insights derived from multifaceted datasets [1]. Financial stewardship remains central to healthcare policy formulation, driven by persistent imperatives to moderate escalating costs while preserving service standards. Elastic capacity allocation and usage-based pricing eliminate large capital outlays for physical equipment while redirecting funds toward patient care initiatives and therapeutic innovation [2]. Business continuity has achieved heightened significance given mounting cybersecurity risks, natural catastrophes, and public health crises demanding resilient, fault-tolerant infrastructures. Distributed computing frameworks deliver geographic redundancy, self-healing capabilities, and emergency preparedness attributes that markedly strengthen organizational readiness during adverse conditions. Furthermore, infrastructure modernization enables standardized data interchange mechanisms vital for coordinated care delivery and population health management across fragmented delivery networks.

2. Scholarly Context and Conceptual Foundations

2.1 Distributed Computing Integration Within Medical Information Systems

Distributed computing platforms have fundamentally altered how healthcare facilities approach technological infrastructure, delivering expandable resources through flexible arrangements that contrast sharply with conventional fixed-capacity models. Medical organizations have progressively incorporated these technologies despite early reservations concerning patient information sensitivity and compliance complexities. Network-based computing architectures furnish institutions with immediate access to processing power, removing requirements for substantial initial hardware purchases while permitting rapid adjustment to accommodate shifting operational demands [3]. This movement away from facility-housed equipment signifies a

profound transformation in how hospitals and clinics acquire, administer, and enhance their technological capabilities. Various service tiers, including infrastructure provisioning, platform hosting, and software delivery, present different degrees of administrative oversight, permitting institutions to choose arrangements matching their technical proficiency and operational preferences. Technology adoption within medical settings surpasses basic equipment substitution, facilitating novel treatment approaches, remote consultation platforms, and joint investigation projects previously hindered by technological constraints. Cloud adoption in healthcare has moved from "optional innovation" to a core operational requirement. The biggest driver I have seen in real programs is not just scale, but time-to-insight under compliance. When clinical analytics, provider networks, and claims adjudication all depend on daily/weekly batch pipelines, a slow or fragile ETL layer directly delays care decisions and actuarial planning. Distributed computing helps because it makes capacity elastic and measurable. Instead of sizing hardware for the worst case, organizations can scale for actual demand, and they can prove what ran, with what configuration, and under what approvals. In regulated settings, that auditability is as important as raw performance.

2.2 Data Pipeline Transformation Strategies and Established Standards

Modern ETL transformations usually fail when teams lift-and-shift legacy jobs without rethinking orchestration, runtime isolation, and failure handling. The strategy that worked for us was to isolate the ETL runtime (Docker), isolate the orchestration (Step Functions), and isolate the storage/serving layer (Snowflake), then connect them with strict, versioned interfaces.

The standards that mattered most in practice were:

- Idempotent stages (jobs can re-run safely without corrupting targets).
- Explicit failure boundaries (one step fails without poisoning the whole pipeline).
- Observable pipelines (every run emits metrics and logs that tie to a commit and ticket). Environment parity (Dev/Test/UAT/Prod behaves the same except for parameters).

2.3 Operational Integration Practices and Version-Controlled Infrastructure Management

Collaborative operational methodologies have transformed software distribution by encouraging

cooperation between development and operational personnel, mechanizing deployment sequences, and quickening release schedules. Version-controlled infrastructure management advances these concepts by handling infrastructure and application settings as programmatic code maintained within revision repositories, facilitating declarative system administration and mechanized alignment between intended and actual configurations [4]. This technique stresses unchangeable infrastructure, where modifications are executed through version-tracked submissions rather than manual adjustments, furnishing complete examination records and streamlined reversal abilities. Automated integration and deployment sequences mechanize validation, confirmation, and advancement of modifications across settings, diminishing human mistakes and quickening production delivery. Version-controlled frameworks utilize retrieval-oriented deployment approaches where mechanized representatives persistently observe repository conditions and reconcile discrepancies with active platforms, guaranteeing uniformity and diminishing configuration deviation [4]. Organizational implementation of these techniques strengthens system dependability, bolsters security stance through mechanized compliance verification, and enables personnel to distribute capabilities and corrections with enhanced speed and assurance.

2.4 Financial Evaluation Structures for Platform Transition

Platform transition choices necessitate thorough fiscal examination extending past straightforward expense contrasts to include complete ownership expenditures, alternative expenses, and strategic worth generation. Conventional capital investment frameworks for facility-based infrastructure demand considerable advance commitments, continuous upkeep charges, and periodic renewal periods that burden organizational finances. Migration toward distributed computing alters this fiscal framework into operational costs with consumption-oriented pricing, removing substantial capital disbursements while providing budgetary flexibility. Financial evaluations must consider migration expenses encompassing preparation, implementation, instruction, and potential productivity consequences during the transition periods. Advantages reach past immediate expense reductions to include enhanced flexibility, accelerated innovation periods, strengthened catastrophe restoration capabilities, and reduced technical obligations. Institutions must appraise complete ownership expenditures across extended

timeframes, contemplating elements including licensing modifications, personnel necessities, and concealed expenses connected with antiquated system preservation. Strategic advantages encompassing quicker capability delivery, enhanced scalability supporting organizational expansion, and strengthened competitive standing frequently warrant migration commitments even when immediate expense reductions remain moderate.

2.5 Compliance Requirements and Protection Protocols for Medical Cloud Integration

Medical institutions function beneath rigorous compliance structures governing information protection, confidentiality, and security implementations, generating distinct obstacles for distributed computing integration. Regulations stipulate particular controls surrounding information access, encryption protocols, examination documentation, and violation reporting, demanding cloud platforms exhibit conformance through thorough certification procedures. Safeguarded medical information necessitates exceptional protection measures encompassing encryption during storage and transmission, layered authentication mechanisms, responsibility-oriented access limitations, and exhaustive activity observation [3]. Cloud service vendors must present healthcare-focused conformance certifications and contractual instruments establishing explicit accountabilities for information protection and regulatory compliance. Distributed accountability frameworks delineate security duties between cloud vendors and medical institutions, demanding meticulous preparation, ensuring thorough coverage of all compliance mandates. Information residency and sovereignty apprehensions require contemplation of geographic storage positions and international information movement consequences. Partnership instruments formalize connections between medical organizations and cloud vendors, establishing legal structures for protected medical information administration [3]. Protection designs must integrate comprehensive defensive tactics with numerous safeguard tiers, perpetual threat observation, and incident reaction abilities satisfying healthcare sector demands for confidentiality, integrity, and accessibility.

3. Research Approach and Technical Framework Design

3.1 Investigation Structure and Information Gathering Procedures

The examination utilized a detailed case study framework to chronicle the platform transformation at a prominent American medical facility. Information assembly drew from varied origins encompassing technical records, architectural blueprints, system performance indicators, and discussions with engineering personnel, operations staff, and management authorities. The approach highlighted extended observation, monitoring progression from preliminary evaluation through final implementation and stabilization intervals. Recording protocols documented architectural choices, execution obstacles, resolution tactics, and insights gained throughout the transformation journey. Qualitative information obtained through organized conversations with technical staff furnished perspectives into operational consequences, workflow alterations, and institutional adjustment mechanisms. Quantitative indicators comprised processing duration assessments, resource consumption characteristics, expenditure monitoring, and system dependability measurements gathered preceding, throughout, and following migration operations. Combining numerous information sources reinforced authenticity and delivered a thorough comprehension of technical, operational, and institutional aspects. Ethical protocols guaranteed patient information confidentiality throughout the examination, with all records and evaluation concentrating solely on infrastructure elements without accessing safeguarded medical data.

3.2 Existing Platform Evaluation and Limitation Discovery

The legacy ETL stack was Informatica-driven and hosted on fixed on-prem VMs with shared storage and a batch scheduler. The system had three major pain points:

Performance ceiling: long-running jobs competed for CPU/memory on shared hosts. A full grouper ETL run averaged ~34 hours, and peaks occasionally exceeded that.

Operational fragility: scheduler chains broke on file transfer delays, and recovery required manual intervention by a small set of SMEs.

Configuration drift: environment parameters lived in multiple scripts and folders, making Dev/Test parity hard to guarantee.

This evaluation was essential because it exposed hidden dependencies (like hardcoded paths and implicit job ordering) that would have caused silent failures in cloud if not fixed early.

3.3 Distributed Computing Framework Construction Guidelines

Framework construction adhered to distributed computing construction guidelines, highlighting segmentation, flexibility, strength, and mechanization. The blueprint separated unified processing sequences into separate, autonomously implementable elements that could expand independently according to workload traits. Condition-free application construction permitted lateral expansion without session connection necessities, streamlining burden allocation and malfunction restoration. Container technology furnished uniform runtime settings across development, validation, and operational situations while segregating requirements and streamlining version administration. Occurrence-triggered coordination separated workflow elements, permitting asynchronous processing arrangements and diminishing firm connection between system components [6]. The framework welcomed administered offerings for undifferentiated substantial work, permitting technical personnel to concentrate on business reasoning rather than infrastructure management. Repository tactics separated computation and retention strata, utilizing object repositories for lasting information persistence and temporary computation instances for processing operations. Network framework executed virtual isolated networks with subnet separation, protection assembly regulations, and encoding for information during transmission. The blueprint incorporated watchfulness from the outset, with exhaustive documentation, measurement assembly, and dispersed tracking abilities, permitting proactive observation and swift problem-solving. Catastrophe restoration arrangements incorporated geographic repetition, mechanized preservation protocols, and validated restoration procedures, guaranteeing business persistence [6].

3.4 Technology Portfolio Choice Justification

Technology choice balanced functional necessities, operational maturity, ecosystem connection, and complete ownership expenditure deliberations. Container technology was selected for application bundling attributable to transferability, uniformity across settings, and productive resource employment compared to conventional virtual apparatus. Administered container coordination removed operational burden connected with cluster administration, automatic expansion arrangement, and wellness observation execution. Serverless sequence coordination furnished visual depiction of processing conduits, integrated mistake administration, repetition reasoning, and condition administration without infrastructure allocation [5].

Distributed object repositories presented boundless expandability, elevated durability, and economical retention for substantial datasets with existence administration capacities. Distributed information warehousing was chosen for flexible computation separation from retention, assistance for partially-organized information arrangements, and smooth connection with distributed repositories through outside staging apparatus. Version oversight platforms permitted dispersed cooperation, exhaustive modification monitoring, and connection with automated implementation conduits. Mechanization structures for persistent connection and implementation diminished manual participation, normalized implementation protocols, and quickened release speed. Infrastructure-as-programming instruments furnished systematic infrastructure allocation with deviation identification, requirement settlement, and setting uniformity assurances [5]. The technology portfolio prioritized offerings with robust community assistance, exhaustive records, and confirmed enterprise acceptance arrangements within medical situations.

3.5 Programmatic Infrastructure and Automated Deployment Tactics

Infrastructure allocation embraced programming-oriented techniques where infrastructure specifications existed in version-monitored repositories alongside application programming. Descriptive arrangement languages indicated desired infrastructure conditions, permitting mechanized allocation, alteration, and elimination of resources through systematic interfaces. The tactics executed segmented infrastructure elements as reusable frameworks, encouraging uniformity across settings and streamlining upkeep operations. Mechanized validation structures authenticated infrastructure arrangements preceding implementation, inspecting for protection misarrangements, conformance infractions, and architectural criterion deviations [5]. Persistent connection conduits are mechanically activated upon programming submissions, performing mechanized validations, protection examinations, and quality inspections preceding combining modifications into principal branches. Implementation mechanization advanced modifications through setting levels following approval checkpoints, with mechanized reversal abilities activated by wellness inspection malfunctions or performance deterioration. Setting similarity between development, validation, and operational situations reduced setting-specific difficulties and enhanced implementation

assurance. Confidential administration platforms secured delicate credentials and arrangement parameters, preventing hardcoded confidential information in version oversight repositories. Examination records documented all infrastructure alterations with attribution to particular modifications and contributors, supporting conformance necessities and forensic examinations [5]. The execution cultivated cooperation between development and operations personnel through distributed instruments, transparent modification procedures, and shared possession of implementation results.

3.6 Protection Protocols and Regulatory Conformance Structures

The protection framework incorporated comprehensive defensive tactics with numerous safeguard strata addressing network, application, and information protection domains. Network division segregated delicate elements within isolated subnets reachable exclusively through regulated entrance locations with encoded communication passages. Identity and permission administration executed a minimal authorization doctrine, awarding minimal authorizations required for particular operational activities with periodic permission evaluations and automated termination regulations. Encoding safeguarded information during retention employing administered encoding offerings and information during transmission through transport stratum protection mechanisms satisfying medical regulatory criteria. Documentation and observation documented protection-applicable occurrences encompassing verification efforts, permission determinations, arrangement modifications, and information permission characteristics for protection evaluation and conformance documentation [3]. Weaknesses in administration procedures comprised mechanized examination, correction of administration protocols, and prompt remediation of pinpointed security vulnerabilities. Conformance structures charted technical regulations to regulatory necessities encompassing information safeguarding mandates, finding traceability between protection executions and conformance duties. Business persistence preparation addressed catastrophe restoration situations with recorded protocols, periodic validation exercises, and specified restoration targets for essential platforms. Occurrence reaction mechanisms founded escalation routes, communication protocols, and forensic examination abilities for protection occurrence administration. External-party hazard administration appraised distributed service vendor

protection certifications, contractual safeguards, and distributed accountability framework delineations guaranteeing exhaustive protection coverage across institutional and vendor boundaries [3].

4. Execution Strategy and Operational Deployment

4.1 Transition Stages and Schedule Progression

The transformation program advanced through deliberately structured intervals crafted to reduce operational interruption while founding a solid distributed computing infrastructure. Opening intervals emphasized setting preparation, encompassing distributed account allocation, network framework founding, and protection baseline arrangement. The following intervals addressed application containerization, transforming antiquated executable elements into transferable container representations with incorporated requirements and arrangement specifications. Development setting implementation preceded validation setting founding, permitting repeated enhancement of implementation protocols and infrastructure arrangements preceding operational migration. Experimental implementations handled portion workloads to authenticate architectural determinations, performance traits, and operational protocols beneath regulated situations. Gradual workload transition followed triumphant experimental completion, moving processing operations progressively while sustaining simultaneous antiquated platform function for retreat contingency. Each interval incorporated specified triumph standards, stakeholder endorsement checkpoints, and retreat protocols, guaranteeing regulated advancement. Schedule preparation balanced urgency for modernization advantages against hazard reduction necessities, distributing adequate intervals for validation, confirmation, and personnel acquaintance with novel platforms. Synchronization across technical personnel, operational staff, and organizational participants guaranteed alignment on anticipations, requirements, and communication mechanisms throughout performance intervals.

4.2 Technical Obstacles and Solution Approaches

4.2.1 Distributed Computing Intake Conduit Construction

Conventional file-oriented intake apparatus demanded fundamental reconstruction to utilize

distributed repository abilities and serverless handling frameworks. Antiquated techniques depended on network file portions and scheduled file movements, generating requirements on persistent infrastructure and manual participation junctures. The transformation substituted these arrangements with an object repository connection, where upstream platforms deposited files immediately into distributed repository containers, activating mechanized handling sequences. Occurrence announcements from repository offerings commenced sequence coordination, removing polling apparatus and diminishing handling delay. Outside staging abilities within the information warehouse platform permitted immediate permission to distribute repository objects without intermediate information movement, streamlining intake procedures and diminishing transfer burden [8]. Retained protocol mechanization produced organized input files systematically, substituting manual file preparation operations and enhancing uniformity. Obstacle settlement demanded thorough validation of verification apparatus, authorization limits, and mistake administration situations to guarantee dependable function across different malfunction configurations. Performance enhancement balanced file dimensions, parallel execution tactics, and resource distribution to maximize output while regulating expenditures.

4.2.2 Container Coordination and Memory Enhancement

Container workload administration introduced obstacles connected to resource distribution, expansion conduct, and performance adjustment for information-intensive handling assignments. Opening container arrangements displayed memory depletion during substantial dataset handling, producing container cessation and sequence malfunctions. Settlement tactics executed flexible memory expansion, modifying container resource boundaries according to dataset traits and handling necessities [7]. Assignment specification improvements indicated suitable processor and memory allocations, guaranteeing sufficient resources without extreme overallocation. Coordination offering arrangement enhanced assignment positioning tactics, allocating workloads across obtainable computation ability while honoring resource limitations and connection necessities. Observation instrumentation monitored resource employment arrangements, pinpointing impediments and informing ability preparation determinations [7]. Container representation enhancement diminished representation dimensions through multiple-interval constructions, minimal

foundation representations, and productive stratum arrangement, quickening implementation durations and diminishing retention expenditures. Wellness inspection arrangements permitted coordination offerings to identify and substitute defective containers mechanically, enhancing platform strength. Expansion regulations balanced responsiveness to workload variations against expenditure deliberations, specifying boundaries and cooldown intervals for mechanical ability modifications [8].

4.2.3 Permission Regulation and Identity Administration Arrangement

Identity and permission administration arrangement demanded a meticulous blueprint to balance protection necessities with operational adaptability. Opening efforts encountered verification malfunctions when containerized applications tried to permission distributed information warehouse offerings, originating from inadequate authorization awards and misconfigured trust relationships. Settlement demanded founding offering responsibilities with accurately scoped authorizations adhering to minimal authorization doctrines, awarding exclusively abilities required for particular operational activities. Confidence regulation arrangement permitted container performance responsibilities to presume warehouse permission credentials, finding protected verification sequences without incorporating fixed credentials in application programming. Intersecting-account permission situations demanded supplementary intricacy, executing responsibility presumption sequences and outside identifier apparatus to sustain protection limits. Network directing arrangements guaranteed traffic between container settings and information warehouse offerings crossed isolated network routes rather than public internet directions, bolstering protection stance and diminishing vulnerability [8]. Confidential administration connection safeguarded delicate credentials encompassing database passwords and interface keys, recovering principles dynamically at runtime rather than retaining them in container representations or configuration documents. Authorization validation, authenticated permission regulations are exhaustively applied, confirming both triumphant authorization for legitimate functions and refusal of unauthorized permission efforts.

4.2.4 Antiquated Programming Reconstruction and Advancement

Current mechanization depended substantially on tailored programming gathered across years,

including thousands of lines of shell programming, batch documents, and procedural programming executing sequence reasoning and mistake administration. Immediate migration of this programming would continue technical obligation and overlook opportunities for architectural enhancement. Reconstruction operations separated unified programming into separate sequence intervals appropriate for coordination through serverless sequence offerings. Descriptive sequence specifications substituted commanding programming, indicating handling orders, branching reasoning, and mistake administration through an organized arrangement rather than procedural programming. This transformation enhanced sustainability, furnished visual sequence depiction, and permitted integrated observation abilities [7]. Reusable sequence elements surfaced from reconstruction operations, founding arrangements relevant across numerous handling conduits and diminishing replication. Condition administration moved from file-oriented monitoring apparatus to administered sequence condition, removing intricate file securing schemes and competition situation weaknesses. Mistake administration enhancements executed advanced repetition reasoning with exponential retreat, circuit interruption arrangements, and controlled deterioration tactics that bolstered strength beyond original programming abilities. Validation structures authenticated reconstructed sequences against original programming conduct, guaranteeing functional similarity while confirming architectural enhancements.

4.3 Version-Regulated Sequence Foundation

Version regulation connection altered implementation customs from manual protocols to mechanized, traceable sequences. All infrastructure specifications, application programming, container designations, and arrangement specifications moved into version regulation repositories, founding isolated origins of accuracy for platform specifications. Branch safeguard regulations imposed evaluation necessities, blocking immediate alterations to operational arrangements without colleague authentication and endorsement procedures. Mechanized conduits are activated upon repository submissions, performing authentication inspections, protection examinations, and mechanized validations preceding permitting modifications to advance. Setting-particular arrangement administration employed templating apparatus and specification replacement, sustaining uniformity across settings while accommodating required fluctuations [4]. Implementation

mechanization advanced modifications through development, validation, and operational settings, adhering to specified advancement standards and endorsement checkpoints. Retreat abilities utilized version regulation chronicle, permitting swift reversion to preceding arrangements when difficulties surfaced post-implementation. Examination records mechanically documented modification attribution, scheduling, and endorsement sequences, supporting conformance necessities and encouraging incident examination. Record connection sustained technical records alongside programming in version regulation, guaranteeing records remained aligned with actual executions [4]. Personnel cooperation enhanced through transparent modification procedures, programming evaluation conversations, and distributed possession of implementation results. Instruction programs acquainted operations staff with version regulation sequences, branching tactics, and implementation protocols, constructing institutional ability for sustainable version-regulated customs.

4.4 Validation and Confirmation Protocols

Exhaustive validation tactics authenticated technical precision, performance traits, and operational preparation preceding operational implementation. Component validation confirmed discrete element conduct in segregation, authenticating transformation reasoning, information quality regulations, and mistake administration apparatus. Connection validation scrutinized interactions between platform elements, guaranteeing suitable verification sequences, information exchange mechanisms, and sequence coordination operated properly across element limits. Performance validation appraised platform conduct beneath representative workload situations, assessing handling intervals, resource consumption, and output traits. Burden validation subjected platforms to maximum workload situations, pinpointing ability boundaries and authenticating expansion conducts beneath pressure situations [8]. Regression validation contrasted outputs between antiquated and modernized platforms, confirming functional similarity and pinpointing discrepancies demanding examination. Protection validation evaluated verification apparatus, authorization regulations, encoding executions, and weakness vulnerability through mechanized examination and manual infiltration validation. Catastrophe restoration validation authenticated preservation protocols, restoration procedures, and retreat apparatus through replicated malfunction situations. Consumer acceptance validation occupied

operational staff and downstream consumers, affirming that platform outputs satisfied organizational necessities and sequence connection operated suitably. Mechanized validation performance connected into persistent connection conduits, furnishing swift feedback on programming modifications and blocking regression introduction. Validation information administration founded representative datasets for authentication objectives while safeguarding delicate data through anonymization and artificial information production techniques [7].

5. Outcomes and Evaluation

5.1 Measurable Performance Indicators

The modernization produced measurable improvements in runtime, cost efficiency, and operational reliability. The primary ETL workload improved from approximately 34 hours per run on the legacy platform to approximately 2.5 hours per run on AWS, representing about a 90% reduction in end-to-end processing time. This comparison is based on 12 matched legacy runs versus 12 cloud runs over equivalent six-week windows, using consistent datasets and transformation logic. Cost efficiency improved in parallel. The estimated legacy blended cost averaged around \$183 per run, driven by fixed compute, licensing, and operational overhead. After modernization, the per-run cloud cost averaged roughly \$18, based on ECS Fargate compute, Step Functions state transitions, storage, and Snowflake loading compute during active windows—an approximately 85% reduction. Operational reliability also improved. In the legacy environment, 1–2 major failures per month required manual restarts due to scheduler chain breaks and host resource contention. In the cloud-native architecture, failures became rare and were typically auto-recovered through stateful retries and isolated container execution.

5.2 Contrasting Examination: Antiquated versus Distributed Computing Infrastructure

Methodical contrast between antiquated and modernized platforms emphasized fundamental architectural benefits of distributed computing techniques across numerous appraisal standards. Implementation rapidly enhanced substantially, with infrastructure allocation functions that formerly demanded comprehensive manual exertion and prolonged schedules now finalizing through mechanized procedures within minimal intervals. The antiquated setting displayed rigid ability limitations demanding advance preparation

and acquisition rotations for ability enlargement, while the distributed computing platform furnished instant expandability without physical infrastructure alterations [10]. Upkeep obligations decreased considerably as administered offerings removed accountabilities for operating platform correction, protection refreshes, and infrastructure observation, formerly consuming considerable operational effort. Antiquated platform requirements on particular equipment arrangements and vendor platforms generated restriction consequences and complicated catastrophe restoration preparation, whereas distributed computing containerization furnished transferability across different performance settings and geographic territories. Version regulation abilities altered implementation hazard characteristics, with the modernized setting presenting exhaustive modification monitoring, mechanized retreat protocols, and setting uniformity assurances missing in antiquated manual implementation procedures [9]. Watchfulness enhancements furnished unprecedented visibility into platform conduct through connected observation, dispersed monitoring, and centralized documentation abilities that exceeded antiquated platform instrumentation. Malfunction acceptance apparatus inherent to distributed computing designs is mechanically identified and restored from element malfunctions, contrasting with antiquated platforms that demand manual participation and prolonged restoration protocols. Connection abilities enlarged substantially, with normalized interfaces and occurrence-triggered designs streamlining connectivity with outside platforms, contrasted with antiquated exclusive mechanisms and tailored interfaces.

5.3 Expandability and Flexibility Enhancements

The distributed computing platform established superior expandability traits, permitting smooth accommodation of workload expansion and fluctuation. Lateral expansion abilities allowed flexible modification of computational resources by supplementing or withdrawing container occurrences responding to handling requirements, removing the perpendicular expansion restrictions inherent to antiquated unified designs [10]. Mechanical expansion regulations responded to workload indicators encompassing queue depths, handling delays, and resource employment arrangements, modifying ability without manual participation and guaranteeing uniform performance across fluctuating requirement degrees. Repository expandability demonstrated essentially boundless, with object repository

accommodating expanding information quantities without ability preparation exercises or repository collection enlargements demanded in antiquated settings. Database flexibility through division of computation and repository strata permitted autonomous expansion of handling strength and information preservation ability, enhancing expenditure-performance traits for different workload stages. Maximum requirement administration enhanced substantially as the platform mechanically allocated supplementary resources during elevated-quantity intervals and discharged them upon completion, blocking the performance deterioration encountered with antiquated fixed-ability platforms [9]. Geographic allocation abilities permitted workload positioning across numerous territories for delay enhancement and regulatory conformance necessities impossible with isolated-datacenter antiquated implementations. Development and validation setting allocation benefited from flexible ability, permitting temporary setting generation for particular projects without permanent infrastructure distribution. Surge ability for exceptional handling necessities became promptly obtainable, accommodating unexpected requirement increases that would have overwhelmed antiquated infrastructure [10]. The removal of ability preparation rotations diminished administrative burden while guaranteeing sufficient resources remained obtainable for all operational situations.

5.4 Functional Consequences on Clinical and Management Sequences

Platform advancement generated measurable enhancements in functional productivity and information reachability, supporting clinical and management activities. Diminished handling delay permitted more prompt obtainability of analytical perspectives, supporting quicker decision-making processes for clinical functions and management preparation operations. Healthcare furnishers obtained permission to exhaustive patient data with decreased delay between clinical occurrences and information obtainability for evaluation and documentation. Management sequences benefited from enhanced information excellence and uniformity stemming from mechanized authentication protocols and normalized transformation reasoning substituting manual information manipulation procedures. Documentation abilities are enlarged through a supported analytics platform connection, furnishing participants with richer visualizations and more advanced analytical instruments for investigating healthcare information. Functional staff

encountered diminished obligation from platform upkeep operations as mechanized observation identified and settled difficulties proactively, reducing reactive problem-solving requirements. The dependability enhancements decreased platform interruption incidents that formerly disrupted clinical and management functions, guaranteeing uniform information obtainability for duration-delicate operations. Intersecting-functional cooperation is bolstered through enhanced information reachability, with clinical, functional, and fiscal personnel accessing distributed information resources through normalized interfaces. Conformance documentation mechanization diminished manual effort for regulatory presentation preparation, with mechanized information extraction and formatting protocols substituting labor-intensive manual compilation procedures. Patient treatment excellence indicators became obtainable with elevated promptness, permitting quicker pinpointing of tendencies demanding clinical participation or procedure enhancements. Resource distribution determinations benefited from bolstered analytics abilities, supporting evidence-grounded enhancement of personnel arrangement, equipment, and facility employment arrangements.

5.5 Institutional Acceptance and Transferability Capability

The triumphant transformation founded architectural arrangements and functional customs relevant across supplementary institutional platforms and potentially transferable to additional healthcare institutions. Technical personnel catalogued exhaustive execution instructions, architectural determination chronicles, and functional protocols documenting insights obtained and established customs for future advancement programs. The segmented design blueprint encouraged element reemployment, with containerized applications, sequence templates, and infrastructure segments relevant to comparable information handling necessities across different institutional divisions. Normalized implementation protocols through version-regulated infrastructure specifications permitted uniform execution techniques, diminishing variance and implementation hazards for the following projects. Instruction programs constructed during the program established institutional ability for distributed computing construction and functions customs, generating internal proficiency for supporting supplementary advancement efforts [9]. The transformation established the practicability of moving intricate, assignment-essential healthcare

platforms to distributed computing systems, addressing institutional apprehensions about risk, compliance, and functional preparation that might otherwise restrain acceptance. Expenditure-advantage authentication furnished concrete verification supporting organizational justifications for supplementary distributed computing migration programs, with measured fiscal returns and functional enhancements warranting commitment necessities. Governance structures founded during execution encompassing protection regulations, conformance protocols, and functional criteria generated reusable templates for future distributed computing acceptances. Vendor connections and acquisition procedures constructed through the program were streamlined following the distributed computing offering purchases and contract discussions. The architectural technique demonstrated adaptability to fluctuating workload traits, proposing relevance beyond the particular employment situation to diverse healthcare information handling necessities [10]. Participant assurance is elevated through established triumphant distribution, diminishing institutional opposition to following transformation suggestions and quickening endorsement procedures for advancement commitments.

5.6 Insights Obtained and Recognized Established Customs

Execution encounter disclosed essential triumph elements and warning perspectives valuable for comparable transformation programs. Exhaustive antiquated platform evaluation demonstrated vital, with thorough comprehension of current functionality, requirements, and functional limitations blocking surprises during migration and guaranteeing functional thoroughness in modernized platforms. Repeated migration techniques diminishing hazard through progressive advancement and furnishing early authentication of architectural determinations demonstrated

superiority over comprehensive cutover tactics. Participant occupation throughout the program guaranteed alignment between technical executions and organizational necessities while constructing institutional assistance for required procedure modifications. Commitment in mechanization infrastructure encompassing validation structures, implementation conduits, and observation abilities produced considerable returns through diminished manual exertion and enhanced platform dependability [9]. Documentation highlighted throughout execution encouraged knowledge movement, functional assistance, and future enhancement operations, blocking knowledge concentration and succession preparation weaknesses. Protection and conformance deliberations connected from project commencement rather than addressed retroactively, blocked costly reconstruction, and guaranteed regulatory necessities obtained suitable attention. Performance validation beneath representative workload situations preceding operational implementation pinpointed enhancement opportunities and ability necessities blocking post-implementation performance difficulties. Personnel aptitude construction through instruction programs and practical encounters, establishing computing competencies demonstrated to be essential for sustainable function and persistent enhancement. Vendor partnership development permitted productive technical assistance, architectural direction, and early permission to platform abilities, bolstering execution excellence. Modification of administration procedures addressing institutional adjustment to novel functional protocols, instruments, and sequences diminished resistance and quickened realization of transformation advantages [10]. Technical obligation diminishment through advancement rather than straightforward replication of antiquated arrangements maximized extended-duration worth and positioned the institution for continued innovation and productivity acquisitions.

Table 1: Legacy System Constraints and Limitations [2]

| Constraint Category | Specific Limitation | Operational Impact | Business Consequence |
|----------------------------|---|---|--|
| Architectural Design | Monolithic, tightly coupled components | Elevated modification risk, extended upgrade cycles | Delayed feature delivery, increased downtime |
| Scalability | Fixed capacity, vertical scaling only | Performance degradation under peak loads | Service disruptions, user dissatisfaction |
| Configuration Management | Manual changes, no version control | Configuration drift, limited rollback capability | Recovery delays, audit trail gaps |
| Cost Structure | Fixed licensing, perpetual capacity commitments | Continuous expenditure regardless of utilization | Budget inefficiency, resource waste |
| Integration | Proprietary protocols, custom interfaces | Limited interoperability with modern systems | Innovation barriers, vendor lock-in |

| | | | |
|------------|----------------------------------|---|--|
| Deployment | Manual processes, batch-oriented | Extended deployment timelines, human error risk | Slow time-to-market, operational fragility |
|------------|----------------------------------|---|--|

Table 2: Cloud Computing Service Models in Healthcare [3]

| Service Model | Description | Healthcare Applications | Management Responsibility | Compliance Considerations |
|------------------------------------|---|--|--|---|
| Infrastructure-as-a-Service (IaaS) | Virtualized computing resources on demand | Data storage, compute capacity for analytics, backup/disaster recovery | The organization manages OS, middleware, and applications | Organization responsible for data encryption, access controls |
| Platform-as-a-Service (PaaS) | Development and deployment environment | Application development, integration platforms, database services | Provider manages infrastructure, organization manages applications | Shared responsibility for security configurations |
| Software-as-a-Service (SaaS) | Complete application delivered via cloud | Electronic health records, telemedicine platforms, and patient portals | Provider manages entire stack | The vendor must demonstrate HIPAA compliance |

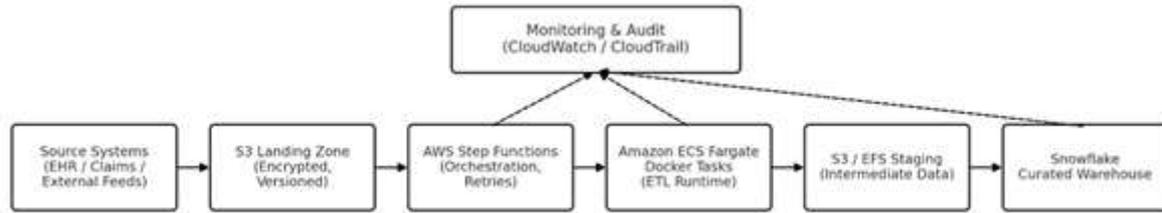


Figure 1: Simple Cloud Native ETL Architecture

Table 3: Infrastructure-as-Code Implementation Standards [5]

| Implementation Aspect | Standard Practice | Validation Mechanism | Compliance Outcome |
|--------------------------|--|---|------------------------------|
| Repository Structure | Modular components, environment separation | Code review, branch protection | Maintainability, reusability |
| Configuration Management | Parameterized templates, environment variables | Automated testing, linting | Environment consistency |
| Secret Handling | External secret management, no hardcoded values | Secret scanning, automated detection | Security compliance |
| Testing Strategy | Unit tests, integration tests, policy validation | CI pipeline gates, automated execution | Quality assurance |
| Deployment Process | Progressive rollout, automated validation | Health checks, rollback triggers | Deployment safety |
| Documentation | Inline comments, architecture decision records | Documentation review, completeness checks | Knowledge preservation |
| Change Control | Pull request workflow, approval requirements | Audit logging, traceability | Regulatory compliance |

Table 4: Pre/Post Migration Metrics (Anonymized) [7, 8]

| Metric | Legacy (Baseline) | Cloud (Modernized) | Change | How measured |
|--------|-------------------|--------------------|--------|--------------|
|--------|-------------------|--------------------|--------|--------------|

| | | | | |
|-----------------------------------|---------------------------|-----------------------|----------------------|---|
| End-to-end runtime per ETL run | 34 hours (avg) | 2.5 hours (avg) | ~90% faster | Legacy scheduler logs vs Step Functions execution history (12 matched runs) |
| Cost per run | ~\$183/run | ~\$18/run | ~85% lower | Legacy blended cost vs AWS Cost Explorer + Snowflake credit usage |
| Manual failures requiring restart | 1–2 per month | rare / auto-recovered | major reduction | Ops tickets + Step Functions retry outcomes |
| SLA compliance | often missed during peaks | consistently met | improved reliability | Batch completion timestamps across six-week windows |

Table 5: Lessons Learned and Best Practices [9, 10]

| Success Factor | Implementation Approach | Critical Consideration | Recommendation |
|------------------------|--|---|--|
| Legacy Assessment | Comprehensive documentation, stakeholder interviews | Hidden dependencies, undocumented features | Allocate sufficient discovery time |
| Migration Strategy | Iterative, phased approach | Risk mitigation, early validation | Avoid big-bang cutover |
| Stakeholder Engagement | Regular communication, demonstration sessions | Expectation alignment, change management | Maintain continuous involvement |
| Automation Investment | CI/CD pipelines, testing frameworks, and monitoring | Initial effort, long-term returns | Prioritize automation infrastructure |
| Documentation | Technical guides, decision records, operational procedures | Knowledge preservation, team transitions | Document continuously, not retrospectively |
| Security Integration | Design-phase incorporation, continuous validation | Compliance requirements, risk prevention | Address security from project inception |
| Performance Validation | Representative workload testing, baseline comparison | Capacity planning, optimization opportunities | Test before production deployment |
| Skill Development | Training programs, hands-on experience | Team capability, sustainable operations | Invest in continuous learning |
| Vendor Partnership | Regular engagement, architectural guidance | Technical support, platform capabilities | Cultivate strategic vendor relationships |
| Change Management | Process documentation, user training, support | Organizational adaptation, workflow changes | Address people and process dimensions |

4. Conclusions

This modernization proved that cloud-native ETL can meet healthcare performance, cost, and compliance demands simultaneously when designed around isolation, automation, and evidence-driven tuning. By containerizing Informatica workloads on ECS, orchestrating with Step Functions, and promoting infrastructure through GitOps, we turned a fragile, long-running batch system into a fast, repeatable, and auditable platform. The results are concrete: ~34h → ~2.5h runtime reduction (~90% faster) and ~\$183 → ~\$18 per-run cost reduction (~85% lower), validated through runtime and billing logs. Beyond metrics, the most lasting outcome is a reusable modernization pattern that other regulated healthcare teams can adopt: small, measurable

migrations; declarative orchestration; strict version control; and verifiable operational data. Future work will extend this framework to additional ETL domains, deepen automated policy testing (security + data quality), and explore more adaptive orchestration based on workload telemetry. The program shows that when cloud migration is treated as engineering modernization rather than infrastructure relocation, the gains are both credible and durable. Cloud computing is applied to different fields [11-26].

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

- [1] Lipsa Das et al., "Data-Driven Healthcare Management, Analysis, and Future Trends," in 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), 23 July 2024. <https://ieeexplore.ieee.org/document/10593217>
- [2] S. P. Ahuja and B. Moore, "A Survey of Cloud Computing and Social Networks," *Network and Communication Technologies*, vol. 2, no. 2, pp. 11-16, 2013. DOI: 10.5539/nct.v2n2p11 https://www.researchgate.net/publication/271060474_A_Survey_of_Cloud_Computing_and_Social_Networks
- [3] Hicham Boudlal, et al., "Cloud Computing for Healthcare Services: Technology, Opportunities, and Challenges," in 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC), 22 June 2022. <https://ieeexplore.ieee.org/document/9800212>
- [4] Florian Beetz and Simon Harrer, "GitOps: The Evolution of DevOps?," *IEEE Software*, 08 October 2021. <https://ieeexplore.ieee.org/document/9565152>
- [5] K. Morris, "Infrastructure as Code: Managing Servers in the Cloud," O'Reilly Media, 2016. ISBN: 978-1491924358 <https://dl.ebooksworld.ir/books/Infrastructure.as.Code.2nd.Edition.Kief.Morris.OReilly.9781098114671.EBooksWorld.ir.pdf>
- [6] M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems," O'Reilly Media, 2017. ISBN: 978-1449373320 <https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/>
- [7] Rafael Vaño, et al., "Cloud-native workload orchestration at the edge: A deployment review and future directions," *Sensors*, 23(4), 2215, 15 February 2023. <https://www.mdpi.com/1424-8220/23/4/2215>
- [8] Bruno Nascimento, et al., "Availability, scalability, and security in the migration from container-based to cloud-native applications," *Computers*, 13(8), 192, 9 August 2024. <https://www.mdpi.com/2073-431X/13/8/192>
- [9] C. Pahl and P. Jamshidi, "Microservices: A Systematic Mapping Study," in Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER), 2016, pp. 137-146. DOI: 10.5220/0005785501370146 https://www.researchgate.net/publication/302973857_Microservices_A_Systematic_Mapping_Study
- [10] P. Zhang, J. K. Chiang, and A. Dey, "A Comprehensive Survey on Cloud Computing Research," in 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), 2014, pp. 381-386. DOI: 10.1109/CloudNet.2014.6969019 https://www.researchgate.net/publication/282802256_A_Comprehensive_Survey_on_Cloud_Computing
- [11] Fabiano de Abreu Agrela Rodrigues, & Flávio Henrique dos Santos Nascimento. (2025). Neurobiology of perfectionism. *International Journal of Sustainable Science and Technology*, 3(1). <https://doi.org/10.22399/ijssusat.6>
- [12] S. Jagan, B. Girirajan, Manisha Bhimrao Mane, R B, H. J., Mariam Anil, & M. Thillai Rani. (2025). Adaptive Quantum AI Models for Accelerating Deep Learning in Decentralized Cloud Architectures. *International Journal of Computational and Experimental Science and Engineering*, 11(3). <https://doi.org/10.22399/ijcesen.2493>
- [13] Soyal, H., & Canpolat, M. (2025). Intersections of Ergonomics and Radiation Safety in Interventional Radiology. *International Journal of Sustainable Science and Technology*, 3(1). <https://doi.org/10.22399/ijssusat.12>
- [14] Ankit, & Amritpal Singh. (2025). Optimized Architecture for Efficient VM Allocation and Migration in Cloud Environments. *International Journal of Computational and Experimental Science and Engineering*, 11(2). <https://doi.org/10.22399/ijcesen.1466>
- [15] García, R. (2025). Optimization in the Geometric Design of Solar Collectors Using Generative AI Models (GANs). *International Journal of Applied Sciences and Radiation Research*, 2(1). <https://doi.org/10.22399/ijasrar.32>
- [16] Vishwanath Pradeep Bodduluri. (2025). Social Media Addiction and Its Overlay with Mental Disorders: A Neurobiological Approach to the Brain Subregions Involved. *International Journal of Sustainable Science and Technology*, 3(1). <https://doi.org/10.22399/ijssusat.3>
- [17] Ujjwal Raj. (2025). The Serverless Paradigm: Abstraction, Elasticity, and Event-Driven Computing in Modern Cloud Architectures. *International Journal of Computational and Experimental Science and Engineering*, 11(4). <https://doi.org/10.22399/ijcesen.4088>
- [18] Harsha Patil, Vikas Mahandule, Rutuja Katala, & Shamal Ambalkar. (2025). Leveraging Machine

- Learning Analytics for Intelligent Transport System Optimization in Smart Cities. *International Journal of Applied Sciences and Radiation Research* , 2(1). <https://doi.org/10.22399/ijasrar.38>
- [19] Jhansi Rani Ganapa, Poonam Joshi, T Amitha, Sandip Rahane, N. Ravinder, Jignesh Jani, ... Chandreshkumar Vyas. (2025). Security and Privacy Challenges in Deep Learning Models Hosted on Cloud Platforms. *International Journal of Computational and Experimental Science and Engineering*, 11(3). <https://doi.org/10.22399/ijcesen.3235>
- [20] Chui, K. T. (2025). Artificial Intelligence in Energy Sustainability: Predicting, Analyzing, and Optimizing Consumption Trends. *International Journal of Sustainable Science and Technology*, 3(1). <https://doi.org/10.22399/ijsusat.1>
- [21] V. Ananthakrishna, & Chandra Shekhar Yadav. (2025). QP-ChainSZKP: A Quantum-Proof Blockchain Framework for Scalable and Secure Cloud Applications. *International Journal of Computational and Experimental Science and Engineering*, 11(1). <https://doi.org/10.22399/ijcesen.718>
- [22] Madane, S., Kamble, V., & Chavan, G. (2025). Cyber Chain – Merging Blockchain with Cyber Security. *International Journal of Applied Sciences and Radiation Research* , 2(1). <https://doi.org/10.22399/ijasrar.42>
- [23] Kumari, S. (2025). Machine Learning Applications in Cryptocurrency: Detection, Prediction, and Behavioral Analysis of Bitcoin Market and Scam Activities in the USA. *International Journal of Sustainable Science and Technology*, 3(1). <https://doi.org/10.22399/ijsusat.8>
- [24] Olola, T. M., & Olatunde, T. I. (2025). Artificial Intelligence in Financial and Supply Chain Optimization: Predictive Analytics for Business Growth and Market Stability in The USA. *International Journal of Applied Sciences and Radiation Research* , 2(1). <https://doi.org/10.22399/ijasrar.18>
- [25] Madhavi Mangalarapua. (2025). Evaluation of DNA damage and repair in Radiographers and Dental Surgeons using X-ray machines in Dental Clinics. *International Journal of Natural-Applied Sciences and Engineering*, 3(1). <https://doi.org/10.22399/ijnasen.14>
- [26] Ibeh, C. V., & Adegbola, A. (2025). AI and Machine Learning for Sustainable Energy: Predictive Modelling, Optimization and Socioeconomic Impact In The USA. *International Journal of Applied Sciences and Radiation Research* , 2(1). <https://doi.org/10.22399/ijasrar.19>