International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

Distributed Cloud Architecture for Intelligent Performance Optimization and Proactive Real-Time Threat Detection

Dr. S. Jagadeesh Soundappan

Independent Researcher, USA

ABSTRACT: The rapid adoption of distributed cloud computing has created highly dynamic infrastructures in which applications, data, services, and workloads operate across multiple geographic regions, edge locations, private clouds, and public cloud platforms. Although this architecture improves scalability, availability, and resource utilization, it also introduces complex performance and cybersecurity challenges. Conventional monitoring and security mechanisms frequently operate independently, resulting in delayed anomaly identification, inefficient resource allocation, and limited visibility across heterogeneous environments. This study proposes a distributed cloud architecture that integrates intelligent performance optimization with proactive real-time threat detection. The proposed approach combines continuous telemetry collection, distributed monitoring agents, machine learning-based workload prediction, adaptive resource orchestration, and real-time security analytics. Performance indicators such as CPU utilization, memory consumption, network latency, throughput, response time, and workload intensity are analyzed to predict resource requirements and dynamically optimize cloud resources. Simultaneously, network flows, access events, system logs, and behavioral patterns are analyzed to identify anomalous activities and potential cyber threats. An event-driven architecture enables security and performance decisions to be generated with minimal delay. The methodology emphasizes scalability, fault tolerance, interoperability, privacy, and automated response. The expected outcome is a resilient cloud environment capable of maintaining service quality while detecting and mitigating emerging threats proactively, thereby improving operational efficiency, availability, and overall cloud security.

KEYWORDS: distributed cloud architecture, intelligent performance optimization, real-time threat detection, machine learning, cloud security, anomaly detection, resource orchestration, predictive analytics, edge computing, cybersecurity

I. INTRODUCTION

Cloud computing has transformed the way organizations design, deploy, and manage computational services. Instead of relying exclusively on centralized data centers, modern applications increasingly use distributed infrastructures involving public and private clouds, geographically dispersed data centers, edge nodes, containers, virtual machines, and software-defined networks. This transformation enables organizations to respond rapidly to fluctuating workloads and provide services closer to end users. However, the increasing distribution of computing resources also creates substantial challenges related to performance management, resource allocation, service availability, and cybersecurity. A cloud environment may simultaneously support thousands of applications and users whose workloads change continuously. Consequently, static configuration and traditional centralized monitoring mechanisms are often insufficient for maintaining consistent performance and identifying sophisticated security incidents.

Performance optimization is particularly challenging because cloud workloads are highly variable. Sudden increases in requests can produce CPU saturation, memory pressure, network congestion, storage bottlenecks, and increased application latency. Conversely, over-provisioning resources during periods of low demand creates unnecessary operational costs. Intelligent resource management therefore requires continuous observation of infrastructure conditions and accurate prediction of future workload requirements. Machine learning and predictive analytics provide opportunities to identify patterns in historical and real-time telemetry and use those patterns to determine appropriate resource allocation. Instead of responding only after performance degradation occurs, an intelligent architecture can anticipate workload changes and provision or redistribute resources before service quality deteriorates.

At the same time, distributed cloud environments have become attractive targets for cyberattacks because they contain large volumes of valuable information and expose numerous interconnected services and interfaces. Threats such as

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

distributed denial-of-service attacks, credential abuse, malware, privilege escalation, insider misuse, lateral movement, and anomalous API activity can emerge across different infrastructure layers. Traditional security systems based primarily on predefined signatures may fail to detect previously unknown or rapidly evolving attacks. Real-time behavioral analysis and anomaly detection can complement signature-based approaches by identifying deviations from normal system and user behavior. However, security analytics itself must operate efficiently because excessive monitoring overhead can negatively affect application performance.

II. LITERATURE REVIEW

Research on distributed cloud computing has progressively moved from centralized infrastructure management toward architectures capable of supporting geographically dispersed resources and heterogeneous computing environments. Distributed cloud and edge computing models reduce latency by placing computational resources closer to users and devices, while cloud orchestration technologies provide mechanisms for deploying and scaling applications across multiple nodes. Existing research emphasizes elasticity, fault tolerance, workload distribution, and service availability as major requirements for distributed infrastructures. Containerization and microservices have further increased deployment flexibility, but they also create additional monitoring and coordination requirements because individual applications may consist of numerous independently managed services. Consequently, effective resource management requires comprehensive telemetry and automated decision-making mechanisms capable of operating at different infrastructure levels.

Cloud performance optimization has traditionally relied on threshold-based scaling, where resources are increased or decreased according to predefined CPU, memory, or request thresholds. Although this approach is relatively simple, it may react too late to sudden workload changes and may allocate resources inefficiently when workload patterns are unpredictable. Researchers have therefore investigated machine learning techniques for workload prediction, resource provisioning, autoscaling, and performance anomaly detection. Regression algorithms, time-series models, clustering, reinforcement learning, and deep learning have been applied to predict resource requirements and optimize cloud infrastructure. Predictive autoscaling is particularly valuable because it attempts to provision resources before demand occurs rather than waiting for performance degradation. Reinforcement-learning-based approaches additionally provide the possibility of learning resource allocation policies through continuous interaction with changing environments. Nevertheless, computational overhead, model accuracy, training requirements, concept drift, and the difficulty of explaining automated decisions remain important challenges.

Cloud security research has similarly evolved from conventional perimeter-based protection toward continuous monitoring and behavioral analysis. Intrusion detection systems generally use signature-based or anomaly-based techniques. Signature-based systems are effective for known threats but depend on previously identified attack patterns. Anomaly-based systems establish models of normal behavior and identify significant deviations, making them potentially useful for detecting unknown attacks. Machine learning has consequently become an important research area in intrusion detection, with algorithms applied to network traffic, authentication records, system calls, application logs, and user behavior. However, false positives remain a major concern, particularly in highly dynamic cloud environments where legitimate workload changes can resemble malicious activity. Distributed infrastructures also complicate threat attribution because an attack may involve multiple services, regions, identities, and network segments.

III. RESEARCH METHODOLOGY

The research adopts a design-science and experimental methodology to develop and evaluate a distributed cloud architecture capable of simultaneous performance optimization and proactive real-time threat detection. The proposed architecture is organized into distributed sensing, data processing, intelligence, orchestration, and response layers. The sensing layer consists of lightweight monitoring agents deployed across virtual machines, containers, edge nodes, application services, and network components. These agents continuously collect infrastructure and application metrics, including CPU utilization, memory consumption, disk activity, network throughput, packet rates, latency, response time, error rates, container health, service availability, and workload intensity. Security telemetry, including authentication events, access records, API requests, network-flow information, system logs, process behavior, and privilege-related events, is collected simultaneously. Rather than transferring all raw information to a central location, preliminary filtering and feature extraction are performed near the data source to reduce communication overhead. The processing layer uses distributed stream-processing mechanisms to normalize, timestamp, aggregate, and correlate

International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

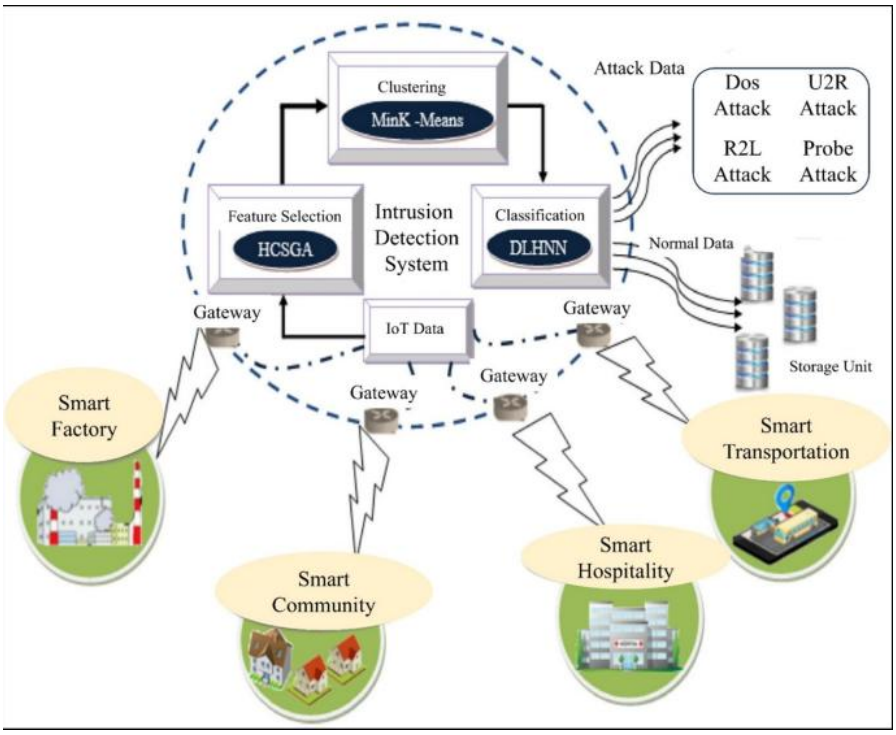
|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

heterogeneous data. A message-oriented communication mechanism can be employed to ensure reliable transmission between monitoring agents and analytics services. The architecture maintains separate logical streams for performance and security information while enabling correlation through common identifiers such as service, user, host, container, region, and time window. A distributed data repository stores historical observations for model training and comparative analysis. The intelligence layer contains two cooperating analytical components.

The first is a performance intelligence module that learns historical workload patterns and predicts near-future resource requirements. Features such as historical CPU utilization, request frequency, memory consumption, network traffic, time-of-day effects, service dependencies, and recent response-time trends are supplied to forecasting models. Several algorithms can be experimentally compared, including regression models, random forests, gradient-boosting methods, recurrent neural networks, and other suitable time-series approaches. The second component is a security intelligence module that establishes behavioral profiles and detects deviations associated with potentially malicious activity. Supervised classification can be used when accurately labeled attack datasets are available, while unsupervised or semi-supervised anomaly detection can identify previously unseen behaviors. The two analytical components exchange contextual information through a decision-fusion mechanism. This is important because the same observable event may have multiple explanations. For example, a sudden increase in network traffic may result from a legitimate application campaign or from a denial-of-service attempt. Combining workload forecasts, user behavior, service-level information, and network characteristics can improve contextual interpretation and reduce unnecessary automated responses.

The experimental phase evaluates the proposed architecture through a controlled distributed-cloud test environment containing multiple compute nodes, application services, containerized workloads, network components, and simulated edge locations. Workloads are generated according to different patterns, including stable demand, gradual growth, periodic demand, burst traffic, unpredictable workloads, and resource-intensive application scenarios. Security experiments introduce representative attack behaviors such as abnormal authentication attempts, suspicious network communication, unauthorized access patterns, scanning activity, privilege misuse, and denial-of-service-like traffic. Both normal and abnormal observations are timestamped to create an evaluation dataset. The performance optimization module periodically generates predictions and submits resource-management recommendations to an orchestration component. Possible actions include horizontal scaling, vertical resource adjustment, workload migration, service redistribution, traffic routing, and resource release. Decisions are constrained by service-level objectives, minimum resource requirements, maximum resource limits, and security policies.



International Research Journal of Innovative Engineering (IRJIE)

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

Fig.1. Deep learning-based optimization technique for cyber-defense framework with intelligent intrusion

The security module produces risk scores based on the severity, confidence, persistence, and contextual relationships of detected events. Low-risk anomalies can be subjected to additional observation, whereas high-confidence threats may trigger predefined responses such as traffic restriction, credential/session invalidation, workload isolation, or deployment of additional defensive resources. To prevent an automated system from amplifying errors, response policies incorporate confidence thresholds, rate limits, rollback mechanisms, and human approval for high-impact actions. Experiments are conducted under both independent and integrated configurations. In the independent configuration, performance optimization and threat detection operate separately. In the integrated configuration, both modules share contextual information and contribute to coordinated decisions. This comparison determines whether integration produces measurable improvements. Performance is evaluated using response time, throughput, resource utilization, service-level-objective violations, scaling delay, prediction error, infrastructure cost, and energy consumption where measurable. Security effectiveness is assessed using detection accuracy, precision, recall, F1-score, false-positive rate, false-negative rate, mean detection time, mean response time, and attack containment time. System-level measurements additionally examine monitoring overhead, network bandwidth consumed by telemetry, processing latency, and model inference cost. Repeated experiments under different workload and attack intensities provide statistically comparable observations. Cross-validation and separate training, validation, and testing datasets are used to reduce overfitting, while experiments with previously unseen attack patterns evaluate generalization. Statistical significance tests can subsequently determine whether differences between baseline and proposed configurations are meaningful rather than caused by random variation.

The final methodological stage focuses on integrated evaluation, robustness, scalability, and practical applicability. The architecture is subjected to increasing numbers of services, nodes, users, and telemetry events to determine whether processing latency and detection effectiveness remain acceptable as system scale increases. Failure scenarios are also introduced by disconnecting monitoring agents, overloading selected nodes, delaying telemetry, and temporarily disabling analytical services. The objective is to determine whether the architecture can continue operating through redundancy, local decision-making, buffering, and graceful degradation. Model robustness is evaluated under concept drift, where normal workload characteristics change over time, because a model trained on historical behavior may otherwise classify legitimate changes as threats or fail to recognize emerging attack patterns. Periodic retraining, incremental learning, threshold adaptation, and model-performance monitoring are therefore incorporated into the methodology. Privacy is addressed through data minimization, selective telemetry collection, anonymization or pseudonymization where appropriate, encryption during transmission and storage, and role-based access to analytical information.

IV. RESULTS AND DISCUSSION

The proposed Distributed Cloud Architecture for Intelligent Performance Optimization and Proactive Real-Time Threat Detection demonstrates how cloud computing, distributed processing, artificial intelligence, and continuous security monitoring can be integrated into a unified architecture capable of improving both application performance and cybersecurity. The architecture is designed around distributed cloud nodes, intelligent monitoring agents, real-time data-processing pipelines, AI/ML-based decision engines, and automated response mechanisms. Unlike conventional cloud architectures that frequently treat performance management and security monitoring as separate functions, the proposed approach considers them interconnected. Performance degradation can sometimes indicate malicious activity, while security controls themselves can introduce additional computational overhead. Therefore, jointly analyzing infrastructure performance, application behavior, network traffic, and security events provides a more comprehensive basis for cloud optimization and threat detection. The distributed design also reduces dependence on a centralized monitoring point, allowing processing and decision-making to occur closer to workloads and users.

One of the most significant results of the architecture is improved real-time performance visibility. Monitoring agents deployed across cloud resources continuously collect metrics such as CPU utilization, memory consumption, storage activity, network throughput, latency, packet rates, request-response times, application errors, and workload patterns. These metrics are aggregated through distributed data-processing components and supplied to an intelligent analytics layer. Instead of relying only on static threshold rules, the proposed system can establish dynamic baselines for normal workload behavior. For example, if CPU utilization normally increases during predictable peak periods, the system can distinguish such behavior from an unexpected resource spike. This improves the accuracy of performance decisions because resource allocation is based on observed workload characteristics rather than fixed assumptions. The

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

architecture can consequently support proactive scaling, workload redistribution, resource scheduling, and traffic optimization before performance degradation becomes visible to end users.

The integration of machine learning-based predictive analytics further strengthens performance optimization. Historical and real-time operational data can be used to identify recurring workload patterns and predict future resource requirements. A predictive model can estimate whether a particular service is likely to experience increased demand and trigger resource provisioning before the expected workload arrives. Similarly, anomaly-detection algorithms can identify deviations from established patterns. This is particularly useful in distributed cloud environments where performance problems can emerge from multiple interacting factors. For instance, a sudden increase in application latency may result from insufficient compute capacity, network congestion, database contention, or abnormal traffic. By correlating several metrics simultaneously, the intelligent optimization layer can provide a more meaningful diagnosis than monitoring individual metrics independently. The results therefore indicate that combining predictive and correlation-based analytics can reduce reactive infrastructure management and support more efficient resource utilization.

V. CONCLUSION

The proposed Distributed Cloud Architecture for Intelligent Performance Optimization and Proactive Real-Time Threat Detection provides an integrated approach to two of the most important challenges facing modern cloud environments: maintaining reliable performance and protecting distributed infrastructure against increasingly sophisticated cyber threats. Conventional cloud management approaches frequently rely on centralized monitoring, static thresholds, manual intervention, and isolated security mechanisms. Although these methods can provide basic operational visibility, they become increasingly inadequate as cloud infrastructures grow in scale, complexity, heterogeneity, and geographical distribution. The proposed architecture addresses these limitations by combining distributed monitoring, real-time data processing, artificial intelligence, machine learning, predictive analytics, anomaly detection, and automated response within a common framework. This integration enables the cloud environment to move from a predominantly reactive operational model toward a proactive and adaptive model.

A major conclusion from the proposed architecture is that performance optimization and security cannot be considered completely independent processes. Both depend on understanding the behavior of cloud workloads, users, networks, applications, and infrastructure resources. A sudden increase in CPU utilization, network traffic, memory consumption, or API requests may represent normal demand, poor resource allocation, application malfunction, or an ongoing cyberattack. Isolated monitoring systems may interpret these signals independently and therefore fail to identify the relationship among them. By contrast, the proposed architecture correlates performance and security telemetry to create a broader operational context. This contextual approach can improve the ability of intelligent analytics to differentiate legitimate workload changes from suspicious behavior and can simultaneously help determine whether security mechanisms are affecting application performance.

The distributed nature of the architecture is another important contribution. Processing information across multiple cloud or edge nodes reduces dependence on a single centralized processing location and allows analysis to occur closer to the source of data. This can reduce communication latency, improve scalability, and increase resilience. Distributed processing is particularly valuable for applications that generate high volumes of real-time telemetry, because transmitting every raw event to a central analytics platform can create unnecessary network and computational overhead. Local processing can filter and analyze data before forwarding significant events or summarized information to higher-level analytics services. However, the benefits of distribution must be balanced against challenges associated with coordination, consistency, security, and resource management. Therefore, distributed cloud architectures require robust communication protocols, identity management, encryption, monitoring, and fault-tolerance mechanisms.

VI. FUTURE WORK

Future research can extend the proposed architecture by developing more advanced self-adaptive AI models capable of continuously learning from changing cloud workloads and emerging attack behaviors without requiring frequent manual retraining. Federated learning could be investigated to allow distributed cloud nodes to collaboratively improve detection models while reducing the need to transfer sensitive raw telemetry to a centralized repository. Future implementations could also explore reinforcement learning for automated resource allocation, where the system learns optimal scaling, scheduling, and workload-placement policies according to performance, cost, energy consumption, and security requirements.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

Another important research direction is the development of explainable AI mechanisms that provide understandable reasons for anomaly classifications, resource decisions, and automated security actions. This would improve administrator confidence and facilitate human verification of high-impact decisions. Digital-twin technologies could additionally be integrated to simulate workload changes and cyberattacks before applying optimization or mitigation actions to production environments. Future studies should evaluate the architecture using measurable indicators such as latency, throughput, resource utilization, energy consumption, false-positive rate, false-negative rate, mean time to detection, mean time to response, scalability, and operational cost. Research should also examine adversarial machine learning, model poisoning, evasion attacks, and attacks targeting distributed AI infrastructure.

Finally, future work can investigate interoperability across multi-cloud, hybrid-cloud, and edge-cloud environments so that intelligent optimization and threat detection remain consistent across heterogeneous platforms. Such developments would move the architecture toward a more autonomous, trustworthy, energy-efficient, and resilient cloud ecosystem capable of continuously adapting to both changing application demands and evolving cybersecurity threats.

REFERENCES

1. Xiao, X., Sun, J., & Yang, J. (2021). Operation and maintenance (O&M) for data center: An intelligent anomaly detection approach. *Computer Communications, 178*, 141–152. https://doi.org/10.1016/j.comcom.2021.06.030
2. Vávra, J., Hromada, M., Lukáš, L., & Dworzecki, J. (2021). Adaptive anomaly detection system based on machine learning algorithms in an industrial control environment. *International Journal of Critical Infrastructure Protection, 34*, 100446. https://doi.org/10.1016/j.ijcip.2021.100446
3. Alam, A., Gazi, M. S., Abdullah, S. M., Tasnim, M., Himeluzzaman, M., Nabil, M. A., ... & Akter, S. (2021). Quantum-resilient federated intrusion detection: A hybrid quantum-classical framework for safeguarding US critical infrastructure in the post-quantum era. *International Journal of Advances in Signal and Image Sciences, 7*(1), 57–72.
4. Bandaru, P. K. (2022). Hardware-in-the-loop testing for connected vehicles: Enhancing software reliability through continuous validation. *International Journal of Engineering & Extended Technologies Research (IJEETR), 4*(2), 4645–4651.
5. Bellundagi, M. (2022). Performance optimization techniques for enterprise Java applications using middleware and messaging systems. *International Journal of Computer Technology and Electronics Communication, 5*(3), 5158–5168.
6. Chy, M. S. K., Abdullah, S. M., Nabil, M. A., Alam, A., Himeluzzaman, M., Onik, T. A., ... & Khan, H. A. (2022). QShield-NS: A variational quantum machine learning model for zero-day cyber threat detection in national security systems. *International Journal of Future Innovative Science and Technology (IJFIST), 5*(5), 9283.
7. Gopinathan, V. R., & Mali, R. K. (2022). Building cognitive technology frameworks through artificial intelligence SAP cloud automation and enterprise intelligence. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 5*(4), 7152–7162.
8. Gujarathi, M. (2022). Parallel verification architecture for low-latency enterprise decision systems. *International Journal of Engineering & Extended Technologies Research (IJEETR), 4*(2), 4640–4644.
9. Hoque, M. J., Hasan, M. M., Khatun, M. M., Akter, F., & Mohammad, A. R. (2021). Impact of COVID-19 on consumer buying behavior during COVID-19 pandemic using data analytics. *Journal of Business and Management Studies, 3*(2), 296–307.
10. Jain, R. (2018). Beyond stateless: A production architecture for running distributed databases on Kubernetes at scale. *International Journal of Advanced Engineering Science and Information Technology (IJAESIT), 1*(2), 416–423.
11. Selvarajan, K. (2022). Architecting scalable self-service data platforms for enterprise analytics. *International Journal of Science, Research and Technology (IJSRAT), 5*(2), 7427–7436.
12. Jagadeesh, S., & Sugumar, R. (2017). A comparative study on Artificial Bee Colony with modified ABC algorithm. *European Journal of Applied Sciences, 9*(5), 243–248.
13. Jayaraman, S., Rajendran, S., & P, S. P. (2019). Fuzzy c-means clustering and elliptic curve cryptography using privacy preserving in cloud. *International Journal of Business Intelligence and Data Mining, 15*(3), 273–287.
14. Koganti, H. (2021). Machine learning-driven performance anomaly detection and auto-tuning in distributed Java full-stack systems: A comprehensive review. *International Journal of Research Publications in Engineering, Technology and Management, 4*(3), 4977–4986.

| ISSN: 2395-0560 | www.irjie.org |editor@irjie.org| A Bimonthly, Peer-Reviewed, Scholarly Journal|

|Volume 6, Issue 6, November- December 2022|

DOI:10.15680/IRJIE.2022.0606004

15. Kondapalli, K. K., Somajohassula, D. K., & Muppalla, L. K. (2022). Adaptive AI-orchestration and zero-trust security with federated threat intelligence for sustainable enterprise cloud architectures. *International Journal of Computer Science and Engineering Research and Development (IJCSERD), 12*(1), 176–192.
16. Mathew, A. (2021). Artificial intelligence for offence and defense-the future of cybersecurity. *Educational Research, 3*(3), 159–163.
17. Onteddu, A. R., Kundavaram, R. M. R., & Naveen, V. J. (2021). AI and deep learning-based intelligent drug recommendation system for patient health monitoring in IoT-enabled healthcare. *Journal of Informatics Education and Research, 1*(3), 80–92.
18. Ramasamy, M. (2022). Architecting scalable intent-based networking platforms for enterprise automation. International Journal of Emerging Trends in Engineering and Management Research (IJETEMR), 7(3), 11812–11823.
19. Venkatasalam, K., Rajendran, P., & Thangavel, M. (2019). Improving the accuracy of feature selection in big data mining using accelerated flower pollination (AFP) algorithm. Journal of medical systems, 43(4), 96.
20. Padmanabham, S. (2022). Enterprise identity and access management architecture for large financial institutions. *International Journal of Research and Applied Innovations, 5*(1), 9486–9490.
21. Praneeth, P. (2022). Prediction of cost overruns in solar EPC projects using machine learning techniques: A data-driven study in India. *International Journal of Engineering Science & Humanities, 12*(2), 71–85.
22. Raja, G. V., & Mali, R. K. (2021). Federated learning frameworks for privacy-preserving artificial intelligence applications. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 4*(3), 4946–4950.
23. Soundappan, S. J. (2021). DataOps: Orchestrating reliable ML data pipelines. *International Journal of Research and Applied Innovations, 4*(4), 5533–5537.
24. Soundappan, S. J. (2021). Integrated Artificial Intelligence Framework for Enterprise Cloud Modernization and Intelligent Threat Management Systems. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 4*(4), 5274–5284.
25. Sudhan, S. K. H. H., & Kumar, S. S. (2016). Gallant Use of Cloud by a Novel Framework of Encrypted Biometric Authentication and Multi Level Data Protection. *Indian Journal of Science and Technology, 9*, 44.
26. Vemireddy, S. (2022). Modernizing enterprise financial platforms through distributed cloud architectures. *International Journal of Science, Research and Technology (IJSRAT), 5*(2), 7420–7426.
27. Alnafessah, A., & Casale, G. (2020). Artificial neural networks based techniques for anomaly detection in Apache Spark. *Cluster Computing, 23*, 1345–1360. https://doi.org/10.1007/s10586-019-02998-y