

Advanced Computing Techniques for Real-Time Data Processing and High-Performance Computing

Ahmed Mustafa¹ Nadia Hameed²

Department of Computer Science and Engineering, Bahauddin Zakariya University, Multan, Pakistan¹, Department of Computer Vision²
amustafa@bzu-fict.edu.pk¹ nhameed@bzu-fict.edu.pk²

DOI: 10.69987/JACS.2023.30801

Keywords

Real-Time Data Processing, High-Performance Computing, Parallel Processing, Distributed Systems, Edge Computing

Abstract

Real-time data processing and high-performance computing (HPC) have become essential components of modern applications across various industries, including healthcare, finance, telecommunications, and autonomous systems. The increasing volume, variety, and velocity of data generated by the Internet of Things (IoT), artificial intelligence (AI), and big data analytics have created new demands for computing systems that can process vast datasets in real time. Traditional computing architectures often struggle to meet these requirements, necessitating the adoption of advanced computing techniques for efficient real-time data processing and enhanced performance. This paper explores advanced computing techniques that are shaping the landscape of real-time data processing and HPC, including parallel processing, distributed computing, in-memory computing, and edge computing. Each of these methodologies offers unique benefits, such as improved scalability, reduced latency, and enhanced fault tolerance. Furthermore, emerging technologies such as AI and quantum computing present new opportunities for enhancing real-time processing capabilities. In addition to examining key techniques, this research addresses the challenges faced by real-time data processing systems, such as data management at scale, minimizing latency, ensuring fault tolerance, and optimizing energy efficiency. The integration of novel solutions, such as edge computing to minimize data transfer delays and in-memory computing for faster access to data, is also discussed. This paper provides a comprehensive overview of the advancements in computing architectures and strategies that enable real-time data processing and high-performance computing. Through detailed analysis, the research offers insights into current trends, the critical challenges faced, and future directions for optimizing real-time and HPC systems to meet the demands of increasingly complex applications. Three tables are presented to summarize techniques, their applications, and the challenges they address in the context of real-time computing and HPC.

1. Introduction

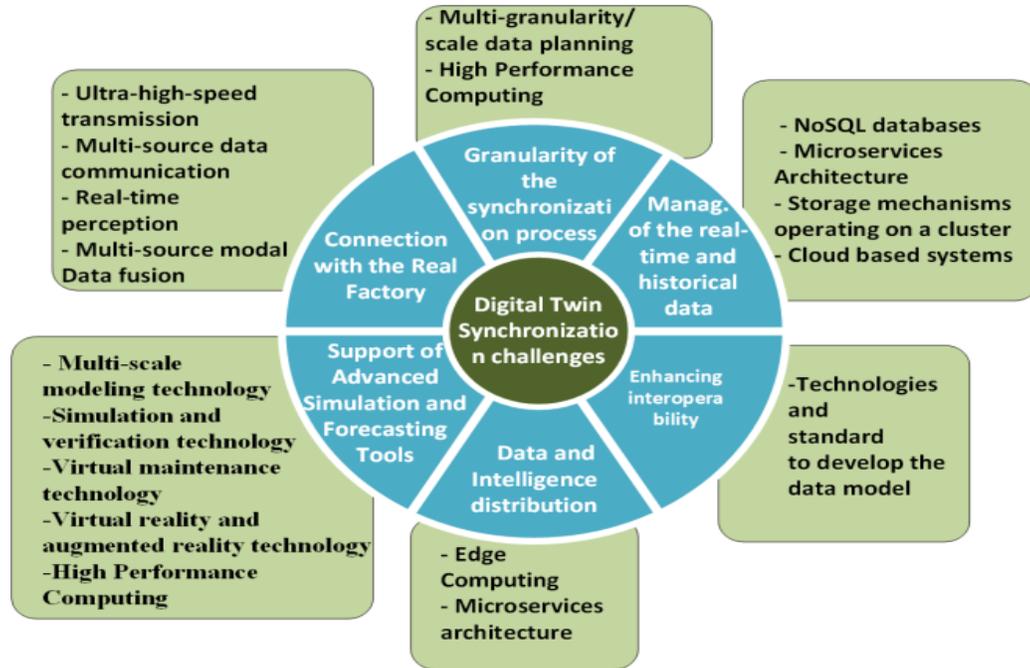
Real-time data processing and high-performance computing (HPC) have become foundational pillars in a wide array of applications, from scientific research to financial services, telecommunications, and autonomous systems. The rapid growth in data volumes, driven by the Internet of Things (IoT), artificial intelligence (AI), and big data analytics, demands computing systems capable of processing massive amounts of data in real time. Traditional computing

architectures often fall short when faced with the scale, complexity, and speed required by modern applications. Therefore, advanced computing techniques are essential to meet the real-time processing demands without sacrificing efficiency, reliability, or scalability.

High-performance computing, traditionally associated with supercomputers and large-scale scientific simulations, is increasingly merging with real-time data processing systems. The convergence of these domains is driven by the need for real-time analytics in areas such as healthcare, financial markets, autonomous vehicles,

and large-scale environmental modeling. These applications demand both fast computation and the ability to handle massive data streams instantaneously.

This paper provides an in-depth exploration of the advanced computing techniques that are shaping the future of real-time data processing and HPC [1].



In this research, we examine key technologies such as parallel processing, distributed computing, in-memory computing, and edge computing, which have revolutionized the field [2]. We will also analyze the growing role of AI and quantum computing in enhancing real-time data processing capabilities. Additionally, we discuss the challenges that these systems face, such as latency, fault tolerance, and the need for energy-efficient computing architectures. Through detailed case studies, technical analyses, and comparisons of different methodologies, this research provides a roadmap for future advancements in the field of real-time data processing and HPC [3].

2. Evolution of High-Performance Computing and Real-Time Data Processing

The history of high-performance computing is rooted in the development of supercomputers, which were initially designed for complex scientific and military simulations. Over the decades, these systems have evolved to handle increasingly larger datasets and more sophisticated models, particularly with the advent of parallel processing techniques and distributed systems. As the computational demands of real-time data

processing have grown, so has the need for combining these capabilities with high-performance computing [4].

2.1 Early Supercomputers and Parallel Processing

Early supercomputers, such as the **Cray-1** introduced in 1976, revolutionized computational capacity by utilizing **vector processing** techniques. These machines could process multiple data points simultaneously, allowing for faster computations than traditional scalar processors. However, even with these advancements, early supercomputers were limited by their centralized architectures, which struggled to scale efficiently with increasing computational demands [5].

The shift towards **parallel processing** was a major milestone in the evolution of HPC. By allowing multiple processors to work on different parts of a task simultaneously, parallel processing enabled supercomputers to achieve previously unimaginable levels of performance. **Massively parallel processors (MPPs)** and **symmetric multiprocessing (SMP)** systems became standard architectures for HPC by the late 1990s. These systems divided tasks among numerous processors, enabling them to solve complex problems such as climate modeling, fluid dynamics, and protein folding more efficiently.

Table 1: Evolution of High-Performance Computing Architectures

Era	Key Technologies	Characteristics	Limitations
-----	------------------	-----------------	-------------

1970s - Early 1980s	Vector Processing	Sequential operations with enhanced speed	Limited scalability
1980s - 1990s	Parallel Processing, MPP, SMP	Multiple processors working concurrently	High inter-processor communication overhead
2000s - 2010s	Distributed Systems, Grid Computing	Distributed workload across geographically distant nodes	Complex fault tolerance and synchronization
2010s - Present	Cloud Computing, Edge Computing	Virtualized resources with real-time scaling	Latency challenges, energy consumption

2.2 Rise of Distributed Systems and Grid Computing

In the late 1990s and early 2000s, as data volumes began to expand exponentially, HPC systems evolved to incorporate **distributed computing** architectures. Distributed systems allow multiple machines, often located in different geographic locations, to work together as a cohesive unit. This approach introduced significant scalability advantages, enabling HPC systems to handle larger datasets and more complex computations [6].

Grid computing emerged as a prominent model during this period, using multiple computer systems to form a virtual supercomputer. By pooling resources across a distributed network, grid computing enabled organizations to perform tasks that would have otherwise required expensive, dedicated supercomputers. The **SETI@home** project, which used a distributed network of home computers to analyze radio signals for signs of extraterrestrial life, is a well-known example of grid computing [7].

While distributed systems and grid computing provided a significant boost to the scalability of high-performance computing, they also introduced new challenges related to **fault tolerance**, **synchronization**, and **data consistency**. The reliance on network-based communication between nodes led to concerns over **latency**, particularly in applications requiring real-time processing.

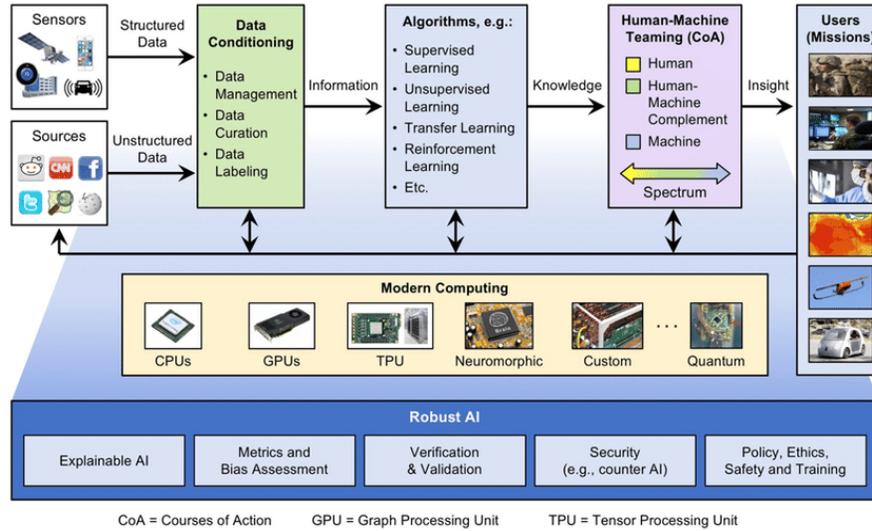
2.3 Emergence of Cloud Computing and Virtualization

The emergence of cloud computing and virtualization in the late 2000s marked a significant turning point in the landscape of high-performance computing (HPC) and

real-time data processing. This paradigm shift transformed the way organizations approach resource allocation, application deployment, and data management. By providing on-demand access to a virtually limitless pool of computational resources, cloud computing alleviated many of the constraints associated with traditional computing infrastructures, such as high costs, rigid resource allocation, and scalability limitations.

At the heart of cloud computing is the principle of **virtualization**, which allows multiple virtual machines (VMs) to run on a single physical server. Virtualization abstracts hardware resources, enabling them to be dynamically allocated based on workload demands. This technology enables organizations to optimize resource usage, as they can scale their computational resources up or down in response to changing workloads without incurring significant hardware costs [8]. For instance, during peak usage times, organizations can deploy additional VMs to handle increased demand and subsequently reduce resources during off-peak periods, thereby enhancing operational efficiency and cost-effectiveness.

Moreover, cloud computing facilitates collaboration and accessibility, as users can access applications and data from anywhere with an internet connection. This capability has been particularly beneficial in supporting remote work and global collaboration, allowing teams distributed across various geographical locations to work seamlessly on projects. Additionally, cloud platforms often offer advanced tools and services, such as machine learning, big data analytics, and managed database services, which empower organizations to leverage sophisticated technologies without needing extensive in-house expertise.



However, the shift to cloud computing has also introduced several challenges. **Latency** remains a critical concern, particularly for real-time applications where data needs to be processed quickly. Although cloud providers have invested in building data centers closer to end users to mitigate this issue, the reliance on network connections can still lead to delays that are unacceptable in certain scenarios, such as autonomous driving or high-frequency trading [9].

Security and data privacy are other significant challenges, as organizations must ensure their sensitive data is protected from breaches and unauthorized access in a shared cloud environment. Regulatory compliance issues, particularly in sectors such as healthcare and finance, also necessitate that organizations implement robust security measures and adhere to strict guidelines.

In conclusion, the emergence of cloud computing and virtualization has revolutionized the way organizations approach high-performance computing and real-time data processing. By providing scalable, flexible, and cost-effective resources, cloud platforms enable businesses to harness advanced computing techniques while navigating the complexities of modern data demands. However, organizations must remain vigilant in addressing the associated challenges, particularly concerning latency, security, and compliance, to fully realize the benefits of this transformative technology.

2.4 Edge Computing and the Shift Towards Real-Time Processing

The rise of **edge computing** has significantly impacted the landscape of real-time data processing, addressing the limitations of traditional cloud computing models. As the Internet of Things (IoT) continues to proliferate, with billions of devices generating vast amounts of data,

the need for real-time processing capabilities has become increasingly critical. Edge computing refers to the practice of processing data closer to the source, or “edge,” of data generation, rather than relying solely on centralized cloud servers. This paradigm shift is essential for enabling faster data analysis, reducing latency, and improving the overall efficiency of data processing workflows [10].

One of the primary advantages of edge computing is its ability to minimize latency. In traditional cloud computing architectures, data must be transmitted to a centralized data center for processing, which can introduce significant delays, particularly for applications requiring instantaneous responses, such as autonomous vehicles, remote surgery, and industrial automation. By performing computations at the edge, data can be analyzed and acted upon in real time, allowing for immediate insights and actions. For example, in smart manufacturing environments, edge devices can monitor machinery and make real-time adjustments to optimize performance, preventing costly downtime and enhancing operational efficiency [11].

Additionally, edge computing reduces the bandwidth requirements associated with data transmission to centralized servers. Transmitting large volumes of raw data can overwhelm network capacities and lead to increased operational costs. By processing data locally and transmitting only relevant insights or aggregated information to the cloud, organizations can optimize their network usage and reduce data transfer costs. This local processing also enhances privacy and security, as sensitive data can be processed on-site without needing to be sent to the cloud, mitigating potential exposure to cyber threats [12].

Furthermore, edge computing supports the scalability of real-time processing systems. As more IoT devices are deployed, the volume of data generated grows

exponentially. Edge computing allows organizations to distribute computing resources across multiple locations, enabling them to scale their processing capabilities effectively. This distributed architecture is particularly beneficial for applications in smart cities, where numerous sensors and devices operate concurrently, generating vast amounts of data that need to be processed and analyzed in real time.

Despite its numerous advantages, edge computing also presents challenges that organizations must address. The heterogeneity of edge devices and the lack of standardized protocols can complicate deployment and interoperability, making it challenging to integrate edge solutions with existing infrastructure. Additionally, managing and maintaining edge devices can be more complex than centralized systems due to their distributed nature, requiring robust monitoring and management solutions [13].

In summary, edge computing is playing a pivotal role in the shift toward real-time processing by enabling faster data analysis, reducing latency, and optimizing bandwidth usage. As the demand for immediate insights and actions grows across various sectors, edge computing will continue to be a critical component of modern computing architectures. Organizations that embrace this paradigm will be better positioned to harness the power of real-time data and leverage advanced analytics to drive innovation and operational efficiency.

3. Key Techniques in Advanced Computing for Real-Time Data Processing

Real-time data processing requires a combination of advanced computing techniques to achieve both high performance and low latency. The techniques used in real-time computing environments must be capable of handling large volumes of data efficiently while maintaining the ability to respond to incoming data streams in real time [14].

3.1 Parallel and Distributed Computing

Parallel computing, as discussed earlier, plays a critical role in real-time data processing by dividing tasks across multiple processors to reduce processing time. Distributed computing extends this concept by using multiple machines to process data concurrently. In real-time systems, both parallel and distributed computing are essential for managing workloads efficiently and reducing delays.

MapReduce and **Apache Spark** are prominent frameworks used in distributed computing environments to handle real-time data analytics. **MapReduce** enables large datasets to be processed in parallel by dividing the data into smaller chunks, while **Spark** enhances this capability with in-memory processing, significantly speeding up real-time data analysis.

One of the primary challenges in real-time parallel and distributed systems is **task scheduling**. Scheduling algorithms must ensure that tasks are distributed optimally across processors and machines to minimize idle time and maximize throughput. Techniques like **priority-based scheduling** and **load balancing** are commonly employed to address these challenges [15].

Table 2: Comparison of Parallel and Distributed Computing Techniques for Real-Time Data Processing

Technique	Application	Advantages	Challenges
MapReduce	Large-scale data processing, batch jobs	Scalable, efficient for batch processing	Limited real-time capabilities, disk I/O overhead
Apache Spark	Real-time analytics, machine learning	In-memory processing, faster than MapReduce	Memory-intensive, requires tuning for performance
Load Balancing	Distributed systems	Ensures even task distribution across nodes	Requires real-time monitoring and adjustments

3.2 In-Memory Computing

In-memory computing has emerged as a crucial technique for real-time data processing due to its ability to significantly reduce data access times. Traditional disk-based storage systems introduce latency due to the

time required to read and write data to disk. In-memory computing eliminates this bottleneck by storing data directly in the system's RAM, allowing for near-instantaneous data retrieval and processing [14].

Apache Ignite and **Redis** are examples of in-memory data grids used to speed up real-time processing by keeping frequently accessed data in memory. This technique is particularly beneficial for applications requiring low-latency data access, such as real-time analytics, fraud detection, and high-frequency trading.

While in-memory computing offers significant performance improvements, it also poses challenges related to **data persistence** and **fault tolerance**. Because data is stored in volatile memory, there is a risk of data loss in the event of a system failure. To mitigate this, many in-memory systems implement redundancy and backup mechanisms to ensure data durability [16].

3.3 Stream Processing and Complex Event Processing (CEP)

For real-time data streams, traditional batch processing techniques are inadequate. Instead, **stream processing** and **complex event processing (CEP)** frameworks are used to handle continuous data flows. Stream processing allows data to be processed as it arrives, rather than waiting for entire datasets to be collected. This is essential for applications such as online recommendation systems, IoT monitoring, and social media analytics, where data must be processed in real time [17].

Apache Flink and **Apache Kafka** are popular frameworks for stream processing. These tools enable real-time analysis by processing data streams in a distributed fashion across multiple nodes. **Kafka** acts as a distributed messaging system, delivering real-time data streams to applications, while **Flink** processes these streams to generate real-time insights [18].

CEP extends stream processing by identifying patterns within the data streams. For example, a CEP system might detect a series of events that indicate potential fraud in a financial transaction system. By analyzing complex sequences of events, CEP systems can generate real-time alerts and automate decision-making processes.

4. Challenges and Opportunities in Advanced Computing for Real-Time Data Processing

While the techniques discussed provide significant advancements in real-time data processing and HPC, they also introduce new challenges. These include managing the scalability of systems, handling latency, ensuring fault tolerance, and optimizing energy consumption. Additionally, the integration of new technologies such as artificial intelligence (AI) and quantum computing opens up new opportunities for enhancing real-time processing capabilities [19].

4.1 Scalability and Data Management

One of the primary challenges in real-time data processing is the ability to scale systems to accommodate growing data volumes. As the number of connected devices increases, particularly in IoT environments, the volume of data generated in real time becomes overwhelming. Techniques like **data partitioning**, **sharding**, and **replication** help distribute this data across multiple nodes, but efficient management of these distributed datasets remains a critical challenge.

Table 3: Key Challenges and Mitigation Strategies in Real-Time Data Processing

Challenge	Description	Mitigation Strategy
Scalability	Increasing data volumes in IoT and big data	Data partitioning, sharding, replication
Latency	Delays caused by network communication	Edge computing, data caching, in-memory computing
Fault Tolerance	Ensuring system reliability during failures	Redundancy, data replication, task re-execution
Energy Efficiency	High power consumption in HPC environments	Energy-efficient algorithms, hardware optimization

4.2 Latency Reduction

Latency reduction is a critical consideration in real-time data processing systems, where even microsecond delays can significantly impact the performance of applications such as financial trading, autonomous vehicles, healthcare monitoring, and industrial control systems. Latency, in this context, refers to the delay between data input and the system's response. Minimizing this delay is essential to ensuring real-time systems function correctly and efficiently [20].

One of the most effective strategies for reducing latency is the use of **edge computing**. Edge computing moves data processing closer to the data source, reducing the time it takes for data to travel to centralized servers and back. For instance, in autonomous vehicles, edge computing allows sensor data to be processed on-board rather than relying on a distant data center. This local processing enables faster decision-making, which is critical for real-time systems where split-second decisions can be a matter of safety.

In-memory computing is another powerful technique for latency reduction. By storing data in RAM instead of on slower disk-based storage, in-memory computing allows for rapid data access and processing. Systems such as **Redis** and **Apache Ignite** are designed to enhance performance in real-time applications by keeping data in memory, eliminating the delays associated with disk I/O.

Additionally, the use of **high-speed, low-latency communication protocols** such as **Remote Direct Memory Access (RDMA)** can further reduce communication delays between nodes in a distributed system. RDMA enables data to be transferred directly between the memory of different machines without involving the CPU, significantly lowering the overhead of data movement across the network [21].

Efficient scheduling algorithms also play a key role in minimizing latency. Algorithms that prioritize real-time tasks and balance the load across processors or nodes in a distributed system can help ensure that time-sensitive operations are executed promptly. Techniques like **preemptive scheduling** allow high-priority tasks to interrupt lower-priority ones, reducing response times for critical operations [22].

In conclusion, reducing latency in real-time data processing systems requires a combination of architectural innovations, efficient communication protocols, and intelligent resource management strategies. As real-time applications continue to proliferate, particularly in fields where timing is critical, optimizing these latency-reduction techniques will be essential to maintaining system performance and reliability [23].

5. Conclusion

The demand for real-time data processing and high-performance computing (HPC) has surged with the rise of data-intensive applications in fields such as healthcare, finance, and autonomous systems. Advanced computing techniques, including parallel processing, distributed computing, in-memory computing, and edge computing, have become essential to address the challenges posed by these large-scale, real-time applications [24]. These technologies have transformed how data is processed, enabling faster computations, reduced latency, and enhanced scalability. However, they also introduce new complexities, particularly in terms of managing large data volumes, ensuring fault tolerance, and optimizing system performance.

This research has examined the evolution of computing architectures and explored how emerging technologies like artificial intelligence (AI) and quantum computing could further enhance real-time data processing and

HPC. The integration of AI can optimize decision-making processes in real time, while quantum computing offers the potential to solve computational problems currently beyond the reach of classical computers [25].

Despite these advancements, significant challenges remain, particularly in terms of latency reduction, energy efficiency, and the continued scalability of systems as data volumes grow. Addressing these challenges will be critical for the future of real-time processing in applications that require instantaneous responses, such as autonomous vehicles, IoT networks, and financial trading systems [26].

In conclusion, the future of real-time data processing and high-performance computing will rely on continued innovation in computing techniques, optimized architectures, and the strategic integration of emerging technologies to meet the growing demands of complex, data-driven applications.

References

- [1] M. Rathi, C. Gupta, R. Shukla, and R. Raubins, "Discernment of malaria-infected cells in the blood streak images using advanced learning techniques," in *Advanced Computational Techniques for Sustainable Computing*, Boca Raton: Chapman and Hall/CRC, 2022, pp. 139–152.
- [2] J. G. C. Ramírez, "Integrating AI and NISQ technologies for enhanced mobile network optimization," *QJETI*, vol. 5, no. 1, pp. 11–22, Jan. 2020.
- [3] T. Luo, L. Zuo, G. Jiang, W. Gao, H. Xu, and Q. Jiang, "Security of MVD-based 3D video in 3D-HEVC using data hiding and encryption," *J. Real Time Image Process.*, vol. 17, no. 4, pp. 773–785, Aug. 2020.
- [4] M. Atiquzzaman, J. Li, and W. Pedrycz, "Special issue on new advanced techniques in security of artificial intelligence," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 3, pp. 1255–1257, Mar. 2022.
- [5] J. G. C. Ramírez, "Quantum control and gate optimization in graphane-based quantum systems," *J. Appl. Math. Mech.*, vol. 4, no. 1, pp. 69–79, Oct. 2020.
- [6] B. Liu *et al.*, "Integrating advanced soft computing techniques with experimental studies for pore structure analysis of Qingshankou shale in Southern Songliao Basin, NE China," *Int. J. Coal Geol.*, vol. 257, no. 103998, p. 103998, May 2022.

- [7] J. G. C. Ramírez, “Vibration analysis with AI: Physics-informed neural network approach for vortex-induced vibration,” *Int. J. Radiat. Appl. Instrum. C Radiat. Phys. Chem.*, vol. 11, no. 3, Mar. 2021.
- [8] R. Idmayanti, D. Meidelfi, A. Erianda, F. Sukma, and R. Zazkia, “Boundary value analysis techniques for testing Geographic Information System applications for Public Facilities,” *ijasce*, vol. 4, no. 3, pp. 155–160, Dec. 2022.
- [9] N. Tantalaki, S. Souravlas, and M. Roumeliotis, “A review on big data real-time stream processing and its scheduling techniques,” *Int. J. Parallel Emergent Distrib. Syst.*, vol. 35, no. 5, pp. 571–601, Sep. 2020.
- [10] R. Gautam and M. Sharma, “Stress diagnosis among academic fraternity using bird-based soft computing techniques,” *Nternational J. Adv. Res. Comput. Commun. Eng.*, vol. 11, no. 4, Apr. 2022.
- [11] S. Misra, S. Weinberger, R. Orlando, and J. Sharp, “Development of laser-free, protein oxidation system, real-time radical dosimeter and data processing software for HOS of proteins using HRPf,” *J. Biomol. Tech.*, vol. 31, no. Suppl, p. S2, Aug. 2020.
- [12] J. G. C. Ramírez, “The role of graphene in advancing quantum computing technologies,” *Annu. Rep. - Aust. Inst. Criminol.*, vol. 4, no. 1, pp. 62–77, Feb. 2021.
- [13] K. L. Lakshmi, Bhargavi, and Jyothi, “A hybrid optimization approach for breast cancer gene sequence analysis using population based soft computing techniques,” *J. Adv. Res. Dyn. Control Syst.*, vol. 11, no. 11, pp. 132–141, Nov. 2019.
- [14] U. W. Humphries *et al.*, “Runoff estimation using advanced soft computing techniques: A case study of Mangla watershed Pakistan,” *Water (Basel)*, vol. 14, no. 20, p. 3286, Oct. 2022.
- [15] A. *et al.* and Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia, “Biometrics authentication techniques: A comparative study,” *Int. J. Adv. Appl. Sci.*, vol. 7, no. 9, pp. 97–103, Sep. 2020.
- [16] P. Sirivanth *et al.*, “A SVM based wine superiority estimation using advanced ML techniques,” in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2021.
- [17] J. G. C. Ramírez, “Enhancing temporal quantum coherence in graphene-based superconducting circuits,” *International Journal of Applied Machine Learning and Computational Intelligence*, vol. 11, no. 12, Dec. 2021.
- [18] R. Sreedhar and M. Ranjitha, “Fog computing – A survey on scheduling, optimization techniques and applications,” in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2021.
- [19] M. VenkataNaresh and I. Kullayamma, “An advanced Machine Learning Techniques on Agriculture fields for future level cultivations,” in *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, Bangalore, India, 2022.
- [20] M. Choraś *et al.*, “Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study,” *Appl. Soft Comput.*, vol. 101, no. 107050, p. 107050, Mar. 2021.
- [21] J. G. C. Ramírez, “Struggling Small Business in the US. The next challenge to economic recovery,” *IJBIBDA*, vol. 5, no. 1, pp. 81–91, Feb. 2022.
- [22] M. M. Murtala, School of IT & Computing, American University of Nigeria, Yola Academic City University College, Accra, Ghana, and O. B. Longe, “Data security in decentralized cloud systems: A case for the deployment of Blockchain technology,” in *Proceedings of the 27th iSTEAMS Multidisciplinary & Inter-tertiary Research Conference*, 2021.
- [23] J. G. C. Ramirez, “Comprehensive exploration of the CR model: A systemic approach to Strategic Planning,” *International Journal of Culture and Education*, vol. 1, no. 3, Aug. 2023.
- [24] J. G. C. Ramírez, “Incorporating Information Architecture (ia), Enterprise Engineering (ee) and Artificial Intelligence (ai) to Improve Business Plans for Small Businesses in the United States,” *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, vol. 2, no. 1, pp. 115–127, 2023.
- [25] Y. C. Bangar, A. Magotra, B. S. Malik, Z. S. Malik, and A. S. Yadav, “Evaluating advanced computing techniques for predicting breeding values in Harnali sheep,” *Trop. Anim. Health Prod.*, vol. 53, no. 2, p. 313, May 2021.
- [26] J. G. C. Ramírez, M. Hassan, and M. Kamal, “Applications of artificial intelligence models for computational flow dynamics and droplet microfluidics,” *JSTIP*, vol. 6, no. 12, Dec. 2022.