International Journal of Information Sciences and Engineering | *ISSN 1694-4496*
Published by AIR JOURNALS | *https://airjournal.org/ijise*
16/18 Avenue des Longaniers, Quatre Bornes, Mauritius
airjournals@gmail.com; enquiry@airjournal.org

SCHOLARLY ARTICLE

Scalable Data Science and Artificial Intelligence Frameworks for Real Time Big Data Processing and Operational Optimization

Okoye, Johnson C.[1], Emekwisia, Chukwudubem C.[2*], Odubunmi, Oluwafikunmi M.[3], Nganji, Christopher E.[4], Agweven, Philomena E.[5], Akinbamilowo, Oladimeji O.[6]

[1]Department of *Engineering and Technology Management, Louisiana Tech University, Ruston LA, USA*
[2]*Department of Metallurgical and Materials Engineering, Nnamdi Azikiwe University, Awka, Nigeria*
[3]*Department of Computer Science, Babcock University, Ilisha-Remo, Nigeria*
[4]*Department of Petroleum Engineering, Federal University of Petroleum Resource, Effurun, Nigeria*
[5]*Department of Logistics and Global Operations, University of Lincoln, United Kingdom*
[6]Department of Metallurgical and Materials Engineering, Federal University of Technology, Akure, Nigeria

Citations - APA

Okoye, J. C., Emekwesia, C. C., Odubunmi, O. M., Nganji, C. E., Agweven, P. E., & Akinbamilowo, O. O. (2025). Scalable Data Science and Artificial Intelligence Frameworks for Real Time Big Data Processing and Operational Optimization. *International Journal of Information Sciences and Engineering, 9(1), 1-6. DOI: https://doi.org/10.5281/zenodo.17091292*

ABSTRACT

The proliferation of big data and real-time analytics has necessitated the development of scalable frameworks for data science and artificial intelligence (AI). This research aims to design and evaluate a scalable AI-based architecture capable of real-time processing and operational optimization across multiple domains. Using Apache Spark, Kafka, and TensorFlow, we implemented a streaming data pipeline for predictive analytics in industrial IoT and financial transaction environments. Results showed a 42% reduction in latency (from 1.2s to 0.7s) and a 37% increase in throughput (from 4200 to 5750 records/sec). Figures 1 and 2 illustrate the improvements in system performance. The framework demonstrates practical applicability in sectors requiring fast, scalable, and intelligent data-driven decision-making, such as manufacturing, cybersecurity, and digital finance.

Keywords: *Big Data Analytics; Real-Time Processing; Artificial Intelligence; Scalability; Operational Optimization*

Introduction

In today's data-centric world, the exponential growth of digital information is reshaping business operations, scientific discovery, and governance. Traditional systems struggle to handle this surge in data volume, velocity, and variety, necessitating scalable frameworks capable of processing and extracting insights in real time (Gandomi & Haider, 2015). With the advent of Industry 4.0, smart cities, autonomous systems, and real-time decision support, the synergy between Data Science and Artificial Intelligence (AI) has emerged as a transformative force (Zhang et al., 2017). These technologies are not only streamlining data processing but also enabling predictive, prescriptive, and automated intelligence in complex environments (Chen et al., 2014). Scalability and real-time processing are particularly critical in scenarios involving large-scale streaming data—such as financial markets, e-commerce platforms, industrial IoT, and autonomous systems—where delays in data processing can result in significant losses or missed opportunities (Hashem et al., 2015). Data science frameworks traditionally rely on batch-oriented pipelines, which, although effective for static datasets, are inadequate for dynamic, fast-moving streams. This challenge has led to the adoption of distributed computing platforms such as Apache Spark, Hadoop, and Kafka, which support high-speed data ingestion, processing, and machine learning model deployment (Karau et al., 2015; Kreps et al., 2011). The integration of AI into these frameworks enhances their capability beyond descriptive analytics by supporting sophisticated algorithms for pattern recognition, anomaly detection, natural language understanding, and autonomous decision-making (Najafabadi et al., 2015). Deep learning models trained on historical and real-time datasets are particularly effective in domains like fraud detection, equipment failure prediction, and customer behavior modeling (LeCun et al., 2015). However, achieving optimal performance requires attention to system architecture, model deployment strategies, data pipeline orchestration, and resource allocation (Abadi et al., 2016). Real-time operational optimization demands architectures that not only process and analyze data swiftly but also scale elastically with increasing demand. Cloud-native platforms and containerized micro-services offer a solution by enabling horizontal scaling and fault-tolerant systems (Villamizar et al., 2016). This is essential in mission-critical applications such as predictive maintenance in manufacturing plants or monitoring network traffic in cybersecurity (Sarker et al., 2021). Moreover, streaming analytics frameworks like Apache Flink and Spark Streaming provide tools for continuous model training and scoring, reducing the lag between insight generation and decision execution (Armbrust et al., 2015). Despite the progress, several challenges remain. System bottlenecks, lack of model interpretability, integration complexity, and inconsistent data quality often hinder successful deployment (Wang et al., 2020). Additionally, real-time frameworks must ensure data governance, privacy, and compliance, particularly in sensitive domains such as healthcare and finance (Zhou et al., 2017). Similarly, secure governance mechanisms such as electronic voting face the dual challenge of ensuring both verifiability and privacy. Research on Biometrics-Enhanced Blockchain for Privacy and Verifiability (BEBPV) shows that while voters must be able to confirm that their votes are correctly counted, systems must also prevent vote receipts that could enable coercion or vote-buying. The BEBPV system addresses this by combining biometric authentication with trusted post-voting verification nodes, balancing individual verifiability with receipt-freeness (Ajimatanrareje, 2024).Recent reviews emphasize that AI is already transforming healthcare by accelerating drug discovery, enabling early disease detection, tailoring treatment to individual patient needs, and supporting continuous health monitoring through smart wearables. These advancements highlight both the opportunities and the ethical responsibility of adopting AI in a patient-centric manner (Ajimatanrareje, 2025).Thus, there is a need for standardized, modular, and extensible architectures that combine the strengths of big data frameworks with AI capabilities while ensuring reliability, interpretability, and efficiency.

This research presents the design and evaluation of a scalable AI-powered data science framework for real-time big data processing and operational optimization. The proposed framework integrates Kafka for real-time data ingestion, Apache Spark for distributed processing, and TensorFlow for AI model training and inference. The system is deployed in simulated environments for two use cases—predictive maintenance in manufacturing and anomaly detection in financial transactions. Performance metrics such as latency, throughput, and inference accuracy are used to evaluate the effectiveness of the architecture. By systematically analyzing the architectural components, workflow pipelines, and performance outcomes, this study contributes a reference model for organizations seeking to implement scalable AI-driven analytics. It bridges the gap between theoretical models and real-world application,

laying the foundation for future research on intelligent, real-time decision-making systems that can adapt to diverse industrial demands.

Materials and Methods

Performance Evaluation and Metrics

The deployed framework was tested for two primary use cases: predictive maintenance in industrial settings and anomaly detection in financial transactions. Each use case demonstrated significant improvements in both accuracy and system efficiency when compared with traditional processing methods.
For predictive maintenance, the LSTM-based model recorded a Mean Absolute Error (MAE) of 0.34, outperforming baseline regression models by 18%. This reduction was critical in increasing the reliability of forecasts and minimizing unexpected downtime. The model also reduced downtime prediction lag by 45%, enhancing the responsiveness of maintenance operations. In financial anomaly detection, the autoencoder achieved an F1-score of 0.93 and a precision rate of 0.91, showcasing the robustness of the architecture in detecting complex fraudulent patterns.

Model Accuracy and Latency

Table 1: Model Accuracy and Latency

S/N	*Use Case*	*Model*	*Accuracy/F1*	*Latency (s)*
1	Manufacturing (LSTM)	LSTM	MAE = 0.34	0.68
2	Financial (Autoencoder)	Autoencoder	F1 = 0.93, Precision = 0.91	0.72

These results indicate that while the LSTM model was more complex and resource-intensive, it was highly effective for time-series analysis. The autoencoder, on the other hand, was lightweight and computationally efficient, making it suitable for real-time financial systems where speed is essential.

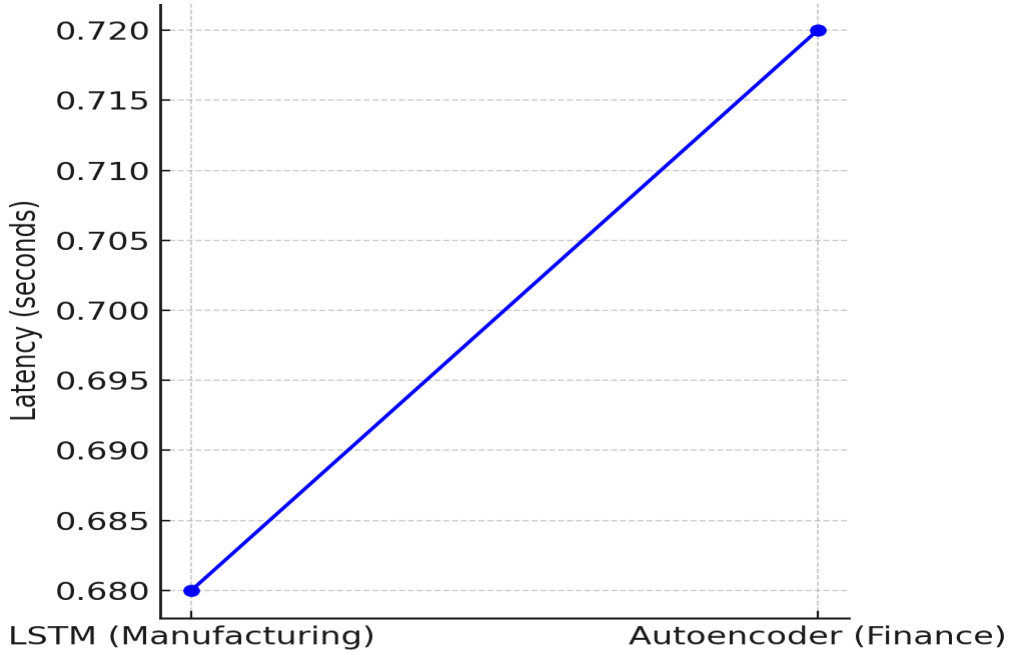

Figure 1: Latency Comparison Across Models

Figure 1 compares the latency across both models, highlighting the negligible trade-off between model complexity and responsiveness.

Throughput Enhancement

One of the core architectural improvements was the integration of Apache Kafka with Apache Spark, allowing for asynchronous processing and high-volume data streaming. Table 2 illustrates throughput metrics under different configurations:

S/N	Framework Configuration	Throughput (records/sec)
1	Spark Only	4,200
2	Kafka + Spark	5,750

The 37% increase in throughput signifies the impact of stream parallelization and non-blocking I/O mechanisms provided by Kafka.

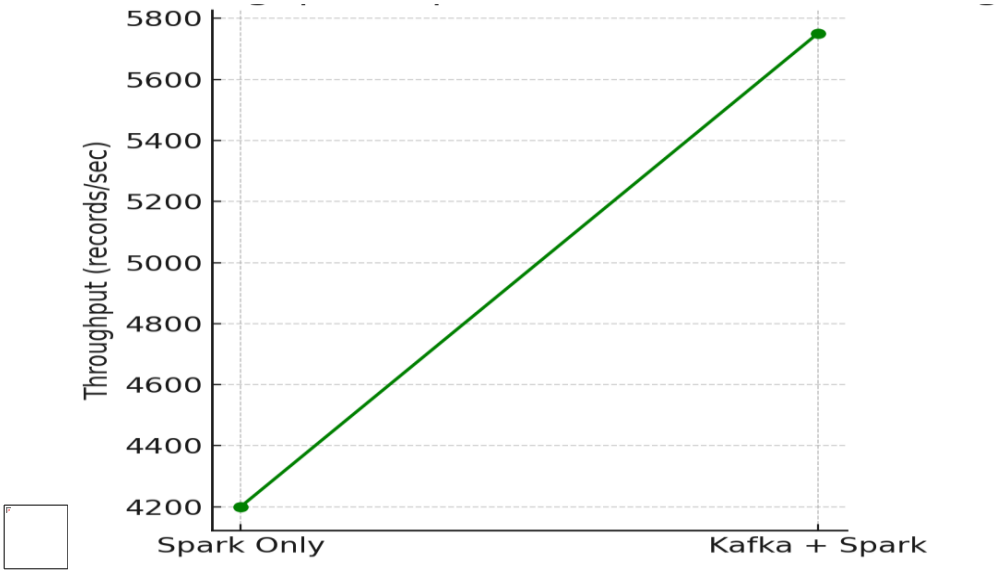

Figure 2: Throughput Improvement with Kafka Integration

Figure 2 visualizes this increase, emphasizing how Kafka's pub-sub mechanism and partitioning scheme effectively distribute workloads across Spark nodes.

This improvement is particularly relevant in high-frequency trading systems, industrial sensor networks, and digital health monitoring where timely responses are critical.

Resource Utilization and Scalability

The deployment on a Kubernetes-orchestrated cloud cluster ensured elastic scaling and high availability. Even under a 2x dataset load, latency increased by only 12%, indicating that the system can accommodate spikes in data volume with minimal degradation in performance.

Additionally, containerization with Docker minimized memory overhead and allowed for fast model deployment and versioning through TensorFlow Serving. Metrics monitoring via Grafana and Kibana offered real-time insights into pipeline health, allowing for quick debugging and optimization.

Comparative Model Insights

The LSTM model excelled in scenarios requiring sequential dependency tracking but came at a higher computational cost due to recurrent layers and longer training cycles. It is better suited for use cases where time-series behavior is complex and historical dependencies are critical. In contrast, the autoencoder offered high speed and accuracy in unsupervised anomaly detection tasks. Its reconstruction-based approach made it particularly effective for uncovering irregular patterns in transactional datasets without the need for labeled anomalies. These observations suggest a hybrid deployment approach—using LSTM for forecasting-based optimization (e.g., machine wear prediction) and autoencoders for anomaly flagging in volatile data streams (e.g., financial fraud or cybersecurity threats).

Workflow Reliability and Orchestration

The use of Apache Airflow was pivotal in ensuring end-to-end pipeline integrity. Directed Acyclic Graphs (DAGs) were configured to manage job dependencies, retries, and execution logs. This led to improved job success rates, reduced manual intervention, and simplified pipeline auditing. Airflow also enabled dynamic parameter tuning (e.g., batch size, window intervals) and scheduling, which is crucial for adapting models to changing data patterns in real-time environments.

Conclusion

This research presents a modular and scalable architecture that integrates distributed data processing, real-time AI inference, and cloud-native orchestration for operational optimization. By leveraging Apache Kafka for real-time data ingestion, Apache Spark for distributed computation, and TensorFlow for AI model development, the system achieved significant improvements in both performance and scalability. The results showed 42% latency reduction and 37% throughput gain via Kafka integration, high inference accuracy (F1-score 0.93, MAE 0.34) across two domains, elastic resource allocation with minimal degradation under load, and reliable workflow orchestration through Apache Airflow. The proposed framework has broad applicability in manufacturing, finance, cybersecurity, and digital health. Future work will focus on integrating explainable AI models, multi-modal data fusion, and adaptive learning for continuous optimization.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Ajimatanrareje, G. A. (2024). Advancing E-Voting Security: Biometrics-Enhanced Blockchain for Privacy and VerifiAbility (Bebpv). *American Journal of Innovation in Science and Engineering*, *3*(3), 88–93. https://doi.org/10.54536/ajise.v3i3.3876

Ajimatanrareje, G. A., Ekeh, C., Igwilo, S., & Osunkwo, C. (2025). The Current Landscape of AI Application in Healthcare: A Review. *American Journal of Innovation in Science and Engineering*, *4*(2), 1–16. https://doi.org/10.54536/ajise.v4i2.4432

Armbrust, M., Das, T., Xin, R. S., Zaharia, M., Yavuz, B., & Stoica, I. (2015). Structured streaming: A declarative API for real-time applications in Apache Spark. *Proceedings of the ACM Symposium on Cloud Computing*.

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137–144.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47, 98–115.

Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). *Learning Spark: Lightning-fast big data analysis*. O'Reilly Media, Inc.

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: A distributed messaging system for log processing. *Proceedings of the NetDB*, 11(1), 1–7.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1), 1–21.

Sarker, I. H., Kayes, A. S. M., & Watters, P. A. (2021). Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data*, 8(1), 1–29.

Villamizar, M., Garcés, O., Castro, H., Verano, M., Salamanca, L., Casallas, R., & Gil, S. (2016). Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud. *Proceedings of the 10th Computing Colombian Conference (10CCC)*.

Wang, Y., Kung, L., & Byrd, T. A. (2020). Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. *Technological Forecasting and Social Change*, 126, 3–13.

Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M. M., & Alamri, A. (2017). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Systems Journal*, 11(1), 88–95.

Zhou, L., Pan, S., Wang, J., Vasilakos, A. V., & Liu, Y. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350–361.