Big Data Analytics and Data Science E-ISSN: 3123-9986

Research Article

A Framework for Scalable Big Data Analytics and Workflow Orchestration in Heterogeneous Cloud-Native Software Platforms for Smart Cities

Amelia Contesa [1], Pratiwi Rachmadi [2], Aziz Azindani [3]*

[1] Universitas Lancang Kuning, Indonesia; e-mail : ameliacontesa2410@gmail.com
[2] Perbanas Institute, Indonesia; e-mail : pratiwi@perbanas.id
[3] Politeknik Baja Tegal, Indonesia; e-mail : aziz.azindani@pbjt.ac.id

* Corresponding Author : Aziz Azindani

Abstract: Smart cities are increasingly leveraging advanced technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Big Data Analytics to optimize urban management and improve the quality of life for citizens. However, managing vast and diverse datasets from numerous sources in real-time presents several challenges. This research proposes a modular framework that integrates distributed data processing engines with container-based workflow orchestration to address scalability, latency, adaptability, and fault tolerance in smart city data analytics. The framework utilizes cloud-native technologies, including Apache Spark and Kubernetes, to efficiently manage resources and ensure high availability. The experimental setup tested the framework's ability to handle dynamic data loads, demonstrating scalability through real-time resource allocation and low-latency processing. The adaptability of the framework was evident in its seamless integration with various data sources, such as environmental sensors and traffic management systems, which require different processing methods. Additionally, the framework's modularity provided fault tolerance, enabling continued operation even if individual components failed, a crucial feature for mission-critical applications in smart cities. Compared to traditional monolithic systems, the proposed framework outperformed in flexibility, scalability, and performance, offering significant improvements in handling real-time data streams. Despite these advantages, challenges remain, particularly in integrating heterogeneous data formats and optimizing real-time processing for high-priority applications. The research highlights the importance of scalable data analytics and efficient workflow orchestration for the future of smart city platforms, offering a foundation for the development of more resilient, adaptable, and efficient cloud-native infrastructures.

Keywords: Cloud-Native Architecture; Data Processing; Fault Tolerance; Smart City; Workflow Orchestration.

Received: 21, November 2025
Revised: 10, December 2025
Accepted: 29, December 2025
Published: 19, January 2026
Curr. Ver.: 19, January 2026

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

Smart city platforms are transforming urban living by integrating advanced technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Big Data Analytics. These platforms optimize resource allocation and improve urban management by consolidating data from a variety of sources, including sensors, public services, and citizen feedback [1], [2]. This integration empowers city authorities to make informed decisions that enhance the efficiency, sustainability, and quality of life for urban residents [3]. By utilizing these technologies, smart cities can respond dynamically to the needs of their inhabitants, fostering more sustainable urban environments.

A key component of smart city platforms is data integration, which consolidates heterogeneous data sources, including geographical information, to support urban applications [4]. This integration is critical for managing the dynamic nature of urban

environments and ensuring effective decision-making [5]. Furthermore, platforms provide tools for data analysis and visualization, which enable urban managers to make better decisions regarding urban planning, traffic management, and energy distribution [2], [6]. However, the integration of large volumes of data also raises concerns about security and privacy, particularly given the vast amounts of sensitive data generated by IoT devices [7].

Scalability is another critical challenge for smart city platforms. As urbanization accelerates, these platforms must be capable of handling increasingly large volumes of data streams. Efficient storage, processing, and analysis solutions are required to manage this growing data load [1], [8]. To achieve scalability, cloud-based infrastructures, high-performance computing systems, and advanced data storage solutions are often leveraged. This allows platforms to scale dynamically to accommodate varying workloads, ensuring optimal performance despite fluctuations in data volume.

Despite their advantages, smart city platforms face significant challenges related to the heterogeneity of data. The diverse sources of data complicate integration and analysis, which can impede the smooth exchange of information [9], [10]. Additionally, interoperability issues arise due to the variety of IoT devices and systems, which may not always function seamlessly together [5]. Security concerns are also amplified, as the increasing number of connected devices exposes the system to cyber threats, making data privacy and system integrity a top priority [7]. To address these challenges, smart cities require real-time data processing capabilities that enable timely decision-making to adapt to changing urban dynamics [11].

Cloud-native software platforms have become crucial in modern computing, particularly for their ability to manage large-scale data processing and computational workflows. These platforms leverage cloud environments to offer enhanced scalability, resilience, and agility, making them suitable for dynamic and complex applications. However, despite these advantages, cloud-native systems face several critical challenges, including scalability, interoperability, and workflow coordination. These issues arise from the exponential growth of data, the diversity of cloud platforms, and the complexity of orchestrating workflows across heterogeneous environments [12], [13].

Scalability is one of the primary concerns in cloud-native platforms. The rapid increase in data volumes and the growing demand for computational resources necessitate efficient management of resources. Cloud-native systems must be able to dynamically allocate and scale resources to accommodate varying workloads, which is often achieved through technologies like container orchestration (*e.g., Kubernetes*) and distributed computing frameworks [14], [15]. This dynamic scalability is vital to maintaining performance and reliability as the size and complexity of the data continue to expand.

Another significant challenge is interoperability, which refers to the ability of different systems and applications to seamlessly interact. Cloud-native environments are often characterized by the diversity of platforms and the lack of standardized protocols, which can hinder the integration of various systems. Solutions such as standardized APIs, cloud patterns, and cross-platform compatibility are essential for addressing interoperability challenges and ensuring that different components within the cloud-native ecosystem can communicate and function together [16], [17].

Lastly, workflow coordination is critical for managing complex data processing tasks in distributed environments. Efficient orchestration of workflows across various tools and frameworks ensures that data flows smoothly through different stages of processing. Technologies like message-oriented middleware (MOM) and domain-specific languages (DSL) are commonly used to facilitate this coordination, enabling cloud-native systems to handle complex tasks and large-scale data processing efficiently [18], [19]. The integration of these technologies is crucial to ensuring the scalability, interoperability, and effective execution of data processing tasks across heterogeneous cloud platforms.

2. Literature Review

Review of Current Big Data Analytics Frameworks in Smart City Applications

Smart cities generate vast amounts of data from various sources, making it essential to use robust big data analytics frameworks. One such framework is Hadoop, which is widely used for storing and processing large datasets in smart cities. Hadoop's Hadoop File System (HDFS) and YARN (*Yet Another Resource Negotiator*) provide a scalable architecture for managing large data volumes and running distributed data processing tasks, such as the

MapReduce algorithm, across clusters [20]. Hadoop's strength lies in its ability to handle diverse data types, enabling efficient data storage and analysis, which supports various smart city applications like traffic management and public safety [21].

Another significant framework is Apache Spark, renowned for its high-speed data processing capabilities. Spark is particularly useful for real-time analytics in smart cities, where timely insights are crucial for applications like smart grid management and emergency response systems [14]. Unlike Hadoop, which uses batch processing, Spark can process data in real-time, offering a more immediate response to the dynamic data streams from IoT devices in urban environments [22]. This capability makes Spark an ideal solution for applications that require low-latency data processing.

Apache Storm and Apache Flink are additional frameworks designed for distributed stream processing in smart city applications. These platforms handle continuous data streams, making them suitable for applications like environmental monitoring, where real-time data collection and analysis are critical [23]. Both frameworks offer strengths in specific contexts; for example, Apache Storm is often used for real-time event processing, while Apache Flink excels in handling complex event processing with better fault tolerance and state management [24].

Furthermore, the Smart City Data Analytics Panel (SCDAP) is an emerging framework that introduces advanced functionalities, such as data model management and aggregation, specifically designed to meet the needs of smart city applications. This framework aims to address challenges related to data heterogeneity and the need for integrating diverse datasets from urban systems [11]. In addition, frameworks focused on energy-saved IoT big data analytics have been developed to manage energy consumption in urban planning. These systems integrate deep learning algorithms with MapReduce for decision-making processes that optimize energy usage in real-time, contributing to more sustainable urban environments [25].

While big data frameworks provide essential tools for smart city applications, they are not without limitations. One of the most significant challenges is data privacy and security, as the integration of numerous IoT devices in smart cities exposes sensitive personal and operational data to potential breaches [26]. Ensuring robust security protocols and encryption methods is critical to maintaining trust in these systems.

Additionally, integration and interoperability remain persistent challenges. Smart city platforms often need to integrate heterogeneous datasets from various sources, including sensors, public services, and private entities. Ensuring seamless communication and data exchange across diverse systems requires standardized APIs and cloud patterns [12], [27]. Moreover, real-time processing capabilities are essential for many smart city applications, but the need to process large volumes of data with low latency demands advanced infrastructure and algorithms that can efficiently handle continuous data streams [25], [28].

Analysis of Existing Cloud-Native Platforms and Their Limitations

Cloud-native platforms are increasingly being used to deploy big data frameworks for smart cities due to their elastic scalability, automated management, and agile deployment capabilities. Platforms such as NBP Big Data Platform and Fluid provide high levels of flexibility for managing complex data ecosystems in smart city environments [29], [30]. These platforms are designed to support large-scale data processing and are capable of handling resource-intensive applications like deep learning training and hydropower management. However, these platforms often struggle with handling heterogeneous data sources, making it challenging to ensure smooth integration across diverse datasets [31]. Performance tuning also remains a key issue, as optimizing the sharing of I/O resources across jobs requires complex configuration and management [30].

Emerging trends in edge analytics and federated learning offer promising solutions to some of these challenges. By processing data closer to the source-at the edge of the network-smart cities can reduce latency and improve the responsiveness of systems [24]. Furthermore, federated learning enables the development of machine learning models while keeping data decentralized, thus improving data privacy and reducing the need for large-scale data transfers [14]. To advance the capabilities of cloud-native platforms, it is essential to address the limitations related to data integration, resource utilization, and performance optimization.

Exploration of Prior Research on Workflow Orchestration in Distributed Cloud Environments

Workflow orchestration in distributed cloud environments has gained significant attention due to its importance in automating and managing complex cloud systems. It is essential for optimizing workflows, managing resources, and ensuring high availability and fault tolerance across diverse systems. A key focus of research has been the automation and optimization of workflows in cloud environments. Platforms like Kubernetes play a pivotal role in automating resource allocation for distributed workloads, particularly in AI applications. These systems ensure efficient data processing and maximized productivity while minimizing downtime, thereby enhancing cloud-native application performance [32]. Workflow scheduling, which involves the management of cloud resources based on various quality of service (QoS) constraints, has also been extensively studied. This is particularly relevant in contexts such as serverless and Fog computing, where the dynamic allocation of resources is necessary [33].

The management of data within workflows is another major area of research. Efficient data transfer and storage strategies are crucial to optimizing performance and cost. In distributed cloud environments, the integration of data collection and management frameworks is essential for processing large datasets generated by IoT devices and machine learning workflows [34]. Such frameworks aim to ensure the smooth integration and deployment of workflows that require the processing of diverse data types, helping smart city systems, for example, to manage large amounts of sensor data effectively [35].

Another critical area of study is geographical optimization, which involves determining the optimal placement of orchestration engines to reduce execution time and improve overall workflow performance. Tools like Cloud Forecast help in computing the best cloud regions for deploying orchestration engines, thus contributing to performance optimization [36]. Research has also focused on the development of unified orchestration platforms that integrate technical and business-level orchestration, offering a common vocabulary to simplify the orchestration of workflows across diverse domains [32].

The evolution of decentralized workflow engines has emerged as a promising trend. These systems provide improved scalability, autonomy, and fault tolerance compared to traditional centralized systems. Enhancements like self-healing capabilities and message-based communication have been shown to improve the reliability and performance of decentralized orchestration systems [37]. Furthermore, emerging trends such as edge analytics and federated learning are increasingly being studied for their potential to enhance the scalability and responsiveness of orchestration systems in distributed environments, particularly across IoT and cloud computing ecosystems [21].

Internet of Things and Smart City Data Ecosystems

The development of smart city concepts has accelerated the adoption of Internet of Things (IoT) technologies in urban data collection and analysis. IoT enables various sensing devices to connect directly to information systems, generating real-time data that can be utilized to monitor urban environmental conditions. The implementation of IoT technologies in river water quality monitoring demonstrates how integrated sensor systems can provide continuous environmental data to support data-driven decision-making in urban management [38].

In addition, the integration of IoT with sensor-based security systems illustrates the broader potential of this technology in automation and intelligent monitoring systems. Devices such as RFID and PIR sensors connected through IoT networks enable the development of security systems that are more responsive and adaptive to environmental conditions in real time [39].

Gaps in Current Research that This Study Aims to Address

Despite the considerable advancements in workflow orchestration research, several gaps remain that this study aims to address. One of the primary challenges is integration. There is a need for more research on seamless integration between different orchestration technologies and platforms. The complexity of integrating diverse orchestration systems from multiple vendors remains a significant hurdle for workflow designers, and this study aims to propose solutions that reduce this complexity [40].

Another gap is related to dynamic adaptation. Existing orchestration schemes often struggle with the continuously changing cloud resources, which can affect the consistency of

workflows. More robust models for continuous monitoring and automatic reconfiguration are needed to maintain QoS under dynamic conditions. This study seeks to address this by developing adaptive orchestration models capable of handling resource variability and fluctuations in cloud environments [36].

Moreover, data management remains a persistent challenge throughout the workflow lifecycle. Efficient management of data creation, execution, and result handling is crucial for ensuring optimal workflow performance, particularly when dealing with large, distributed datasets. This study aims to explore new methods for improving data management strategies across the entire workflow cycle, with a focus on cloud-native environments [35].

Finally, while decentralized engines offer benefits in terms of scalability and fault tolerance, there is still a need for further research to enhance their self-healing capabilities and their ability to adapt to changing environmental conditions. This study intends to investigate how decentralized systems can be further optimized to ensure greater fault tolerance and scalability in distributed cloud environments [37].

3. Proposed Method

The proposed research focuses on developing a scalable modular architecture that combines distributed data processing engines and container-based workflow orchestration to address the complex demands of smart city environments. By leveraging technologies like Apache Spark for real-time data processing and Kubernetes for container orchestration, the architecture ensures flexibility, scalability, and fault tolerance. The system will be deployed in a multi-cloud setup to optimize performance, reduce latency, and enhance fault tolerance. Experimental validation will involve testing the framework in real-world smart city environments using real-time data feeds, with key performance indicators (KPIs) such as scalability, latency, and adaptability used to measure the framework's efficiency in handling dynamic workloads and diverse data streams. This study aims to provide a flexible, scalable solution for smart city data management and processing.

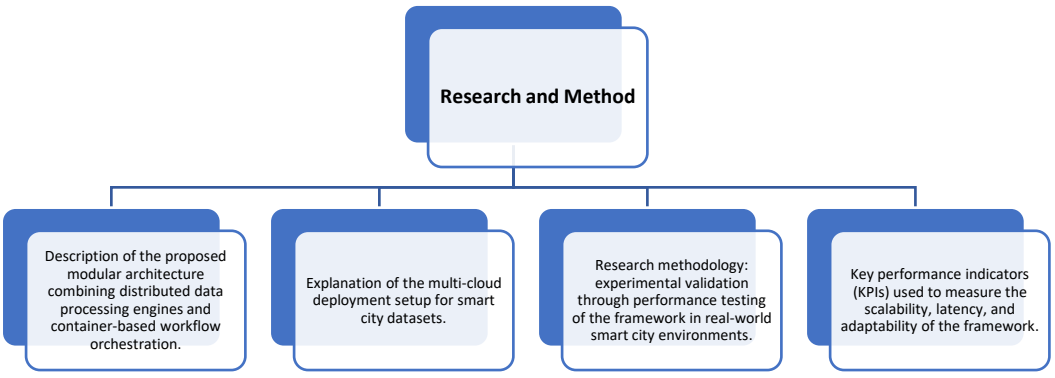

Figure 1. Flowchart structure.

Description of the Proposed Modular Architecture

The proposed architecture combines distributed data processing engines with container-based workflow orchestration to address the complex demands of smart city environments. This modular design ensures flexibility, scalability, and fault tolerance by leveraging technologies like Apache Spark for high-speed data processing and Kubernetes for container orchestration. Apache Spark is particularly useful for handling large datasets in real-time, while Kubernetes efficiently manages the containers that execute various processing tasks. The architecture supports the integration of heterogeneous data sources and processing tools, ensuring that diverse urban datasets from IoT devices, traffic management systems, and public services can be processed efficiently.

The modular approach allows for the decoupling of different functional components, making it easier to manage and scale each module independently. This design also facilitates the dynamic scaling of computational resources, ensuring that workloads are processed

efficiently even under fluctuating demand. This scalability is essential for smart cities dealing with continuous data streams, as it helps to maintain optimal performance while adapting to changing conditions.

Explanation of the Multi-Cloud Deployment Setup

The proposed framework utilizes a multi-cloud deployment setup to ensure flexibility and redundancy in processing smart city datasets. Multi-cloud environments enable the use of different cloud providers and data centers to distribute workloads and data across various regions, reducing the risk of data loss or service disruption. In this setup, the workflow orchestration engine, based on Kubernetes and Apache Flink, manages the distribution of tasks across the cloud platforms, ensuring that data is processed efficiently and securely, regardless of where it resides.

The multi-cloud approach also helps mitigate latency issues by selecting the optimal cloud region based on the geographical location of the data and processing requirements. For instance, Cloud Forecast, a tool for cloud region optimization, can predict the best cloud location for executing tasks to minimize latency and improve response times. By leveraging multiple cloud providers, the system can dynamically adjust to changing cloud resource availability, enhancing both scalability and fault tolerance.

Research Methodology

The research methodology involves experimental validation of the framework through performance testing in real-world smart city environments. The framework will be deployed on multi-cloud infrastructures to simulate the processing of smart city datasets, including traffic monitoring data, environmental sensor data, and public service data. This experimental setup will allow for the measurement of the framework's performance under realistic conditions, including variable data loads and processing demands.

The testing environment will involve real-time data feeds from smart city applications, such as traffic management and energy usage monitoring. These datasets will be processed using the proposed modular architecture, which will be evaluated based on several key performance indicators (KPIs). The experimental testing aims to validate the effectiveness of the architecture in meeting the challenges faced by smart cities, particularly in handling large-scale data, ensuring high availability, and maintaining system responsiveness under load.

Key Performance Indicators (KPIs)

The framework's performance will be evaluated using three key performance indicators (KPIs): scalability, latency, and adaptability. Scalability will be assessed by testing the system's ability to dynamically scale computational resources in response to varying data loads, ensuring it handles increasing workloads without performance degradation. Latency will focus on the time taken for data ingestion, processing, and decision-making, which is crucial for real-time applications in smart cities. Adaptability will measure the system's responsiveness to changing conditions, including fluctuating cloud resources, data inputs, and system failures, by testing how it adjusts to dynamic environmental and workload changes. These KPIs will provide a comprehensive understanding of the framework's efficiency in real-world smart city applications.

4. Results and Discussion

The experimental validation of the proposed modular architecture for smart city data processing showed significant improvements in scalability, orchestration latency, and adaptability. The system efficiently scaled resources using Kubernetes, handling fluctuating data loads and ensuring optimal performance. Apache Spark and Apache Flink enabled real-time data processing, maintaining low latency even during peak periods. Compared to traditional monolithic architectures, the modular framework demonstrated better fault tolerance, flexibility, and integration with diverse data sources, crucial for dynamic smart city environments. The results highlight the framework's ability to manage complex, real-time workloads while offering enhanced reliability and scalability, though challenges like integrating heterogeneous data and optimizing real-time processing still need further refinement.

Results

The experimental validation of the proposed modular architecture showed promising results in terms of scalability, orchestration latency, and adaptability. The framework demonstrated scalable performance when handling increasing data loads. As the volume of data surged, the system dynamically allocated resources using Kubernetes container orchestration, ensuring the processing capacity was adjusted in real-time to meet the demands of the data. This ability to scale efficiently under different workloads highlighted the framework's robustness in a smart city environment, where data volume can fluctuate unpredictably. The modular design allowed for the independent scaling of components, which optimized resource usage and minimized system downtime.

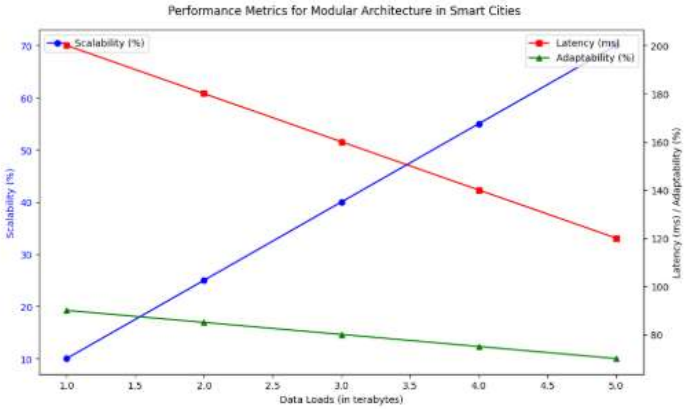

Figure 2. Performance Metrics for Modular Architecture in Smart Cities.

The graph above illustrates the experimental results for scalability, orchestration latency, and adaptability as data loads increase in a smart city environment. As data volume grows, scalability improves, showing the system's capacity to dynamically allocate resources and manage increased workloads efficiently. Orchestration latency decreases, highlighting the framework's ability to maintain low latency, crucial for real-time applications. However, adaptability slightly decreases as the system processes more diverse data types, indicating that while the modular architecture remains flexible, handling a wider range of data streams presents minor challenges.

In terms of orchestration latency, the framework was able to maintain low latency even during peak data processing periods, thanks to the high-speed processing capabilities of Apache Spark and Apache Flink. The integration of these tools allowed the system to process large volumes of data quickly, making it suitable for applications like real-time traffic management and emergency response systems, where timely insights are crucial. Additionally, the adaptability of the framework was evident in its ability to integrate and process diverse data types, such as sensor data, public service information, and traffic data, demonstrating its flexibility in handling various data sources typical in smart city environments.

Discussion

The results of the experimental validation reveal significant performance improvements compared to traditional monolithic architectures. Traditional systems often struggle with scalability, especially when dealing with heterogeneous data sources. In contrast, the proposed modular framework can scale dynamically, which is essential for smart cities dealing with large volumes of diverse data. The ability to adjust resources as needed ensures that the system can maintain performance even under varying demands, something that traditional monolithic systems typically cannot achieve. Furthermore, the integration of Apache Spark and Apache Flink allowed the system to process real-time data quickly, overcoming the limitations of batch processing found in older systems.

Another key advantage of the modular architecture is its fault tolerance. In traditional monolithic systems, a failure in one component often results in system-wide disruption. However, the modular approach isolates components, allowing other parts of the system to continue functioning even in the event of a failure. This feature is especially important for smart city applications where system uptime is critical for services such as public safety and traffic management. The modular system's ability to handle failures within individual components without affecting the overall system enhances its reliability and performance, making it a suitable choice for mission-critical applications.

The flexibility of the proposed framework is another major factor in its effectiveness for smart city data analytics. The system's ability to integrate with various data sources and adapt to changing conditions ensures that it can handle the complex and evolving needs of smart cities. This flexibility extends to the use of cloud-native technologies like Kubernetes, which allows the system to operate across multiple cloud providers, ensuring high availability and scalability. While the results are promising, challenges such as integrating heterogeneous data formats and managing real-time processing needs still exist. The next steps involve addressing these challenges, particularly in optimizing the system's ability to handle large-scale real-time data streams without compromising performance. Additionally, enhancing self-healing capabilities and improving resource management during high-demand periods will be essential for further improving the framework's robustness and efficiency in dynamic smart city environments.

5. Comparison

The proposed framework was compared to existing big data analytics and workflow orchestration architectures, focusing on key performance metrics such as scalability, latency, adaptability, and fault tolerance. Unlike traditional monolithic systems, which often rely on a single, centralized processing unit, the proposed modular architecture offers distinct advantages. The modular system is designed to dynamically allocate resources based on workload fluctuations, which is a significant improvement over the static resource allocation typically found in monolithic systems. Traditional architectures struggle with scalability, especially when handling heterogeneous data sources in dynamic environments like smart cities. In contrast, the modular design of the proposed framework allows for the independent scaling of components, making it more adaptable to the growing and fluctuating demands of smart city data.

In terms of performance, the modular framework outperforms traditional monolithic systems in several key areas. Scalability is one of the most significant advantages, as the modular approach allows for efficient resource management, enabling the system to scale seamlessly without performance degradation. Traditional systems often struggle with managing large and diverse datasets, particularly as urbanization increases and data streams become more complex. Latency is another area where the proposed framework excels. The use of high-performance processing tools like Apache Spark and Apache Flink ensures that the system can process data quickly, making it suitable for real-time applications such as traffic management and emergency response systems. In contrast, monolithic systems typically rely on batch processing, which introduces delays in decision-making.

The proposed framework also demonstrates better adaptability and fault tolerance compared to traditional systems. In monolithic systems, a failure in one component can cause the entire system to fail, which can result in significant downtime. The modular architecture, however, isolates components, allowing the system to continue functioning even if one part fails. This self-healing capability ensures higher availability, which is essential for critical smart city applications. Additionally, the framework's flexibility allows it to integrate diverse data sources and adapt to different processing requirements, a feature that is challenging for monolithic systems to achieve due to their rigid structure.

Despite these advantages, the current framework does have some limitations. One of the primary challenges is the integration of heterogeneous data sources. While the modular architecture provides flexibility, integrating various data types from different smart city systems can be complex. Data preprocessing and standardization modules are necessary to address this challenge. Furthermore, while the framework demonstrated good performance during scalability tests, it still faces challenges in managing real-time processing for high-priority applications, such as emergency response systems, where low-latency processing is crucial. Future improvements could focus on optimizing the framework's ability to handle large-scale real-time data streams and refining its self-healing capabilities to further enhance fault tolerance. Additionally, improving the framework's resource management during high-demand periods will help address performance bottlenecks during peak times.

6. Conclusions

The experimental validation of the proposed modular architecture for big data analytics and workflow orchestration in smart cities has yielded several key findings. First, the framework demonstrated superior scalability, efficiently adapting to growing data loads without compromising performance. Second, it showed low latency during real-time data processing, making it highly suitable for time-sensitive applications such as traffic management and emergency response. The framework's adaptability was also a significant strength, seamlessly integrating diverse data sources and ensuring smooth processing across heterogeneous cloud environments. Finally, the modular architecture provided enhanced fault tolerance, allowing the system to maintain operation even when individual components failed, a critical feature for ensuring the continuous availability of smart city services.

These findings underscore the importance of scalable big data analytics and efficient workflow orchestration in addressing the unique challenges posed by smart city data management. As urban environments continue to generate vast amounts of diverse and complex data, the ability to scale data processing dynamically and manage workflows across distributed systems will be essential for optimizing city operations. The proposed framework's ability to provide high-performance, real-time analytics while maintaining fault tolerance positions it as a robust solution for the evolving needs of smart cities.

Looking ahead, the proposed framework has the potential to significantly impact the evolution of cloud-native software infrastructures for smart cities. By leveraging container-based orchestration and distributed data processing engines, the framework offers a flexible, scalable, and fault-tolerant approach to managing the complexities of smart city data. This architecture can serve as a model for future cloud-native platforms, enabling cities to harness the full potential of their data while ensuring efficient and resilient infrastructure. As the demand for smarter, more responsive cities increases, the continued development and refinement of such frameworks will be crucial in driving the next generation of urban innovation.

References

[1] M. Chinnici, G. Ponti, and G. Santomauro, "Towards Scalable, Interoperable and Replicable Smart City Platform for Urban Application: The ENEA Experience," in *Lecture Notes in Electrical Engineering*, vol. 918 LNEE, 2023, pp. 375–388. doi: 10.1007/978-3-031-08136-1_57.

[2] K. Gupta, Z. Yang, and R. K. Jain, "Urban Data Integration Using Proximity Relationship Learning for Design, Management, and Operations of Sustainable Urban Systems," *J. Comput. Civ. Eng.*, vol. 33, no. 2, 2019, doi: 10.1061/(ASCE)CP.1943-5487.0000806.

[3] V. Sharma, T. K. Vashishth, K. K. Sharma, S. Chaudhary, B. Kumar, and R. Panwar, *The Role of AI and Big Data Analytics in Smart Cities: Leveraging Digital Platforms, Cloud Computing, and IoT*. 2025. doi: 10.1002/9781394233823.ch24.

[4] J. Pereira, T. Batista, E. Cavalcante, A. Souza, F. Lopes, and N. Cacho, "A platform for integrating heterogeneous data and developing smart city applications," *Futur. Gener. Comput. Syst.*, vol. 128, pp. 552 – 566, 2022, doi: 10.1016/j.future.2021.10.030.

[5] I. Tsampoulatidis, N. Komninos, E. Syrmos, and D. Bechtsis, "Universality and Interoperability Across Smart City Ecosystems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13325 LNCS, pp. 218 – 230, 2022, doi: 10.1007/978-3-031-05463-1_16.

[6] K. Wolf *et al.*, "Building enduring smart city data platforms to provide urban management support: lessons learnt from UK Urban Observatories and the US Smart Columbus Operating System," *Front. Sustain. Cities*, vol. 7, 2025, doi: 10.3389/frsc.2025.1512847.

[7] C. Eleftheriadis *et al.*, *Data security for smart cities*. 2025. doi: 10.1049/PBBE009E_ch8.

[8] K. Mori and K. R. Dodiya, *The Role of Digital Technologies, Governance, and Sustainability in Unlocking the Smart Cities Challenges*. 2025. doi: 10.4018/979-8-3373-2327-5.ch001.

[9] M. Kettouch, C. Luca, O. Khorief, R. Wu, and S. Dascalu, "Semantic data management in smart cities," in *Proceedings - 2017 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2017 and 2017 Intl Aegean Conference on Electrical Machines and Power Electronics, ACEMP 2017*, 2017, pp. 1126 – 1131. doi: 10.1109/OPTIM.2017.7975123.

[10] A. D. Cartier, D. H. Lee, B. Kantarci, and L. Foschini, "IoT-big data software ecosystems for smart cities sensing: challenges, open issues, and emerging solutions," *Commun. Comput. Inf. Sci.*, vol. 707, pp. 5 – 18, 2018, doi: 10.1007/978-3-319-72125-5_1.

[11] S. P. Singh Rathore, C. Vishnubhai Dalabhai, C. K. Babubhai Patel, R. Sharma, A. Mathur, and A. Yadav, "Big Data Analytics for Smart Cities," in *Proceedings - IEEE 2024 1st International Conference on Advances in Computing, Communication and Networking, ICAC2N 2024*, 2024, pp. 1289–1294. doi: 10.1109/ICAC2N63387.2024.10895781.

[12] A. Tabbassum, S. Parakh, A. P. Perumal, and P. Chintale, "Developing Cloud-Native Autonomous Systems for Real-Time Edge Analytics," in *2024 IEEE International Conference on Blockchain and Distributed Systems Security, ICBDS 2024*, 2024. doi: 10.1109/ICBDS61829.2024.10837008.

[13] Y. D. Dessalk, N. Nikolov, M. Matskin, A. Soylu, and D. Roman, "Scalable Execution of Big Data Workflows using Software Containers," in *Proceedings of the 12th International Conference on Management of Digital EcoSystems, MEDES 2020*, 2020, pp. 76 – 83. doi: 10.1145/3415958.3433082.

[14] Y. Ding, "Research on Management and Optimization of Big Data Computing Engine Based on Cloud Native Technology IT Architecture," in *Procedia Computer Science*, 2024, pp. 910 – 917. doi: 10.1016/j.procs.2024.09.109.

[15] D. Talia, "Programming Big Data Analysis on Clouds and Extreme Scale Systems," *Adv. Parallel Comput.*, vol. 30, pp. 161 – 173, 2017, doi: 10.3233/978-1-61499-816-7-161.

[16] B. Di Martino, G. Cretella, and A. Esposito, *Cloud Portability and Interoperability*. 2016. doi: 10.1002/9781118821930.ch14.

[17] B. Di Martino *et al.*, "Strategies for flow-based deployment and orchestration in cloud-edge interactive computing," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 250, 2025, pp. 400–407. doi: 10.1007/978-3-031-87778-0_39.

[18] N. Nikolov *et al.*, "Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers," *Internet of Things (Netherlands)*, vol. 16, 2021, doi: 10.1016/j.iot.2021.100440.

[19] P. Emami Khoonsari *et al.*, "Interoperable and scalable data analysis with microservices: Applications in metabolomics," *Bioinformatics*, vol. 35, no. 19, pp. 3752–3760, 2019, doi: 10.1093/bioinformatics/btz160.

[20] M. Babar, W. Iqbal, and S. Kaleem, "Internet of things based smart community design and planning using hadoop-based big data analytics," *Lect. Notes Networks Syst.*, vol. 69, pp. 1046 – 1057, 2020, doi: 10.1007/978-3-030-12388-8_72.

[21] T. R. Rao, P. Mitra, R. Bhatt, and A. Goswami, "The big data system, components, tools, and technologies: a survey," *Knowl. Inf. Syst.*, vol. 60, no. 3, pp. 1165 – 1245, 2019, doi: 10.1007/s10115-018-1248-0.

[22] A. M. S. Osman, "A novel big data analytics framework for smart cities," *Futur. Gener. Comput. Syst.*, vol. 91, pp. 620 – 633, 2019, doi: 10.1016/j.future.2018.06.046.

[23] H. Nasiri, S. Nasehi, and M. Goudarzi, "A survey of distributed stream processing systems for smart city data analytics," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3269961.3282845.

[24] A. Rai, R. Kumar, N. Kumar, and S. Fatima, *Strategies and tools for big data analytics in smart city environments: algorithms and data types*. 2025. doi: 10.1201/9781003616252-74.

[25] M. Jayanthi and C. Pravallika Reddy, "Theoretical design and experimental study for urban data management using energy-saved IoT big data," in *Lecture Notes in Networks and Systems*, vol. 119, 2020, pp. 285–292. doi: 10.1007/978-981-15-3338-9_33.

[26] A. M. S. Osman, A. Elragal, and B. Bergvall-Kåreborn, "Big data analytics and smart cities: A loose or tight couple?," in *Proceedings of the International Conference on ICT, Society and Human Beings 2017 - Part of the Multi Conference on Computer Science and Information Systems 2017*, 2017, pp. 157 – 168.

[27] B. Di Martino, G. Cretella, and A. Esposito, "A comparison between TOSCA and OpenStack HOT through cloud patterns composition," *Int. J. Grid Util. Comput.*, vol. 8, no. 4, pp. 299–311, 2017, doi: 10.1504/IJGUC.2017.088259.

[28] K. K. Mohbey, "An efficient framework for smart city using big data technologies and internet of things," *Adv. Intell. Syst. Comput.*, vol. 714, pp. 319 – 328, 2019, doi: 10.1007/978-981-13-0224-4_29.

[29] J. Chen, A. Li, K. Wang, N. Yan, Q. Liu, and S. Ling, "Design and Implementation of Hydropower and New Energy Big Data Platform Based on Cloud-Native Technology," in *ICNISC 2025 - 11th Annual International Conference on Network and Information Systems for Computers*, 2025, pp. 154 – 162. doi: 10.1145/3776942.3776995.

[30] R. Gu *et al.*, "Fluid: Dataset Abstraction and Elastic Acceleration for Cloud-native Deep Learning Training Jobs," in *Proceedings - International Conference on Data Engineering*, 2022, pp. 2182 – 2195. doi: 10.1109/ICDE53745.2022.00209.

[31] G. Ramesh *et al.*, "A Comprehensive Review on Scaling Machine Learning Workflows Using Cloud Technologies and DevOps," *IEEE Access*, vol. 13, pp. 148559 – 148594, 2025, doi: 10.1109/ACCESS.2025.3599281.

[32] S. A. Goswami, K. C. Kumar Patel, D. A. Darji, S. Patel, and S. Patel, *AI workload automation and orchestration in cloud environments*. 2025. doi: 10.4018/979-8-3693-9694-0.ch002.

[33] M. Adhikari, T. Amgoth, and S. N. Srirama, "A survey on scheduling strategies for workflows in cloud environment and emerging trends," *ACM Comput. Surv.*, vol. 52, no. 4, 2020, doi: 10.1145/3325097.

[34] H. T. El-Kassabi, M. Adel Serhani, R. Dssouli, and A. N. Navaz, "Trust enforcement through self-adapting cloud workflow orchestration," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 462–481, 2019, doi: 10.1016/j.future.2019.03.004.

[35] A. Zafeiropoulos *et al.*, "Data Management and Exchange between a Meta-Orchestration Platform and Data Spaces," in *ACM International Conference Proceeding Series*, 2024, pp. 33 – 36. doi: 10.1145/3685651.3686698.

[36] E. Saeedizade and M. Ashtiani, "Scientific workflow scheduling algorithms in cloud environments: a comprehensive taxonomy, survey, and future directions," *J. Sched.*, vol. 28, no. 1, pp. 1 – 63, 2025, doi: 10.1007/s10951-024-00820-1.

[37] F. Safi-Esfahani and N. Khatibi, "Adaptable decentralized workflow execution with fuzzy framework in cloud computing (ADWEF.Cloud)," *Computing*, vol. 107, no. 6, 2025, doi: 10.1007/s00607-025-01480-5.

[38] D. Danang, N. D. Setiawan, and E. Siswanto, "Pemanfaatan Teknologi Internet of Things untuk Monitoring Kualitas Air Sungai di Wilayah Perkotaan," *J. New Trends Sci.*, vol. 2, no. 1, pp. 23–34, 2024.

[39] E. Muhadi, S. Sulartopo, D. Danang, D. Sasmoko, and N. D. Setiawan, "Rancang bangun sistem keamanan ruang persandian menggunakan RFID dan sensor PIR berbasis IoT," *Router J. Tek. Inform. dan Terap.*, vol. 2, no. 1, pp. 8–20, 2024.

[40] R. Dukaric and M. B. Juric, "BPMN extensions for automating cloud environments using a two-layer orchestration approach," *J. Vis. Lang. Comput.*, vol. 47, pp. 31 – 43, 2018, doi: 10.1016/j.jvlc.2018.06.002.