

Augmenting Smart City Management through the Integration of Digital Twins with Fog Computing

Zainab Awami and Muhammad Bilal*

Cite <https://doi.org/10.64589/juri/231537>

Submitted: April 26, 2026 Revised: May 10, 2026 Accepted: May 17, 2026

ABSTRACT

Fog computing has emerged as a solution to bandwidth and latency constraints in dynamic Internet of Things (IoT) environments; however, large-scale deployments are still limited. Today, abundant urban IoT data, mature 5G networks, and the real-time demands of digital twin cities (DTCs) have created a landscape well-suited to fog computing. This study examines the transition from centralized cloud computing to fog computing as a foundational enabler of real-time DTC systems in urban environments. By redistributing computational workloads across fog nodes, a framework is proposed that significantly reduces data transmission latency by up to 67% and enhances network efficiency by approximately 65–67% compared with conventional cloud-computing architectures. Central to this study is the integration of digital twin technology with a three-layer edge-fog-cloud hierarchy, which enables continuous synchronization between virtual and physical urban environments and supports autonomous real-time decision-making through DTC technologies. The proposed model was validated using iFogSim to systematically compare fog- and cloud-based scenarios in a modeled 25 km² residential city, with surveillance camera counts scaled from 16 to 80 across four operational zones. The results demonstrate the potential of fog computing to enhance the quality and efficiency of smart urban monitoring and management, thereby providing a scalable and sustainable foundation for highly efficient infrastructure in next-generation cities.

Keywords: smart cities, fog computing, internet of things, digital twin, iFogSim, middle east, latency optimization

1. INTRODUCTION

Rapid advancements in information and communication technology (ICT) have driven urbanization, leading to overcrowding in cities. This influx of people has placed increased pressure on various urban services and facilities, particularly infrastructure designed to accommodate a limited number of users. Consequently, numerous technological approaches have been developed to address these challenges in current and future urban cities. A digital twin (DT) is a prominent technology that is dynamically maintained; it constitutes a cyber-physical simulation framework that synthesizes physical models, Internet of Things (IoT) sensors, and heterogeneous data sources from historical, operational, and spatial data records, and enables virtual objects to autonomously influence and manage physical systems without human involvement. Although DT is a promising technology with the potential to make a real difference in modern urban environments, it faces inherent challenges, particularly in optimizing data transmission efficiency, which is critical for maintaining real-time, continuous data streams with minimal latency, loss, and system downtime. To overcome these challenges and support partial integration of data and DT systems, smart cities frequently use cloud computing infrastructure for scalable data storage and processing. At its core, cloud computing

is a type of parallel and distributed system consisting of a collection of interconnected, virtualized computers that are dynamically provisioned and presented as one or more unified computing resources¹. However, cloud architectures suffer from high latency and bandwidth constraints, which limit their suitability for real-time applications². Proposed solutions to this problem include edge computing and mobile edge computing, which push computation closer to data sources to reduce latency³. Although edge computing addresses issues that cloud computing alone cannot fix, such as bandwidth and connectivity, it remains constrained by the limited computational capacity of edge devices⁴. This limitation eventually led to the emergence of fog computing by Cisco in 2012, which combines the strengths of both cloud and edge computing, offering a more balanced solution for complex and dynamic environments, particularly in the context of IoT applications; however, even after a decade of research, large-scale deployments have not materialized. Fog computing has not presented a clear business case; its market is projected at only \$343 million by 2030 compared to cloud's \$791 billion by 2028, and challenges around infrastructure incentives, energy requirements, and resource management remain unresolved³. Today, however, the exponential growth of urban IoT data, the maturation of 5G networks, and the real-time demands of DT cities (DTCs) have created the precise conditions for

which fog computing was designed, making it an architecturally suitable intermediary layer between cloud scalability and edge proximity⁵. Fog computing is designed to distribute and store large datasets across local or regional networks, reducing the load on the cloud while maintaining high responsiveness and greater flexibility in data processing. The integration of fog computing with DT technologies has enhanced the scalability and efficiency of deploying DT systems in modern cities, particularly in smart city environments and applications. Cities today have greater opportunities to adopt DT concepts more efficiently and on a wider scale by integrating these technologies with fog computing, particularly in smart-city contexts and applications. This study investigates the actual returns from integrating fog computing with DT technologies in cities.

2. METHODOLOGY

This study employed a test-and-simulation methodology to systematically evaluate and compare the performance of fog computing and traditional cloud computing. The primary purpose was to quantify the performance differences between the two approaches to identify their respective benefits. All simulations were performed using iFogSim, an open-source simulator⁶. iFogSim is based on the CloudSim framework, which was chosen for its native support of hierarchical IoT environments encompassing edge devices, middle fog computing nodes, and central cloud computing servers, directly mirroring the architectural layers adopted in this study. Two experimental scenarios were independently defined and simulated under identical baseline conditions across a modeled of 25 km² city divided into 4 operational zones, starting with 16 surveillance cameras (4 per zone) scaled up to 80, each transmitting 500-byte tuples every 5 seconds, with identical hardware parameters derived from the iFogSim benchmark toolkit (Cloud: 44,800 MIPS, 40,000 MB RAM; Router: 2,800 MIPS, 4,000 MB RAM). To ensure structural symmetry in comparison, the presence or absence of a fog layer was the sole independent variable driving the performance differences between them. In the fog-computing scenario, the urban environment was modeled as four operational zones, each served by a dedicated fog node responsible for local image processing, alert generation, and data caching, with an intermediary server managing the connection to the cloud layer. In cloud-based scenarios, the fog nodes were disabled to evaluate the changes in system performance. The hardware parameters for this configuration are presented in Tables 1 and 2 in Section 3. In both scenarios, the number of cameras initially started at 16, with four cameras in each zone, and gradually increased during the simulations, while all other parameters remained constant. To ensure reliable outcomes, this study integrated iterative simulations in the iFogSim environment with a mathematical model that calculates both latency and network usage. The first metric, latency, is defined as the time elapsed between data acquisition and processing. It is calculated by summing three components: processing execution time (α), data transfer delay (μ), and operator response time (ϕ), as follows:

$$\text{Latency} = \alpha + \mu + \phi, \quad (1)$$

where α represents CPU execution time computed as the product of tuple length and number of cameras per zone divided by the node MIPS capacity; μ captures the full network traversal path comprising sensor to camera data acquisition (40 ms), camera to fog uplink (2 ms), zone-specific fog to proxy uplink, and network transmission overhead (0.05 ms); and ϕ denotes the operator response latency (1 ms) in this model.

The second metric, network usage, is calculated by multiplying response time by network data volume, reflecting the total connectivity burden imposed by the system architecture, and is expressed as

$$\text{Network Usage} = \text{Latency} \times \text{tupleNWSize (500 bytes)} \\ \times \text{numCameras}. \quad (2)$$

The test environment and DT-based simulation framework, along with the proposed system architecture, are described in Section 3.

2.1. Adopted Architecture. The proposed model comprises a three-layer architectural system that integrates a DT mechanism with fog computing. The model combines cloud, fog, and edge computing into a unified interdependent system, as illustrated in Figure 1. The system operates across five hierarchical tiers. At the lowest level (Tiers 4 and 5), edge computing nodes are represented as cameras equipped with a microcontroller that continuously captures visual data from the physical urban environment and transmits it wirelessly to their assigned fog nodes at Tier 3. The fog nodes receive and process data in real time, detect anomalies, and coordinate with one another without relying on the cloud. Tier 2 contains the proxy server, which organizes and manages the upstream flow of processed data from the fog layer to the cloud. At Tier 1, the cloud receives data and is responsible for long-term storage, historical data management, and offline analytics, whereas all processing tasks are intentionally restricted to the fog layer to minimize both latency and network usage.

Cameras are strategically deployed at the edge layer across the target geographic area, and each camera is integrated with a dedicated microcontroller that serves as a local intermediary. The cameras capture and transmit data to the fog layer. At regular intervals, the microcontroller wirelessly transmits the captured frames to the fog node over Wi-Fi, ensuring that the virtual representation of the city remains synchronized with its physical counterpart. Any changes in the physical environment, such as a structural defect, traffic accident, or resource leak, would be instantly captured, encoded, and transmitted upward through the system without human intervention. The data captured at the edge layer are transmitted to the fog layer, which serves as the computational core of the proposed architecture and is the tier at which the DT transitions from passive data representation to active environmental management. The fog layer consists of geographically distributed fog nodes, each assigned to a specific operational zone within the modeled urban environment. This structure enables localized decision-making independent of communication with a central server, thereby minimizing response time and reducing the communication overhead that characterizes cloud-only systems. Upon receiving image data from

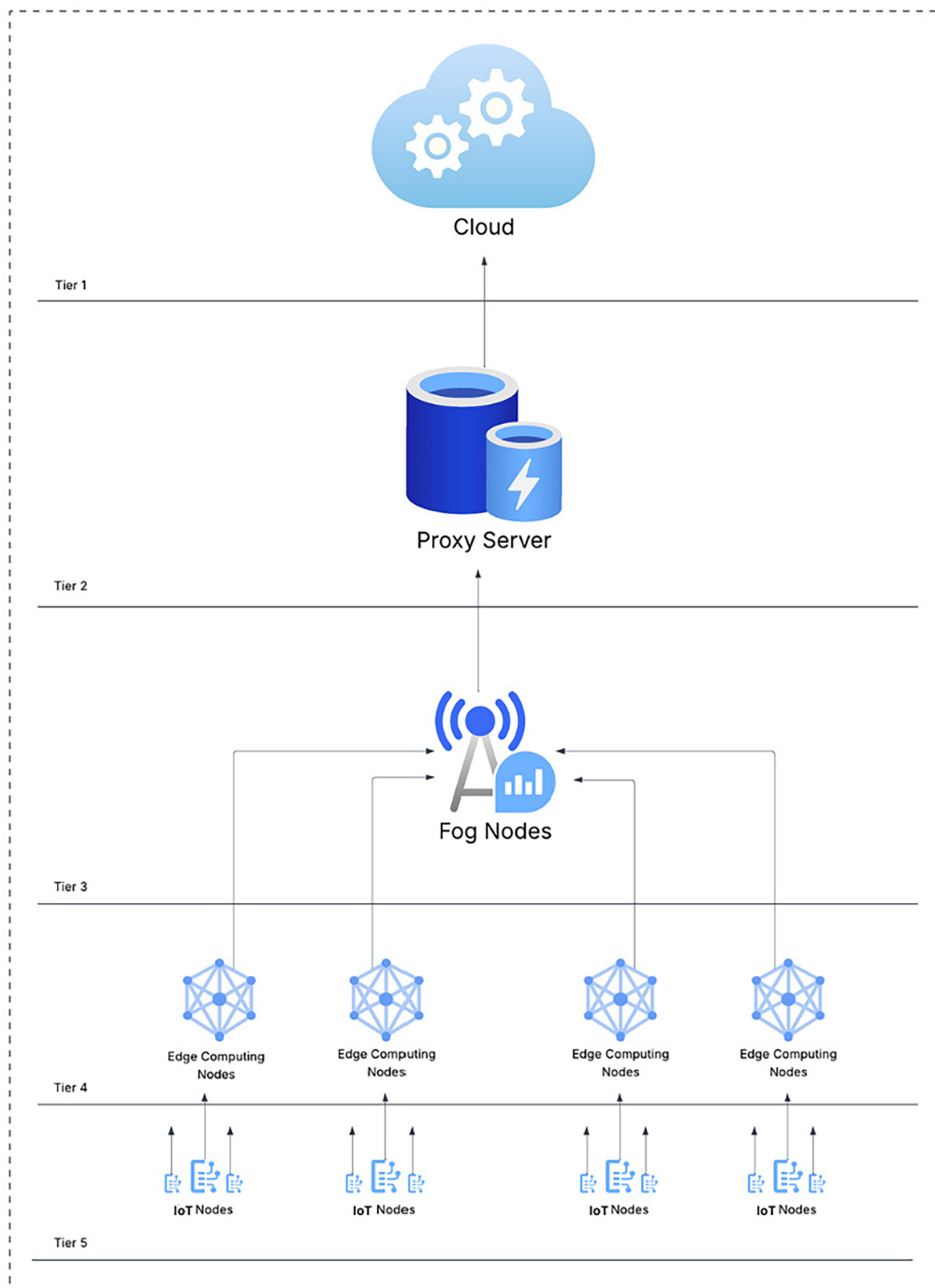


Figure 1. Tiers 1 to 5 of the adopted architecture

the edge layer, each fog node applies preprogrammed image-processing models to analyze captured frames and determine the current spatial condition of its assigned zone. It identifies whether intervention is required based on the detected state, such as a water leak, road obstruction, or unauthorized access event, and triggers the appropriate alert within the system automatically and in real time. Beyond alert generation, each fog node temporarily caches the processed data to maintain a short-term operational record of its zone and exchanges state information with neighboring fog nodes through a peer communication mechanism⁶. At this layer, the predictive and prescriptive capabilities of DT are operationalized. The fog node does not merely report events

occurring in the physical environment; rather, by integrating DT with existing nodes, the system can determine the next step and automatically initiate the corrective response. This capability distinguishes the proposed architecture from simpler sensor alert systems, which lack the semantic understanding of urban conditions that a DT framework provides.

2.2. Experimental Scenarios. To operationalize this architecture in the iFogSim simulation environment, the urban environment was modeled as four distinct operational zones, each served by a single fog node. The initial deployment

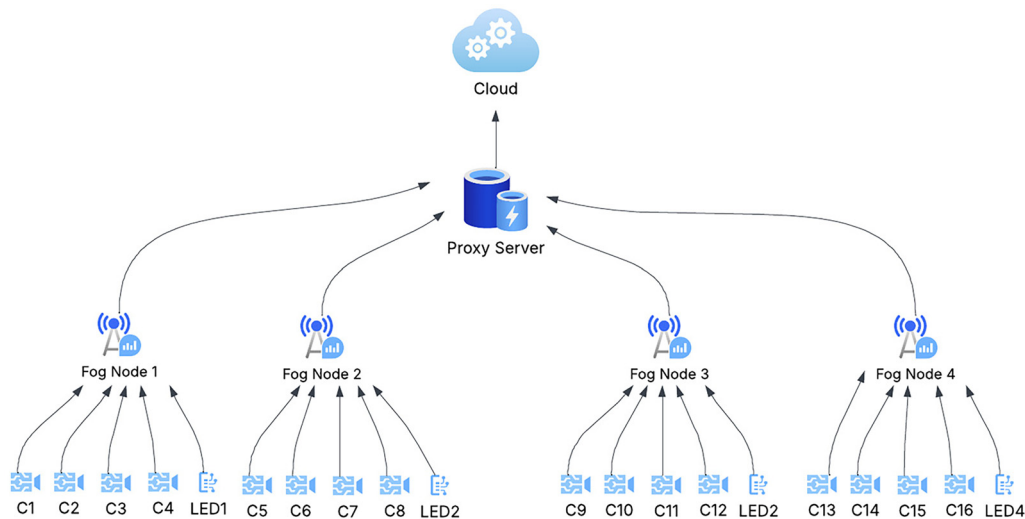


Figure 2. Architecture of proposed fog-based scenario

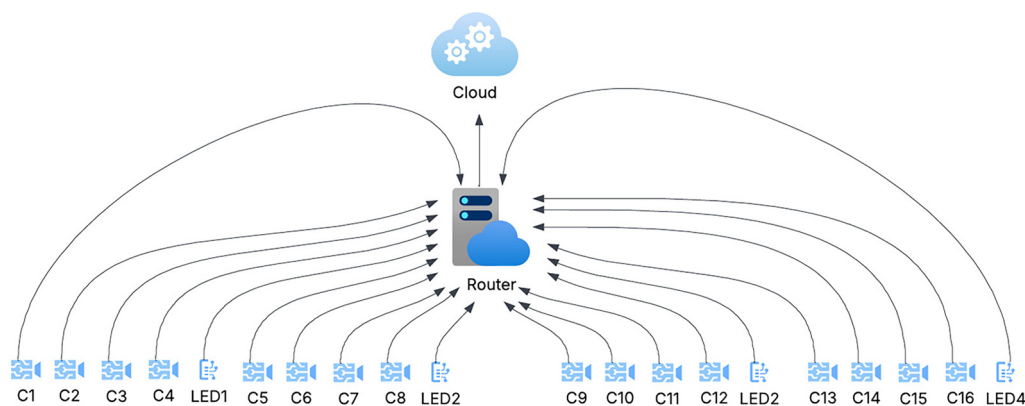


Figure 3. Architecture of proposed cloud-based scenario

consisted of 16 cameras, four per zone, and each group was connected directly to its corresponding fog node. In the fog scenario, a proxy server was introduced between the fog and cloud layers to manage upward communication, as shown in Figure 2. In the cloud scenario, all cameras were connected directly to a centralized cloud server via a router, with processing and memory specifications equivalent to those of the proxies, as illustrated in Figure 3. To ensure that the research outcomes aligned with global standards for comparing and evaluating fog computing

models, the hardware parameters used in the scenarios were derived from the iFogSim benchmark toolkit published by Gupta et al. 7, as shown in Tables 1 and 2 for the cloud-based scenario parameters.

3. RESULTS AND DISCUSSION

The simulation modeled a 25 km² smart residential city divided into four heterogeneous zones integrated with a fog computing

Table 1. Simulation parameters for the fog-integrated urban monitoring scenario⁶

Parameter	Cloud	Router (CPU)	Parameter
CPU length (MIPS)	44,800	2,800	CPU length (MIPS)
RAM (MB)	40,000	4,000	RAM (MB)
Uplink bandwidth (Mbps)	100	10,000	Uplink bandwidth (Mbps)
Downlink bandwidth (Mbps)	10,000	10,000	Downlink bandwidth (Mbps)
Hierarchy level	0	1	Hierarchy level
Rate per MIPS (MIPS)	0.01	0.00	Rate per MIPS (MIPS)
Busy power (W)	1,648.00	107.34	Busy power (W)
Idle power (W)	1,332.00	83.43	Idle power (W)

Table 2. *Cloud-based scenario simulation parameters*⁶

Parameter	Cloud	Router (CPU)	Parameter
CPU length (MIPS)	44,800	2,800	2,800
RAM (MB)	40,000	4,000	4,000
Uplink bandwidth (Mbps)	100	10,000	10,000
Downlink bandwidth (Mbps)	10,000	10,000	10,000
Hierarchy level	0	1	2
Rate per MIPS (MIPS)	0.01	0.00	0.00
Busy power (W)	1,648.00	107.34	107.339
Idle power (W)	1,332.00	83.43	83.34

Table 3. *Fog-based scenario latency results per zone across all camera configurations (ms)*

Cameras /Zone	Total Cameras	Latency Zone 1 (ms)	Latency Zone 2 (ms)	Latency Zone 3 (ms)	Latency Zone 4 (ms)	Fog Average (ms)
4	16	47.764	46.764	48.883	45.764	47.294
8	32	48.479	47.479	49.717	46.479	48.038
12	48	49.193	48.193	50.550	47.193	48.782
16	64	49.907	48.907	51.383	47.907	49.526
20	80	50.621	49.621	52.217	48.621	50.270

architecture to support a real-time DT system. The zones were distributed into four: Zone 1 covered 35% of the total area (8.75 km²) with a standard 2,800 MIPS fog node and a 4 ms uplink latency; Zone 2 covered 20% (5.00 km²) with a 2,800 MIPS node and an uplink latency of 3 ms; Zone 3 covered 30% (7.50 km²) with a 2,400 MIPS and fog node coupled with a 5 ms uplink, reflecting both its larger geographic footprint and degraded hardware infrastructure; and Zone 4 covered the remaining 15% (3.75 km²) with a 2,800 MIPS node and the lowest uplink latency of 2 ms. Each zone was initially equipped with four surveillance cameras that operate at a 5 s transmission interval, resulting in

16 cameras for the city at the initial base. However, to evaluate system scalability under increasing data loads and assess the system's operations under harsh conditions, the number of cameras per zone was deliberately increased during execution in iFogSim. This increase was adjusted, tested, and simulated to monitor and measure the effects of adding additional cameras to the system. The number of cameras was gradually increased across five levels: 4, 8, 12, 16, and 20 cameras per zone, for a maximum of 80 cameras citywide, whereas the number of fog nodes remained fixed at four across all scenarios. Each camera generated 500-byte data tuples that were transmitted to the processing layer

Table 4. *Cloud-based scenario latency and network utilization results*

Total Cameras	Cloud Latency Zone 1 (ms)	Cloud Latency Zone 2 (ms)	Cloud Latency Zone 3 (ms)	Cloud Latency Zone 4 (ms)	Cloud Latency Average (ms)	Cloud Network Usage (byte•ms)
16	143.229	143.229	143.229	143.229	143.229	1,145,832
32	143.407	143.407	143.407	143.407	143.407	2,294,512
48	143.586	143.586	143.586	143.586	143.586	3,446,064
64	143.764	143.764	143.764	143.764	143.764	4,600,448
80	143.943	143.943	143.943	143.943	143.943	5,755,720

Table 5. *Comparative latency and network utilization between fog and cloud scenarios*

Total Cameras	Fog Average (ms)	Cloud Latency (ms)	Fog Network Usage (byte•ms)	Cloud Network Usage (byte•ms)	Network Reduction %
16	47.294	143.229	378,352	1,145,832	67.0%
32	48.038	143.407	768,608	2,294,512	66.5%
48	48.782	143.586	1,170,768	3,446,064	66.0%
64	49.526	143.764	1,584,832	4,600,448	65.5%
80	50.270	143.943	2,010,800	5,757,720	65.1%

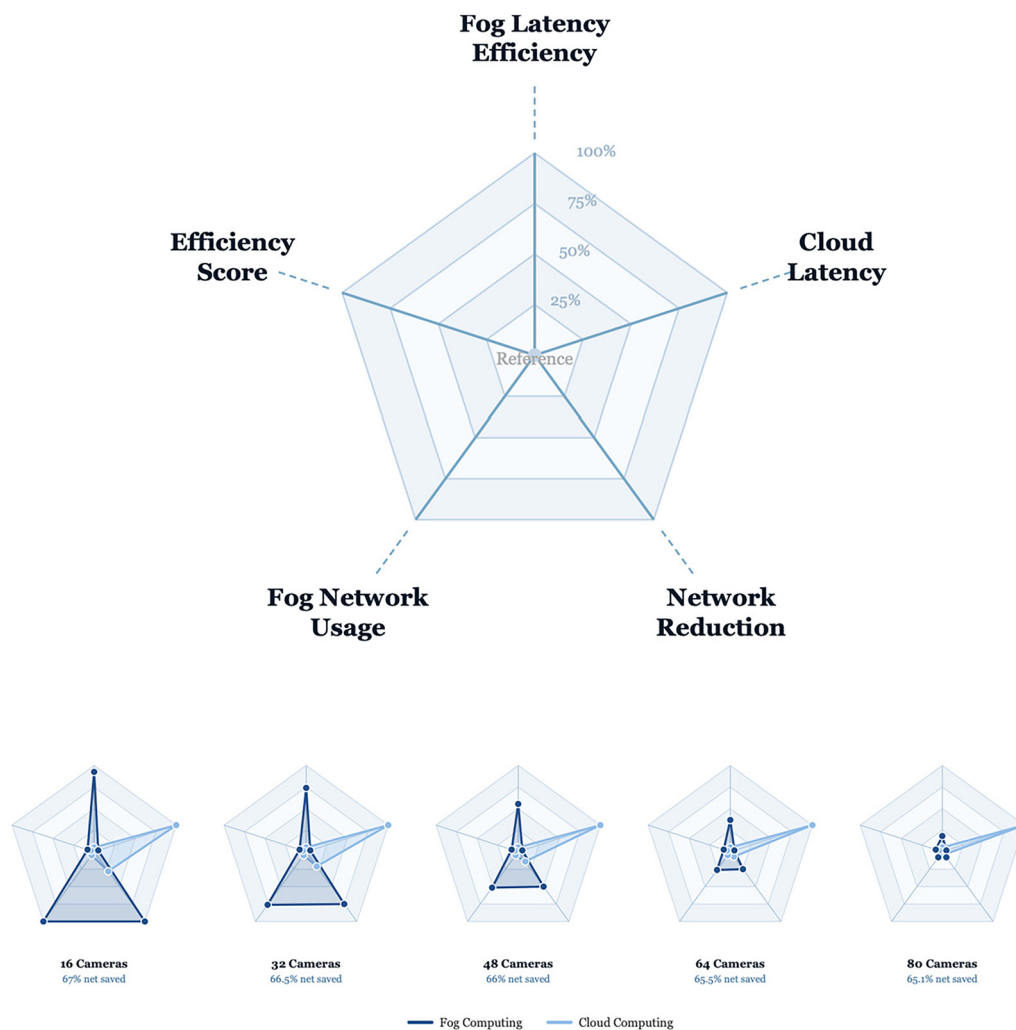


Figure 4. Performance comparison of fog vs. cloud across scaling scenarios

for image analysis and DT state computation. The evaluation was based on the latency and network usage metrics mentioned in Section 2. However, these constants are not fixed absolutes: they vary with the specific hardware features and capabilities, network conditions, and deployment environments in each urban scenario. Because the simulation was conducted under standardized benchmark conditions that mirror these values, these values were adopted as representative defaults to ensure consistency and reproducibility. The results were obtained through repeated simulations by varying the number of cameras and extracting the outcomes for each zone. The results for the fog- and cloud-based scenarios are presented in Table 3 and 4, respectively. To measure the variance and differences between these scenarios, Table 5 presents the numerical performance differences. The differences are also illustrated graphically in Figure 4, which shows the fog system's superiority in achieving lower latency and reducing network congestion from 67% to 65.1%, demonstrating its overall efficiency in minimizing bandwidth consumption. Furthermore, the network reduction metric remained above 65% even at peak capacity (80 cameras), underscoring the robustness of the proposed fog architecture in handling large-scale, intensive data. This performance consistency highlights the decentralized nature of the fog layer, which demonstrates how the

system compensates for localized hardware limitations, such as lower MIPS capacity and higher latency in Zone 3, thereby preventing a total system bottleneck. Consequently, these findings collectively validate that integrating a fog-based layer is not merely an optimization task but also a fundamental requirement for DT systems. Such systems inherently consume significant energy, undermining their effectiveness in cloud computing environments with high power demands. Therefore, the combination of these factors may result in an unsustainable cumulative energy footprint, rendering the solution economically infeasible. Because urban areas are defined by their highly complex and stochastic nature along with high density and data demands, sustainable, scalable, and automated solutions are in demand, which is precisely what this study aims to provide with the integration of fog computing and DT.

4. CONCLUSIONS

This study underscores the transformative potential of integrating fog computing with DT technology to overcome the inherent latency and bandwidth constraints of centralized cloud-based architectures in urban environments. The benefits were demonstrated using iFogSim simulations, which extracted

a quantifiable performance leap. The fog-node architecture achieved up to 67% reduction in network usage and maintained a stable sub-51 ms response threshold even under high-density conditions involving 80 cameras. Beyond the immediate technical gains in reducing latency and network usage, the broader adoption of this integrated solution may also offer socioeconomic and environmental benefits. By minimizing data transmission redundancy, the proposed framework has the potential to reduce carbon emissions and mitigate electrical grid strain, thereby supporting global sustainability goals. In addition, improvements in operational efficiency may enable the strategic reallocation of financial capital to other developmental sectors, thereby enhancing the overall economic viability of smart city initiatives. Consequently, the integrated framework may be particularly relevant for nations pursuing sustainable smart-city development, including The Line in Saudi Arabia, Khalifa City and Masdar City in the UAE, as well as international benchmarks, such as Songdo (South Korea), Nusantara (Indonesia), and Tatu City (Kenya). Notwithstanding these significant performance enhancements, certain limitations inherent in the current study warrant acknowledgment, primarily that the empirical evaluation relies on synthetic workloads within a simulated environment which may not fully replicate the dynamic network and hardware heterogeneity of empirical deployments, nor does it explicitly incorporate the complex cybersecurity protocols essential for distributed cloud-edge-fog networks within the urban areas. To facilitate this transition from theoretical modeling to empirical implementation, subsequent research might overcome these limitations by deploying localized, hardware-in-the-loop physical testbeds to validate the framework to be fully fit within the real-world conditions, expanding the architectural simulation to model complex, multi-modal urban data streams under maximum network stress, and developing lightweight, decentralized cryptographic protocols optimized for resource constrained edge devices to guarantee data integrity across the distributed ecosystem. Thus, the accelerating momentum of urban evolution and large-scale infrastructure deployment present an unprecedented opportunity to redefine digital capabilities by integrating fog computing and DTs. This integration represents a significant shift in traditional urban design by introducing a systemic paradigm for perceiving city dynamics and data ecosystems. This provides policymakers and stakeholders with a high-fidelity, actionable platform for fostering a new era of resilient, automated, and intelligently managed urban environments. Ultimately, the future of research rests in the hands of those empowered to translate empirical evidence into a tangible framework. Hence, the transition from simulation to practical application requires concerted efforts to measure practical outcomes in real-world urban

testing environments, ensuring that technical standards align with socioeconomic objectives. By bridging the gap between theoretical data, governance, and stakeholders, we can eventually translate these findings into tangible results and establish a new global standard for city design.

AFFILIATIONS AND AUTHOR DETAILS

Undergraduate Author

Zainab Awami – Department of Architecture and City Design, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; [0009-0003-8512-6883](mailto:s202260780@kfupm.edu.sa)
Email: s202260780@kfupm.edu.sa

Corresponding Author

Muhammad Bilal – Research Mentor, Assistant Professor, Department of Architecture and City Design & Interdisciplinary Research Center for Aviation & Space Exploration, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia; [0000-0003-1022-3999](mailto:muhammad.bilal@kfupm.edu.sa)
Email: muhammad.bilal@kfupm.edu.sa

REFERENCES

- (1) Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). *Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility*. *Future Generation Computer Systems*, 25(6), 599–616. <https://doi.org/10.1016/j.future.2008.12.001>
- (2) Perera, C., Qin, Y., Estrella, J. C., Reiff-Marganiec, S., & Vasilakos, A. V. (2017). *Fog computing for sustainable smart cities: A survey*. *ACM Computing Surveys*, 50(3), Article 32. <https://doi.org/10.1145/3057266>
- (3) Kocyigit, A. (2023). *Latency estimation and computational task offloading in vehicular mobile edge computing applications*. *IEEE Transactions on Vehicular Technology*.? <https://doi.org/10.1109/TVT.2023.3334192>
- (4) Angel, N., Ravindran, D., Vincent, P. M. D. R., Srinivasan, K., & Hu, Y.-C. (2022). *Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies*. *Sensors*, 22(1), 196. <https://doi.org/10.3390/s22010196>
- (5) Badidi, E., Mahrez, Z., & Sabir, E. (2020). *Fog computing for smart cities' big data management and analytics: A review*. *Future Internet*, 12(11), 190. <https://doi.org/10.3390/fi12110190>
- (6) Gupta, H., Dastjerdi, A. V., Ghosh, S. K., & Buyya, R. (2017). *iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments*. *Software: Practice and Experience*, 47(9), 1275–1296. <https://doi.org/10.1002/spe.2509>