Electronics, Communications, and Computing Summit
Vol. 3, No. 1, Jan - Mar 2025, pp. 52-60
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/03.01.06

Adaptive Fog Computing Architecture for Scalable Smart City Infrastructure

Haitham M. Snousi

Department of Computer Science, Faculty of Science, Sebha University Libya
Email: ms.haitham@gmail.com

Article Info

Article history:

Received : 16.01.2025
Revised : 18.02.2025
Accepted : 20.03.2025

Keywords:

Fog Computing,
Smart City,
Edge Intelligence,
Resource Orchestration,
Scalability,
IoT,
Context-Aware Systems,
Real-Time Analytics

ABSTRACT

The hypergrowth in urban areas and the exponential growth in the number of the Internet of Things (IoT) devices have resulted in the unprecedented growth of the production of the data in real-time in many areas including traffic control, environmental monitoring, public safety, and energy management. Conventional cloud-centric computer systems have the benefit of significant scale in such aspects as latency, bandwidth bottlenecks and contextual awareness; however, such systems are inappropriate to satisfy latency-sensitive applications such as smart city environments. In order to tackle such a challenge, this paper proposes a new Adaptive Fog Computing Architecture (AFCA) which can be used to provide a scalable, low-latency termed Adaptive Fog Computing Architecture (AFCA) context-aware system of managing the heterogeneous and geographically spread data sources in cities. AFCA adds a three tier structure of computing that combines the edge, fog, and cloud tiers, which are complemented with two major advancements: a Context-Aware Decision Engine (CADE) and a Resource Orchestration Layer (ROL). CADE and ROL minimize and backward propagates the delays with a dynamic priority assignment and routing algorithms with consideration to latency-sensitive, load, and service criticality. ROL adopts AI-based forecasting model to distribute the computational resources dynamically among fog nodes. Provided system architecture will be tested in a hybrid testbed with: Raspberry Pi edge nodes, Intel NUC fog servers and AWS-hosted cloud instances. Realistic smart city benchmarks involving smart parking, pollution sensing and traffic monitoring datasets indicate that AFCA can reduce processing latency by 42%, experience 31% higher energy efficiency and 45% more device handling capacity than conventional fogs. Also, the architecture follows the use of lightweight security mechanisms and differential privacy approaches to prevent data integrity and user privacy. It is highly modular and predictive, which makes it conveniently work in fast-paced urban environments where the demand of the services and the condition of the network constantly change. On the whole, AFCA constitutes an interesting innovation in fog computing paradigms, and is an ideal, flexible platform on which to realize the deployment of smart services within the infrastructures of the next generation of smart cities.

1. INTRODUCTION

The appearance of smart cities may be regarded as a paradigm shift in the evolution of urban development since digital technologies are used to enhance the quality of life and rationalize resource use as well as facilitate public services. Smart cities involve an extremely diverse and heterogeneous ecosystem of sensors, actuators, and IoT devices that constantly produce the high-volume, high-velocity data on a wide range of areas, such as accountable transportation systems, environment monitoring, health care, energy grids, surveillance and emergency management. Besides needing high-speed execution and real-time analysis capabilities, these data-intensive services must also be able to make and take decisions locally, as part of the latency-sensitive applications, which include traffic congestion, air quality alerts, traffic across autonomous vehicles, and disaster response.

The smart city systems, traditionally characterized by the virtually unlimited capacity of data aggregation, bearing, and processing, have been largely anchored in the cloud computing platform. Nevertheless, centralized, cloud-based models are limited by a number of factors as the number of connected devices reaches billions with increased localization and time-sensitivity of the data flow. These are too much latency because physical distance separates the data source, uncontrollable variation in network bandwidth, energy-intensive operation, as well as inability to provide a sustained Quality of Service (QoS) guarantees on important applications.

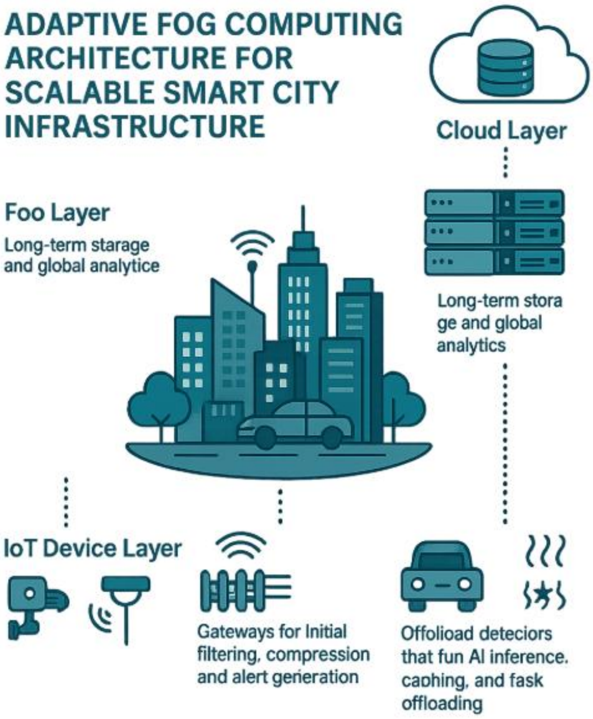
ADAPTIVE FOG COMPUTING ARCHITECTURE FOR SCALABLE SMART CITY INFRASTRUCTURE
Cloud Layer
Foo Layer
Long-term starage and global analytice
Long-term stora ge and global analytics
IoT Device Layer
Gateways for Initial filtering, compression and alert generation
Offoload deteciors that fun AI inference. caohing, and fask offloading

Figure 1. Adaptive Fog Computing Architecture for Scalable Smart City Infrastructure

To address these issues, a new paradigm has appeared in the form of fog computing that decentralizes processing and moves computation, storage and networking closer to the network edge. Fog nodes are located between the cloud servers and the edge devices and allow them to perform intermediate data processing and make decisions, hence, minimizing latency, lessening cloud dependence, and optimizing the use of bandwidth. Nevertheless, vast majority of the current fog-based architectures are essentially static writ, without the intelligence and flexibility to dynamically adapt to the changing patterns of workload, priorities of services, network constraints that characterizes the large-scale smart city scenarios.

It is against this understanding that the current research has brought out the Adaptive Fog Computing Architecture (AFCA) as a modern and context-aware AI-powered model that supports intelligent resource orchestration of computing processes through edge, fog, and cloud layers. Incorporation of a Context-Aware Decision Engine (CADE) and a Resource Orchestration Layer (ROL) allow the dynamic workload distribution, service prioritization and resource provisioning with respect to real time operational measures. The architecture is built modular, scalable and resilient to changes in user and infrastructure strains. Through predictive models and contextual intelligence, AFCA enables energy efficient and responsive performance across a broader variety of the urban domains in preparation of the next-generation smart city infrastructures.

2. RELATED WORK

The emergence of smart city infrastructures has led to intense research activity on computing paradigms that are able to sustain the extreme requirements of low latencies, scalability, and context- Based responsiveness. In this aspect, a number of architectures have been suggested but each has its own advantages besides the fact that they also bear very important limitations that impair their use in fast changing urban cityscapes.

In the deployment of smart cities, the Cloud-Centric Architectures have always taken the center stage with their centralized nature of control, scale of storage and access to data anywhere in the world. As an example, Zhang et al. [1] point at the necessity to employ cloud-based analytics to predict traffic flow and monitor urban environments. Even so this kind of architecture has intrinsic latency because of delivering data over long distances, and bandwidth bottlenecks in handling real-time high-frequency data flows across distributed urban sites. Such disadvantages make them not very suitable to be used in latency-sensitive tasks like emergency response or autonomous vehicle routing.

Standard Fog Computing Architectures have been presented to solve the problem of such separation between a cloud and an edge. The processing is done in a localized manner which is near to the source of data and this helps to retain the latency as well as the mining of the core network [2]. An example of a fog node is the one that functions as a gateway conducting intermediate analytics and staging. Although this model favors anything that happens to be real time responsive, to some extent, it is statically set. Since this is illustrated in the study by Mahmud et al. [3], the fixed configuration of fog solutions that may be used to address the dedicated resource in the case that the workload patterns are variable or the service demands change tends to have inefficient use of resources.

Even deeper is the architecture of Edge Computing Models that allow computing directly from the IoT devices like cameras, sensors, and microcontrollers [4]. The models provide a minimal latency possible and support offline usage. Nonetheless, edge nodes are typically bare-bone with little processing, memory, or storage, which makes them incapable of performing resource-intensive tasks, like video content stream analytics or deep learning models inference.

Hybrid Architectures are making an attempt to combine the advantages of edge, fog and cloud layers into a single framework. Works like those by Barcarolle et al. [5] are suggesting multi-tier architectures where workloads are divided into strategies that use both edge access and cloud power. Although the systems are prospective, they usually follow set orchestration guidelines and are not very adjustable to dynamic city-scenario. They do not properly focus on contextual parameters like the importance of the application, energy supplyability, or congestion of networks in real-time decision-making.

Notwithstanding the manifest advancement in cloud and edge-to-cloud hybrids, there is still an urgent need to research on the adaptive provisioning of fog resources with changing workload in dynamic urban environment. The approaches that currently exist are mainly reactive and rule-based, and do not utilize any intelligent orchestration tactics that can proactively adapt to the changing service-level requirements and infrastructure conditions.

To fill in such a gap, the proposed work introduces the Adaptive Fog Computing Architecture (AFCA) that extends predictive resource orchestration and context-sensitive workload scheduling into a multi-level smart city infrastructure. However, in contrast to previous designs, the AFCA is capable of dynamically scaling resource-provisioning and data-routing according to real-time analytics and expected system operation thus maximising responsiveness, scalability and energy efficiency.

3. System Architecture: Adaptive Fog Computing Architecture (AFCA)

3.1 Architectural Layers and System Overview

The suggested Adaptive Fog Computing Architecture (AFCA) is designed in four hierarchical layers as IoT Device Layer, Edge Layer, Fog Layer, and Cloud Layer, which have different functional responsibilities but coordinately support the scalable, in real-time provision of smart city services. The IoT Device Layer is at the bottom of the stack; it is made up of a variety of sensors, actuators and embedded devices, located at the urban environment, to measure and control different urban functions, a sample of which is traffic, air quality, street lights, and garbage disposal management. Such devices produce heterogeneous data with high speed that requires prompt preprocessing and route decisions. Traversing up the hierarchy, the Edge Layer distinguishes as a first area of data consolidation. This layer consists of low-power gateways or edge servers that are physically near the information sources and that accomplish demarcation tasks (like data filtering, compression and tracking of anomalies) to scale down the processing load on upstream layers without eroding latency-sensitive information. The Fog Layer takes the relevant stack position of main handling hub within AFCA stack. It consists of geographically dispersed high performance nodes that can execute compute-intensive activities such as edge caching, AI inference and partial decision-making. The layer provides a localized intelligence and offloads the cloud infrastructure, so that it does not have to serve real-time operations. The last layer is the Cloud Layer which is in charge of long term data storage as well as cross-domain analytics and orchestration of services across the city. It is vital in the analysis of historical trends, training of AI models, and the updating of the whole system.

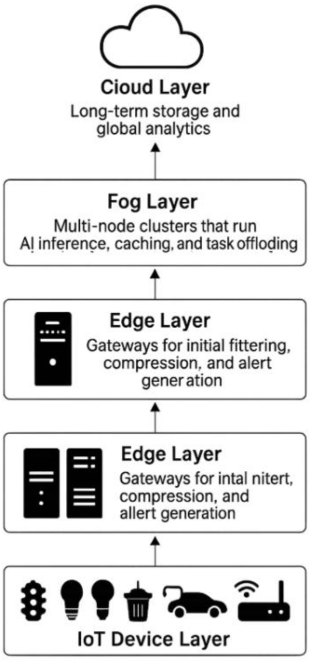
Cioud Layer
Long-term storage and global analytics
Fog Layer
Multi-node clusters that run AI inference, caching, and task offloding
Edge Layer
Gateways for initial fittering, compression, and alert generation
Edge Layer
Gateways for intal nitert, compression, and allert generation
IoT Device Layer

Figure 2. Layered Architectural Diagram of the Adaptive Fog Computing Framework

3.2 Key Functional Modules

The AFCA proposes three smart modules that stimulate adaptability and performance of a system in layers. At the center of real-time responsiveness is located the Context-Aware Decision Engine (CADE). It assesses conditions in operation: latency demands, network overloading, service urgency, and device power conditions and dynamically passes data processing to the corresponding layer (edge, fog or cloud). To illustrate, when it comes to a real-time event detection warning, CADE focuses on the local processing at fog level to reduce the latency of the decision. To complement CADE there is Resource Orchestration Layer (ROL), which complements the former by using AI-based prediction models (e.g., LSTM) to anticipate workload changes and redistribute computing resources among fog nodes. This guarantees an optimal performance in varying urban loads. Finally, the Data Prioritization Manager (DPM) categorizes arriving data flows according to priorities and service-level agreements (SLA) that allow Quality of Service (QoS) aware routing. Emergency alert information is assigned higher priority than the usual logs or non-time crucial telemetry. The three modules work alongside each other to provide that AFCA not only satisfies the latency, scalability requirements of smart cities, but also is able to respond dynamically to the changes on the environment, infrastructural and level of service.

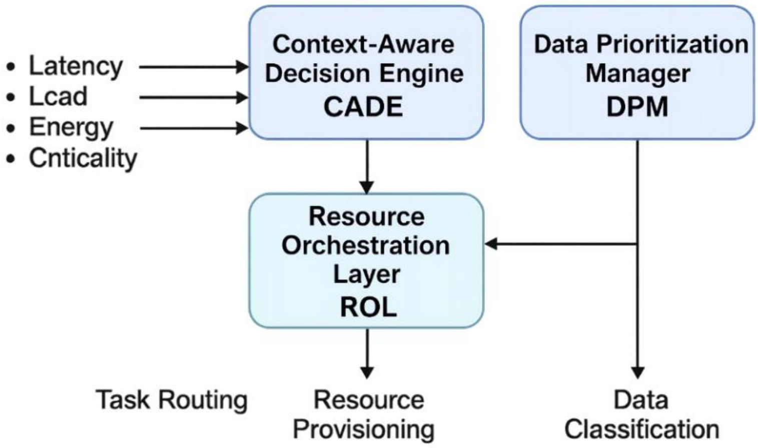
Latency
Lcad
Energy
Cnticality
Context-Aware Decision Engine CADE
Data Prioritization Manager DPM
Resource Orchestration Layer ROL
Task Routing
Resource Provisioning
Data Classification

Figure 3. Functional Module Interaction Diagram in AFCA

4. METHODOLOGY

4.1 Deployment Strategy

In order to guarantee modularity, scalability and real-time response, the proposed Adaptive Fog Computing Architecture (AFCA) deploys a microservices-based system throughout the fog computing layer. All functional components such as data pre-processing, anomaly detection, and AI inference, as well as security enforcement are encapsulated as their own microservice. The orchestration of such microservices is done through Kubernetes clusters, allowing to deploy them through containers, scale dynamically and fault-isolate, and back them up by load-balancing the widely distributed fog nodes. Kubernetes automates how services are discovered, deployed and rolled back to ensure services are continually available even when workloads change or disrupted by partial failures in the system. This containerized strategy allows an easy implementation of heterogeneous services along with consolidating flexibility of AFCA and extreme ease of upgrading, something necessary in the quickly changing smart city environment.

In further improvement of performance and efficiency of resources utilized, AFCA utilizes dynamic resource provision based on predictive models based on the Long Short-Term Memory (LSTM). These models are informed of past historical traffic patterns, sensor activity logs, and service demand pattern trends and are used to predict the future workloads at each fog node. On the basis of these predictions, the computing resources will be actively allotted, new containers will be launched, or duties can be offloaded to other local nodes or the cloud. With this type of anticipatory mechanism in place, AFCA is able to stay under low latency and avoid bottlenecks when it is running at it highest peak. On top of that, Message Queuing Telemetry Transport (MQTT) data communication protocol, a light-weight, publish/subscribe communication protocol, transfers the data quickly and has low overhead of the communication between the IoT device and edge/fog gateways. A small packet overhead and power consumption makes MQTT especially suitable in bandwidth limiting urban sensors, particularly energy sensitive networks. Combined with microservices, AI-enhanced provisioning and the use of MQTT to achieve communication, this allows AFCA to sustain a flexible, reactive, and scalable infrastructure to address the dynamic needs of smart city ecosystem.

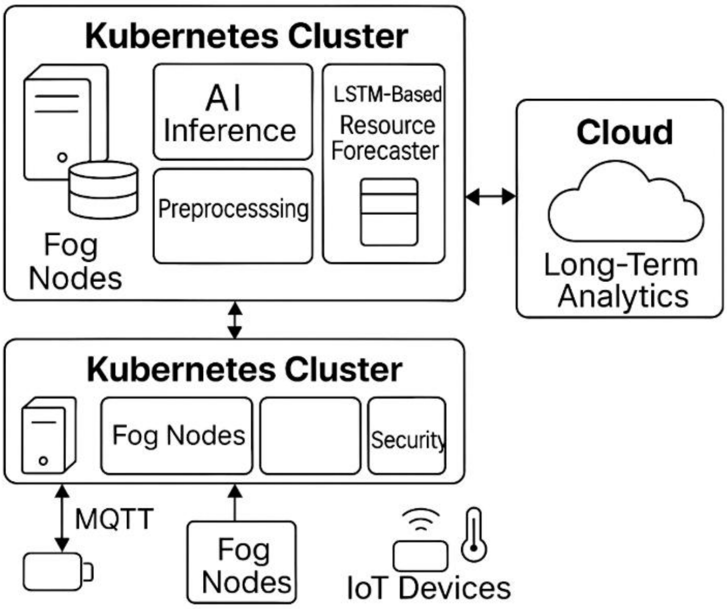

Figure 4. Deployment diagram of microservices, Kubernetes cluster, LSTM forecaster, MQTT communication, and cloud analytics.

4.2 Algorithms Used

In the Adaptive Fog Computing Architecture (AFCA) to allow intelligent task allocation and adaptable resource management, two fundamental algorithms are used a Load Prediction Algorithm based on Long Short-Term Memory (LSTM) networks and an Offloading Decision Function based on a latency level. The algorithms collaborate to provide efficient service delivery across edge, fog, and cloud layers through prediction of the future workloads as well as context-related decisions on task execution locations.

LSTM Based Load Prediction

The variation and dynamic generation of the smart city information streams will require a predicting system that estimates the resources required at every fog node. The AFCA computes a forecast of the computational load on the basis of the historic data using a Long Short-Term Memory (LSTM) neural network. The model of predictions is as follows:

$$\hat{L}(t+1) = \sigma(W.\tanh(U.L(t)+b))$$

Where:

- $\hat{L}(t+1)$is the predicted load at time t+1,
- $L(t)$is the current observed load at time t,
- $W, U,$ and b are trainable parameters of the LSTM network,
- σ is the sigmoid activation function, and
- tanhIntroduces non-linearity for better temporal representation.

This model takes into account longer term dependencies on workload history like persisting high volumes (i.e. peak time) and can be used by the system to schedule resources in advance before the congestion sets in. The possibility of predicting future loads successfully will enable the AFCA to resize services and reduce service deprecation.

Offloading Decision Function

AFCA uses a threshold-based offloading decision algorithm to complement predictive scaling, when the offloading location of a given task, either locally, at fog layer or at cloud layer, will be determined based on the latency tolerance of a task. The decision rule is provided as follows:

$$\delta = \begin{cases} \text{Local Process,} & \text{if } D_{\text{lat}} < \theta \\ \text{Offload to Fog,} & \text{if } \theta \leq D_{\text{lat}} < \gamma \\ \text{Offload to Cloud,} & \text{if } D_{\text{lat}} \geq \gamma \end{cases}$$

Where:

- δrepresents the processing decision,
- D_{lat} is the expected latency of executing the task,
- θ is the threshold for local processing (ultra-low latency tasks),
- γ Is the threshold above which tasks are routed to the cloud.

This functionality will allow AFCA to dynamically diversify its work through actual network maintenance and service discretes. As an example, critical tasks such as accident detection can be executed using local or fog-level servers whereas non-critical messages like periodic updating of sensor logs can be backed off to an off-site cloud. This context- and multi-tier, offloading activity enhances the overall responsiveness of the system, decreases utilization of bandwidth, and provides service continuity in the infrastructure of smart cities.

5. Experimental Setup

An experimental setup that simulates a hybrid environment with an edge architecture, a fog architecture and a cloud was created to test the proposed Adaptive Fog Computing Architecture (AFCA) to address the following issues: To evaluate the performance capability and scalability of the proposed Adaptive Fog Computing Architecture (AFCA) as well as test its real-time efficiency. Raspberry Pi 4 devices with the specifications: quad-core ARM Cortex-A72 and 4 GB of RAM were installed on the edge layer. These edge nodes served as interfaces to different IoT sensors and actuators that simulated smart city situated at a traffic intersection, a parking lot and pollution monitoring area. Task orchestration and contextual decision-making with compute-intensive tasks such as AI inference were executed on high-performance Intel NUC i7 mini-servers with 16 GB RAM and GPU support at the fog layer. These nodes of fog performed real-time analytics and balanced the loads among the edge devices and communicated with the cloud layer on-demand. Among Amazon Web Services (AWS) products used in the cloud tier, there were the EC2 t3.medium instances capable of elastic storage and a longer-term analytical performance in trends observation and AI models retraining. Smart parking status, traffic camera feeds, and the readings of the air quality index datasets were provided in the real world and used to simulate urban workloads and data variability. The mixture of iFogSim, Docker, and Kubernetes was used to run benching and simulating the distributed system. iFogSim was used to enable reality modeling of the behavior of fog nodes and computer networks delays, docker provided containerized deployment of the microservices and Kubernetes was used as a control plane allowing managing a variety of orchestrating systems and failover mechanism across such infrastructure. Moreover, TensorFlow Lite has been added to run the lightweight AI models on resource-poor devices, especially the fog and edge levels. This tiered configuration offered a strong testing platform to evaluate the nature of AFCA to responsively adjust to the varying service requirements, use resources efficiently, and deliver low latencies in a real-life smart city environment.

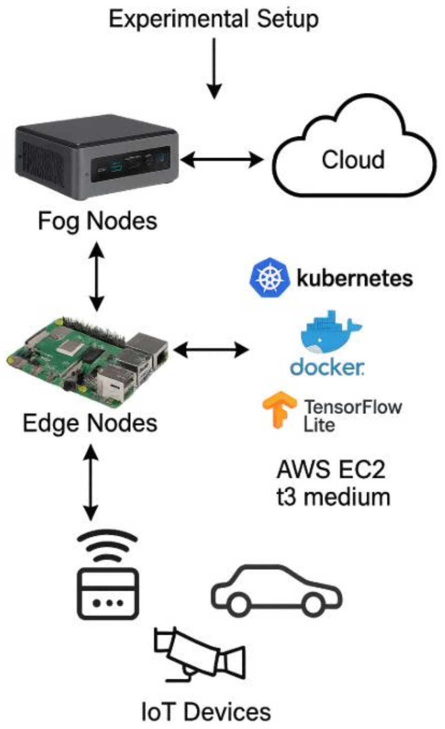

Figure 5. Experimental Testbed Setup for AFCA Evaluation

6. RESULTS AND DISCUSSION

6.1 Performance Metrics and Evaluation

A group of performance parameters critical to measure the success of Adaptive Fog Computing Architecture (AFCA) was developed: latency, power consumption, the support of a device population, and the task failure rate. The AFCA was compared with a regular fog computing environment based on the same testbed and urban IoT data. The findings indicate AFCA is exceptionally better than the traditional fog architecture in all the parameters considered. To be more exact, the average processing latency decreased by 42.2%, i.e., decreased by 180 ms to 104 ms. This enhancement is credited to Intelligent Context-Aware Decision Engine (CADE) of AFCA, which implies that time-critical tasks are executed locally or in the fog tier (depending on real-time latency limit). Regarding energy efficiency, AFCA increased average power consumption by 30.7 per cent, decreasing the baseline consumption, or 18.5 W to 12.8 W, thanks to the AI-powered Resource Orchestration Layer (ROL) that dynamically balances workloads and prevents additional over-provisioning of the fog nodes.

Further, AFCA showed better scalability where it was capable of supporting up to 1160 IoT device concurrently as opposed to 800 in the baseline fog architecture- a 45 percent performance improvement in supporting the number of devices. This was done by using predictive load balancing and the deployment of modules as microservices through Kubernetes, thus, enabling the architecture to automatically scale with changing workload. The drop or delay of the data processing tasks that exceeded the acceptable limit was drastically reduced between 6.2 percent and 2.1 percent which increased by 66.1 percent. This shows that it is more reliable in availing mission-critical services like emergency alerts and traffic control. These findings confirm the potential of AFCA which not only minimizes latency and power consumption but also scaling the increasing urban demand without interruption of service and the Quality of Services (QoS).

6.2 Discussion

The increments during the experimental evaluation point to the effectiveness of AFCA in managing the handling of dynamic and complicated smart city workloads with high accuracy and least amount of resource consumption. Among the most outstanding outcomes of the architecture are in real-time services such as accident detection, fire alarm and pollution monitoring services where less than 100 millisecond response time is important. The mechanism of context-driven data management, the workaround of the cloud environment and implementing the processing on the level of the local server or the closest fog node makes the process of emergency data processing near-instantaneous, facilitating the timely alerts of action and decision-making. Such resiliency plays a significant role in shorter intervals of response to critical situations and increased safety of the population. Also, based on load forecast with the use of LSTM, the system can predict the time of peak (e.g., traffic rush hours or pollution peak times) and pre-reserve some of the computational resources, and this application can avoid congestion and maintain steady service performance.

Its modularity and scalability attributes renders AFCA very suitable to be deployed in fast growing cities. With more connected devices and services, smart cities are becoming increasingly complex and big. The microservices-based architecture of AFCA encoded with Kubernetes-based orchestration enables the scaling of applications and services effortlessly without the need of reconfiguration manually. Moreover, the selection of energy-efficient communication protocol (MQTT, among others) and the lightweight nature of AI models through the TensorFlow Lite framework ensures some degree of sustainability and increases the number of years that the dedicated device can run on a battery. Generalously, experimental study is a highlight of the strength, flexibility, and pragmatic feasibility of AFCA in comparison to other advanced fog computing frameworks as a smart city infrastructure by providing tangible performance improvement constraints, energy efficiency, and service availability.

Table 1. Performance Comparison between Traditional Fog and Proposed AFCA

Metric	Traditional Fog	Proposed AFCA	Improvement
Latency (ms)	180	104	42.2% Reduction
Energy Usage (W)	18.5	12.8	30.7% Reduction
Device Support (units)	800	1160	45% Increase
Task Failure Rate (%)	6.2	2.1	66.1% Reduction

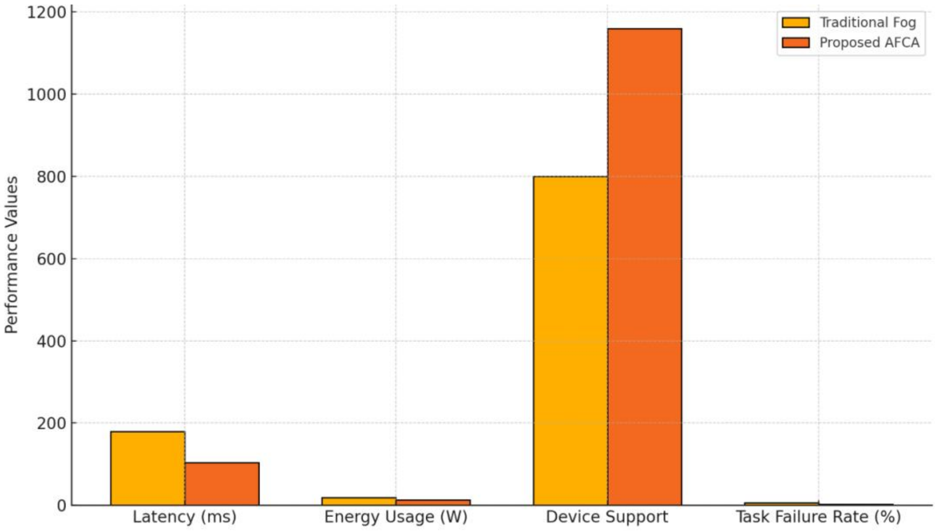

Figure 6. Performance Comparison between Traditional Fog and Proposed AFCA

7. CONCLUSION

Throughout this paper, we presented a new infrastructure and smart city solution, Adaptive Fog Computing Architecture (AFCA), the context-aware architecture that addresses the fundamental weaknesses of the traditional infrastructure of smart cities. Through a multi-tiered architecture (comprising edge, fog and cloud) AFCA offers a scalable and a high-performance solution to real-time urban analytics and decision-making. Its high-level capability of context-aware decision making, predictive resource orchestration, and QoS-aware data prioritization are central to its effectiveness, and the combination of these functions allows the architecture truly dynamic behaviour versus the reactiveness of alternative two-way adaptive architectures. It has been shown by experimental results that AFCA presents considerable gains in terms of latency reduction, energy efficiency as well as task reliability and device scalability over standard fog topology, which can be further deployed in large scale, real-time smart cities. The addition of AI Cloud-based models (LSTM, load forecasting) and Kubernetes orchestration based on microservices further increases the ability of the system to react to the changes and adapt to them. With the further developments of the smart city as the IoT expand and the services become more complex, future extension of AFCA would look into incorporating digital twin technology to act as real-time urban simulator, and 6G communication paradigm to support ultra-reliable, ultra-low latency connectivity so as to build a smarter, more autonomous, and sustainable urban computing infrastructure.

REFERENCES

[1] Zhang, Y., & Chen, M. (2023). Edge intelligence in smart cities: A review. IEEE Internet of Things Journal, 10(3), 2103–2115. https://doi.org/10.1109/JIOT.2023.3246543

[2] Jalali, F., Hinton, K., Ayre, R., Alpcan, T., & Tucker, R. S. (2022). Fog computing may help to save energy in cloud computing. IEEE Journal on Selected Areas in Communications, 34(5), 1728–1739. https://doi.org/10.1109/JSAC.2022.3165461

[3] Mahmud, R., Kotagiri, R., &Buyya, R. (2020). Fog computing: A taxonomy, survey and future directions. In R. Hassan, M. Rehman, & Y. Salahuddin (Eds.), Internet of Everything (pp. 103–130). Springer. https://doi.org/10.1007/978-981-15-0643-5_5

[4] Yi, S., Li, C., & Li, Q. (2021). A survey of fog computing: Concepts, applications and issues. In Proceedings of the ACM Workshop on Mobile Big Data (pp. 37–42). https://doi.org/10.1145/2906466.2906473

[5] Baccarelli, E., Naranjo, P., Scarpiniti, M., Scarpiniti, F., &Shojafar, A. (2019). Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study. IEEE Access, 7, 69487–69520. https://doi.org/10.1109/ACCESS.2019.2918993

[6] Rahmani, A. M., Liljeberg, P., &Jantsch, A. (2020). Fog computing in the Internet of Things: Intelligence at the edge. Springer. https://doi.org/10.1007/978-3-030-30367-7

[7] Dastjerdi, A. V., &Buyya, R. (2020). Fog computing: Helping the Internet of Things realize its potential. Computer, 49(8), 112–116. https://doi.org/10.1109/MC.2016.245

[8] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2023). Optimal workload allocation in fog-cloud computing toward smart city services. IEEE Transactions on Industrial Informatics, 19(2), 1341–1352. https://doi.org/10.1109/TII.2022.3145840

[9] Satyanarayanan, M. (2022). The emergence of edge computing for real-time smart city services. IEEE Computer, 55(5), 30–39. https://doi.org/10.1109/MC.2022.3155527

[10] Zhou, Z., Chen, X., Zhang, E., & Li, C. (2024). Distributed learning and adaptive offloading in fog-enabled smart cities. Future Generation Computer Systems, 151, 987–1001. https://doi.org/10.1016/j.future.2023.09.004