Smart City Digital Twin Edge-Core Deployment: A Case Study on Traffic and Air Quality Management

Manoj Herath, Hrishikesh Dutta, Roberto Minerva, Noel Crespi, Maira Alvi, Syed Mohsan Raza
Samovar, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France.
Email: {manoj.herath, hrishikesh.dutta, roberto.minerva, noel.crespi, maira.alvi, syed-mohsan_raza}@telecom-sudparis.eu

***Abstract*—The increasing demand for smart cities calls for advanced solutions to enhance urban sustainability. Digital twin technology offers transformative potential by synchronizing virtual and physical environments in real time. However, existing approaches struggle with scalability due to the reliance on numerous specialized prediction models for individual urban components, the lack of a unified framework for different use cases limits generalization, and high latencies in real-time synchronization. To address these, this paper presents a comprehensive software architecture for smart city DT and integrates correlation-aware model reduction and dynamic adaptive forecasting to support diverse urban applications to improve generalizability and scalability. This is done while adapting a smart distribution of DT software components between edge and core servers to ensure a low-latency performance. Validated using real-world traffic and air quality data, the system demonstrates significant improvements in traffic flow, emissions reduction, and public transportation efficiency, and enhances air quality monitoring, forecasting, and pollutant management. Key contributions include a scalable and generalizable DT architecture, AI-driven adaptability, edge-core deployment, and extensive validation through predictive analytics. This work establishes a replicable blueprint for metropolitan-scale DTs, balancing computational efficiency with responsive urban analytics.**

***Index Terms*—Digital Twin, Smart City, Edge Core Deployment, Artificial Intelligence**

I. Introduction

The rapid urbanization and increasing demand for sustainable cities have driven global interest in smart city solutions. Digital infrastructure & connectivity, effective urban planning, intelligent management of utilities and smart city resources, and real-time decision making are essential to ensure efficient city operations. Among emerging technologies, Digital Twin (DT) has gained significant attention due to its ability to create a virtual counterpart of a physical city, continuously updated—often in real time—with data reflecting its behavior, status, and characteristics [1]. By enabling monitoring, forecasting, and simulation, DTs empower cities to improve traffic management, optimize infrastructure, and enhance environmental sustainability.

Effective traffic and air quality management, a critical aspect of smart city initiatives, depends on precise and real-time monitoring and forecasting. These capabilities enable city planners to mitigate congestion, reduce vehicle emissions, improve air quality, and promote sustainable urban mobility. In this context, Digital Twin technology provides a powerful solution by facilitating data-driven modeling of traffic patterns and pollution levels, including vehicle density, emission rates, and air quality indices. However, while existing smart city DTs address traffic or environmental monitoring in isolation, they often lack an integrated, scalable framework for comprehensive pollution control and traffic optimization across diverse urban landscapes.

Existing smart city digital twins are typically designed with an application-specific focus [2], [3]. However, these approaches often lack a comprehensive software architecture that can generalize across different smart city applications. Additionally, existing solutions face scalability challenges, making it difficult to expand them for city-wide use cases with advanced functionalities. Moreover, current smart city digital twin architectures primarily rely on cloud-based deployments [4]–[6], failing to leverage the potential benefits of edge-core capabilities, such as reduced latency and lower carbon footprint.

This paper addresses these limitations by introducing a scalable and generalizable smart city digital twin architecture, incorporating edge-core computing for low-latency processing. Our approach focuses on real-time traffic and air quality management, validated through an IoT sensor network deployed in Madrid, demonstrating its effectiveness in managing large-scale heterogeneous smart city applications.

Specifically, we propose a comprehensive software architecture for smart city Digital Twins. Validated using data collected from Madrid's traffic and air quality monitoring stations, the architecture processes these data to update deep learning models dynamically, perform correlation analysis as needed, and generate predictions. Then the software components are deployed in a laboratory edge-core architecture. Through extensive experimentation, we evaluate the prediction performance of trained DL models on both traffic and air quality data, the effectiveness of the correlation analysis in identifying key traffic stations, and the suitability of the edge-core deployment strategy for smart city digital twin applications.

The key contributions of this paper are as follows.

1) A modular and scalable software architecture is introduced for Smart City Digital Twins. Such a modular architecture allows the DT framework to be adapted for heterogeneous environments, ensuring generalizability.
2) The different modules of the proposed DT framework are distributed across an edge-core deployment architecture in an in-house prototype hardware, to ensure performance while keeping latency and carbon footprint in check.

3) An AI-driven framework is incorporated into the software architecture that adjusts prediction models based on real-time urban dynamics and changing conditions.
4) The developed DT architecture also supports a mechanism for efficient management of massive time series data generated from a large-scale sensor network.
5) The feasibility of the proposed methodology is validated through a case study of real-time traffic and air quality monitoring using an IoT network deployed in the smart city of Madrid.

II. Related Work

In recent years, smart city Digital Twin implementations have gained global attention, with notable examples in Helsinki, Finland [4], Rennes, France [5], Berlin, Germany [6], and Florence, Italy [7] in Europe; Singapore [1] in Asia; and Victoria, Australia[2]. However, these solutions are typically designed for specific use cases, limiting their adaptability to broader smart city applications. Most of these DTs primarily focus on 3D virtual modeling for visualization rather than leveraging key digital twin functionalities, particularly real-time prediction, dynamic simulation, and prescriptive analytics. Moreover, these smart city DTs rely on centralized cloud-based architectures, leading to challenges in latency, computational efficiency, and energy consumption. Emerging research in DT has demonstrated significant advantages through edge-core computing paradigms in other domains [8]. This suggests that incorporating edge and core computing could substantially enhance both the operational quality and system responsiveness of smart city digital twins.

Prediction, as a core functionality of Digital Twins, enables proactive management of traffic and air quality. Since both domains rely on time-series data, Recurrent Neural Networks (RNN) and their variants—particularly Long Short-Term Memory (LSTM) networks have emerged as state-of-the-art solutions for forecasting. Although Gated Recurrent Units (GRU) are also employed, LSTM networks often outperform them in prediction performance [9]. Prior studies have validated LSTM's effectiveness in traffic prediction, such as combining traffic and noise pollution data for improved accuracy [10], and utilizing surrounding traffic flow data to enhance model performance [11].

Scaling LSTM-based prediction models for city-wide applications presents significant computational and resource challenges. While some approaches, such as traffic sensor redundancy reduction for Madrid [12], have aimed to address these issues, the proposed solution exhibits limited generalizability. Specifically, the model's performance may degrade when applied to different road networks or dynamic traffic conditions, restricting its broader applicability.

In contrast, existing solutions face several key limitations, including a lack of a unified architecture, application-specific designs, limited generalizability, scalability constraints, and underutilization of edge-core resources. To address these challenges, we propose a modular, scalable, and generalizable smart city digital twin architecture. Our approach integrates diverse smart city applications, addressing both scalability and generalizability while optimizing edge-core resource distribution.

[1]https://oecd-opsi.org/innovations/virtual-twin-singapore/

[2]https://www.land.vic.gov.au/maps-and-spatial/digital-twin-victoria

III. Smart City Digital Twin for Traffic and Air Quality Management

This section details the architectural structure of the smart city digital twin. This architecture extends our previous work [13], [14] by introducing a generic services module for seamless integration of new components, exploration of different smart city applications, and deployment in the edge-core environment. Figure 1 illustrates the proposed architecture, which comprises six subsystems: edge, data management, DT, event management, resource management, and system management.

A. Edge Subsystem

The edge subsystem connects IoT devices to the data management subsystem. The IoT sensors deployed across Madrid[3] gather real-time traffic data and air quality data with timestamps and geographical information. The city has 60 permanent traffic stations and 24 air quality monitoring stations, as shown in Figure 2. These stations record hourly data throughout the day. The edge subsystem comprises the IoT sensors, actuators, and enablers, such as Raspberry Pi and Jetson Orin Nano developer kits.

B. Data Management Subsystem

IoT sensor data from the edge subsystem undergoes processing, conversion, and storage in the data management subsystem. The preprocessor validates the time-series data to ensure sufficient information for conversion. The data converter transforms the data into the NGSI-LD format[4], using the "Traffic Flow Observed" and "Air Quality Observed" models from Smart Data Models[5]. These data models include attributes such as observed date and time, intensities, and location. The data injector transmits the NGSI-LD-formatted data to the context broker via its REST API, updating entities for each traffic and air quality monitoring station. For scalable, persistent storage and retrieval, we utilize Stellio[6], an open-source, NGSI-LD-compliant context broker built on linked-data principles, which facilitates a standardized API for CRUD operations.

C. Digital Twin Subsystem

The digital twin subsystem consists of three primary modules: generic services, model library, and API gateway. Generic Services provides core functionalities, including a correlation analyzer for identifying key traffic stations for updating DL models, a predictor for traffic and air quality forecasting, and a module for real-time visualization. Model Library serves as a repository for both data models and DL models. It stores the

[3]https://datos.madrid.es/portal/site/egob/

[4]https://www.etsi.org/committee/cim

[5]https://smartdatamodels.org/

[6]https://stellio.readthedocs.io/en/latest/

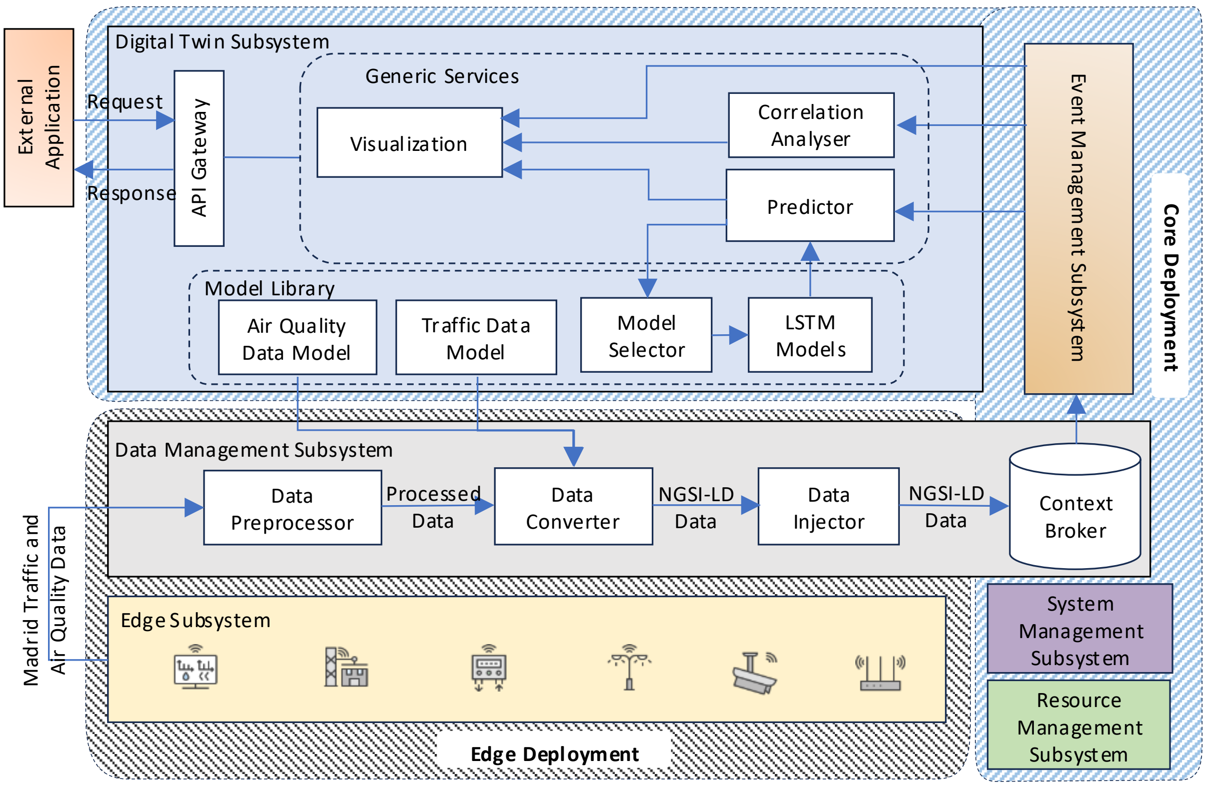

Fig. 1: Overview of the System Architecture and Deployment Strategy

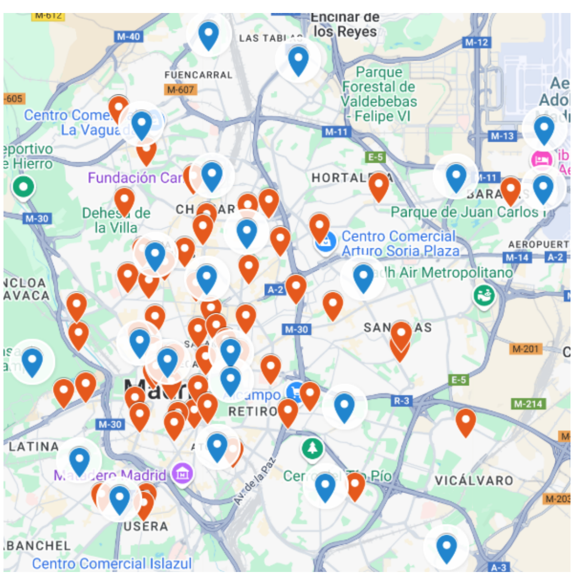

Fig. 2: Distribution of 60 Permanent Traffic Stations (red) and 24 Air Quality Monitoring Stations (blue) in Madrid.

LSTM models, with the model selector choosing the appropriate model for prediction. API Gateway enables external users and applications to access the DT subsystem's generic services. We implemented this interface using FastAPI[7], an open-source framework that provides efficient REST API capabilities.

Correlation Analysis: Correlation analysis determines the similarity between two datasets. Figure 3 shows that most traffic stations have a correlation coefficient exceeding 0.75, indicating a high similarity of their data. The analysis identifies key stations that can be used in prediction for other traffic stations. Stations are selected for predefined correlation ranges $1 > r > 0.95, 1 > r > 0.90, 1 > r > 0.85, 1 > r > 0.80,$ and $1 > r > 0.75$. The selection process prioritizes the station with the highest number of correlated stations first, followed by the station with the next highest number of unique correlated stations (excluding already covered stations). This repeats until all traffic stations are covered.

Traffic and Air Quality Prediction: LSTM models are trained using the data from traffic stations identified through correlation analysis. Correlation analysis is not performed for air quality monitoring stations due to their smaller number and variations in measured pollutants, making it difficult to use one station's data to predict another.

D. *Event Managment Subsystem*

This subsystem accesses stored data through the context broker's REST API and handles all data exchange requirements for different processes. It retrieves data from the context broker for correlation analysis, LSTM model training, real-time traffic and air quality monitoring, and prediction.

[7]https://fastapi.tiangolo.com/

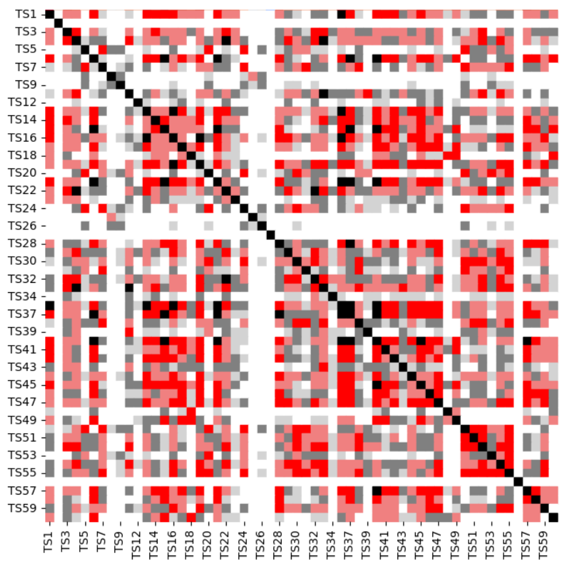

Fig. 3: Correlation Matrix for Jan to Mar 2024 Traffic Data: Light grey (0.75-0.8), grey (0.8-0.85), light red (0.85-0.9), red (0.9-0.95), black (>0.95).

E. System Management Subsystem

The system management subsystem ensures overall security through access control mechanisms. Keycloak[8], an authentication and authorization manager, secures context broker access by enforcing security rules and policies.

F. Resource Management Subsystem

The resource management subsystem manages the distribution of computing resources. Containerized software components are orchestrated using K3S[9]. Further deployment setup and details are presented in Section IV.

The proposed system provides a holistic framework and a modular DT architecture that enhances scalability, generalizability, and reusablity. Compared to existing architectures, it streamlines development and places a stronger emphasis on DT modeling, making it well-suited for smart city DTs.

IV. Edge Core Deployment Setup

The implemented architecture follows a two-tier edge-core deployment model, as illustrated in our laboratory setup, Figure 4. The edge layer comprises: five Raspberry Pi 4 Model B devices (Cortex-A72, 1.8GHz, 8GB RAM), four Jetson Orin Nano boards (Cortex-A78AE, 2.2GHz, 8GB RAM), and an edge server (Intel Core i3-3220, 3.3GHz, 8GB RAM). The core consists of a single server (Intel Xeon W-2245, 3.9GHz, 128GB RAM). As illustrated in Figure 1, the edge subsystem and the data management subsystem modules: the data preprocessor, data converter, and data injector, are deployed at the edge, while the remaining subsystems and the context broker operate at the core. Deploying the data management modules at the edge enables data preprocessing and conversion to occur closer to the source, which reduces the volume of data transmitted to the core. Additionally, keeping these components at the edge allows for real-time processing, thereby reducing latency. Centralizing the context broker and the remaining subsystems in the core simplifies overall system management and resource allocation.

[8]https://stellio.readthedocs.io/en/latest/admin/keycloak_integration.html
[9]https://k3s.io/

TABLE I: Specifications of the Learning Models and Hyperparameters

Parameters	Traffic Prediction	Air Quality Prediction
layer 1 units	100	150
layer 2 units	50	100
dropout rate	0.02	0.03
learning rate	0.01	0.01
time steps	24	24
epochs	100	100
batch size	32	32
Activation Function	ReLU	ReLU
Optimizer	Adam	Adam
Loss Function	MSE	MSE

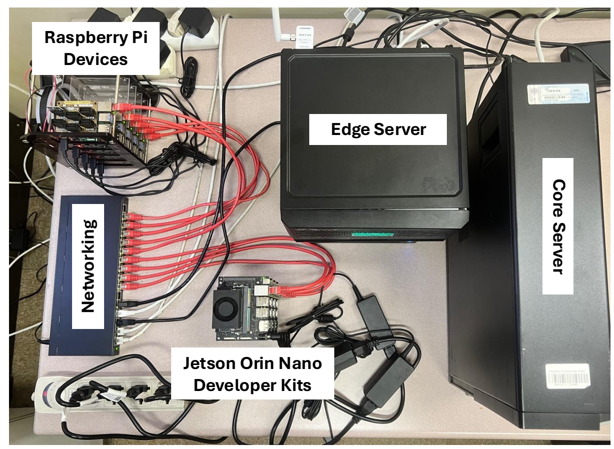

Fig. 4: Laboratory Setup for Edge and Core Deployment

V. Experiments and Results

The smart city digital twin implementation was validated using traffic and air quality data from Madrid[10], collected between January and April 2024. Table I presents the complete deep learning model configurations, including LSTM architectures and optimized hyperparameters that were systematically tuned to maximize predictive performance.

As explained in section III, the DT subsystem hosts all learning models. The IoT sensors in the edge subsystem monitor traffic and air quality, transmitting raw data to the data management subsystem for preprocessing, conversion, and storage. This processed data are then used for real-time visualization, correlation analysis, DL model training, and real-time prediction in the DT subsystem. The system management subsystem handles authentication and authorization, while the event management subsystem provides seamless data flow.

[10]https://datos.madrid.es/portal/site/egob/

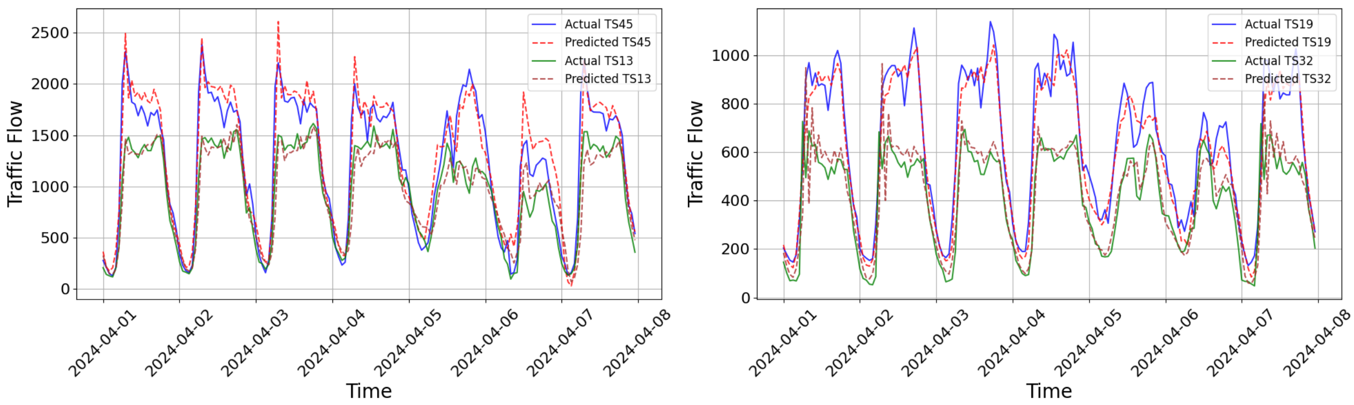

Fig. 5: Actual and Predicted Traffic Data for Different Traffic Stations

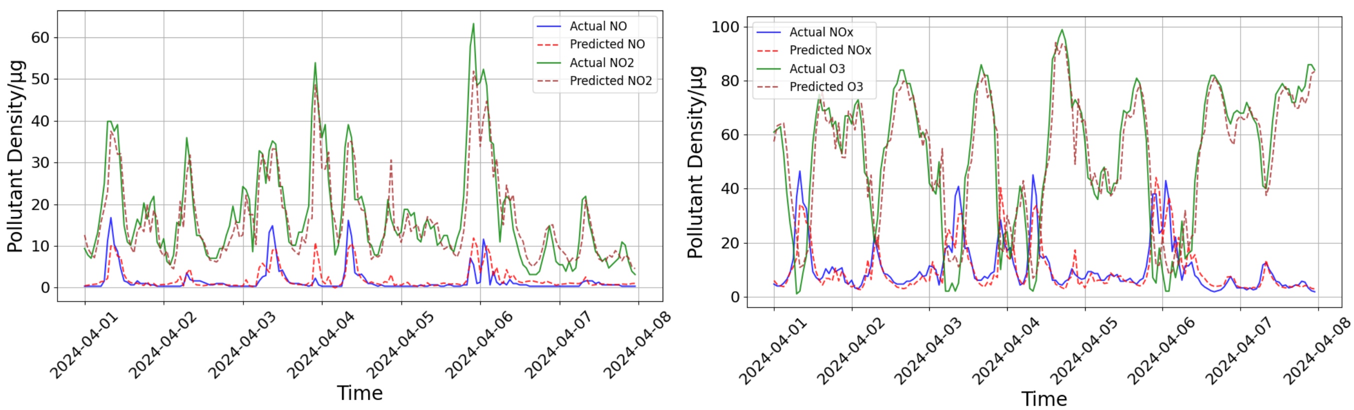

Fig. 6: Actual and Predicted Air Quality Data for 28079016 Air Monitoring Station Different Pollutants

The experiments evaluate the prediction accuracy for traffic and air quality data; the correlation analyser's ability to select the right sensors for prediction; and the effectiveness of edge-core deployment. To achieve this, we conducted comprehensive testing using LSTM models trained on traffic station data selected through the correlation analyzer as well as on the pollutant concentration readings from air quality monitoring stations, and measured latency and energy consumption metrics in edge-core deployment.

Correlation analysis revealed a systematic reduction in the number of required LSTM models as the correlation threshold was broadened. Specifically, when expanding the correlation range from $1 > r > 0.95$, the number of required models decreased to 47, representing a 19% reduction. Further increasing the range to reduced the model count to 18 (69% reduction), followed by $1 > r > 0.85$ with 12 models (79% reduction), $1 > r > 0.80$ with 7 models (88% reduction), and finally, $1 > r > 0.75$, which required only 6 models (90% reduction). This reduction occurs because broader correlation thresholds encompass more traffic stations exhibiting statistically similar patterns, thereby minimizing redundancy in model deployment.

Figure 5 illustrates the comparison between the predicted and actual traffic intensity for the correlation range $1 > r > 0.95$. The results indicate that for all traffic stations, the predicted traffic flow closely aligns with the observed data. This strong agreement is attributed to the high correlation between traffic stations used for LSTM model training. Specifically, traffic station TS13 shows a strong correlation with TS36, TS45 with TS40, TS19 with TS16, and TS32 with TS22, demonstrating the effectiveness of leveraging correlated stations for predictive modeling.

Table II presents the prediction errors, quantified using Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), for traffic stations within the correlation range $1 > r > 0.95$. The observed trend of improved predictive performance across all traffic stations is further validated by these error metrics, reinforcing the efficacy of the correlation-based model reduction approach.

For the case of air quality management, Figure 6 compares the predicted and true intensities of the pollutants NO, NO2, NOx, and O3 at the air quality monitoring station 28079016. The results demonstrate that the predicted values are closer to the observed data across all pollutants. This is because the fact that the temporal dependency-capturing capability of LSTM networks effectively models pollutant dispersion patterns.

Table III presents the latency measurements for traffic data

TABLE II: Prediction Performance based on MAE, MSE, and RMSE for Traffic Data

Prediction Model	Station	MAE	MSE	RMSE
TS40	TS6	0.057	0.007	0.083
	TS15	0.052	0.006	0.077
	TS21	0.077	0.011	0.104
	TS45	0.062	0.007	0.082
	TS57	0.069	0.008	0.091
TS36	TS1	0.081	0.010	0.101
	TS13	0.070	0.009	0.093
T37	TS28	0.079	0.010	0.098
TS16	TS19	0.049	0.004	0.065
TS22	TS32	0.077	0.016	0.126
TS33	TS11	0.057	0.009	0.095

TABLE III: Latency Measurements of System Operations

Station	Preprocess Time (ms)	Conversion Time (ms)	Injection Time (ms)	Retrieval Time (ms)	Total Time (ms)
TS1	1.36	0.22	20.68	0.28	22.54
TS6	1.38	0.22	18.07	0.33	20.00
TS11	1.38	0.22	18.01	0.45	20.06
TS13	1.38	0.23	17.18	0.37	19.16
TS15	1.37	0.22	17.41	0.36	19.36
TS19	1.24	0.16	18.40	0.31	20.11
TS28	1.08	0.18	18.82	0.29	20.37
TS32	1.24	0.16	17.65	0.33	19.38
TS45	1.26	0.16	17.90	0.31	19.63

preprocessing, data conversion, context broker injection, and retrieval operations, along with their total time. This total latency represents the time required for edge data to reach the digital twin subsystem. Once received, the data can be utilized for real-time visualization, correlation analysis, and prediction. Among these operations, data injection exhibits the highest values for all devices. This is due to the additional network delay introduced during edge-to-core transmission. The maximum observed total latency was 22.54 ms, resulting in an end-to-end latency (from data capture to visualization/prediction) of slightly over 20 ms. The reason for low latency is due to local data preprocessing, conversion, and injection at the edge devices eliminates cloud-bound processing delays, and the physical proximity between edge devices and the core server in the deployment minimizes network transmission latency. The average energy consumption was measured at 105 mJ (70 mJ average for Raspberry Pis and 147 mJ average for Jetsons).

VI. Conclusions

In this paper, we propose a modular and scalable software architecture for smart city digital twins. Our framework addresses key limitations of existing implementations, including application specificity, lack of generalizability, scalability issues, and absence of edge-core deployment capabilities. The architecture has been validated through two use cases—traffic management and air quality management—for the city of Madrid, using real-world data collected from an IoT sensor testbed. We present a comprehensive system characterization by evaluating the architecture across multiple use cases, assessing various predictive deep learning models, and testing edge-core deployment in a laboratory environment. The results demonstrate the generalizability of both the software architecture and the DL models, providing scalable solutions for diverse smart city digital twin applications. Results demonstrate that the traffic and air quality prediction can be performed with high accuracy, while having a maximum latency of 22 ms and average energy consumption of 105 mJ. Future work will focus on integrating additional smart city applications, incorporating data from varied sensing modalities, and refining model generalization to enhance further the broad applicability of digital twins in smart city environments.

Acknowledgment

This research work is supported by Project CLOUD CONTINUUM SOUVERAIN ET JUMEAUX NUMRIQUES under Grant AMI CLOUD-1 C2JN (DOS0179613/00, DOS0179612/00).

References

[1] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020.

[2] X. He, Q. Ai, J. Wang, F. Tao, B. Pan, R. Qiu, and B. Yang, "Situation awareness of energy internet of things in smart city based on digital twin: From digitization to informatization," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7439–7458, 2022.

[3] M. Jafari, A. Kavousi-Fard, T. Chen, and M. Karimi, "A review on digital twin technology in smart grid, transportation system and smart city: Challenges and future," *IEEE Access*, vol. 11, pp. 17 471–17 484, 2023.

[4] Helsinki digital twin. [Online]. Available: https://kartta.hel.fi/?setlanguage=en\#,andhttps://kartta.hel.fi/3d/\#/

[5] Rennes en 3d: Rennes 2030 la vill à votre idée. [Online]. Available: https://rennes2030.fr/rennes-en-3d/

[6] 3d city db. [Online]. Available: https://www.3dcitydb.org/3dcitydb/visualizationberlin/

[7] L. Adreani, P. Bellini, M. Fanfani, P. Nesi, and G. Pantaleo, "Smart city digital twin framework for real-time multi-data integration and wide public distribution," *IEEE Access*, 2024.

[8] F. P. Knebel, R. Trevisan, G. S. do Nascimento, M. Abel, and J. A. Wickboldt, "A study on cloud and edge computing for the implementation of digital twins in the oil & gas industries," *Computers & Industrial Engineering*, vol. 182, p. 109363, 2023.

[9] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth academic annual conference of Chinese association of automation (YAC)*. IEEE, 2016, pp. 324–328.

[10] F. M. Awan, R. Minerva, and N. Crespi, "Using noise pollution data for traffic prediction in smart cities: experiments based on lstm recurrent neural networks," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20 722–20 729, 2021.

[11] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuhera, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[12] L. Ding, P. Rajapaksha, R. Minerva, and N. Crespi, "Deep learning for reducing redundancy in madrid's traffic sensor network," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*. IEEE, 2024, pp. 1–9.

[13] M. Herath, M. Alvi, R. Minerva, H. Dutta, N. Crespi, and S. M. Raza, "Smart city digital twins: A modular and adaptive architecture for real-time data-driven urban management," in *20th International Conference on Network and Service Management*, 2024.

[14] M. Herath, H. Dutta, R. Minerva, N. Crespi, M. Alvi, and S. M. Raza, "An ai-driven, scalable, and modular digital twin framework for traffic management," in *IEEE Wireless Communications and Networking Conference*, 2025.