An AI-driven, Scalable, and Modular Digital Twin Framework for Traffic 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 growing need for intelligent tools to support urban planning and resource management has positioned Digital Twin (DT) technology as a cornerstone of smart city development. DTs, as dynamic virtual replicas of physical systems, offer capabilities that extend beyond mere representation, enabling monitoring, diagnostics, forecasting, and optimization. In the context of urban traffic management, DTs provide a robust solution for real-time traffic monitoring and predictive analytics. However, existing approaches often lack a systematic design methodology, leading to challenges in scalability and adaptability, particularly in heterogeneous environments. This paper presents a novel methodology for developing scalable and adaptive smart city DT architectures, with a focus on real-time traffic management. A modular and unified software framework is proposed, leveraging AI-driven approaches to address the complexity of managing diverse traffic data sources. A sequential learning model is integrated into the architecture to enhance the DT's adaptability to evolving traffic conditions and congestion patterns. The proposed framework is validated using real-world traffic data from an IoT network deployed in Madrid, demonstrating its scalability and low-latency performance. Experimental results highlight the effectiveness of the framework in handling heterogeneous traffic scenarios and its ability to deliver accurate predictions while minimizing resource overhead.**

***Index Terms*—Digital Twin, Smart City, Artificial Intelligence**

I. Introduction

The growing global trend towards smart cities highlights the crucial need for advanced tools and services to facilitate urban planning, optimize resource management, and support decision-making. In this context, Digital Twin (DT) is one of the imperative technologies with promising capabilities and a wide range of applications [1]. DT is a virtual representation of a physical system that reflects its behavior, status, and properties. This software counterpart of the physical system is continuously updated, often in real time, with relevant data characterizing the physical system in its operating environment [2]. DT's features and abilities go beyond the mere representation of the smart city by offering monitoring, diagnostics, forecasting, predicting operational problems, simulation, and optimization [3].

Effective traffic management, a cornerstone of smart city initiatives, relies heavily on accurate and timely predictions. Such capabilities empower urban planners to optimize travel routes, improve traffic flow, reduce operational costs, and enhance overall system efficiency. In this context, DT technology offers a robust solution by enabling data-driven modeling of urban road networks, including traffic intensity, speed, and flow. However, while several approaches to smart city DTs exist, they are often narrowly focused on specific domains such as traffic or energy systems and lack a holistic, scalable architecture for practical deployment in diverse urban environments.

There are existing works on smart city digital twins, with many approaches considered, each focusing on different applications such as traffic, energy, etc. [4], [5] However, these approaches do not propose a holistic sofware architecture or a systematic methodology necessary to design and deploy a smart city digital twin. As a result, it is challenging for developers to apply a standard method to achieve their objectives. In addition, due to the lack of generalizability, these data-driven DT design approaches often face the challenges of scalability in heterogeneous environments. For instance, in the traffic management scenarios, dedicated DT modeling, with separate training procedures for each of them, needs to be done for traffic monitoring in each road within the city. These architectures require unique AI models for traffic predictions trained for each traffic flow sensor station/road. Considering the smart city use case of Madrid, there are more than 7,000 strategically positioned vehicle detectors across the city, operating at over 4,000 measurement points [6]. In such a scenario, developing unique DT software and training separate AI models for each station is impractical, as it is both time- and cost-inefficient. Additionally, creating a large model capable of predicting traffic on any road would negatively impact the performance of a digital twin, which needs to operate in real time with minimal latency.

This paper addresses these challenges by introducing a systematic and scalable methodology for designing smart city DTs, with a focus on real-time traffic management. The proposed framework employs a modular software architecture, incorporating an AI-driven sequential learning model to adapt dynamically to changing traffic conditions and ensure scalability across heterogeneous environments. The methodology is validated using real-world traffic data collected via an IoT network deployed in Madrid, demonstrating its effectiveness in managing large-scale urban traffic systems.

Specific contributions of this paper are as follows.

1) A modular software architecture for smart city Digital Twin is proposed with the key focus on traffic management.
2) An AI-driven approach is adopted for making the DT software architecture scalable, thus making it suitable for heterogeneous traffic scenarios with a large number of

sensors.

3) A sequential learning model is developed and integrated with the proposed software architecture to make the DT informed and adaptive to future traffic congestion.
4) The viability of the proposed methodology is demonstrated through a case study of real-time traffic monitoring using an IoT network deployed in the smart city of Madrid.
5) A detailed characterization of the developed system is presented by means of extensive experiments performed using a wide range of predictive learning algorithms.

The remainder of the paper is structured as follows. In Section II, we introduce the related work, Section III describes the smart city digital twin architecture, and Section IV illustrates the experiments carried out and results. Finally, we describe the conclusion and future work in Section V.

II. Related Work

There have been many smart city digital twin implementations in recent years such as Helsinki, Finland [7]; Rennes, France [8]; Berlin, Germany [9]; Zurich, Switzerland [10]; Singapore [11]; and Florence, Italy [12]. However, Most of these are tailored to specific use cases and greatly focus on virtual 3D representations while insufficiently addressing some of the important functionalities in DT such as analytics, prediction and what if analysis. Moreover they are lacking of a systematic approach and modular framework where they have limited possibility to deploy on different scenarios.

In smart city use cases, traffic prediction has received significant attention, and Deep Learning (DL) methods have gained popularity over traditional statistical and machine learning methods. Recurrent Neural Networks (RNNs) in [13], Long Short Term Memory (LSTM) and Gated Recurrent Units (GRUs) in [14], and graph convolutional networks in [15] are proposed for traffic flow prediction. To improve the accuracy of DL models, attempts have been made to include more information from the surroundings. In [16], LSTM is utilized for traffic prediction by using both traffic and noise pollution data. Similarly, in [17], the authors propose a traffic flow prediction method using LSTM, incorporating surrounding traffic flow data. However, the DL models enhance prediction accuracy, these models are not easily generalizable, requiring the training of a large number of models to predict city-wide traffic flow, which can be a significant challenge for smart city DTs. Therefore, such solutions are not ideal for smart city DTs which are large-scale solutions. This issue is partially addressed in [18], where the authors introduce a traffic sensor redundancy reduction method and use an LSTM model to predict traffic flow in Madrid. This method attempts to tackle the problem by reducing redundant sensor data. However, the model's adaptability to other roads remains limited, potentially leading to inaccurate predictions.

In contrast, the existing solutions described above face several limitations, including a lack of a systematic approach and modular architecture, insufficient generalizability for different use cases, and the limited scalability of DL solutions for traffic prediction. To address these challenges, we propose a systematic methodology for implementing smart city DTs. This methodology features a modular architecture that can be applied to various smart city use cases while enhancing scalability for smart city DTs.

III. Smart City Digital Twin Architecture

In this section, we describe the proposed software architecture for smart city DT. This architecture is an extension of our previous work [19] by introducing a correlation analyzer that is capable of calculating the correlation coefficient with the aim of reducing the number of prediction models, and by adding a model library that serves as a repository for prediction models so the prediction can be more effecient. As shown in Figure 1, the proposed architecture consists of six subsystems: edge, data management, digital twin, event management, resource management, and system management, each with its constituent components. An architectural view of the proposal and the components involved in the different segments for smart city traffic management is detailed below.

A. Edge Subsystem

The edge subsystem connects the IoT data sources to the data management subsystem, enabling seamless data acquisition. The IoT sensors located across the city that gather traffic data with timestamps and geographical data are used for data collection at the edge subsystem. Traffic data were collected from the open data platform of the city of Madrid [6]. Madrid has 60 permanent traffic stations distributed all over the city as shown in Figure 2. These stations record the number of vehicles per hour throughout the day. The dataset includes hourly traffic data for January and February 2024.

B. Data Management Subsystem

The IoT sensor data from the edge subsystem are processed, converted, and stored in the data management subsystem. The data management subsystem contains software components for each of these tasks. The preprocessor checks the time-series data and confirms that the data contains sufficient information for conversion. It resolves common issues in time-series data, such as irregular timestamps. Before sending to the data converter, it scales the data using equation 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

The data converter transforms this scaled data into the NGSI-LD [20] format. The data model used by the converter is defined by the digital twin subsystem. As the descriptive model that can accurately represent time-series data, a transportation data model called "Traffic Flow Observed" from the Smart Data Models [21], which adheres to NGSI-LD principles, is used. This data model includes attributes such as date observed, intensity, location, and address, which effectively represent date and time information, traffic intensity, and geographical

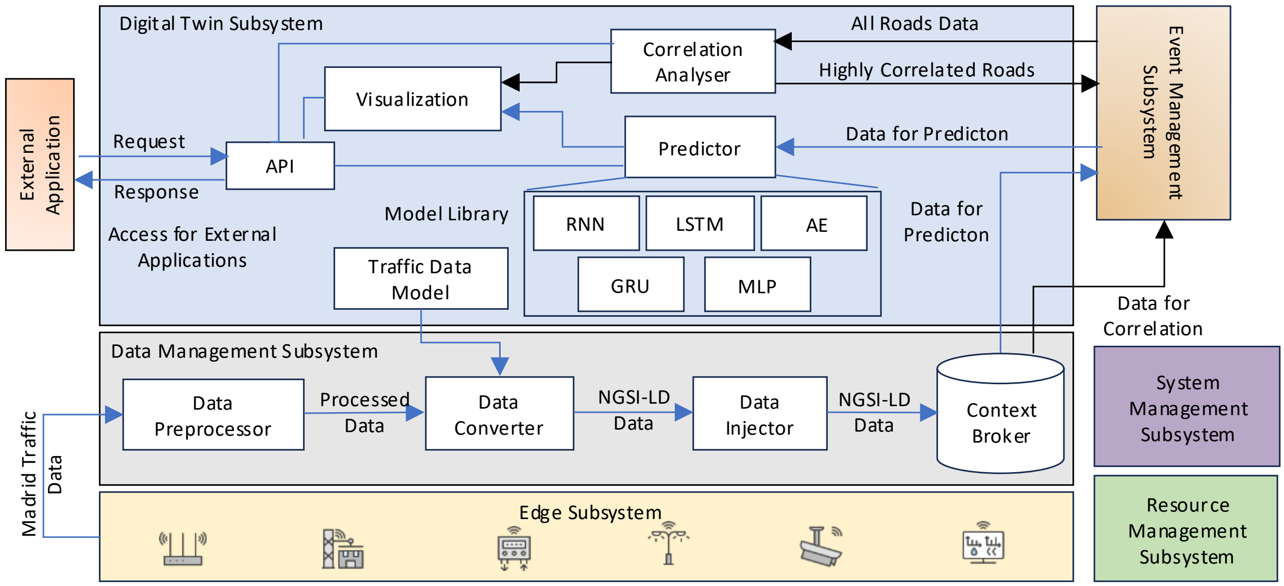

Fig. 1: Overview of the System Architecture

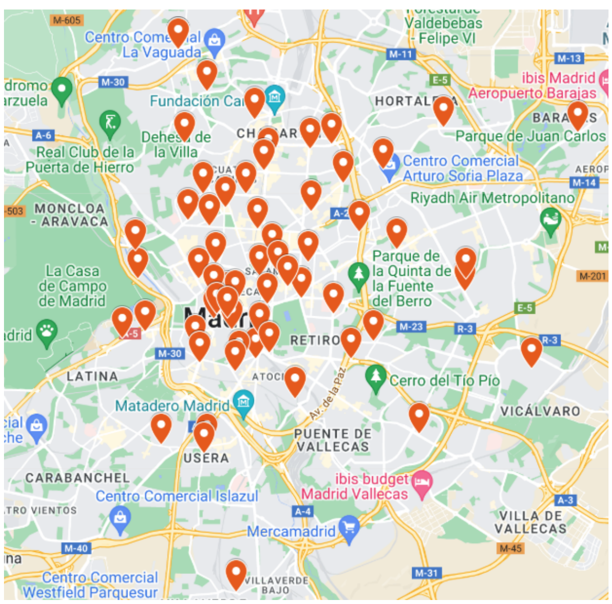

Fig. 2: Distribution of Permenant Traffic Stations in Madrid.

location in a smart city. This step ensures that the data conforms to the structure required for injection into the context broker.

The data injector injects the NGSI-LD data received from the data converter into the context broker. The injector updates the entities created in the context broker for each traffic station using it's REST API. Persistent storage and retrieval require a scalable and generalized mechanism. To achieve this, Stellio [22], an open-source context broker, is used. Stellio operates on linked-data principles and adheres to the NGSI-LD standard. It offers a standardized API that supports data retrieval, entity querying, and subscription mechanisms, ensuring compatibility with various data models across different domains in smart cities.

C. Digital Twin Subsystem

The digital twin subsystem is composed of modules for modeling, analytics, visualization, and access to external users and applications. It uses the data in the context broker for correlation analysis, training the DL models, and real-time prediction.

1) Correlation Analysis: Correlation measures the similarity between two data sets. Correlation anlyser module calculates the correlation and identifies highly correlated traffic stations, which are then used for prediction with AI models trained on one of the traffic stations. This approach aims to reduce the need for training separate AI models for each traffic station, thus minimizing both time and cost. Figure 3 shows the highly correlated traffic stations. Traffic stations TS13, TS21, TS36, and TS45 have a correlation value exceeding 0.95 with TS1, while TS3, TS6, TS14, TS15, TS16, TS17, TS19, TS22, TS37, TS40, TS41, TS42, and TS57 have correlation values ranging from 0.9 to 0.95. The traffic stations with the highest correlation to TS1 are used to test the DL models trained on TS1.

2) Traffic Prediction: Traffic Station TS1 data for January 2024 were used to train five well-known DL models: RNN, LSTM, Multilayer Perceptron (MLP), GRU, and Autoencoder (AE). These models were then tested on the traffic stations identified through the correlation analysis. For all models, the hyperparameters were tuned to enhance prediction accuracy. The details of the DL models, including their hyperparameters and architectural specifications, are presented in Table I. The four stations with the highest correlation to TS1, exceeding a correlation value of 0.95—TS13, TS21, TS36, and TS45—were used to test the DL models trained on TS1. For these stations, the DL models forecast the traffic flow from 2 February to 9 February 2024.

Fig. 3: Correlation Matrix results for Jan 2024. Red markers represent correlations ranging from 0.9 to 0.95 and black markers indicate correlations exceeding 0.95.

TABLE I: Specifications of the Learning Models and Hyper-parameters

Model	RNN	LSTM	MLP	GRU	AE
Arch.	1 RNN 1 Dense	2 LSTM layers	3 Dense layers 1 Dense output	2 GRU layers 1 Dense output	1 Dense encoding 1 Dense output
Units in Layers	64 -	64 each -	128,64,32 -	64 each -	16 -
Activation Functions	ReLU	ReLU	ReLU	ReLU	ReLU Sigmoid
Epochs	20	20	20	20	20
Batch size	32	32	32	32	32
Optimizer	Adam	Adam	Adam	Adam	Adam
Loss Function	MSE	MSE	MSE	MSE	MSE

3) Visualization and API: The visualization module displays the prediction and correlation results, while the API component enables external applications and users to access the digital twin models for visualization, correlation, and prediction purposes. We adopted the FastAPI [23] library to provide a REST interface. For example, through this API, an external application can request traffic predictions.

D. Event Managment Subsystem

A mechanism is required for data exchange and control between subsystems and modules. The event management subsystem ensures the smooth flow of data for processes such as correlation analysis and real-time prediction. It extracts data from the context broker for correlation analysis and, based on the highly correlated roads, ensures that the correct real-time traffic station data are sent for prediction. In this case, data from traffic stations TS13, TS21, TS36, and TS45 are sent for prediction.

E. System Management Subsystem

The system management subsystem ensures overall security for the system and emphasizes the importance of addressing various security requirements. Access to the data stored in the data management subsystem is controlled through appropriate mechanisms. In the architecture, Keycloak [24], an authentication and authorization manager, is employed to secure access to the context broker by enforcing basic rules and functions.

F. Resource Management Subsystem

Apart from the above subsystems, we use a resource management subsystem to manage the distribution of computing resources. Container images of the software components are utilized, with the context broker and DT subsystem modules, including the correlation analyzer and predictor, running on the core servers, while other software components operate on Jetsons and Raspberry Pis for operational simplicity. This setup is orchestrated using K3S [25].

In comparison with existing DT implementations, the proposed implementation provides a holistic framework and a systematic approach. Most of the components are reusable, significantly saving implementation time. This modular architecture places a greater focus on DT modeling. Moreover, this architecture promotes generalizability and scalability. Since our DT implementation offers a comprehensive framework and a clear perspective on the constituent components, it supports scalability and generalizability, making it suitable for the design and implementation of smart city DTs.

IV. Experiments and Results

The smart city digital twin has been evaluated for traffic data collected for the city of Madrid [6]. The data collection spanned from January to February 2024. The details of the DL models used, including the hyperparameters and architectural specifications, are tabulated in Table I.

The learning models for traffic prediction are implemented in the DT subsystem. The data collected using the IoT traffic sensors and devices in the edge are processed and stored in the data management subsystem, which is used for training of the learning models, correlation analysis, and real-time prediction at the DT subsystem. The authentication and authorization are handled by the system management subsystem, and the event calls for prediction and correlation by the modules in the DT subsystem are handled using the event management subsystem.

The experiments primarily focused on analyzing the performance of the prediction models on traffic data. To that end, experiments were conducted for different DL models trained on the traffic data of traffic station TS1, at which the data was collected by the other traffic stations. The comparison of the predicted traffic intensity and the true traffic intensity is demonstrated in Figure 4. The general observation here is that,

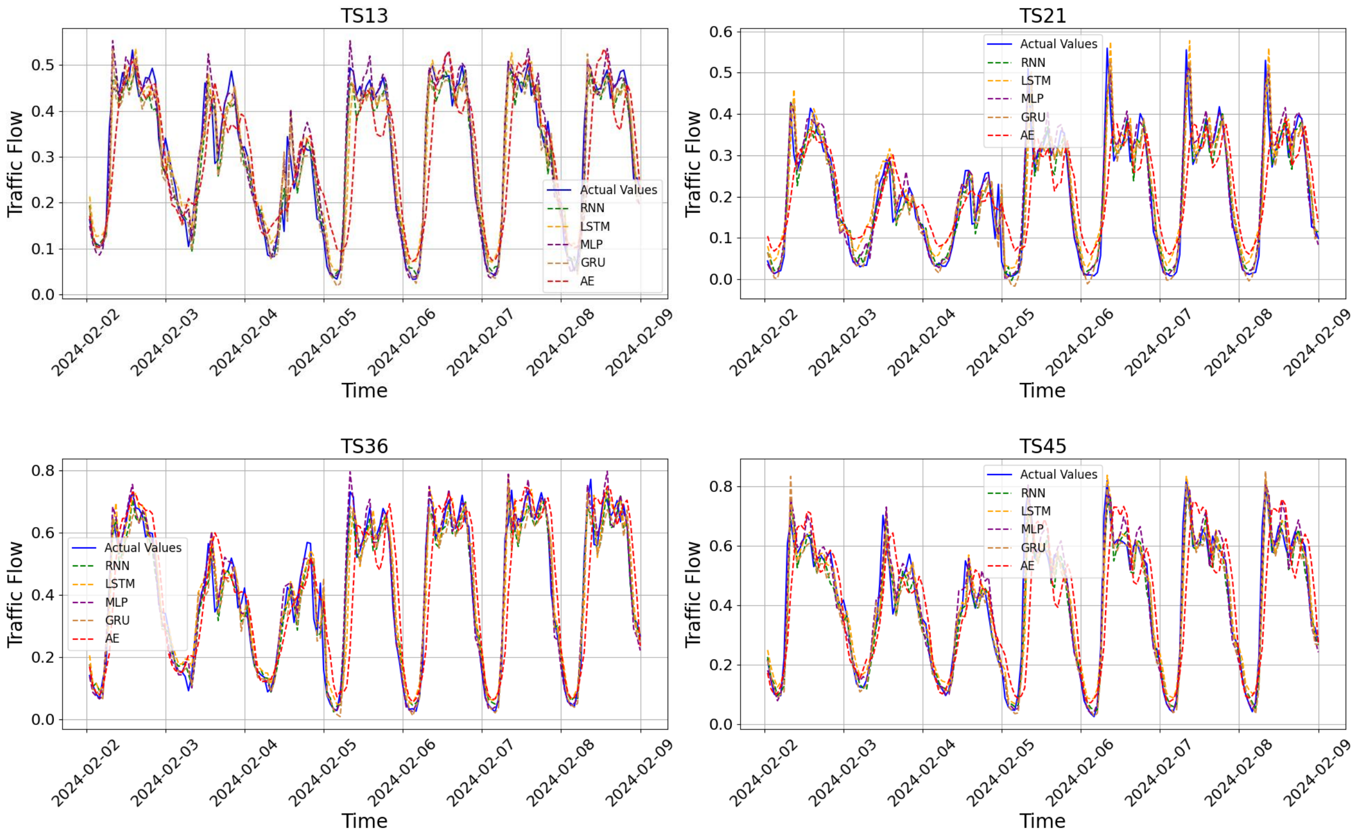

Fig. 4: Actual and Predicted Traffic Data for Different Prediction Models.

for all DL models, the predicted traffic flow is closer to the ground truth. This is because of the fact that for all four traffic stations, the data have a high correlation with traffic data from traffic station TS1. The prediction errors of the DL models, in terms of MAE, MSE, and RMSE, for different stations are reported in Table II. The observation reported above, that is, better performance for all the models, can also be visualized here. However, compared to other models a notably lower performance for AE is observed, which was not clearly depicted in Figure 4. Specifically, the following points can be noted: (1) The MLP model is consistently superior to the others in terms of prediction error for all stations. This can be because the traffic flow data exhibit relatively simple and stable temporal patterns over the period, and MLPs are efficient at capturing such patterns. (2) RNN and LSTM show less performance than MLP. This can be due to RNN and LSTM models requiring large datasets to exploit their sequential modeling capabilities effectively. With limited data, they may fail to leverage their strengths and end up underperforming compared to models like MLP. (3) GRU performs better than LSTM. This can be due to GRUs having fewer gates compared to LSTMs, making them computationally more efficient and less prone to overfitting on smaller datasets. (4) AE performs least. This can be due to they do not inherently model temporal dependencies, leading to weaker performance compared to models explicitly designed for sequential data.

TABLE II: Prediction Performance based on MAE, MSE, and RMSE for different Deep Learning Models

Deep Learning Model	Traffic Station	MAE	MSE	RMSE
RNN	TS13	0.030	0.002	0.041
	TS21	0.028	0.002	0.040
	TS36	0.039	0.003	0.051
	TS45	0.040	0.004	0.061
LSTM	TS13	0.035	0.002	0.047
	TS21	0.046	0.004	0.062
	TS36	0.052	0.005	0.070
	TS45	0.046	0.005	0.068
MLP	TS13	0.028	0.002	0.041
	TS21	0.026	0.001	0.038
	TS36	0.033	0.002	0.048
	TS45	0.038	0.003	0.053
GRU	TS13	0.032	0.002	0.044
	TS21	0.036	0.002	0.050
	TS36	0.047	0.004	0.064
	TS45	0.038	0.003	0.054
AE	TS13	0.061	0.007	0.083
	TS21	0.068	0.008	0.088
	TS36	0.086	0.015	0.123
	TS45	0.094	0.018	0.133

The R-squared scores (R^2) of the DL models are reported in Table III. The observation reported above, that is, better performance for MLP followed by RNN, GRU, and LSTM and the worst performance for AE, can also be visualized here. The

MLP achieved the highest average R^2 score of 0.939, superior to the others: 0.930, 0.880, 0.914, and 0.665 for RNN, LSTM, GRU, and AE, respectively.

TABLE III: Prediction Performance based on R^2 score for different Deep Learning Models

Traffic Station	R^2 value				
	RNN	LSTM	MLP	GRU	AE
TS13	0.929	0.906	0.930	0.919	0.710
TS21	0.919	0.806	0.928	0.877	0.611
TS36	0.952	0.909	0.957	0.925	0.718
TS45	0.920	0.900	0.940	0.936	0.619
Average	0.930	0.880	0.939	0.914	0.665

These findings demonstrate that the prediction models trained on TS1 performed well on traffic stations TS13, TS21, TS36, and TS45. The models exhibited low errors and high adaptability for these stations identified during the correlation analysis. This suggests that models trained on one station can be effectively applied to multiple stations, in this case, for four stations, reducing the need for explicit training across all stations. This approach highlights the potential for transferring prediction models across locations in urban environments, thereby reducing the number of models required for smart city DT implementations and enhancing scalability.

V. Conclusions

In this paper, we propose a modular and scalable software architecture for smart city digital twins. The developed framework addresses the shortcomings of existing implementations, which are often overly complex, lack generalizability and scalability, and are not modular enough, causing limitations on design and implementations. The developed architecture has been tested for the use case of traffic management for the city of Madrid, using real-world data collected using an IoT sensor testbed. The research presents a detailed system characterization by evaluating the architecture using different predictive sequential learning models. The results demonstrate the generalizability of the adaptive models, offering scalable solutions for urban mobility while minimizing developer overhead. Future work on this research includes incorporating data from diverse sensing modalities, integrating multi-road predictions, and refining model generalizations to enhance the broader applicability of digital twins in smart city contexts.

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] G. White, A. Zink, L. Codecá, and S. Clarke, "A digital twin smart city for citizen feedback," *Cities*, vol. 110, p. 103064, 2021.

[2] 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.

[3] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, 2021.

[4] 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.

[5] 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.

[6] (2024) Open data portal of the madrid city council. [Online]. Available: https://datos.madrid.es/portal/site/egob/

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

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

[9] 3d city db. [Online]. Available: https://www.3dcitydb.org/3dcitydb/visualizationberlin/

[10] G. Schrotter and C. Hürzeler, "The digital twin of the city of zurich for urban planning," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 1, pp. 99–112, 2020.

[11] Virtual singapore - a 3d city model platform for knowledge sharing and community collaboration. [Online]. Available: https://www.sla.gov.sg/articles/press-releases/2014/virtual-singapore-a-3d-city-model-platform-for-knowledge-sharing-and-community-collaboration

[12] 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.

[13] T. Gräber, S. Lupberger, M. Unterreiner, and D. Schramm, "A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 39–47, 2018.

[14] 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.

[15] X. Qi, G. Mei, J. Tu, N. Xi, and F. Piccialli, "A deep learning approach for long-term traffic flow prediction with multifactor fusion using spatiotemporal graph convolutional network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8687–8700, 2022.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] ETSI. (2024) Gs cim 009 - v1.5.1 - context information management (cim); ngsi-ld api. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.08.01_60/gs_CIM009v010801p.pdf

[21] "Smart data models," https://smartdatamodels.org/, accessed: 2024-04-01.

[22] FIWARE. (2024) Stellio context broker. [Online]. Available: https://stellio.readthedocs.io/en/latest/

[23] FastAPI. (2024) Fastapi. [Online]. Available: https://fastapi.tiangolo.com/

[24] FIWARE. (2024) Keycloak integration. [Online]. Available: https://stellio.readthedocs.io/en/latest/admin/keycloak_integration.html

[25] "K3s," https://k3s.io/, accessed: 2024-04-01.