GenTwin: Generative AI-powered Digital Twinning for Adaptive Management in IoT Networks

Kubra Duran, *Graduate Student Member, IEEE,* Hyundong Shin, *Fellow, IEEE,* Trung Q. Duong, *Fellow, IEEE,* and Berk Canberk, *Senior Member, IEEE.*

***Abstract*—The dramatic increase in smart services makes adaptive management of communication networks more critical. Especially for Internet of Things (IoT) networks, adaptive management faces several challenges, like fluctuating network conditions, heterogeneity in data sources, and rapid response capabilities. These challenges lead to performance degradation and data losses in IoT applications if not handled. Even though traditional AI algorithms are applied in most network topologies, they fall short of handling these adaptive management challenges without requiring additional software developments. Therefore, we propose a Generative AI-powered Digital Twinning (GenTwin) framework to create digital twin models with generative AI algorithms. In this framework, we design two novel mechanisms: Priority Pooling and Twin Adapter. Priority Pooling is to extract the dynamic relations within the topology before performing model training. We theoretically formulate the priority levels and corresponding weights with a novel presence parameter to present a modular architecture to increase training efficiency. The Twin Adapter is to interact with the GAI architecture and fine-tune the model for the adaptive twin modelling task in IoT networks. After creating the adaptive twin models, we test the rapid response capabilities of GenTwin with what-if analysis. According to our simulation results, we note that the proposed pooling mechanism extracts the data relations 19% more by enhancing the training accuracy. In addition, we show that GenTwin surpasses the traditional twin performance in terms of rapid response capabilities by reducing the response time significantly.**

***Index Terms*—digital twin, IoT, adaptive management, modelling, generative AI**

I. Introduction

In recent years, the integration of Generative Artificial Intelligence (GAI) into communication networks has revolutionized the concept of effective automation and management strategies. The current developments on Digital Twins (DT) merged with traditional AI algorithms have significantly advanced the efficiency of communication networks from 5G mobile networks to Internet of Things (IoT) networks [1]–[4]. Nevertheless, in the face of the diverse challenges IoT networks present, these solutions lose their effectiveness. This emphasizes the necessity for enhanced digital twining methods in the context of adaptive management strategies [5]. At this point, GAI has significantly affected communication networks by enabling more dynamic, intelligent, and adaptable solutions to handle complex challenges [6]. Regarding the GAI applications, Generative Pretrained Transformers (GPTs), like GPT 3.5, 4, and Large Language Models (LLMs), are catching the interest of both academia and industry [7] due to their successive performance in the generation of related contents. These models provide innovative tools and opportunities for human-computer interaction, model interpretation, and automation systems [8]. Their capabilities in generating multimodal content, applying professional knowledge, and solving problems are particularly noteworthy. However, applying these to the specific concepts for specific tasks requires fine-tuning. For instance, to use a GAI model for the adaptive management of IoT networks requires specific alignments to respond to the management demands [9].

K. Duran is with the School of Computing, Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, U.K., and also with the BTS Group, Istanbul, Turkey. (E-mail:kubra.duran@napier.ac.uk).

H. Shin is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, Yongin-si, Gyeonggido 17104, Republic of Korea (E-mail: hshin@khu.ac.kr).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and also with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT7 1NN Belfast, U.K. (E-mail: tduong@mun.ca).

B. Canberk is with the School of Computing, Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, U.K. (E-mail: b.canberk@napier.ac.uk).

A. Main Challenges in the Adaptive Management of IoT Networks

1) *Fluctuating network conditions*: In an IoT network, fluctuating network conditions cause communication delays and data losses. This variability in network performance presents significant challenges in ensuring real-time data processing. In this circumstance, maintaining adaptive management becomes difficult.
2) *Heterogeneity in data sources*: Divergent types of IoT devices draw a picture of different data formats and software at the backend. These differences require efficient modelling to handle this heterogeneity among the data. In addition, this requirement poses a challenge regarding the scalability of such data.
3) *Lack of rapid responsiveness*: In the adaptive management of IoT networks, creating responses to the queried tasks, such as requesting metadata for a specific IoT sensor, is critical to maintaining smooth IoT network operations. For this reason, the lack of rapid response capabilities degrades the network services, even causing system inefficiencies and potential failures.

These problems require enhanced management capability and adaptability to real-time changes. At this juncture, GAI-based digital twinning strategies are becoming a candidate to hit this management requirements. For instance, a GAI can simulate network conditions and predict fluctuations by allowing for real-time adjustments to bandwidth, latency, and routing strategies. These models also enable networks to respond intelligently to external conditions like weather, congestion, and emergency situations by adjusting configurations to maintain quality of service.

B. State of the Art

1) Traditional AI in IoT and Digital Twins: Digital twins utilize traditional modelling techniques and are limited in their awareness of deployment situations [10]. These rely on historical data by covering only scenarios the system has previously encountered [11]. Related to this issue, [12] introduces a proactive application modelling system for IoT networks with three focuses: incentive, profit, and latency, using a fully distributed edge network. The proposed algorithm ensures optimal deployment and profit maximization with low latency. Besides this approach, the current DT developments combined with AI methods have shown significant performance, especially for wireless networks [13]. For instance, a DT-native AI-driven service modelling is proposed in [14] implementing a TCP-based data flow pipeline and a Reinforcement Learning (RL)-based learner model for IoT-enabled Internet of Vehicles (IoV) networks. Similarly, there are examples utilizing a Deep Learning (DL) classification framework [15], and a Federated Learning (FL)-based DT management scheme for low-carbon industrial IoT networks [16]. The main goal of these studies is to increase connectivity within the IoT topology via the intelligent methods. Furthermore, the energy consumption perspective is considered for edge-based IoT networks to maximize energy efficiency during dynamic topology maintenance via Markov and Multilayer Perceptron-based recommendations [17], [18] and [19]. Regarding the 5G-infrastructured smart city scenarios, [20] and [21] propose advanced encoding and sampling techniques to model mobile networks by employing a spatio-temporal-social multi-feature extraction framework enhanced by an edge-enhanced Graph Convolutional Network (GCN) and Long-Short Term Memory (LSTM). To maintain adaptive network management with Digital Twins, [22] proposes a new qualifier called the age of twin (AoT) to measure digital twin data freshness. This framework aims to enhance twin-to-twin interactions and optimize digital twin modelling for 6G-enabled network deployments across various topology, service, and traffic types. Besides that, [23] proposes a DT-based architecture for efficient data processing for edge-based IoT networks. With this, the anomaly detection phase is presented in an intelligent manner via eXtreme Gradient Boosting (XGBoost), an autoencoder-based algorithm. In [24], a Deep Reinforcement Learning-based (DRL) queue management scheme is proposed to tackle the fluctuating network conditions for IoT networks. Also, the Age-of-information (AoI) metric is utilized to ensure stable performance under the changing data update triggers. Moreover, [25] proposes an optimization framework for changing QoS requirements in next-generation IoT applications. It utilizes different metrics from three dimensions such as computing, networking and application. Furthermore, [26] introduces an RL-based predictive dynamic bandwidth allocation framework to address QoS degradations in heterogeneous IoT networks.

2) Generative AI in IoT and Digital Twins: Despite significant advancements in adaptive network management [35], the Digital Twin with GAI is limited in the telecom network literature. Similarly, its implementation has just started in smart city scenarios [36]. In this circumstance, [27] explores the potential of GAI techniques and their applications in IoT. It addresses challenges in network management and proposes a secure incentive mechanism using Generative Diffusion Models (GDMs) and blockchain for managing IoT securely. Also, [28] introduces a novel modelling method to achieve semantic interoperability in digital twins by focusing on creating Asset Administration Shell (AAS) models within Industry 4.0. The system is powered by LLMs to generate standardized AAS models. Similarly, [29] proposes a collaborative cloud-edge approach, NetGPT, to effectively coordinate diverse communication and computing resources to optimize LLM deployment within the communication networks. The paper notes that fine-tuning open-source models like GPT-2-base and LLaMA, NetGPT demonstrates superior performance over alternative cloud-only or collaboration methods. [30] surveys the advances in communication technology and AI for industrial Internet of Things (IIoT) networks. The paper emphasizes the potential of deep generative models (DGMs) in IIoT applications like anomaly detection, trust-boundary protection, network traffic prediction, and platform monitoring. On the other hand, there are efforts to address the intrinsic challenges of GAI's. For instance, [31] proposes using Generative Adversarial Networks and Self-taught Learning to mitigate the data generation challenges in LLMs in the context of IoT applications. Moreover, [32] investigates the application of LLMs in intelligent network control for 6G terrestrial and non-terrestrial networks. The paper highlights their ability to learn from extensive data and adapt through fine-tuning smaller datasets by proposing control algorithms. Besides, [33] thoroughly explores GAI's applications in physical layer communications. It compares GAI's capabilities with traditional AI. Furthermore, [34] reviews GAI techniques and discusses resource challenges at the edge. It proposes a deep reinforcement learning-based service selection algorithm by demonstrating efficiency in content quality and minimizing task failures compared to standard policies.

All of these efforts related to AI and GAI-based DT developments for network management are summarized in Table I." However, none focuses on the adaptive management of IoT networks with the twin modelling task using generative AI technology. To the best of our knowledge, this is the first study to develop a generative AI-based digital twin modelling for IoT-based smart city applications. Therefore, our research question in this study is *"How can we design an intelligent digital twin modelling scheme by presenting an efficient training phase, highly accurate twin models and rapid responses to handle the adaptive management challenges of IoT networks?"* To address this, we propose a Generative AI-powered Digital Twinning (GenTwin) framework that can create twin models in the form of knowledge graphs. Moreover, at GenTwin, we design two novel mechanisms: Priority Pooling and Twin Adapter. Priority Pooling is to extract the dynamic relations within the topology. To do this, we theoretically formulate the

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2025.3527719

TABLE I
PROPOSED GENTWIN FRAMEWORK AND CURRENT STATE OF THE ART STUDIES

Literature	Adaptive Mgmt	AI method	GAI model	Topology	Mgmt. context
[12], [14], [15], [16], [17], [18], [19]	✓	MLP, RL	-	IoT, IoV	optimal deployment, energy efficiency
[20], [21], [22], [23], [24], [25], [26]	✓	LSTM, GCN, XGBoost	-	5G, 6G, IoT	resource allocation
[27], [28], [29], [30]	-	-	AAS, GPT-2, LLaMA	IoT	security, resource optimization
[31], [32], [33], [34]	-	-	GAN	IoT, 6G	resource management
Our work	✓	-	LLaMA	IoT smart city	adaptive modelling, rapid responses

priority levels and their corresponding weights to present a modular architecture to prepare the data for training. On the other hand, the Twin Adapter is to interact with the LLM architecture and fine-tune the model for the adaptive twin modelling task. After creating the adaptive twin models, we test the rapid response capabilities of the adaptive models with what-if analysis scenarios.

C. Contributions

We summarize the contributions of this study below:

- We design a novel priority pooling mechanism to extract the dynamic relations within the IoT topology. For this, we theoretically derive the priority levels and their respective weights by enabling a modular architecture for the external pool operations to increase the efficiency of the training phase.
- We introduce the twin adapter layer to interact with the core of generative AI and perform fine-tuning for the adaptive twin modelling task. For this, we utilize down and up projection phases by freezing particular parts of an open-source LLM architecture. With the implementation of the twin adapter layer, we create adaptive digital twin models in the form of knowledge graphs.
- We implement a what-if analysis module to define IoT-based smart city scenarios and run these with the specific twin requirements. With this, we test the rapid response capabilities of the adaptive twin modelling against changing update trigger values.

The remainder of the article is organized as follows: In Section II, we explain the details of the proposed GenTwin framework. We give our simulation results in Section III, and finally, we conclude the paper in Section VI by presenting future work cases.

II. GENTWIN: GENERATIVE AI-AIDED DIGITAL TWINING

The proposed GenTwin framework consists of three distinct spaces: Physical Space, Cyber Space, and Service Space. Each is specialized to perform an efficient end-to-end GenTwin for generative AI-aided adaptive IoT management. The details of these spaces are explained below.

A. Architectural Framework

1) Physical Space: This space consists of an IoT topology structure for smart city applications. Therefore, in this space, there are several types of IoT devices serving different smart city services. For instance, air quality sensors are present within the topology for an air-quality measurement service. If the smart service is transportation management, then road infrastructure sensors are located throughout the lanes. Furthermore, gateway devices are presented within the IoT topology that are standing in the fog and communicating with edge sensors.

TABLE II
NOMENCLATURE

Notation	*Explanation*
$\mathcal{P}$	Priority pool
G	Knowledge graph
S	Sensor set
Gw	Gateway set
$\perp$	Presence coefficient
P	Priority level
s_i	i^{th} sensor in sensor set
g_i	i^{th} gateway in gateway set
g_{s_i}	Gateway connected to node s_i
d_i	Total number of connected sensors to i^{th} gateway
m	Number of instances within the dataset
M	Total number of relations
N_i	Number of relations for the i^{th} node
n	Number of smart devices
r	Total number of pools

2) Cyber Space: The LLM-aided modelling framework is embedded within this space to handle the dynamic changes within the topology. We design the Priority Pooling and Twin Adapter modules within this space to create the twin models in the form of knowledge graphs. We also perform prompt generation by using the adaptive twin models to enhance the GenTwin framework.

3) Service Space: We test the rapid response capabilities of GenTwin via the what-if analysis module within this space. For this, we define the smart city service-specific scenarios and their twin requirements to query the required twin version. In this way, the dynamic adaptation in terms of IoT node presence and deletion is maintained rapidly.

Furthermore, GenTwin has two closed control loops to maintain external and internal control flows within the three spaces:

- *External closed-loop:* This control loop consists of four sub-flows to serve the information for an end-to-end updated GenTwin. Namely, the external closed-loop serves the data flow from Physical Space to Service Space. For this, the *data collection* is performed to convey the data from Physical Space to Cyber Space. In our developments, we use an IoT simulation environment as

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2025.3527719

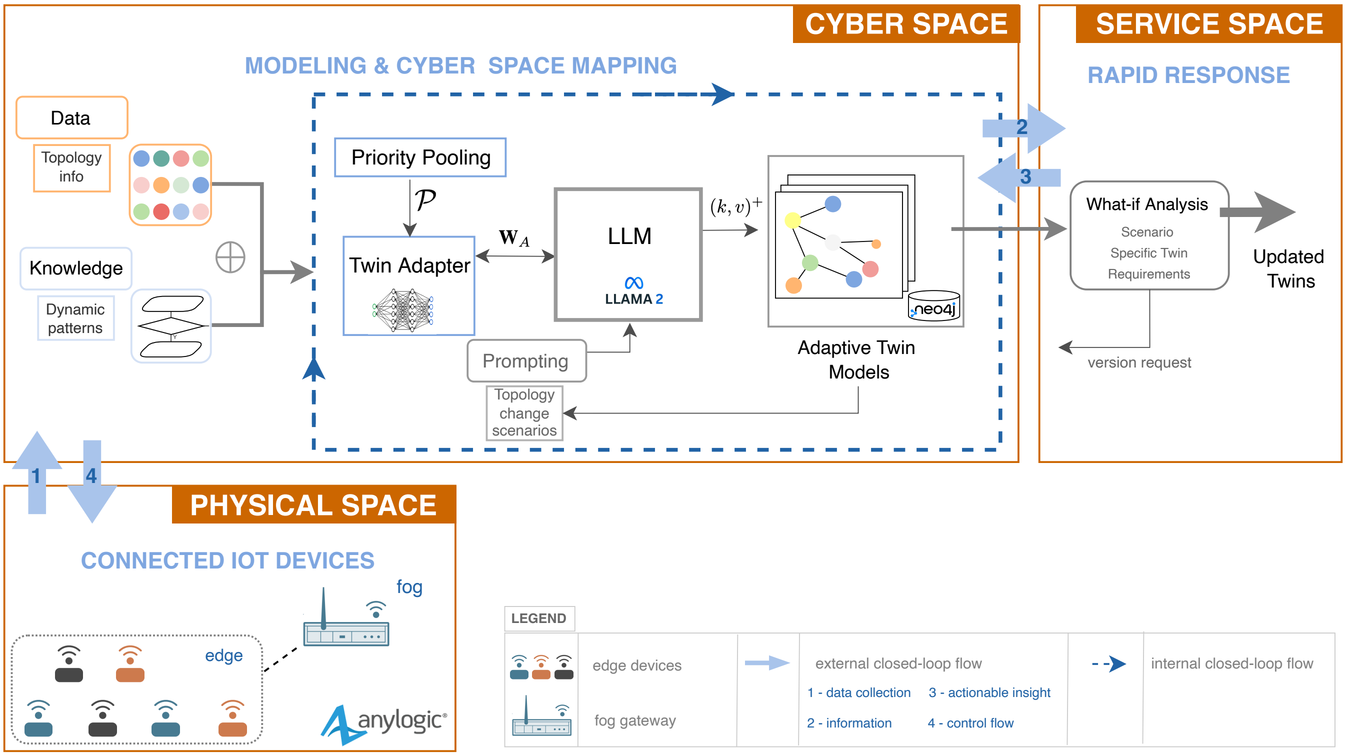

Fig. 1. Architectural Framework of Generative AI-aided Digital Twining - GenTwin.

our Physical Space and feed this space with the smart city application scenarios. Then, the *information* created within the Cyber Space is conveyed to the Service Space to be evaluated for a divergent set of smart city scenarios and the specific twin requirements. After that, depending on the scenario, *actionable insights* are sent to the Cyber Space and the *control flows* to the Physical Space as feedback.

- *Internal closed-loop:* This control flow ensures the stability and the efficiency of the created adaptive twin models in the form of knowledge graphs. Therefore the control flow is performed within the Cyber Space, starting from the output of priority pooling and ending at the creation of adaptive twin models in the knowledge graph database.

B. Modelling

While the number of connected devices and, thus, the relations are growing, the modelling and management methods should be capable of supporting this situation [37], [38]. Here, to prepare for this, we first start with data preprocessing to ensure the interoperability of the heterogeneous data sources for the IoT topology. We perform data parsing and extraction of unstructured data as sensor logs to represent the topology information. Also, we take the historical dynamic patterns as a knowledge base. To ensure that both datasets, topology and historical patterns, are in a unified structure, we transform both into relational formats by storing them in a knowledge graph.

1) Priority Pooling: In generative models, a context window (usually tokens) refers to the amount of input data the model receives. Usually, this window has a limit because of the

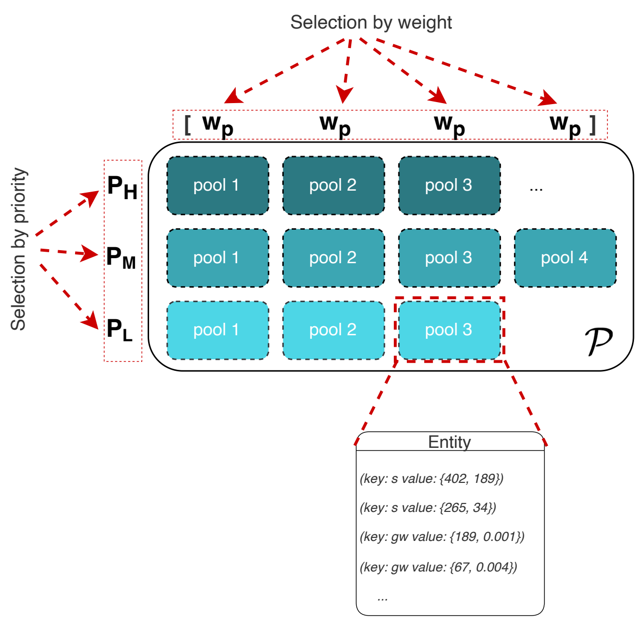

Fig. 2. Priority pooling mechanism

computational resources. Therefore, in generative model applications, it is crucial to feed the model with the most important information at first. Therefore, this mechanism decides the most important relations of the IoT topology. The architecture of this mechanism is given in Fig. 2. It is indicated with $\mathcal{P}$ and has two tuples: $\{P, \vec{W_p}\}$, priority levels, and weights. In our design, there are three distinct priority levels. We first define a presence coefficient, $\perp$ to mark the active devices within the

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2025.3527719

topology. We assign the values for the presence as:

$$\begin{cases} \perp = 1, & \text{if node is active} \\ \perp = 0, & \text{o.w.} \end{cases} \tag{1}$$

Afterwards, we construct the key, k_i, and value, v_i pairs, by quantifying the occurrence of the entities within the data. Here, a (k, v) pair provides information on the device id and its connectivity about a network device (a sensor or a gateway). Depending on the node type, we initialize (k_i, v_i) values as:

$$(k_i, v_i) = \begin{cases} k_i = [\text{:s}], v_i = \{s_{i_{ID}}, g_{s_i}\}, & \text{if } n_i \in S \\ k_i = [\text{:gw}], v_i = \{g_{i_{ID}}, d_i\}, & \text{if } n_i \in Gw \end{cases} \tag{2}$$

where $\forall\ i = 1, ..m$, and m is the number of instances within the dataset. In (2), the key values are assigned as labels, $[: s]$ or $[: gw]$, which indicates the key types. For the values, if a node is a kind of sensor, then the related sensor ID and the ID of its connected gateway are added to the value set. Conversely, if a node is a kind of gateway, then its ID and the total number of sensors connected to this gateway are added to the key-value set. Afterwards, we calculate the priority values at time t as follows:

$$p_{i_t} = \frac{(\sum_i N_i)\perp_i}{M}, i \leq m \tag{3}$$

In (3), N_i is the number of relations for the i^{th} node, M is total number of relations in the sets S, and Gw, and m is the total number of instances within the dataset. Afterwards, we construct three priority pools: high-priority, medium-priority, and low-priority, P_H, P_M, and P_L, respectively, as given:

$$\begin{cases} P_H = \{(k_i, v_i)^+ \mid p_{i_t} > 3Avg(p)/4\} \\ P_M = \{(k_i, v_i)^+ \mid 3Avg(p)/4 \leq\ p_{i_t} < Avg(p)/2\} \\ P_L = \{(k_i, v_i)^+ \mid p_{i_t} \leq Avg(p)/2\} \end{cases} \tag{4}$$

In (4), the $(y)^+$ operator means the set of all y instances fulfilling the given condition, and p is the priority set including all calculated p_i values. We then proceed with the weight calculation. For this, we know that the gateway is important in an IoT topology, standing at the fog and communicating with edge sensors. For this reason, the number of connected IoT sensors to a gateway implies its importance within the topology in terms of service continuity. To interpret this, we use the node centrality measure to calculate the weight values of each device within the topology. Here, the bigger the average node centrality value, the higher the weight of a device.

$$w_i = \begin{cases} \frac{1}{n}, & \text{if} \quad k_i = [\text{:s}] \\ \frac{d_i}{n}, & \text{if} \quad k_i = [\text{:gw}] \end{cases} \tag{5}$$

In (5), we assume each sensor is connected to one gateway only. We then calculate the weight values of each sub-pool, p as:

$$W_p = \frac{1}{n} \sum_i 1 + d_i, \ \forall i \in \mathcal{P} \tag{6}$$

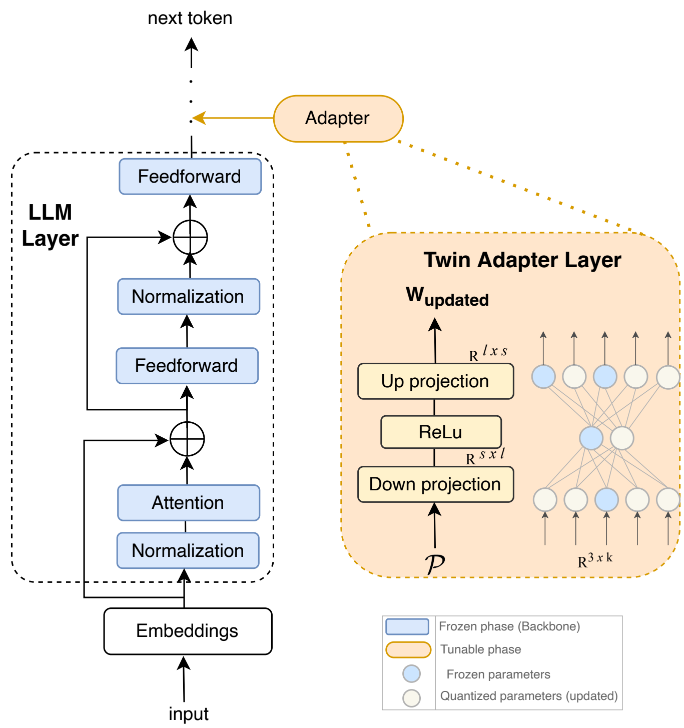

Fig. 3. LLM Layer and designed Twin Adapter Layer in GenTwin.

Afterwards, we construct the weight values of priority levels as $\vec{W_p} = \{W_{p_j} | j = 1, 2, ...r\}$, where r is the total number of pools. And then, we create the calculated weight matrix by stacking the weight vectors as $\mathbf{W}_C = \bigcup_{i=1}^{3} W_{p_i},\ W = W_{p_1} \cup\ W_{p_2} \cup W_{p_3}$. Here, the priority pool indexes indicate our three main priority levels: P_H, P_M, and P_L. We define an update trigger parameter, α for the calculation of weight values for the pool as given below:

$$\alpha_i = |\Delta(w_{p_{i(t)}} - w_{p_{i(t+1)}})| \tag{7}$$

where Δ operator checks for the changes in the weight values for the i^{th} pool within the $\mathcal{P}$.

2) Twin Adapter: As seen in Fig. 3, the twin adapter works after the LLM Layer. In LLM applications, a token is the smallest unit of text that a model processes. In GenTwin, we define the context window, or the token as a word including the entity's relations that will be embedded. Within the Twin Adapter Layer, we freeze some parts of the parameters to prevent overfitting and feed the layers with different combinations of calculated sub-pools. Here, the model that we generate based on the open-source LLM outputs the knowledge graph of the IoT topology based on the weight values which are calculated for the specific IoT context. We explain the details of the Twin Adapter Layer below:

- *Frozen phase:* Frozen phase does not require running the embedded algorithm within the LLM. In Fig. 3, we give the common architecture of LLMs' on the left with the frozen phases, which means we do not run the LLM model from scratch. Instead, we use an open-source LLM and use its trained parameters.

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2025.3527719

- *Frozen parameter:* Freezing the parameters refers to keeping certain model parts unchanged during the fine-tuning process. This means the weights (parameters) are not updated based on the new task-specific data. In our framework, the frozen parameters are located within the adapter layer. And the main target of utilizing frozen parameters is to fine-tune the framework to create the knowledge graph of the IoT topology by embedding all dynamic relations, thus presenting adaptive twin models.
- *Tunable phase:* Unlike the frozen phase, a tunable phase is allowed to run the embedded algorithm. As seen in Fig. 3, the twin adapter is the tunable phase in GenTwin to calculate the weights with the IoT topology data. In this way, GenTwin adapts its parameters to perform better by weighing the relations in IoT data by decreasing computational costs.
- *Quantized parameter:* Unlike frozen parameters, quantized parameters are considered in the training phase and updated accordingly at the end.
- *Down projection:* This layer reduces the dimensionality of embeddings within the model. A subset from $\mathcal{P}$ is extracted and fed into the down projection. We indicate the weights to be updated in the projection as W_D, where $\in R^{kxl}$.
- *Activation Function:* This phase embeds non-linearity into the model via several mathematical function options. The embedded function allows it to learn and represent more complex patterns and relationships in the data. We utilize the Rectified Linear Unit (ReLU) as the non-linear activation function to keep our adapter simpler and nonnegative.
- *Up projection:* Contrary to down projection, up projection transforms the lower dimensions into higher-dimensional vectors. The up projection phase is calculated through a linear transformation: $y = W_A\ .\ x$. Here, we created the higher-dimensional vector y using the weight matrix, W_A, learned through the training within the Twin Adapter Layer. Moreover, we indicate the weights to be updated in the projection as W_U, where $\in R^{lxk}$.

$$\mathbf{W}_A = \mathbf{W}_A + ReLu((\mathcal{P}(.)\mathbf{W}_D)\mathbf{W}_U \tag{8}$$

In (8), the learning weight matrix of the Twin Adapter Layer is constructed by feeding the various combinations of the priority pool to the projection layers. (.) operator shows the extraction operation from the pool $\mathcal{P}$. During the training, we observed the results against Mean Absolute Error (MAE), also known as L1 loss:

$$L_1 = \frac{\sum_M |r_i - \hat{r}_i|}{M} \tag{9}$$

After all these steps, adaptive twin models are created as knowledge graphs. In a smart city application, there are several sub-services, each with a specific target. And knowledge graphs are efficient for divergent sets of use cases in modelling. That's why we choose knowledge graphs to store the generated dynamic IoT dataset. We denote our created knowledge graphs as $G = (V, E)$, where V is the set of nodes, and E is the set of directed edges. As seen in Fig. 4, the knowledge graph of an IoT-based smart city application is a knowledge collection of labelled nodes, such as entity categories (i.e. gateway, sensor, smart city service, etc.) linked by relationships, such as *includes*, *serves in*, etc. to structure the directed paths. Moreover, all the nodes have their own metadata, giving the smart city application details.

3) Prompting: This phase is required to use the specific inputs to guide the GenTwin model to generate desired outputs. We indicate the generated output as:

$$O = g_{(k,v)}(q, R) \tag{10}$$

where, the function g extracts the respected key-value pairs with the retriever R, and perform verbalizing depending on the query q. In the backend, the generative model creates all possible relations that may occur between the entities. Therefore, GenTwin can respond to different prompting questions via the generated information at the backend. As given in Fig. 4, prompt generation is performed on the produced adaptive twin models in the first iteration of GenTwin. We apply specific questions by filtering the knowledge graph and then extracting the relevant facts within the IoT-based smart city scenario. We only show one result in the figure for only the question, *"What is a PM2.5 sensor?"*.

Algorithm 1 GenTwin: Generative-AI aided Digital Twining

Require: topology information, dynamic patterns
Ensure: G, $\mathcal{P}$, P

```
Initialize W_A, P_H, P_M, P_L, α_i;
if change in α_i
    Decide presence values                               ▷ (1)
    Construct (k, v) pairs, and priority values          ▷ (2),(3)
    Create priority levels, P_H, P_M, P_L                ▷ (4)
    Assign the weight values, create 𝒫                   ▷ (5),(6)
    Update α_i ← Δ(w_{p_{i(t)}} − w_{p_{i(t+1)}})        ▷ (7)
end
Perform projections in Twin Adapter
Update the weights, W_A                                  ▷ (8)
Generate G
Calculate L_1 = Σ_M |r_i − r̂_i| / M                      ▷ (9)
Generate prompts O = g_{(k,v)}(q,R)                      ▷ (10)
Update the LLM with the generated new prompts
Define specific smart city service scenarios
foreach episode
    Decide twin requirements
    List (k, v) pairs
    Request respective twin versions
    Measure the response time of adaptive twin models
end
```

After generating adaptive twin models and storing them in a graph database, a rapid response testing service is activated within the Service Space. We design the what-if analysis module to change the input conditions representing the dynamic changes in the IoT topology. Depending on the required twin models, a version request is sent to the graph database to get the most updated version of the twins.

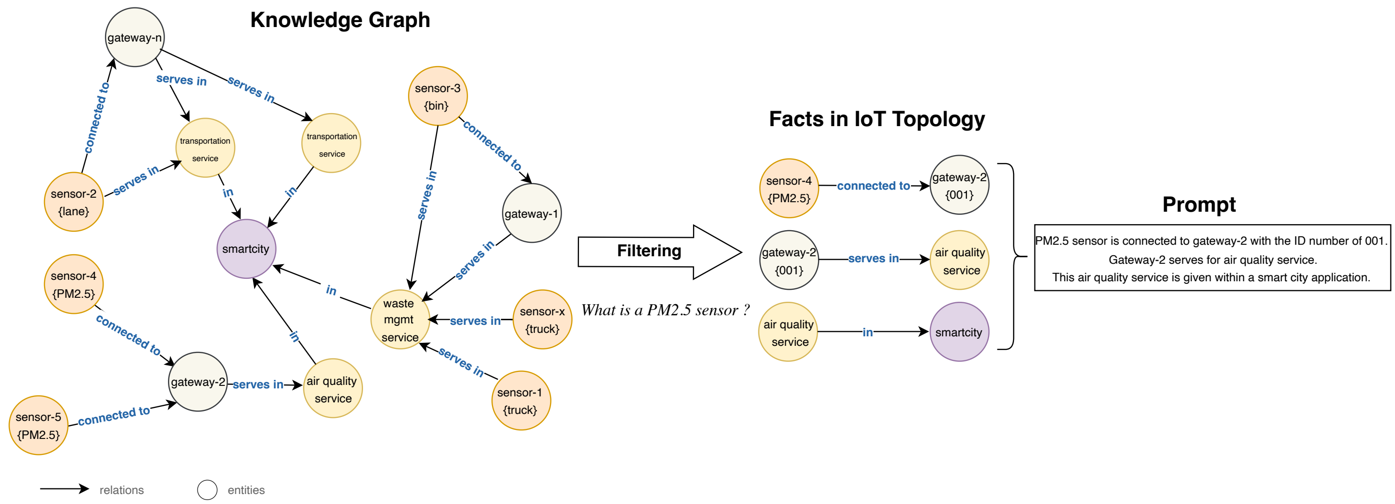

Fig. 4. Knowledge graph representation of IoT integrated smart city application (on the left), Filtering results according to the specific question (at the middle), Generated prompt by verbalizing the extracted facts (on the right).

As seen in Alg. 1, GenTwin comprises two main parts: Priority Pooling (implemented in lines 3-7) and Twin Adapter Layer (implemented in lines 9-13). Before implementing these, the weight matrix, priority levels and update trigger parameters are initialized in line 1. Afterwards, the presence values are calculated, (k, v) values are created, and the priority values are assigned in lines 2-4. The priority pool is constructed in lines 5-6 using the priority values and the weights. As the final step of Priority Pooling, the trigger parameter is updated in line 7. After that, the Twin Adapter Layer is activated in line 9, and projection layers are run. In each epoch, the weight values are updated (line 10) by observing the loss function results (line 11). To enhance the LLM model, prompts are generated for the IoT data (line-12) and feedback to the model. Then, the knowledge graph of the topology is constructed with the LLM. To test the rapid responsiveness of the twin models, a what-if analysis module is implemented, and response time is measured (lines 15-20). Overall, the priority pool calculates the importance of each nodes considering the number of relations they have and highlights the more critical ones within the IoT topology. Thanks to the created pools, the datasets are fed into the Twin Adapter according to two criteria: selection by priority and selection by weight for fine-tuning. These two options make the training process more efficient as the most weighted and prior ones will be used first. Afterwards, the Twin Adapter performs fine-tuning and learns the embedded relations within the dataset. This step prepares the model to produce knowledge graph data.

III. Performance Evaluation

A. Simulation Environment:

We construct the Physical Space by using the DT-based IoT simulator AnyLogic©. Moreover, we create Cyber Space including Priority Pooling and the Twin Adapter and perform training and model creation operations by using Python and PyTorch©. As an open-source LLM, we use Llama-2-8B. As the graph database to store the created knowledge graphs, we use Neo4j. The data communication between physical space and the cyberspace is managed by feeding the txt files to the Anylogic environment. Also, graph queries are utilized to communicate data between the graph database and physical space. The end-to-end GenTwin simulation environment is shown in Fig. 5. Also, all simulation parameters are given in Table III.

TABLE III
Simulation Parameters

Parameters	Values
Number of smart city services	{2, 3}
Number of IoT sensors	250
Number of gateways	50
Learning rate	0.08
Discount factor	0.99
Batch size	{32, 128, 256}
Confidence interval	95%

B. Dataset:

We created our IoT-based smart city scenario form scratch by using open source datasets[1]. We assume all the smart city service types are identical and have the same topology. For instance, all air-quality services for different locations have the same number of sensors and gateways within the topology. The raw data we use in our simulations includes 1000 instances, 300 nodes, and 580 unique relations. We divide the data into three parts: 70% as training data, 15% as test data, and 15% as validation data to use in the performance evaluation process.

C. Performance Results:

In this section, our target is to investigate the performance of GenTwin considering (i) the extraction of dynamic relations with the proposed priority pooling mechanism, (ii) the accuracy of model generation and corresponding L1 loss values

[1]https://github.com/smart-data-models/SmartCities?tab=readme-ov-file

This article has been accepted for publication in IEEE Transactions on Cognitive Communications and Networking. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCCN.2025.3527719

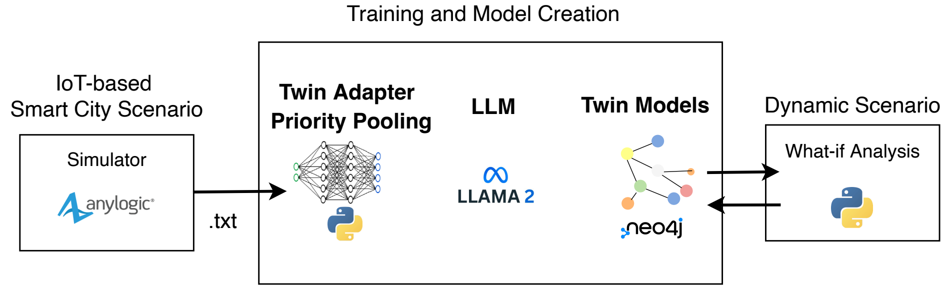

Fig. 5. Simulation environment.

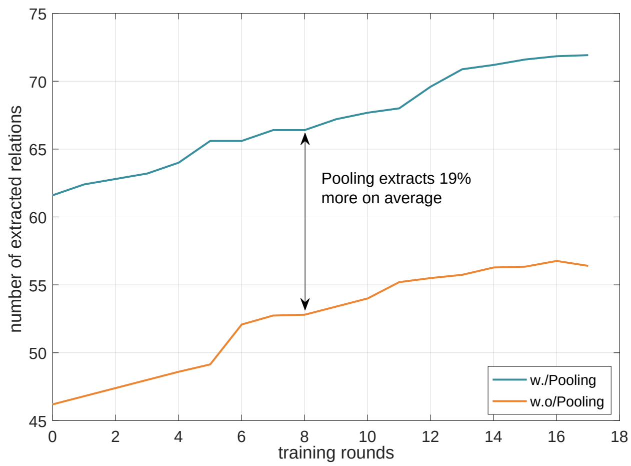

Fig. 6. The number of extracted relations for training rounds.

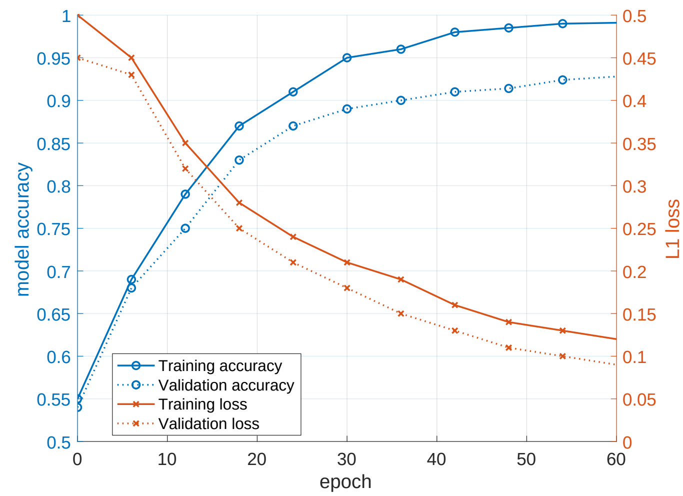

Fig. 7. Training and validation results in terms of accuracy and loss.

for the training and validation phases, and (iii) response time of the adaptive twin models against increasing update trigger parameter.

We first test the performance of the Priority Pooling mechanism by examining its capability of extracting dynamic relations within the IoT topology. In this test, we observe the total number of extracted relations for each training round (up to eighteen) for pooling-enabled and pooling-disabled cases. In the pooling-disabled case, GenTwin directly runs the Twin Adapter Layer operations fed by the data and knowledge data. Conversely, in the pooling-enabled case, GenTwin performs prioritization on the data and fed these groups at first to the Twin Adapter Layer. As seen in Fig. 6, the pooling-enabled case of GenTwin performs 19% better on average than the pooling-disabled case. The main reason is the weight calculation and prioritization of the data before performing the model training phase. According to the recorded results, it is clear that performing priority pooling enhances the performance of the trained model.

Furthermore, we test the Twin Adapter's performance by observing the model generation's efficiency. As a starting point, we use Llama's learned weights and then calculate the Twin Adapter Layer's weights. Afterwards, we run the model creation step for sixty epochs. We record the model accuracy and L1 loss values for the training and validation data. As seen in Fig. 7, the model accuracy and L1 loss values behave conversely, meaning that the training and validation phases converge at a particular epoch value. We record the accuracy of models converging to the maximum value of 1. Even though the accuracy of models in the validation does not reach the highest value, it converges to ~ 0.9, which is a notable achievement for the normalized accuracy interval of $[0, 1]$. This mainly stems from the successive performance of the projection layers within the Twin Adapter Layer in fine-tuning the LLM model to achieve an adaptive twin modelling task.

In addition, we test the rapid responsiveness of the GenTwin for dynamical changes within the IoT topology. For this, we created several smart city scenarios using Python within the What-if analysis module. When a scenario is run as the main scenario, it queries the required twin models from the graph database, Neo4j in our case. We observe the response times in ms against increasing update trigger parameter, α, which means dynamicity within the topology. We compare the recorded response times of GenTwin with the Traditional Twin [39]. As seen from Fig. 8, when the update trigger parameter is 0.2, the response time for GenTwin is $\sim 400ms$, for the traditional twin is $\sim 550ms$. The difference between the response times for GenTwin and traditional twins increases when the dynamicity value increases. Namely, when the dynamicity has its lowest value, the difference between the response times of GenTwin and the traditional twin is 13%. When the dynamicity has its highest value, the difference reaches up to $\sim 53\%$

values. The main reason for this is observing the α value and updating it at the end of each iteration of Priority Pooling given in line-7 in Alg. 1. As the traditional twin does not perform such dynamicity control, it shorts the fall of GenTwin.

TABLE IV
GENTWIN PERFORMANCE UNDER CHANGING NETWORK CONDITIONS

	Data Completeness (%)		
	small topology	*medium topology*	*large topology*
Scenario-1	99.8	99.8	99.7
Scenario-2	99.7	99.7	99.6

In the last section of our experiments, we test GenTwin's performance in generating knowledge graphs under different topology sizes and changing network conditions. For this, we measure the data completeness metric in percentage, meaning how much GenTwin can maintain data in cyberspace, even in the interrupted data flow from physical space. We also define two distinct scenarios to test the data completeness performance as detailed below:

- *Scenario-1:* This is the base scenario where the network conditions do not degrade and cause an interruption in the data flow from the physical IoT topology. In this scenario, the data is fed to GenTwin every 2 seconds. This behaviour means GenTwin will receive a periodic update.
- *Scenario-2:* In this scenario, the data flow from physical IoT topology is cut for a time interval of 10 seconds to create a fluctuating network condition. In this behaviour, GenTwin will not be updated during this time duration.

We run these scenarios for three different IoT topology sizes: small (30 sensors), medium (100 sensors) and large (250 sensors). In this test, we assume that each sensor will be sending only its unique ID to cyberspace, and this information will be considered enough to contribute to data completeness. We run each scenario on the respective topologies and measure the data completeness values. According to the simulation results, we note that the data completeness values are similar in both scenarios. The main reason for this is that GenTwin can generate information about an entity within the topology based on the information provided for priority pooling. The exact values for data completeness are given in Table IV.

IV. CONCLUSION AND FUTURE DIRECTIONS

Adaptive management plays a crucial role in today's communication networks. However, generative models are required to combat the adaptive management challenges in IoT networks. With this motivation, we introduce a Generative AI-powered Digital Twining (GenTwin) framework in this study. We first design a Priority Pooling mechanism to extract dynamic relations within the data and thus increase the efficiency of the training phase. Then, we design a Twin Adapter Layer to interact with the generative AI and perform fine-tuning for the adaptive twining modelling of IoT networks. Afterwards, we create what-if analysis scenarios to test GenTwin's rapid response capabilities. We test our proposed framework on

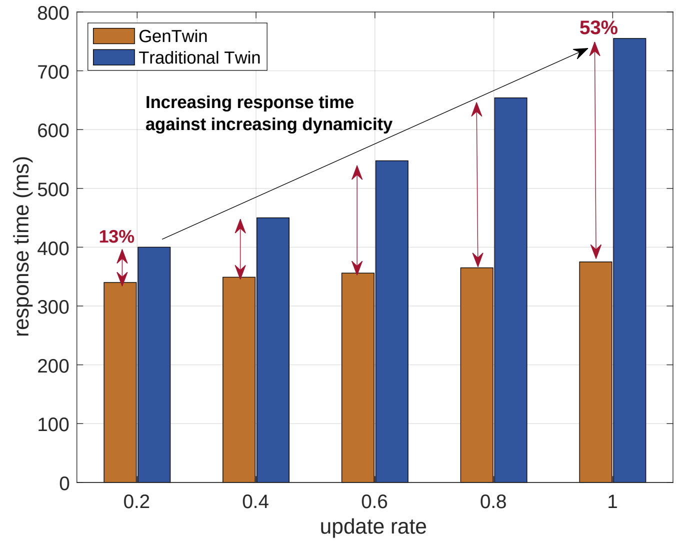

Fig. 8. Measured response times for GenTwin and Traditional Digital Twin against increasing dynamicity.

an IoT-based smart city application topology. Our simulation results show that GenTwin surpasses the traditional twin models in terms of training efficiency, adaptive modelling, and rapid responsiveness.

For future work, we aim to investigate the long-term adaptability of GAI-enhanced DTs in evolving IoT ecosystems, focusing on continuous learning and self-improvement. We also plan to test GenTwin's scalability against an increasing number of IoT devices.

ACKNOWLEDGMENTS

This work was partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) 1515 Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab. Project 5239903. The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. The work of K. Duran was supported in part by DeepMind.

REFERENCES

[1] L. Zeng, S. Ye, X. Chen, and Y. Yang, "Implementation of big ai models for wireless networks with collaborative edge computing," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 50–58, 2024.

[2] E. Ak, K. Duran, O. A. Dobre, T. Q. Duong, and B. Canberk, "T6conf: Digital twin networking framework for ipv6-enabled net-zero smart cities," *IEEE Communications Magazine*, vol. 61, no. 3, pp. 36–42, 2023.

[3] A. Celik and A. M. Eltawil, "At the dawn of generative ai era: A tutorial-cum-survey on new frontiers in 6g wireless intelligence," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2433–2489, 2024.

[4] M. A. Ferrag, O. Friha, B. Kantarci, N. Tihanyi, L. Cordeiro, M. Debbah, D. Hamouda, M. Al-Hawawreh, and K.-K. R. Choo, "Edge learning for 6g-enabled internet of things: A comprehensive survey of vulnerabilities, datasets, and defenses," *IEEE Communications Surveys Tutorials*, vol. 25, no. 4, pp. 2654–2713, 2023.

[5] K. Duran, E. Ak, G. Yurdakul, and B. Canberk, "6g-enabled dtaas (digital twin as a service) for decarbonized cities," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2023, pp. 421–426.

[6] O. Friha, M. A. Ferrag, B. Kantarci, B. Cakmak, A. Ozgun, and N. Ghoualmi-Zine, "Llm-based edge intelligence: A comprehensive survey on architectures, applications, security and trustworthiness," *IEEE Open Journal of the Communications Society*, pp. 1–1, 2024.

[7] S. Bengesi, H. El-Sayed, M. K. Sarker, Y. Houkpati, J. Irungu, and T. Oladunni, "Advancements in generative ai: A comprehensive review of gans, gpt, autoencoders, diffusion model, and transformers," *IEEE Access*, vol. 12, pp. 69 812–69 837, 2024.

[8] B. Qin, H. Pan, Y. Dai, X. Si, X. Huang, C. Yuen, and Y. Zhang, "Machine and deep learning for digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 21, pp. 34 694–34 716, 2024.

[9] M. Xu, H. Du, D. Niyato, J. Kang, Z. Xiong, S. Mao, Z. Han, A. Jamalipour, D. I. Kim, X. Shen, V. C. M. Leung, and H. V. Poor, "Unleashing the power of edge-cloud generative ai in mobile networks: A survey of aigc services," *IEEE Communications Surveys Tutorials*, vol. 26, no. 2, pp. 1127–1170, 2024.

[10] L. V. Cakir, C. J. Thomson, M. Özdem, B. Canberk, V.-L. Nguyen, and T. Q. Duong, "Intelligent digital twin communication framework for addressing accuracy and timeliness tradeoff in resource-constrained networks," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2024.

[11] Y. Yigit, K. Duran, N. Moradpoor, L. Maglaras, N. Van Huynh, and B. Canberk, "Machine learning for smart healthcare management using iot," in *IoT and ML for Information Management: A Smart Healthcare Perspective*. Springer, 2024, pp. 135–166.

[12] S. Deng, Y. Chen, G. Chen, S. Ji, J. Yin, and A. Y. Zomaya, "Incentive-driven proactive application deployment and pricing on distributed edges," *IEEE Transactions on Mobile Computing*, vol. 22, no. 2, pp. 951–967, 2023.

[13] C. Tunc, T. X. Tran, and K. Joshi, "Digital Twins for Beyond 5G," in *AI in Wireless for Beyond 5G Networks*, 1st ed. CRC Press, 2024, pp. 169–190.

[14] K. Duran, M. Broadbent, G. Yurdakul, and B. Canberk, "Digital twin-native ai-driven service architecture for industrial networks," in *2023 IEEE Globecom Workshops (GC Wkshps)*, 2023, pp. 1297–1302.

[15] J. Pan, N. Ye, H. Yu, T. Hong, S. Al-Rubaye, S. Mumtaz, A. Al-Dulaimi, and I. Chih-Lin, "Ai-driven blind signature classification for iot connectivity: A deep learning approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6033–6047, 2022.

[16] H. Liao, Z. Zhou, N. Liu, Y. Zhang, G. Xu, Z. Wang, and S. Mumtaz, "Cloud-edge-device collaborative reliable and communication-efficient digital twin for low-carbon electrical equipment management," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1715–1724, 2023.

[17] E. Bozkaya, M. Erel-Özçevik, T. Bilen, and Y. Özçevik, "Proof of evaluation-based energy and delay aware computation offloading for digital twin edge network," *Ad Hoc Networks*, vol. 149, p. 103254, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870523001749

[18] Z. Ding, L. Shen, H. Chen, F. Yan, and N. Ansari, "Energy-efficient topology control mechanism for iot-oriented software-defined wsns," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13 138–13 154, 2023.

[19] K. Duran and B. Canberk, "Digital twin enriched green topology discovery for next generation core networks," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 4, pp. 1946–1956, 2023.

[20] S. Huang, H. Zhang, X. Wang, M. Chen, J. Li, and V. C. M. Leung, "Fine-grained spatio-temporal distribution prediction of mobile content delivery in 5g ultra-dense networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 1, pp. 469–482, 2024.

[21] H. Du, J. Liu, D. Niyato, J. Kang, Z. Xiong, J. Zhang, and D. I. Kim, "Attention-aware resource allocation and qoe analysis for metaverse xurllc services," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 7, pp. 2158–2175, 2023.

[22] K. Duran, M. Özdem, T. Hoang, T. Q. Duong, and B. Canberk, "Age of twin (aot): A new digital twin qualifier for 6g ecosystem," *IEEE Internet of Things Magazine*, vol. 6, no. 4, pp. 138–143, 2023.

[23] B. Bolat-Akça and E. Bozkaya-Aras, "Digital twin-assisted intelligent anomaly detection system for internet of things," *Ad Hoc Networks*, vol. 158, p. 103484, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870524000957

[24] T. Song and Y. Kyung, "Deep-reinforcement-learning-based age-of-information-aware low-power active queue management for iot sensor networks," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 700–16 709, 2024.

[25] J. L. Herrera, J. Galán-Jimnez, J. Garcia-Alonso, J. Berrocal, and J. M. Murillo, "Joint optimization of response time and deployment cost in next-gen iot applications," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3968–3981, 2023.

[26] I. Chakour, C. Daoui, M. Baslam, B. Sainz-De-Abajo, and B. Garcia-Zapirain, "Strategic bandwidth allocation for qos in iot gateway: Predicting future needs based on iot device habits," *IEEE Access*, vol. 12, pp. 6590–6603, 2024.

[27] J. Wen, J. Nie, J. Kang, D. Niyato, H. Du, Y. Zhang, and M. Guizani, "From generative ai to generative internet of things: Fundamentals, framework, and outlooks," *IEEE Internet of Things Magazine*, vol. 7, no. 3, pp. 30–37, 2024.

[28] Y. Xia, Z. Xiao, N. Jazdi, and M. Weyrich, "Generation of asset administration shell with large language model agents: Towards semantic interoperability in digital twins in the context of industry 4.0," *IEEE Access*, pp. 1–1, 2024.

[29] Y. Chen, R. Li, Z. Zhao, C. Peng, J. Wu, E. Hossain, and H. Zhang, "Netgpt: An ai-native network architecture for provisioning beyond personalized generative services," *IEEE Network*, pp. 1–1, 2024.

[30] S. De, M. Bermudez-Edo, H. Xu, and Z. Cai, "Deep generative models in the industrial internet of things: A survey," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 5728–5737, 2022.

[31] T. Tu, Z. He, Z. Zheng, Z. Zheng, J. Jiang, Y. Gong, C. Hu, and D. Cheng, "Towards lifelong unseen task processing with a lightweight unlabeled data schema for aiot," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[32] B. Rong and H. Rutagemwa, "Leveraging large language models for intelligent control of 6g integrated tn-ntn with iot service," *IEEE Network*, pp. 1–1, 2024.

[33] N. Van Huynh, J. Wang, H. Du, D. T. Hoang, D. Niyato, D. N. Nguyen, D. I. Kim, and K. B. Letaief, "Generative ai for physical layer communications: A survey," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 3, pp. 706–728, 2024.

[34] H. Du, Z. Li, D. Niyato, J. Kang, Z. Xiong, X. Shen, and D. I. Kim, "Enabling ai-generated content services in wireless edge networks," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 226–234, 2024.

[35] K. Duran, B. Karanlik, and B. Canberk, "Graph theoretical approach for automated ip lifecycle management in telco networks," *Wiley Int J Network Mgmt*, vol. 31, no. 4, e2138, 2021.

[36] C. Savaglio, V. Barbuto, F. Mangione, and G. Fortino, "Generative digital twins: A novel approach in the iot edge-cloud continuum," *IEEE Internet of Things Magazine*, pp. 1–7, 2024.

[37] D. M. Gutierrez-Estevez, B. Canberk, and I. F. Akyildiz, "Spatio-temporal estimation for interference management in femtocell networks," in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, 2012, pp. 1137–1142.

[38] K. Duran, B. Karanlik, and B. Canberk, "Ai-driven partial topology discovery algorithm for broadband networks," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[39] D. R. R. RAJ, T. A. Shaik, A. Hirwe, P. Tammana, and K. Kataoka, "Building a digital twin network of sdn using knowledge graphs," *IEEE Access*, vol. 11, pp. 63 092–63 106, 2023.

Kubra Duran (Graduate Student Member, IEEE) received her B.Sc. degree in Electrical and Electronics Engineering in 2016 and her M.Sc. degree in 2021 in Computer Engineering from Istanbul Technical University. She has five years of experience in the industry, performing in R&D Specialist and Software Engineering roles. She has been awarded the Best MSc Thesis Award'22 by the IEEE Communications Society Turkey Chapter and the IEEE Computer Society Turkey Chapter. She is currently pursuing her Ph.D. in the School of Computing, Engineering and the Built Environment at Edinburgh Napier University. Her research interests are Digital Twin Networks, Intelligent Internet of Things and AI-enabled Communication Frameworks.

Hyundong Shin (Fellow, IEEE) received the B.S. degree in Electronics Engineering from Kyung Hee University (KHU), Yongin-si, Korea, in 1999, and the M.S. and Ph.D. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 2001 and 2004, respectively. During his postdoctoral research at the Massachusetts Institute of Technology (MIT) from 2004 to 2006, he was with the Laboratory for Information Decision Systems (LIDS). In 2006, he joined the KHU, where he is currently a Professor in the Department of Electronic Engineering. His research interests include quantum information engineering, wireless communication, and machine intelligence. Dr. Shin received the IEEE Communications Society's Guglielmo Marconi Prize Paper Award and William R. Bennett Prize Paper Award. He served as the Publicity Co-Chair for the IEEE PIMRC and the Technical Program Co-Chair for the IEEE WCNC and the IEEE GLOBECOM. He was an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS.

Trung Q. Duong (Fellow, IEEE) is a Canada Excellence Research Chair (CERC) and a Full Professor at Memorial University of Newfoundland, Canada. He is also the adjunct Chair Professor in Telecommunications at Queen's University Belfast, UK and a Research Chair of Royal Academy of Engineering, UK. His current research interests include quantum communications, wireless communications, signal processing, machine learning, and realtime optimisation. Dr. Duong has served as an Editor/Guest Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS MAGAZINES, and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He received the Best Paper Award at the IEEE VTC-Spring 2013, IEEE ICC 2014, IEEE GLOBECOM 2016, 2019, 2022, IEEE DSP 2017, IWCMC 2019, 2023, and IEEE CAMAD 2023. He has received the two prestigious awards from the Royal Academy of Engineering (RAEng): RAEng Research Chair (2021-2025) and the RAEng Research Fellow (2015-2020). He is the recipient of the prestigious Newton Prize 2017.

Berk Canberk (Senior Member, IEEE) is a Professor at the School of Computing, Engineering and The Built Environment, Edinburgh Napier University, and he is also an Adjunct Professor at the Department of Artificial Intelligence and Data Engineering, Istanbul Technical University. He serves as an Editor in IEEE Transactions on Vehicular Technology, Elsevier Computer Networks and Elsevier Computer Communications. He is the recipient of IEEE ICC Best Paper Award (2024), IEEE Turkey Research Incentive Award (2018), IEEE INFOCOM Best Paper Award (2018), The British Council (UK) Researcher Link Award (2017), IEEE CAMAD Best Paper Award (2016), Royal Society International Exchanges (UK) (2021), Royal Academy of Engineering (UK) NEWTON Research Collaboration Award (2015), IEEE INFOCOM Best Poster Paper Award (2015), ITU Successful Faculty Member Award (2015) and Turkish Telecom Collaborative Research Award (2013). His current research areas are AI-enabled Digital Twins, IoT Communication, and Smart Wireless Networks.