An Architecture and Realization of a Smart City Digital Twin with eHealth Case Studies

Running Title

Digital Twin Architecture & Realization

Authors

- Nandana Jayachandran[1]
- Atef Abdrabou[1]
- Mohammad Al Bataineh[1]
- Kamarul Ariffin Noordin[2]

Affiliations

1. Department of Electrical and Communication Engineering, College of Engineering, UAE University, Al Ain, Abu Dhabi, UAE, PO 15551.
2. Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia.

Corresponding Author

Atef Abdrabou

Email: atef.abdrabou@uaeu.ac.ae

Tel: +971-3-713-5149

ORCID IDs

Nandana Jayachandran: https://orcid.org/0009-0003-7783-6535
Atef Abdrabou: https://orcid.org/0000-0002-9639-7684
Mohammad Al Bataineh: https://orcid.org/0000-0003-0148-7618
Kamarul Ariffin Noordin: https://orcid.org/0000-0003-4003-4793

Acknowledgments

This work was made possible by the support of the UAE University AUA Grant 12N143.

Data availability statement

No new datasets were created in this study. All datasets used are appropriately cited within the article.

Funding statement

This research was funded by the UAE University UPAR grant number 12N143.

Conflict of interest disclosure

The authors declare no conflict of interest.

Ethics approval statement

Not applicable.

Patient consent statement

Not applicable.

Permission to reproduce material from other sources

Clinical trial registration

Not applicable.

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ORIGINAL RESEARCH PAPER

An Architecture and Realization of a Smart City Digital Twin with eHealth Case Studies

Nandana Jayachandran[1] | **Atef Abdrabou***[1] | **Mohammad Al Bataineh**[1] | **Kamarul Ariffin Noordin**[2]

[1]Department of Electrical and Communication Engineering, UAE University, College of Engineering, Al-Ain, Abu Dhabi, PO 15551, UAE

[2]Department of Electrical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia

Correspondence
Corresponding author: Atef Abdrabou.
Email: atef.abdrabou@uaeu.ac.ae

Present address
Department of Electrical and Communication Engineering, UAE University, College of Engineering, UAE University, Al-Ain, Abu Dhabi, UAE, PO 15551.

Abstract

Digital Twinning is an advanced technology that involves creating virtual replicas of various physical systems. In smart cities, digital twins serve as digital representations that model and simulate various urban elements, such as environment protection (e.g., air quality), critical infrastructure, transportation networks, and other urban management processes. It has recently gained considerable attention for its transformative potential, enabling city authorities to visualize and analyze complex city dynamics for better-informed decision-making. Therefore, this paper proposes a simplified layered architecture for smart city digital twins. The layers of the proposed architecture cover the range of operations required by the functionality of the digital twin and the interaction between them, from data transfer or synthesizing to big data streaming and intelligent analytics. The paper also introduces an open-source software tool that realizes the proposed architecture, with each layer designed as an independent Python module for easy integration and maintenance. Three case studies are used to demonstrate the capabilities of the tool. One use case addresses short-term forecasting of the air quality index, whereas the other use case targets the detection of an individual's respiratory condition based on data received from wearable devices. The third case combines the other two cases to offer a warning system for residents with medical conditions based on air quality. The results of the case studies show the tool's ability to effectively handle environment-related and e-health-related use cases and combine them for the welfare of smart city residents, leading to a more resilient, health-focused urban landscape.

KEYWORDS
Digital Twin; Smart City; e-Health, Machine Learning; Tool; Architecture

1 | INTRODUCTION

The rapid growth of urban populations, coupled with increasing demands for sustainable development, has pushed cities to explore advanced technologies to address complex challenges in urban governance. By 2050, it is projected that 68% of the world's population will reside in urban areas, increasing pressure on governments to find innovative solutions for issues such as traffic congestion, energy distribution, and environmental degradation[1]. In response, cities increasingly leverage digital transformation technologies to enhance urban governance. Among these technologies, Digital Twin (DT) technology has emerged as a crucial tool, promising to revolutionize urban management through real-time data insights and predictive capabilities[1,2].

The concept of smart city digital twins represents a transformative approach to urban management and planning, leveraging advanced technologies to create virtual replicas of physical urban environments. A digital twin is defined as a digital representation of a physical entity, which facilitates real-time monitoring, analysis, and optimization of urban systems and processes[3,4]. This innovative framework serves as a bridge between the physical and digital realms, enabling city planners, policymakers, and stakeholders to visualize and simulate various urban scenarios, thereby enhancing decision-making capabilities[5,6]. The integration of big data and the Internet of Things (IoT) is pivotal in the development of digital twins, as these technologies provide the necessary data streams that feed into the digital models, ensuring their accuracy and relevance[3,7].

The emergence of smart city digital twins is largely driven by the need to address complex urban challenges, such as traffic congestion, resource management, and environmental sustainability. By simulating urban dynamics, digital twins allow for exploring different interventions before implementation, thus minimizing risks and optimizing outcomes[5]. For instance, city officials can utilize these models to assess the impact of new traffic regulations or urban development projects, thereby making informed decisions that align with the city's strategic goals[6]. Furthermore, the ability to integrate real-time sensor data enhances the responsiveness of urban systems, allowing for proactive management of infrastructure and services[8 9].

Integrating digital twin technology into the realm of eHealth within smart cities represents a significant advancement in the management and delivery of healthcare services. A digital twin, as a virtual representation of a physical entity, allows for real-time monitoring, analysis, and optimization of health-related processes and systems[10 4]. This innovative approach not only enhances the quality of healthcare delivery but also fosters a more personalized and efficient healthcare ecosystem. By leveraging data from various sources, including wearable health devices, electronic health records, and environmental sensors, digital twins can create dynamic models that reflect the health status of individuals and populations[10 4]. This facilitates a comprehensive understanding of urban health dynamics.

As urban environments become increasingly complex, the need for integrated health management systems that can respond to real-time data becomes paramount[11 12]. Digital twins enable healthcare providers and city planners to simulate various health scenarios, assess the impact of urban policies on public health, and optimize resource allocation[12 13]. For instance, by modeling the spread of diseases or the effects of environmental factors on health, stakeholders can make informed decisions that enhance community well-being and resilience[10 14].

Moreover, the use of digital twins in eHealth can significantly improve patient outcomes through personalized medicine. By creating individualized digital replicas of patients, healthcare providers can tailor treatments based on real-time data, leading to more effective interventions[10 4]. This capability is particularly relevant in managing chronic diseases, where continuous monitoring and timely adjustments to treatment plans are crucial[10 14]. Furthermore, the integration of artificial intelligence (AI) with digital twin technology can enhance predictive analytics, allowing for early detection of health issues and proactive management of patient care[10 4].

On the other hand, by simulating various health scenarios and interventions, city officials can assess the impact of urban policies on public health outcomes[11 15]. For example, digital twins can be used to model the effects of environmental changes, such as air quality improvements or the introduction of green spaces, on community health[15]. This capability is particularly relevant in the context of addressing public health challenges, such as the spread of infectious diseases or the management of chronic health conditions, where timely and data-driven interventions are essential[16]. However, the implementation of a smart city twin architecture in general and for eHealth in particular is not without challenges. One of the foremost challenges is data management. Smart cities rely on vast amounts of data collected from numerous sources such as sensors, IoT devices, and environment monitoring systems. This data, while abundant, is often fragmented and stored in isolated systems, making it difficult to create a unified, real-time representation of corresponding entities, leading to limiting the ability of cities to create a comprehensive, real-time view of these entities, which is crucial for effective decision-making[17].

Thus, this paper introduces a smart city digital twin architecture with a realization of this architecture in the form of an open-source software tool [†] that can interface with real IoT devices or simulate these devices by generating correlated data from any available dataset. This real or simulated data is streamed through the tool using the message queuing telemetry transport (MQTT) protocol to a big data platform that categorizes this data into relevant topics, stores it, and makes it accessible to intelligent machine learning (ML) analytic models created by the user based on the use case and application demand. The tool is scalable in terms of the number of devices it can receive the data from, either real hardware or simulated ones, and the number of employed ML models. Moreover, the paper introduces three integrated case studies that demonstrate the proposed tool as a realization of a smart city digital twin providing a personalized public health advising system based on environmental factors, such as air quality level and health status of city residents.

The main contributions of this paper are three-fold.

- It introduces a simple smart city digital twin architecture that can be used for different smart city use cases, including environment and eHealth-related ones, by accommodating IoT-sensed data and/or eHealth data from wearable devices of city residents in a big data analytics framework.

[†] https://github.com/nandanajayachandran/SCDTT/tree/main

- It provides a realization of the proposed architecture in the form of a scalable open-source software tool that can receive live data or simulate the generation of realistic data, stream it to a big data platform, and analyze it using various configurable ML models.
- Different case studies are presented demonstrating the ability of the tool to employ AI to predict the impact of the environment on the health of smart city residents with medical conditions.

The rest of the paper is organized as follows. Section 2 reviews the most relevant research works in the literature. Section 3 introduces a smart city digital twin architecture fostering intelligent big data analytics. The structure of the software tool realizing the introduced architecture is described in Section 4. Section 5 presents different case studies that demonstrate the usage of the proposed tool. Finally, Section 6 concludes the paper.

2 | RELATED WORKS

The role of digital twins extends beyond mere simulation. They enable enhanced data integration and real-time analytics, which can lead to improved urban governance and citizen services [17–21].

Indeed, they can play a vital role in enhancing urban resilience and sustainability. As cities face increasing pressures from climate change, population growth, and technological advancements, digital twins provide a framework for understanding and mitigating risks associated with urbanization [22 23]. They facilitate data-driven policy-making by offering comprehensive insights into urban environments, enabling stakeholders to identify vulnerabilities and develop effective countermeasures [24 25]. Huang et al. [26] argue that leveraging digital twin data can significantly enhance urban management by enabling real-time remote control of physical assets.

Thus, digital twins are employed across various domains within smart cities, including urban planning, environment protection, infrastructure management, transportation systems, situational awareness, and public health. Hämäläinen in [11] discusses the early adoption of digital twin technology in Helsinki, highlighting its potential to enhance urban development through improved data-driven decision-making. Moreover, the socio-technical perspective on digital twins emphasizes the importance of incorporating social dimensions into urban planning, ensuring that the needs and behaviors of citizens are considered in the decision-making process [27 28].

In addition to their application in urban planning and management, digital twins are instrumental in disaster risk management. In [22] and [29], the authors argue that the ability to simulate various disaster scenarios allows cities to prepare and respond more effectively to emergencies. Ford and Wolf in [30] propose a conceptual model for using smart city digital twins (SCDT) in disaster management, illustrating how integrated sensing and simulation can improve community resilience. In transportation, digital twins play a crucial role in optimizing traffic management and enhancing mobility systems. Yeon et al. [31] introduce DTUMOS, a digital twin framework designed for urban mobility operating systems, which facilitates the testing of various mobility algorithms and policies.

By integrating data from multiple sources, including weather forecasts and infrastructure conditions, digital twins can provide real-time insights that enhance situational awareness to facilitate coordinated responses during crises [25]. This capability underscores the potential of digital twins not only to improve operational efficiency but also to enhance the overall quality of life for urban residents.

The benefits of employing digital twins in eHealth are manifold. They have been increasingly applied in various eHealth contexts, including personalized healthcare, chronic disease management, and health monitoring systems since they enable healthcare providers to simulate various health scenarios, assess the impact of interventions, and optimize resource allocation [32]. In [33], the authors discuss the potential of digital twins to provide personalized healthcare services, emphasizing the need for ethical considerations in their deployment. Ferko et al. [34] emphasize the importance of architectural solutions for digital twins, which can evaluate "what-if" scenarios using intelligent algorithms. In [16], the authors present an interoperable eHealth reference architecture that facilitates the integration of digital twins into primary care using a service-oriented approach and a communication bus to connect distributed applications. The study by Ven et al. [32] indicates that digital twins can enhance patient engagement by providing tailored health information and empowering patients to take control of their care.

The integration of digital twins with other smart city technologies, such as the Internet of Things (IoT) and artificial intelligence (AI), is critical for enhancing their effectiveness in eHealth applications. Qian et al. [35] discuss the role of IoT in connecting smart devices to collect and analyze health data, enabling effective monitoring and control of health systems. The authors suggest that leveraging machine learning techniques can further enhance the capabilities of digital twins in analyzing complex health data.

The work of Tekinerdoğan and Verdouw in [36] on systems architecture design patterns for digital twins can provide insights into creating adaptable and scalable eHealth solutions.

In short, some research works in the literature discuss the capabilities of digital twins to enable smart cities to improve city governance and citizen welfare by enhancing urban resilience via planning, management, and sustainability. Other works show that digital twins are also envisioned to support community resilience via disaster management and mobility enhancement by providing a conceptual model [30] or a framework [31]. However, a few research works provided insights on integrating IoT and AI in smart city digital twins to enable effective eHealth applications [35] and [36].

To the best of our knowledge, no other research work in the literature offers a tool that provides a realization of smart city digital twin architecture fostering the integration of IoT, big data, and machine learning, allowing modeling the impact of environmental changes on community health.

3 | DIGITAL TWIN ARCHITECTURE

Our study emphasizes using digital twin (DT) technologies within smart cities while supporting e-health for city residents, demonstrating its potential to transform these domains via data-driven decision-making and predictive analytics. In the sequel, we introduce a simplified digital twin architecture for smart city services, including e-health.

3.1 | Architecture of the Digital Twin for Smart Cities

As illustrated in Fig. 1, the proposed architecture consists of multiple layers, each contributing to the overall functionality of the digital twin system. These layers include the Data Emulation Layer, Data Exchange Layer, Data Ingestion Layer, Data Intelligence Layer, Decision Support Layer, and Client Interaction Layer. Each layer plays an essential role, working together to ensure the smooth operation of the system.

In the figure, we also see a depiction of a smart city with sensors, highlighting the extensive use of sensors that represent the cyber-physical world, where everything revolves around data. Smart cities rely on a broad spectrum of IoT sensors, including those for detecting motion, tracking locations, monitoring air quality, temperature, wind speed, humidity, and many more. In addition, there are wearable devices of the city residents that can be in different forms, such as smartwatches and portable activity/health monitoring devices, carrying various sensors, such as heartbeat, respiratory rate, and stress level, to name a few. The data generated by these sensors propagate through the architecture, playing a crucial role in how the system functions. All these sensors hold a massive amount of valuable information. Thus, the architecture is designed to handle and efficiently process this amount of data. Additionally, the architecture allows for flexibility and scalability, making it easier to adapt to different applications and requirements. The specific functions of the layers are discussed below.

3.1.1 | Data Emulation Layer

The Data Emulation Layer is the first layer in the digital twin architecture and plays a critical role in generating data when real-time sensor data from physical sensors is not yet available for the required scenarios. This layer should contain emulated sensors that act like real sensors, providing the ability to be configured to generate synthetic correlated data in real time based on historical datasets. It is mainly used to simulate the data generation for intended use cases, especially when the available data is insufficient for intelligent analytics, helping maintain predictive and classification capabilities of the digital twin system with acceptable accuracy for decision-making functions.

3.1.2 | Data Exchange Layer

This layer will stream the data from the cyber-physical world, or emulated sensors, in real time. Using standard data communication protocols like Message Queuing Telemetry Transport (MQTT) [37] or Constrained Application Protocol (CoAP), this layer ensures that data moves efficiently and securely throughout the system. In smart cities, for instance, data collected from sensors such as those monitoring temperature, traffic, or pollution levels needs to be transmitted reliably. The Data Exchange Layer also ensures this happens without delay, providing real-time transfer to critical information, such as e-health data. Thus, it shall

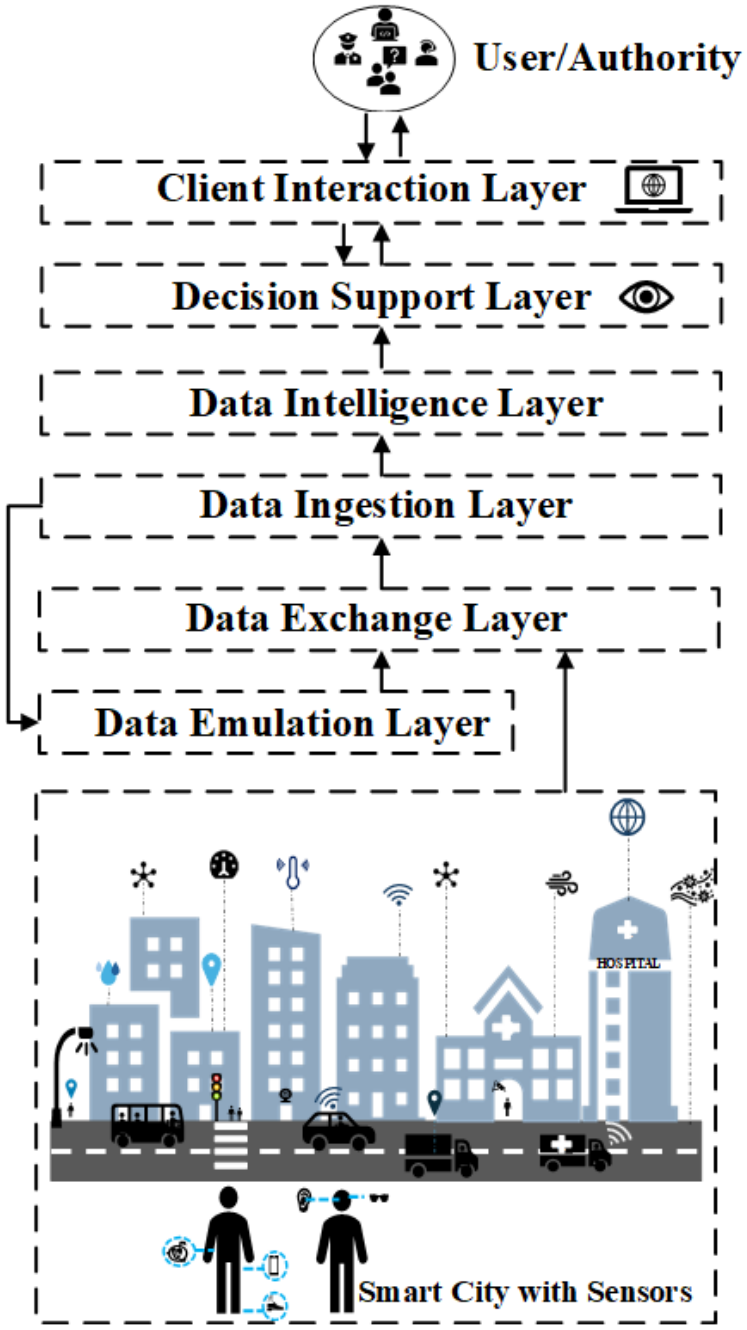

FIGURE 1 Digital Twin Architecture

support communicating with IoT devices, wearables, or portable devices via different wireless technologies. In addition, this layer is responsible for the confidentiality and integrity of the received data as well as the authentication of the hardware devices that send it.

3.1.3 | Data Ingestion Layer

Once data has been collected and transferred to the digital twin system, the Data Exchange Layer passes it to the Data Ingestion Layer. This layer is responsible for dealing with data streamed in real time. It stores this data, organizes it according to its topics or categories, and prepares it to be analyzed effectively. It ensures that the incoming data is structured, valid, and ready for

further processing, thereby maintaining the quality and reliability of the data being used. The datasets stored by this layer can also be used to support the Data Emulation layer to simulate new scenarios or use cases that require the information of this data.

3.1.4 | Data Intelligence Layer

The Data Intelligence Layer is responsible for analyzing and processing data contained in the Data Ingestion Layer to uncover meaningful insights. This layer uses techniques such as machine learning and data mining to build and optimize models, making them available for further evaluation. It is also tasked with training and testing these models to ensure accuracy and performance. Additionally, this layer supports anomaly detection by identifying irregularities or unexpected patterns in the data, which is crucial for detecting potential issues such as faulty sensors or unusual system behavior. Through continuous learning, the models are regularly updated by the Data Ingestion Layer with new data, allowing the system to adapt and improve over time.

3.1.5 | Decision Support Layer

After the data has been processed, this layer provides guidance to users, such as city municipality officials or healthcare professionals, on how to respond to various situations. It leverages the machine learning models from the Data Intelligence Layer to make informed decisions based on the processed data. For example, in an e-health-related scenario, the system may alert doctors to a patient's deteriorating condition, or in another scenario, it could give forecasts on the high energy consumption of some city areas or buildings based on weather conditions or other reasons. Therefore, this layer translates data-driven insights into practical actions that enhance decision-making and overall system efficiency.

3.1.6 | Client Interaction Layer

The Client Interaction Layer is the top layer, where the digital twin system client or user engages directly with it. This layer serves as the interface for users such as smart city officials, healthcare professionals, or city residents to control, monitor, or interact with the system, respectively. It shall offer a portable, user-friendly interface that is intuitive for non-technical users, allowing them to interact with the system effortlessly. Users can visualize information based on their specific needs. For instance, it can offer a forecasting service regarding the anticipated status of certain public parking areas for city residents, recommending the best place to park their cars in the near future. Additionally, the layer includes an alert and notification system that sends timely alerts for any critical changes or issues, enabling immediate action when necessary. For example, in a smart city, users might receive alerts on energy consumption spikes or traffic congestion. Similarly, in an e-health application, doctors could be notified about sudden changes in a patient's condition. This layer is crucial for ensuring decision-makers can act quickly and effectively based on real-time data. Thus, this layer should provide data integrity as well as authentication and access control to all the clients dealing with the digital twin system. Furthermore, this layer can also interact with client devices in a cyber-physical system if these devices are required to be automatically controlled based on the decisions made by the Decision Support Layer, which relies on analytics supported by real or simulated data.

This layered structure of the digital twin architecture allows for seamless integration of different technologies and provides the flexibility needed to manage complex systems, such as smart cities. The layering ensures that each function, from data collection to decision-making, is handled efficiently and can be adapted to suit a wide range of use cases.

4 | A SOFTWARE REALIZATION FOR THE DIGITAL TWIN ARCHITECTURE FOR SMART CITIES

In order to realize the aforementioned simplified architecture, we propose a modular open-source tool called the Digital Twin Realization for Smart City (DTRSC) tool. The tool realizes most of the proposed functionalities of each layer of the layered architecture. For seamless integration and easy maintenance, each layer of the DT architecture is implemented as an independent module using Python code. The tool features a command-line interface (CLI) as the primary means of interaction, allowing users to configure, monitor, and engage with the DT model in a scalable, portable, and computing-resources efficient manner. It enables users to interact directly with smart city services, including e-health applications, making more effective use of real-time data.

The adaptable nature of the DTRSC tool ensures that all components are tightly integrated with the overall DT framework, allowing for smooth transitions from data generation to real-time analysis and predictive modeling. This approach not only optimizes dynamic smart city functions, such as traffic management and environment protection, but also extends its capabilities to healthcare applications, including patient status monitoring and predictive diagnostics.

The DTRSC tool is designed as an open-source platform, providing broader accessibility and customization for various applications. The main components and their interactions are illustrated in Fig. 2, demonstrating how each module integrates into the DT framework to deliver seamless data generation, real-time processing, and predictive analytics. In the sequel, the functionality of each tool module will be presented.

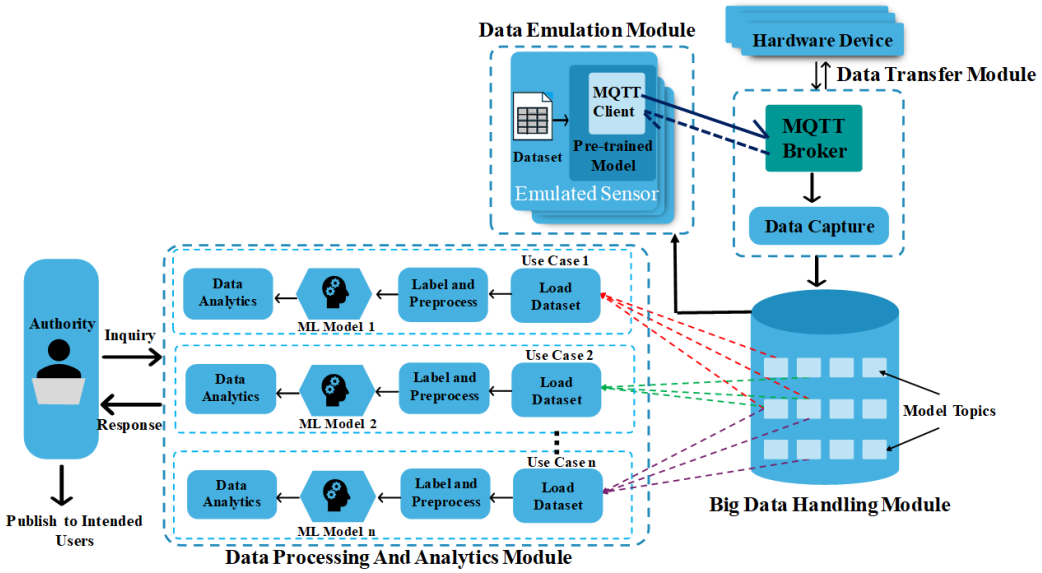

FIGURE 2 The structural design of the DTRSC tool.

4.1 | Data Emulation Module

The Data Emulation Module aligns with the Data Emulation Layer of the proposed DT architecture, containing emulated devices that replicate real-world IoT nodes. Each device can contain more than one sensor of different types. The emulated sensors are developed using pre-trained machine-learning models created from realistic datasets. When real-time sensor data is unavailable or difficult to obtain, the system leverages these models to generate a time series of synthetic correlated data in real time. The generated data closely mirrors real-world conditions, accurately imitating the performance of physical sensors. Similar to physical IoT devices, the emulated devices communicate with the rest of the tool (the Data Transfer module) via the MQTT protocol. Therefore, every emulated device should be assigned an IP address. These emulated devices can be created individually or in groups. This allows the tool to support the scale required for smart cities in terms of the number of devices.

A configurable data generation profile controls the data generation process, allowing users to define the type of emulated sensors they need to attach to any virtual (emulated) device. The list of available sensors can be expanded by integrating more emulated sensors, as the DTRSC tool is open-source. Moreover, the time profile for the data generation is configurable. It can be periodic or random data generation, depending on the user's preference based on the type of data to be generated. Additionally, the emulation time can be customized, providing flexibility for different scenarios.

4.2 | Data Transfer Module

The Data Transfer Module corresponds to the Data Exchange Layer of the DT architecture and is responsible for the efficient streaming and exchange of data. Once data is generated by real and/or emulated sensors, it is encapsulated into MQTT network packets, which are then transmitted to an MQTT broker. The broker manages message routing between the sensors (real or emulated) and the rest of the tool. In this setup, hardware devices and emulated sensors operate as MQTT clients, while the MQTT broker ensures smooth data flow within the system. To function as clients, hardware/emulated devices must be assigned IP addresses. For moving devices, such as e-Health gadgets, a mobile IP scheme is assumed to be in place.

To enhance the data exchange process, the DTRSC uses PyShark [38] to capture and analyze network traffic, providing a unified interface to the Big Data Handling module regardless of the data source (real hardware or emulated sensors). This makes data generated by sensors, whether real or emulated, flow seamlessly through the system, allowing for continuous real-time analysis and predictive modeling across various applications. While MQTT is the primary protocol used for communication, the DTRSC is flexible enough to support alternative data protocols if needed.

4.3 | Big Data Handling Module

The Big Data Handling Module is a critical component that manages the continuous data flow within the digital twin system. The properties of the Data Ingestion Layer are implemented here by leveraging Kafka's [39] robust streaming capabilities. This module captures and processes real-time data from physical devices or emulated sensors. The data captured by PyShark is then stored in pre-assigned Kafka topics, categorized by device ID or IP address and sensor type. These topics shall be created based on the use case or the digital twin service offered by smart city officials. The mapping between device IDs or IP addresses, locations, and their owners is inaccessible to the DTRSC tool and its users. They should be kept secret in confidential records to maintain the anonymity of these devices for any smart city digital twin service, including e-Health.

This module can store data for further analysis or send the information to the Data Emulation Module to simulate devices. Kafka's configuration is optimized for performance, with features such as message compression, idempotency, and batch processing to enhance data throughput and reliability. This ensures that the data pipeline can efficiently handle large volumes of information, which is crucial for maintaining real-time analytics capabilities in different smart city DT services.

4.4 | Data Processing and Analytics Module

The Data Processing and Analytics Module forms the analytical backbone of the DTRSC, representing the Data Intelligence and Client Interaction layers of the DT architecture. It processes real-time data streams and historical datasets to extract actionable insights and generate predictions. Users can select specific machine learning models through the CLI, such as Long Short-Term Memory (LSTM), Linear Regression, Random Forest, Gradient Boosting, or Neural Networks, to perform regression and classification tasks but it can also be extended to include additional models. This module's flexibility allows it to cater to a wide range of analytical needs, supporting complex data-driven decision-making.

During the data processing phase, the system applies data preprocessing techniques to ensure the information is ready for machine learning analysis, addressing aspects such as missing values, feature scaling, and data encoding. Once the data is refined, the models generate predictions, delivering insights that guide strategic actions for city authorities or tool users.

The CLI-based interface empowers authorities or tool users to interact directly with the system, allowing them to configure the machine learning models, make inquiries, and receive immediate responses even with limited programming knowledge.

4.5 | DTRSC Work Flow

The DTRSC tool developed in this study leverages a Command-Line Interface (CLI) designed to manage and configure the entire workflow from data generation to analytics. The DTRSC acts as an intuitive platform that allows users or relevant authorities to set up and control DT applications across various domains, such as e-health, air quality monitoring, and beyond. This section details the workflow experienced while using the DTRSC, illustrating the interaction process at each stage.

```
Welcome to DTRSC Tool
-----------------------------

Please choose an option to get started:
1. Start a new project
2. Load an existing project
3. Exit
-----------------------------

Enter your choice: 1
Generating a new project...
Project created successfully!
Please choose an option to generate real-time data:
1. Capture Hardware Data
2. Synthesize Data
3. Exit
-----------------------------

Enter your choice: 2
Synthesizing data using emulated sensors...
```

FIGURE 3 Creating a new project in DTRSC.

Upon launching the CLI, users are prompted to either start a new project, load an existing project, or exit the application. Selecting "Start a new project" initiates a new project configuration, as shown in Fig. 3. After creating the project, the DTRSC prompts the user to choose between hardware-generated or synthetic data. In this figure, the user selects the option to proceed with synthetic data.

```
Data Emulation Module
-----------------------------

Setting up Module Parameters...
Network Configuration
Enter use case: Case 1
How would you like to add devices?
1. Add Group of Devices
2. Add Single Device
Enter your choice: 1
Enter the number of devices to add in the group: 10
Configuring 'Device_1':
Enter the number of sensors you want to select for Device_1: 10
Sensor selection for Device Device_1
Select sensor 1 for Device_1 by entering the corresponding numbers:
1. PM2.5
2. PM10
3. NO2
4. CO
5. O3
6. SO2
7. Temperature
8. Heart Rate
```

FIGURE 4 Device and network configuration for a group of devices.

The Data Emulation module is responsible for configuring the network of IoT devices or sensor nodes. The user first provides a use case, such as e-health in Fig. 4. This is labeled as "Case 1." The user then chooses how to add devices based on the application requirements, with options to add a group or a single device. Fig. 4 illustrates the process of adding a group of identical devices. Here, the user specifies the number of devices, selects sensors for the first device and the remaining devices, and the MQTT broker will be automatically configured, as shown in Fig. 5.

```
Data Emulation Module
-----------------------------

Setting up Module Parameters...
Network Configuration
Enter use case: Case 1
How would you like to add devices?
1. Add Group of Devices
2. Add Single Device
Enter your choice: 2
Enter the name of the device: D1
Configuring 'D1':
Enter the IP address for MQTT Client 'D1': 127.0.0.2
Enter the number of sensors you want to select for D1: 2
Sensor selection for Device D1
Select sensor 1 for D1 by entering the corresponding numbers
1. PM2.5
2. PM10
3. NO2
4. CO
5. O3
6. SO2
7. Temperature
8. Heart Rate
```

FIGURE 5 Configuration of group devices and MQTT broker.

If the user selects the single device option, it can be configured by naming the device, assigning an IP address for the MQTT client, and choosing the sensors to be used, as shown in Fig. 6. After completing the network configuration, the system sets up the MQTT broker, which manages communication between devices in the data exchange layer. If devices are added individually, the user can view a configuration summary of each device. The user is also prompted to choose a time profile (periodic or random) for data generation, allowing flexibility in emulating real-time sensor device behavior.

```
Select sensor 6 (1-14): 6
Sensor 6 'SO2' selected.
Data Generation Profile
Select data generation time profile for 'SO2':
1. Periodic
2. Random
Enter choice (1/2): 1
Enter time period in second(s): 1
Time period for 'SO2' Sensor 6 set to 1 seconds.
You have selected 6 sensor(s) for Device_1.
Device_1_PM2.5 Sensor1: Periodic, 1 second(s)
Device_1_PM10 Sensor2: Periodic, 1 second(s)
Device_1_NO2 Sensor3: Periodic, 1 second(s)
Device_1_CO Sensor4: Periodic, 1 second(s)
Device_1_O3 Sensor5: Periodic, 1 second(s)
Device_1_SO2 Sensor6: Periodic, 1 second(s)
Device 1 configured with IP: 127.0.0.2
Configured devices from Device 2 to Device 10 as identical to
 Device 1, with IP addresses from 127.0.0.3 to 127.0.0.11.
MQTT Broker is configured with IP Address: 127.0.0.1
```

FIGURE 6 A configuration snapshot based on a single device.

Once the sensor devices are configured, the user is prompted to initiate data streaming. The Big Data Handling Module then begins streaming the generated sensor or IoT data into a Kafka-based big data platform by connecting to the Kafka broker and creating Kafka topics to categorize data by sensor type, as shown in Fig. 7. For instance, topics such as D1_Temperature and D1_HeartRate are created to stream temperature and heart rate data in real time. In this example, we illustrate five topics due to screen limitations, which restrict capturing additional topics in a single image. Kafka's producer configurations ensure that data is efficiently streamed to the correct topics while consumers subscribe to these topics to read data as it is generated. Users can monitor the data stream and choose to view the streaming data in real time through Kafka, as shown in Fig. 7.

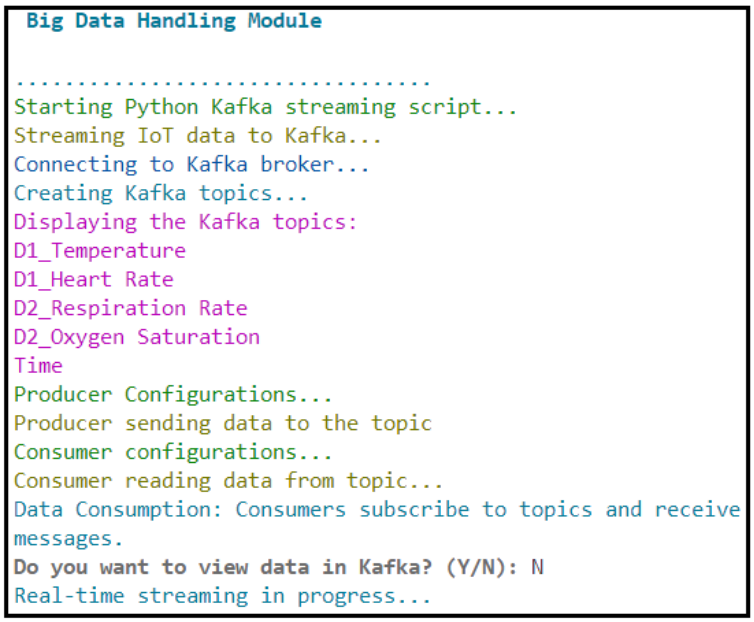

FIGURE 7 Real-time data streaming in the Big Data Handling module.

After the data has been streamed into Kafka, the next step is to analyze it using the Data Processing and Analytics Module. The DTRSC tool allows users to load an existing dataset or generate a new real-time dataset for analysis, as shown in Fig. 8. For real-time dataset generation, users specify the duration for which data will be captured from the real time streamed data. They can then select input variables from the available Kafka topics. For example, a machine learning model may use selected input features from this list to predict the target variable(s). Here, in Fig., 8, respiratory rate is chosen as the target (output) variable.

Once the input and output variables are selected, the system allows the user to choose a machine learning algorithm from a list of available options, including LSTM (Long Short Term Memory), Linear Regression, Gradient Boosting, Random Forest, and Neural Networks, as shown in Fig. 9. The tool can be easily extended to include other models and their respective parameters. The DTRSC tool supports the creation of both regression and classification models, and in this example, regression analysis is performed. The user selects a Neural Network model, and the tool prompts them to configure it by specifying parameters such as the number of layers, neurons per layer, and the optimization algorithm. For instance, a neural network can be configured to include five layers and 64 neurons per layer and to use the Adam optimization algorithm. The model is trained using the dataset generated from the streaming data, with 80% of the data used for training and 20% for testing. The training process involves fitting the model to the data, and the system provides feedback on the performance of the trained model.

Once the model has been trained, users can obtain the output values by providing input values for the selected features. The DTRSC tool then uses the trained model to predict the output variable and display the result to the user.

At this stage, users can choose to continue using the current model, create a new one, or exit the application. This flexibility allows the DTRSC tool to support iterative testing and model refinement. After the analysis is completed, the results generated by the machine learning model are communicated back to the user of the DTRSC tool. Decisions derived from the predictive analysis can be shared with intended stakeholders, such as healthcare providers or city authorities, enabling real-time interventions.

```
Select the model input from the available Kafka streaming topics:
1. D1_Heart Rate
2. D2_Respiration Rate
3. Time
Select a model input (enter the number, or 'done' to finish): 1
Selected model input(s): D1_Temperature, D2_Oxygen Saturation, D1_Heart Rate

Select the model input from the available Kafka streaming topics:
1. D2_Respiration Rate
2. Time
Select a model input (enter the number, or 'done' to finish): 2
Selected model input(s): D1_Temperature, D2_Oxygen Saturation, D1_Heart Rate, Time

Select the model input from the available Kafka streaming topics:
1. D2_Respiration Rate
Select a model input (enter the number, or 'done' to finish): done

Select the target (output) variable from the remaining available topics:
1. D2_Respiration Rate
Select an output variable (enter the number): 1
Selected target variable: D2_Respiration Rate
```

FIGURE 8 Selection of input variables for ML model training.

```
Do you want to perform pre-processing? (Y/N): Y
Performing pre-processing...
Enter the percentage of data for training (0-100): 80
Using 80% of the data for training and 20% for testing.
Please choose the machine learning technique you want to use.
1. LSTM
2. Linear Regression
3. Gradient Boosting
4. Random Forest
5. Neural Network
Enter your choice: 5
Enter the number of layers: 5
Enter the number of neurons per layer: 64

Select an optimization algorithm for the Neural Network:
1. Adam
2. Stochastic Gradient Descent (SGD)
3. RMSprop
Enter your choice: 1

You selected model: Neural Network
Parameters for the model:
Number of Layers: 5
Neurons per Layer: 64
Optimization Algorithm: Adam

The R^2 score of the selected model is: 0.88
```

FIGURE 9 Selection of model, training, and evaluation.

Additionally, users with little background in machine learning and computer programming can leverage this tool to perform predictive analysis effectively.

5 | CASE STUDIES

This section highlights how the DTRSC tool's design provides authorities and users with responsive real-time functionality for managing and adapting to dynamic conditions in a smart city. By leveraging the proposed digital twin layered architecture, the DTRSC tool can support a multitude of applications that aim to improve citizens' quality of life. Here, we demonstrate the

ability of the tool to handle use cases analyzed using different machine-learning models and to combine their results in another use case built on the same data topics streamed to the big data platform.

The first use case focuses on forecasting air quality in smart cities, offering valuable insights for public health management and individual recreation. The second use case involves predicting an individual's respiratory condition based on vital signs collected from wearable devices, facilitating personalized health monitoring. The third use case combines the air quality forecast with the individual's respiratory condition to classify each situation as an "alert" or "no alert." These real-time alerts are generated by analyzing the individuals' vital signs alongside the air quality index (AQI) to assess whether the anticipated outdoor conditions are safe for them. To evaluate the system's effectiveness, we emulated the generation of real-time data from individuals' datasets and assessed the tool's accuracy in generating these alerts.

It is worth noting that these use cases represent examples or case studies. The tool design based on the proposed DT architecture can be adapted to implement various other applications.

5.1 | Case Study 1: AQI Forecasting in Smart Cities

As highlighted by the World Health Organization (WHO), outdoor air pollution remains a significant global public health concern, contributing to different respiratory diseases[40]. Given the impact of air pollution on health, particularly in urban areas, we chose air quality forecasting as the first use case for our study[41]. This use case demonstrates how the DTRSC tool can be used to monitor and forecast the AQI in real time, helping city authorities and individuals take preventive measures when air quality declines.

In this case study, we utilized a time-series dataset[42] containing real-time measurements of key air pollutants, including particulate matter (PM2.5 and PM10), Ozone (O3), Nitrogen Dioxide (NO2), Sulfur Dioxide (SO2), and Carbon Monoxide (CO), as well as meteorological variables such as temperature, humidity, and wind speed. These variables were used as input features for forecasting the AQI, which serves as the output variable. Using the emulation module of the DTRSC tool, we generated real-time data from the dataset. We augmented the dataset with the AQI values, which are calculated using (1) for each pollutant[43],

$$AQI_p = \frac{(I_{\text{high}} - I_{\text{low}})}{(C_{\text{high}} - C_{\text{low}})} \times (C_p - C_{\text{low}}) + I_{\text{low}} \tag{1}$$

where AQI_p represents the AQI for pollutant p, and C_p is the measured concentration of pollutant p. The terms C_{high} and C_{low} denote the breakpoint concentrations for the upper and lower bounds of the pollutant's concentration range that contains C_p. Similarly, I_{high} and I_{low} represent the upper and lower bounds of the AQI scale for pollutant p at the concentration levels C_{high} and C_{low}, respectively. This formula ensures AQI calculations align with EPA guidelines by basing values on pollutant concentrations[43].

The AQI for each pollutant in a given row of data is calculated individually using the formula above, whereas the overall AQI for that row is determined by selecting the maximum AQI value among all pollutants. This approach follows standard practices, where overall air quality is determined by the pollutant with the highest individual AQI value at any given time.

Indeed, the AQI categorizes air quality into different severity levels based on pollutant concentrations. Table 1 outlines the breakpoints for the AQI, following the Environmental Protection Agency (EPA) guidelines[43].

To forecast the AQI levels, we employed the long short-term memory (LSTM) model, which is known to have a unique architecture with input, forget, and output gates. These gates control what information is added, retained, or discarded, allowing the model to focus on meaningful patterns over longer time spans. This capability makes the LSTM model particularly effective for handling the complexities of AQI data, including seasonal trends, sudden fluctuations, and long-term variations[44]. To evaluate the effectiveness of the LSTM predictions, we applied several performance metrics implemented in the tool, which are discussed in detail in the sequel[45–47].

5.1.1 | Performance Metrics

5.1.1.1 | *Mean Absolute Error (MAE)*

The mean absolute error (MAE) measures the average magnitude of the errors between the actual values and the predicted values without considering the error sign. It is defined as

AQI Category	O_3 (ppm)	$PM_{2.5}$ ($\mu g/m^3$)	PM_{10} ($\mu g/m^3$)	NO_2 (ppb)	CO (ppm)	SO_2 (ppb)
Good (0-50)	0.000-0.054	0.0-12.0	0-54	0-53	0.0-4.4	0-35
Moderate (51-100)	0.055-0.070	12.1-35.4	55-154	54-100	4.5-9.4	36-75
Unhealthy for Sensitive Groups (101-150)	0.071-0.085	35.5-55.4	155-254	101-360	9.5-12.4	76-185
Unhealthy (151-200)	0.086-0.105	55.5-150.4	255-354	361-649	12.5-15.4	186-304
Very Unhealthy (201-300)	0.106-0.200	150.5-250.4	355-424	650-1249	15.5-30.4	305-604
Hazardous (301-500)	-	250.5-500.4	425-604	1250-2049	30.5-50.4	605-1004

TABLE 1 Breakpoints for AQI Calculation for Ozone (O_3), $PM_{2.5}$, PM_{10}, NO_2, CO, and SO_2.

$$\text{MAE} = \frac{1}{m}\sum_{j=1}^{m}\left|a_j - \hat{p}_j\right| \tag{2}$$

where m represents the total number of predictions, a_j is the actual value, and $\hat{p}_j$ is the predicted value[45,46]. This metric represents the prediction error by measuring the average absolute difference between actual and predicted values.

5.1.1.2 | *Root Mean Squared Error (RMSE)*

The root mean squared error (RMSE) is the square root of the mean squared error (MSE), providing a measure of error in the same units as the predicted variable. It can be obtained from

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(a_j - \hat{p}_j)^2} \tag{3}$$

where m represents the total number of predictions, a_j is the actual value, and $\hat{p}_j$ is the predicted value. This metric is particularly useful for understanding the standard deviation of prediction errors and is sensitive to larger errors[45,46].

5.1.1.3 | *Mean Absolute Percentage Error (MAPE)*

The mean absolute percentage error (MAPE) represents the prediction error as a percentage, providing a relative measure of the error. It can be calculated from

$$\text{MAPE} = \frac{1}{m}\sum_{j=1}^{m}\left|\frac{a_j - \hat{p}_j}{a_j}\right| * 100 \tag{4}$$

where m represents the total number of predictions, a_j is the actual value, and $\hat{p}_j$ is the predicted value. MAPE is often used to compare prediction performance across models, as it indicates the average percentage error, making it useful for evaluating relative accuracy[45–47].

5.1.1.4 | *R-Squared (R^2)*

The R^2 (coefficient of determination) measures the proportion of variance in the dependent variable that can be explained by the independent variables. It is expressed as

$$R^2 = 1 - \frac{\sum_{j=1}^{m}(a_j - \hat{p}_j)^2}{\sum_{j=1}^{m}(a_j - \overline{a})^2} \tag{5}$$

where a_j is the actual value, $\hat{p}_j$ is the predicted value, and $\overline{a}$ is the mean of the actual values. The R^2 value ranges from 0 to 1, with higher values indicating better model performance, as it shows how well the model explains the variability of the target variable[45–47].

Table 2 provides the results for the performance metrics of these models,

The Case Study 1 model results in Table 2 demonstrate the performance of the LSTM model for forecasting AQI with good accuracy while processing data directly without outlier handling and feature engineering techniques. Note that the LSTM model

Model	MAE	RMSE	R^2	MAPE
LSTM	0.7043	0.8541	0.88	0.2351

TABLE 2 Performance metrics of the LSTM model for AQI forecasting.

is used as an example, where a relatively short-term forecast is assumed for the purpose of public health awareness for outdoor activities. Other forecasting models can be implemented in the tool to provide long-term forecasting with sufficient accuracy.

The current results highlight the tool's capability to process smart city sensor data, showcasing its adaptability in handling unrefined data inputs. By leveraging historical data, the digital twin realized by the tool can provide reliable AQI forecasts, which are crucial for anticipating future air quality trends and supporting public health and environmental management efforts.

5.2 | Case Study 2: Monitoring Respiratory Conditions Using Wearable Devices

Wearable devices offer a convenient way to continuously monitor vital signs such as respiratory rate, which can be indicative of an individual's health status. With the increasing prevalence of smartwatches, fitness trackers, and smart rings, it is now possible to collect real-time data on respiratory rate, heart rate, blood oxygen levels, and other physiological parameters. The accessibility and ubiquity of these devices have made continuous health monitoring more practical and widespread, empowering individuals to take preventive steps in managing their health in a smart city setting where the city residents are always connected.

In this e-health-related case study, the proposed tool enables the early detection of potential health risks, such as the ones related to lung function[48]. We utilized data collected from daily wearable devices[49] to predict respiratory conditions based on an individual's respiratory rate. It is a vital sign that directly reflects the state of an individual's respiratory system. Abnormal respiratory rates can indicate various health issues, such as asthma, chronic obstructive pulmonary disease (COPD), or respiratory infections. If respiratory rate data is unavailable, it can be approximated from heart rate[50] by using

$$\text{Respiratory Rate} \approx \frac{\text{Heart Rate}}{4}. \tag{6}$$

In this case study, respiratory rates are classified into general risk categories based on the risk level for the individual's health[51], as shown in Table 3.

Risk Level	Respiratory Rate (breaths per minute)
Low Risk (Normal)	12-20
Moderate Risk (Elevated)	> 20
High Risk (Below Normal)	< 12
Critical (Severe Abnormalities)	Variable (with symptoms)

TABLE 3 Classification of respiratory rate by risk level.

For the predictive analysis of this case study, we selected skin temperature, heart rate, respiratory rate, and oxygen saturation as input features. We augmented the dataset with the individual's respiratory condition based on the risk level defined in Table 3. The classification models in DTRSC were applied for this purpose, showcasing the tool's capability to handle classification tasks effectively. The available classification models include Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), and Artificial Neural Networks (ANN), with the flexibility to integrate additional models as needed based on application requirements. The performance of these classification models for the used dataset was evaluated using several classification metrics available in the proposed tool. The key metrics[52] are summarized below.

5.2.1 | Key Metrics

5.2.1.1 | *Accuracy*

Accuracy is a measure of the proportion of correct predictions made by the model, combining both true positives (Tp) and true negatives (Tn) relative to the total cases.

$$\text{Accuracy} = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{7}$$

In (7), *Tp* (True Positives) represents cases where the model correctly identified positive instances, and *Tn* (True Negatives) denotes correctly identified negative cases. Conversely, *Fp* (False Positives) are instances incorrectly classified as positive, while *Fn* (False Negatives) are cases wrongly identified as negative.

5.2.1.2 | *Precision*

Precision indicates the proportion of positive predictions that were actually correct, showing the model's effectiveness in selecting relevant positive cases.

$$\text{Precision} = \frac{Tp}{Tp + Fp} \tag{8}$$

In (8), *Tp* refers to True Positives, which are correctly predicted positive cases, while *Fp* denotes False Positives, representing cases mistakenly predicted as positive by the model.

5.2.1.3 | *Recall*

Recall, also known as Sensitivity, measures the proportion of actual positive cases that the model correctly identified.

$$\text{Recall} = \frac{Tp}{Tp + Fn} \tag{9}$$

In (9), *Tp* represents True Positives, or the positive cases correctly identified by the model, and *Fn* represents False Negatives, or cases where the model failed to identify actual positives.

5.2.1.4 | *F1-Score*

The F1-Score provides a balanced metric by taking the harmonic mean of Precision and Recall, especially useful in scenarios with imbalanced datasets. It is calculated as follows:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

5.2.1.5 | *AUC-ROC*

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) assesses the model's effectiveness across various thresholds. The ROC curve is a plot of Sensitivity *Se* versus the False Positive Rate *Fpr*.

$$Se = \frac{Tp}{Tp + Fn} \tag{11}$$

$$Fpr = \frac{Fp}{Fp + Tn} \tag{12}$$

In (11), *Se* is equivalent to the True Positive Rate, while *Fpr* in (12) represents the proportion of negative instances incorrectly classified as positive. Here, *Tp* denotes True Positives, *Fn* indicates False Negatives, *Fp* refers to False Positives, and *Tn* represents True Negatives. The AUC value provides a single measure summarizing the area under the ROC curve, with higher values indicating better model performance.

The following Table 4 provides the results for the performance metrics of the second case study.

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	0.9437	0.9435	0.9438	0.9436	0.9863
Random Forest	0.9319	0.9312	0.9319	0.9301	0.9728
Gradient Boosting	0.9475	0.9469	0.9475	0.9471	0.9839
Artificial Neural Networks	0.9363	0.9369	0.9363	0.9365	0.9817

TABLE 4 Performance of different ML Models for predicting respiratory conditions.

As shown in Table 4, the DTRSC tool allows users to evaluate and compare machine learning models such as Logistic Regression, Random Forest, Gradient Boosting, and Artificial Neural Networks, each assessed across metrics like accuracy, precision, recall, F1-score, and AUC-ROC. To enhance these metrics, we appended additional emulated data[53], achieving improved results with our classification models. This capability enables users to select the most suitable model based on

performance, facilitating accurate predictions for conditions like respiratory health issues. Table 4 shows Gradient Boosting achieves the highest accuracy (0.9475) and F1-score, followed closely by Logistic Regression. Random Forest and Artificial Neural Networks also perform well, with AUC-ROC values above 0.97, indicating strong classification capabilities across all models. With its intuitive, CLI-based interface, the DTRSC tool makes machine learning analytics accessible not only to technical users but also to city authorities, healthcare providers, and others without extensive machine learning expertise.

5.3 | Case Study 3: Outdoor Health Hazard Warning

This use case demonstrates the capability of the DTRSC tool to integrate the usage of the models of other use cases and their underlying datasets managed by the Big Data Handling module into a new use case. Integrating air quality forecasting with wearable health data via the DTRSC tool advances the ability to create a robust health management system for smart cities. By combining environmental and personal health data in real-time, this case study offers individuals tailored health alerts whenever their physiological metrics and surrounding environmental conditions reach critical levels.

Empirical studies[41,54] demonstrate a strong link between air quality and respiratory health, underscoring the importance of monitoring air quality alongside individual respiratory indicators.

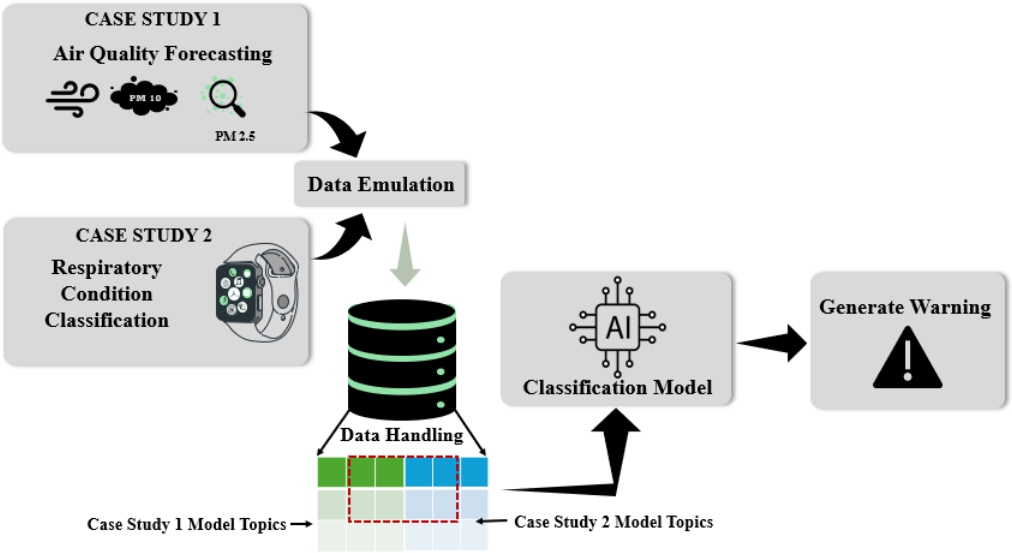

FIGURE 10 Outdoor Health Hazard Warning

Empirical studies[41,54] demonstrate a strong link between air quality and respiratory health, underscoring the importance of monitoring air quality alongside individual respiratory indicators. Thus, in this use case, as illustrated in Fig. 10, we create a synthetic time series dataset that merges air quality data with wearable health metrics, emulating simultaneous data capture from both sources. This dataset provides a foundation for testing the model's ability to issue personalized health alerts that consider both environmental and individual health factors. Here, we consider the previously mentioned case studies, with their data being emulated and transferred via the Data Emulation and Data Transfer modules of the DTRSC tool and then streamed into the Big Data Handling module. As seen in Fig. 10, model topics are created for each case study and then aligned using timestamps generated by the Data Handling module. The DTRSC tool's classification model processes these combined topics effectively, generating timely warnings (alert) or no alert status based on the integrated data considering the respiratory rate and AQI index.

Table 5 presents the performance metrics of the third case study.

The results obtained from the DTRSC tool and presented in Table 5 show the performance of various machine learning models in generating accurate health warnings. Both Logistic Regression and Random Forest models achieve high accuracy, precision, recall, and F1-scores of 0.9725, indicating strong, balanced performance in prediction reliability. Gradient Boosting slightly outperforms these with accuracy and an F1-score of 0.9763, making it particularly effective in this application. Although

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	0.9725	0.9724	0.9725	0.9724	0.9963
Random Forest	0.9725	0.9724	0.9725	0.9724	0.9936
Gradient Boosting	0.9763	0.9761	0.9763	0.9761	0.9882
Artificial Neural Networks	0.9588	0.9592	0.9588	0.9589	0.9931

TABLE 5 Performance of different ML models for outdoors health warning use case.

Artificial Neural Networks have a slightly lower accuracy of 0.9588, they still demonstrate strong performance, especially with an AUC-ROC of 0.9931, underscoring their ability to distinguish between alert and no-alert cases. Notably, all models achieve AUC-ROC values above 0.98, with Logistic Regression leading at 0.9963, indicating excellent classification capability across the models. These results suggest that each model is well-suited for generating timely and accurate health warnings, with Gradient Boosting providing a slight advantage in prediction accuracy and overall consistency.

6 | CONCLUSION AND FUTURE DIRECTIONS

This paper introduces a simple, layered digital twin architecture tailored for smart cities. The proposed architecture breaks down a smart city digital twin functionality into six layers, which allow scalable real-time data emulation, data transfer, data ingestion, intelligent analytics, decision-making support, and portable interaction for users or authorities.

The digital twin architecture is envisioned through the proposed open-source Digital Twin Realization for Smart City (DTRSC) tool, which supports scalable, real-time data generation, handling, and analysis. The tool can handle real-time data from physical devices and features a built-in data emulation module that generates correlated realistic data from virtual devices via a single or batch device network configuration. The tool echoes the proposed architecture by streaming the digital twin data to a big data platform that keeps track of the historical data, enabling longitudinal analysis. Moreover, the tool allows machine learning training and testing via multiple available techniques with easy extension to others, while decision support outcomes can be obtained through a simple user interface.

The practical capabilities of the DTRSC tool have been demonstrated through three case studies focusing on the role of digital twins in enhancing public welfare. The first case, air quality monitoring, uses regression models to assess air quality index levels, providing real-time insights into a smart city's environmental conditions. In the second case, respiratory e-health data from wearable devices is assumed to be transmitted to the digital twin, received, and processed using classification models to assess individuals' respiratory conditions. The third case integrates air quality and wearable e-health metrics data to implement a warning system that can offer a smart alert to city residents based on their health metrics and current AQI levels, showcasing the DTRSC tool's potential in real-world smart city applications that combine environmental and eHealth data. These case studies also illustrate that authorities or government officials can operate the DTRSC tool, allowing users with limited programming knowledge to utilize its functionality.

Despite these advancements, certain limitations remain to be addressed in future work. The current implementation of the interaction layer in the DTRSC tool supports only human interaction. In the future, we target making this interaction with cyber-physical systems or devices to allow the digital twin to enhance the operation of these systems.

AUTHOR CONTRIBUTIONS

N. Jayachandran contributed to developing the digital twin architecture and the proposed tool. She also contributed to paper writing. A. Abdrabou contributed to the design of the proposed architecture and tool. He also participated in writing the paper. M. Al Bataineh and K. Noordin contributed to reviewing and writing the paper.

ACKNOWLEDGMENTS

This work was made possible by the UAE University AUA grant 12N143.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

REFERENCES

1. Deng T, Zhang K, Shen ZJM. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. Journal of Management Science and Engineering. 2021 6;6:125–134.

2. Nikolskaya K, Ivanov S, Radchenko G, Sokolinsky L, Zymbler M. Digital Twin of city: Concept overview. In: 2020 Global Smart Industry Conference (GloSIC). IEEE; 2020. p. 178–186.
3. Mukhacheva AV, Ugryumova MN, Morozova IS, Mukhachyev MY. Digital Twins of the Urban Ecosystem to Ensure the Quality of Life of the Population. In: International Scientific and Practical Conference Strategy of Development of Regional Ecosystems "Education-Science-Industry"(ISPCR 2021). Atlantis Press; 2022. p. 331–338.
4. Botín-Sanabria DM, Mihăiţă AS, Peimbert-García RE, Ramírez-Moreno MA, Ramírez-Mendoza RA, J Lozoya-Santos Jd. Digital Twin Technology Challenges and Applications: A Comprehensive Review. Remote Sensing. 2022;.
5. Campo Gd. Virtual Reality and Internet of Things Based Digital Twin for Smart City Cross-Domain Interoperability. Applied Sciences. 2024;.
6. Dembski F, Wössner U, Letzgus M, Ruddat M, Yamu C. Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany. Sustainability. 2020;.
7. Shen S. Construction of Smart Transportation City System Based on Digital Twins. 2024;.
8. Rantanen T. Open Geospatial Data Integration in Game Engine for Urban Digital Twin Applications. Isprs International Journal of Geo-Information. 2023;.
9. Shamlitsky Y. Using Digital Twins to Manage Traffic Flows. E3s Web of Conferences. 2024;.
10. Kamel Boulos MN, Zhang P. Digital Twins: From Personalised Medicine to Precision Public Health. Journal of Personalized Medicine. 2021;.
11. Hämäläinen M. Urban Development With Dynamic Digital Twins in Helsinki City. Iet Smart Cities. 2021;.
12. D'Hauwers R, Walravens N, Ballon P. From an Inside-in Towards an Outside-Out Urban Digital Twin: Business Models and Implementation Challenges. Isprs Annals of the Photogrammetry Remote Sensing and Spatial Information Sciences. 2021;.
13. Taylor JE, Bennett G. Engineering Smarter Cities With Smart City Digital Twins. Journal of Management in Engineering. 2021;.
14. Brucherseifer E, Winter H, Mentges A, Mühlhäuser M, Hellmann M. Digital Twin Conceptual Framework for Improving Critical Infrastructure Resilience. At - Automatisierungstechnik. 2021;.
15. Laamarti F, Badawi HF, Ding Y, Arafsha F, Hafidh B, Saddik AE. An ISO/IEEE 11073 Standardized Digital Twin Framework for Health and Well-Being in Smart Cities. Ieee Access. 2020;.
16. Nijeweme drHollosy WO, Velsen Lv, Henket A, Hermens H. An Interoperable eHealth Reference Architecture for Primary Care. 2018;.
17. Allam Z, Jones DS. Future (post-COVID) digital, smart and sustainable cities in the wake of 6G: Digital twins, immersive realities and new urban economies. Land Use Policy. 2021 2;101.
18. Wang H, Chen X, Jia F, Cheng X. Digital twin-supported smart city: Status, challenges and future research directions. Expert Systems with Applications. 2023 5;217.
19. Jeddoub I, Nys GA, Hajji R, Billen R. Digital Twins for cities: Analyzing the gap between concepts and current implementations with a specific focus on data integration. International Journal of Applied Earth Observation and Geoinformation. 2023 8;122.
20. Schrotter G, Hürzeler C. The Digital Twin of the City of Zurich for Urban Planning. PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science. 2020 2;88:99–112.
21. Attaran M, Celik BG. Digital Twin: Benefits, use cases, challenges, and opportunities. Decision Analytics Journal. 2023 3;6.
22. Ariyachandra MF, Wedawatta G. Digital Twin Smart Cities for Disaster Risk Management: A Review of Evolving Concepts. Sustainability. 2023;.
23. Ziehl M. Transformative Research in Digital Twins for Integrated Urban Development. International Journal of E-Planning Research. 2023;.
24. Vempati S. Securing Smart Cities: a Cybersecurity Perspective on Integrating IoT, AI, and Machine Learning for Digital Twin Creation. Jes. 2024;.
25. Astarita V. Risk Reduction in Transportation Systems: The Role of Digital Twins According to a Bibliometric-Based Literature Review. Sustainability. 2024;.
26. Huang Y, Peng H, Sofi M, Zhou Z, Xing T, Ma G, et al. The City Management Based on Smart Information System Using Digital Technologies in China. Iet Smart Cities. 2022;.
27. Ravid BY, Aharon-Gutman M. The Social Digital Twin:The Social Turn in the Field of Smart Cities. Environment and Planning B: Urban Analytics and City Science. 2022;.
28. Nochta T, Wan L, Schooling J, Parlikad AK. A Socio-Technical Perspective on Urban Analytics: The Case of City-Scale Digital Twins. Journal of Urban Technology. 2020;.
29. Shaharuddin S, Abdul Maulud KN, Rahman S, Che Ani AI. Digital Twin for Indoor Disaster in Smart City: A Systematic Review. The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences. 2022;.
30. Ford DN, Wolf CM. Smart Cities With Digital Twin Systems for Disaster Management. Journal of Management in Engineering. 2020;.
31. Yeon H, Eom T, Jang K, Yeo J. DTUMOS, Digital Twin for Large-Scale Urban Mobility Operating System. Scientific Reports. 2023;.
32. der Ven Jv, den Bemt Bv, Dijk Lv, Opdam M, Haegens LL, Vriezekolk JE, et al. Preferences of Patients With Musculoskeletal Disorders Regarding the Timing and Channel of eHealth and Factors Influencing Its Use: Mixed Methods Study. Jmir Human Factors. 2023;.
33. Huang PH, Kim KH, Schermer M. Ethical Issues of Digital Twins for Personalized Health Care Service: Preliminary Mapping Study. Journal of Medical Internet Research. 2022;.
34. Ferko E, Bucaioni A, Behnam M. Architecting Digital Twins. Ieee Access. 2022;.
35. Qian C, Liu X, Ripley C, Qian M, Fan L, Yu W. Digital Twin—Cyber Replica of Physical Things: Architecture, Applications and Future Research Directions. Future Internet. 2022;.
36. Tekinerdoğan B, Verdouw CN. Systems Architecture Design Pattern Catalog for Developing Digital Twins. Sensors. 2020;.
37. Kantawong K, Chaichumpa S, Pravesjit S, Yaibuates M. A Lightweight Framework for Retrieve IP Device Status Based on MQTT Protocol. In: 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON). IEEE; 2022. p. 46–49.
38. KimiNewt. PySharkhttps://pypi.org/project/pyshark/ Accessed 12 July 2024.
39. Foundation AS. Appache Kafkahttps://kafka.apache.org/ Accessed 10 August. 2024.
40. World Health Organization. Ambient Air Pollution: A Global Assessment of Exposure and Burden of Disease 2024. Available at: https://www.who.int/health-topics/air-pollution#tab=tab_1.
41. Marino E, Caruso M, Campagna D, Polosa R. Impact of air quality on lung health: myth or reality? Therapeutic Advances in Chronic Disease. 2015;6(5):286–298.
42. Zheng Y, Yi X, Li M, Li R, Shan Z, Chang E, et al. Forecasting fine-grained air quality based on big data. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2015. p. 2267–2276.

43. U S Environmental Protection Agency. 2024. Technical Assistance Document for the Reporting of Daily Air Quality – the Air Quality Index (AQI). Research Triangle Park, NC: Office of Air Quality Planning and Standards, Air Quality Assessment Division. EPA-454/B-24-002Available from: https://www.airnow.gov/publications/air-quality-index/technical-assistance-document-for-reporting-the-daily-aqi/.
44. Yadav A, Jha CK, Sharan A. Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science. 2020;167:2091–2100. International Conference on Computational Intelligence and Data Science. Available from: https://www.sciencedirect.com/science/article/pii/S1877050920307237.
45. Dubey AK, Kumar A, García-Díaz V, Sharma AK, Kanhaiya K. Study and analysis of SARIMA and LSTM in forecasting time series data. Sustainable Energy Technologies and Assessments. 2021;47:101474.
46. Tatachar AV. Comparative assessment of regression models based on model evaluation metrics. International Journal of Innovative Technology and Exploring Engineering. 2021;8(9):853–860.
47. Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science. 2021;7:e623.
48. Liaqat D, Abdalla M, Abed-Esfahani P, Gabel M, Son T, Wu R, et al. WearBreathing: Real-World Respiratory Rate Monitoring Using Smartwatches. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2019;3(2):1–22.
49. Reddy M. Wearables Dataset 2023. Accessed: 2024-07-04. Available from: https://www.kaggle.com/datasets/manideepreddy966/wearables-dataset?select=personal_health_data.csv.
50. Garmin. The Critical Relationship Between Respiratory Rate, Heart Rate, and CadenceAccessed: 2024-08-02. Available from: https://www.garmin.com/en-PH/blog/critical-relationship-between-respiratory-rate-heart-rate-and-cadence/.
51. Ra HK, Salekin A, Yoon HJ, Kim J, et al. Asthmaguide: an asthma monitoring and advice ecosystem. In: 2015 IEEE Wireless Health (WH). IEEE; 2015. p. 1–8.
52. Vakili M, Ghamsari M, Rezaei M. Performance analysis of machine learning models for IoT data classification. arXiv preprint arXiv:220209636. 2022;.
53. Bommela NR. Health Monitoring System Dataset n.d. Accessed: [2024-10-10]. Available from: https://www.kaggle.com/datasets/nraobommela/health-monitoring-system?select=what-is-a-normal-respiratory-rate-2248932-v1-5c1abe6846e0fb0001c6284a.png.
54. Götschi T, Heinrich J, Sunyer J, Künzli N. Long-term effects of ambient air pollution on lung function: a review. Epidemiology. 2008;19(5):690–701.