Cloud-based Digital Twin Framework and IoT for Smart Emergency Departments in Hospitals

Haider Q. Mutashar
Department of Computer Engineering, Mustansiriyah University, Baghdad, Iraq
haider.aasim@uomustansiriyah.edu.iq

Sawsan M. Mahmoud
Department of Computer Engineering, Mustansiriyah University, Baghdad, Iraq
sawsan.mahmoud@uomustansiriyah.edu.iq

Hiba A. Abu-Alsaad
Department of Computer Engineering, Mustansiriyah University, Baghdad, Iraq
eng.hibaakram@uomustansiriyah.edu.iq (corresponding author)

Received: 19 January 2025 | Revised: 15 February 2025 | Accepted: 27 February 2025

 DOI: https://doi.org/10.48084/etasr.10290

ABSTRACT

Emergency response time has always been one of the most important factors in saving lives, yet many healthcare systems struggle with delays and resource inefficiencies. This paper proposes the use of cloud-based digital twin technology, integrated with an Internet of Things (IoT) hub, to enhance patient care and optimize emergency department workflows. The system utilizes an Azure IoT virtual hub and an Azure Digital Twins model for real-time data transmission and processing of bed availability and capacity in a hospital emergency department. Azure Digital Twins can create virtual models of any physical environment, such as an emergency department, which is essential for creating an automated decision-making system using advanced and modern technologies. Unlike previous solutions that lack coherence in resource allocation systems or real-time decision-making mechanisms, our system dynamically updates emergency department bed availability in real-time through embedded system devices in hospitals, and makes automated decisions based on both patient location and resource availability. The system is implemented using a mobile application and validated using a case study. The case study includes data from multiple emergency departments in hospitals within a specific area. The system provides the best options for emergency patients based on the hospital location and bed availability.

Keywords-Microsoft Azure; digital twins; cloud computing; emergency healthcare; IoT; smart emergency

I. INTRODUCTION

The Internet of Things (IoT), a network of interconnected devices that collect and transmit data in real-time, is revolutionizing the way devices communicate and interact, facilitating seamless integration between the physical and digital realms. By incorporating sensors, actuators, and connectivity into common objects, the IoT enables the real-time data acquisition, surveillance, and regulation [1]. This collection of interconnected devices facilitates a wide range of applications, from domestic automation to industrial processes, and is establishing itself as a fundamental component in the development of more intelligent and efficient systems [2].

Smart healthcare environments employ IoT technology to construct spaces that dynamically adapt to the requirements of their occupants. By utilizing the integration of devices, sensors, and software, these environments optimize energy utilization, enhance safety measures, and enrich the user experience [3, 4]. Many researchers are using IoT for a large range of applications like security, emergency services, traffic congestion, structural health, healthcare, industrial control, and many others. For example, smart hospitals can use IoT to monitor patient conditions, manage equipment utilization, and automate essential operations, thereby improving healthcare outcomes and reducing operational costs and inefficiencies [5]. The independent adaptation of IoT is challenging due to its scaling limitations, so in robust and modern structures, cloud computing plays a crucial role in supporting healthcare systems that rely on IoT and smart environments by offering scalable and on-demand computing resources [6]. By acting as a platform for storing, analyzing, and processing large amounts of data, cloud technology ensures the efficient operation of IoT devices, allowing them to provide actionable insights [7]. Additionally, cloud computing facilitates collaboration and innovation by supporting advanced technologies like artificial intelligence, machine learning, and other useful tools discussed in the following sections, which are essential for making intelligent decisions across various fields [8].

In modern healthcare practice, effective management of patients within Emergency Departments (EDs) is critical to patient health, especially for high-demand cases in multiple healthcare facilities [9]. Traditional systems frequently encounter challenges in managing real-time resource allocation and decision-making during emergencies, leading to potential delays in treatment and undesired outcomes [10]. However, this study proposes a centralized platform that leverages digital twin technology, a virtual representation of a physical entity, or a system that continuously updates in real-time, to utilize the emergency response capabilities of multiple EDs. Multiple EDs of the same geographical area are represented as virtual copies of their physical counterparts within a unified emergency center. The system can provide real-time monitoring, resource management, decision-making, and emergency call handling, with digital twin technology serving as the backbone of its structure. We used a digital twin to manage and monitor patient flow during emergencies. In general, digital twins can be developed at various levels within emergency healthcare, including patients, hospital staff, ED equipment, and the ED itself. The application of digital twin technology in emergency healthcare offers significant advantages, as it can minimize errors, inefficiencies, and costs within healthcare systems [8, 11]. The development of a digital twin-based system for an ED requires several preliminary steps. The first step is the collection of comprehensive data concerning the physical counterpart [12]. This includes, but is not limited to, details such as the name, location, bed availability, staffing, and schedules. The accuracy of the digital twin model improves as more data are collected. After gathering and organizing the required data, selecting a suitable platform to host our model is crucial. There are various software solutions for implementing a digital twin model, some of which offer cloud-based platforms, such as Azure Digital Twins [13], AWS IoT TwinMaker [14], and Google Cloud IoT Core combined with Partner Solutions [15]. In this paper, Microsoft Azure Digital Twins was used to implement the proposed system for several reasons. First, it supports the Digital Twin Definition Language (DTDL), a standardized modeling framework that simplifies the modeling process. Furthermore, as a cloud-native technology, it offers a scalable solution, rendering it suitable for real-world applications, including smart cities, healthcare systems, and industrial IoT. Additionally, it provides seamless integration with other Azure services, such as Azure IoT Hub, a managed cloud service that facilitates secure communication between IoT devices and cloud applications, and Azure Function App, a serverless cloud computing service that enables event-driven execution of code without requiring infrastructure management, which are essential for the implementation of our system. It is also particularly suitable for real-time monitoring use cases [16]. However, Microsoft Azure is not the only option for the implementation of our system, other service providers with similar tools such as AWS or Google can be adapted. An advantage of our proposed system is that the protocols have been carefully selected to achieve an optimal balance between efficiency, security, and reliability, thereby ensuring smooth data flow across all system components. Furthermore, a ranking equation was formulated to account for both bed occupancy and estimated arrival time to offer the optimal choice for the user. Additionally, the Azure Digital Twins Explorer is employed to visualize and monitor system performance. This tool provides real-time updates on hospital status and resource availability, enabling administrators to make informed, data-driven decisions. Moreover, our smart ED framework uses a mobile application to assist patients in accessing hospital emergency services. Through this application, patients can get more information about hospitals, saving time and potentially saving lives.

II. RELATED WORK

The concept of the digital twin has received significant attention the healthcare sector within due to its potential to improve patient care [11]. In the context of healthcare, digital twins are primarily used to model individual patients, healthcare facility operations, or medical devices. The utilization of digital twins is progressively being used to develop individualized patient models by merging data from Electronic Health Records (EHRs), wearable devices, and imaging technologies. For example, authors in [17] demonstrated the application of digital twins in the domain of cardiovascular care, where simulations tailored to individual patients assisted in the diagnosis and treatment of heart diseases.

Recent advancements, such as in [18], have underscored the potential of digital twins in personalized healthcare, facilitating the emergence of human digital twins augmented with mobile Artificial Intelligence-Generated Content (AIGC) to transform treatment planning and monitoring. Authors in [8] employed digital twins to provide a storage solution that aims to improve emergency healthcare cases by delivering real-time access to patient data during emergencies. However, both lack emphasis on resource management and IoT integration. Authors in [19] proposed an IoT-based digital twin solution for real-time monitoring of patient vital signs via an Android application. While the study demonstrates the potential of digital twins for real-time monitoring, it lacks a framework for resource allocation and decision-making. Authors in [20] integrated a comprehensive smart hospital emergency system with a global positioning system and a smartphone application; however, it lacks cloud service integration to ensure scalability.

Within the realm of medical device engineering, digital twins are instrumental in the processes of prototyping, testing, and real-time monitoring. Digital twins of medical devices have the ability to predict maintenance needs and enhance device reliability, thereby ensuring patient safety. For instance, authors in [21] employed digital twins to monitor the performance of medical devices, enabling prompt maintenance and minimizing downtime. Digital twins also have the ability to improve hospital operations by simulating workflows and forecasting the efficient management of resources. For example, authors in [22] highlighted how digital twins can enable personalized treatment plans, predictive analytics, and preventive interventions by leveraging data from EHRs, wearables, and IoT devices. Additionally, digital twins can improve clinical workflows, resource allocation, and emergency response training by providing realistic simulations and real-time monitoring. However, these studies primarily focus on individual patient care and either don't have a resource allocation framework or don't mainly focus on decision-making in EDs.

Several studies have addressed healthcare systems related to bed occupancy, regardless of the implementation of digital twin technology. For instance, [23] proposed a real-time bed availability monitoring system to support users through a website application, in addition to a desktop application for administrative access. Although this study has a resource allocation framework, it lacks a good decision-making mechanism and doesn't support cloud services. Authors in [24] proposed a data-driven bed assignment methodology that seeks to optimize the balance between admission times and overflow in hospital wards, thereby maximizing compliance with mandatory patient care targets and minimizing reliance on non-primary beds. Authors in [25] developed a mobile Bed Management Mobile Application (BMMA) to enhance PC-based systems and provide convenience and mobility to hospital staff. Although users found it beneficial and promising, they also expressed concerns regarding data usage and internet connectivity. Authors in [26] presented an IoT-based smart-bed system leveraging deep learning, smart sensors, and computer vision for efficient patient monitoring and posture detection. However, unlike these studies, our research focuses on developing a centralized platform that leverages digital twin technology and an IoT-based cloud system to enhance emergency response capabilities and manage bed occupancy in hospital EDs.

III. DATASET

The dataset used in this study for testing purposes consists of data from 252 hospitals in California, USA. It originally contained 17 fields per hospital covering various departments. The dataset was intentionally sourced from hospitals in a specific region and does not include personally identifiable patient information is included to protect privacy. However, to align with the study's focus, the data were carefully cleaned and filtered to include only information related to EDs. The final dataset retains key attributes relevant to emergency response, including hospital identification number, name, geographic location (latitude and longitude), and ED bed capacity. Other fields were excluded as they were not directly relevant to bed occupancy, which is the core parameter used for decision-making in our system. This dataset is available online [27]. Examples of the data used are presented in Table I. One example instance represents Almeda hospital with an identification number of 106010735. The hospital's ED has a capacity of 12 beds and its geographic location is specified by the coordinates 37.76266 (latitude) and -122.254 (longitude).

IV. PROPOSED SYSTEM IMPLEMENTATION

The proposed system serves as a centralized platform aimed at optimizing patient flow among EDs of distributed hospitals within the same region. At its core, the system manages a virtual emergency center by employing digital twins of various EDs. This center increases the efficiency of managing emergency calls, enables real-time monitoring of hospital resources, supports decision-making processes related to patient allocation, and ensures continuous performance evaluation. As shown in Figure 1, the emergency center serves as the backbone of the system, hosting digital twins from nearby hospital EDs. It receives real-time status updates from IoT devices and performs the necessary computations to identify the optimal choice based on the received data. The emergency center then communicates the optimal solution to the user or updates the status of the digital twins in the monitoring software accordingly.

TABLE I. SAMPLE DATA FROM THE DATASET USED FOR TESTING

Facility ID	Facility name	ED stations	Latitude	Longitude
106010735	Alameda hospital	12	37.76266	-122.254
106010739	Alta Bates summit medical center – Alta Bates campus	22	37.85645	-122.257
106010776	UC San Francisco Benioff children's hospital Oakland	40	37.83722	-122.267
106010846	Highland hospital	65	37.79925	-122.231
06010937	Alta Bates summit medical center	30	37.82106	-122.263

A. System Architecture

The system consists of several interconnected components. As illustrated in Figure 2, certain components are integrated within Microsoft Azure tools, including Azure Digital Twins, Azure IoT Hub, and Azure Functions, whereas other components operate as independent development tools and services, such as OpenStreetMap and Flutter. Specifically, the system components are as follows:

1. Azure Digital Twins: This is the basic framework of the system where digital twin representations of hospital EDs are constructed. Each digital twin includes attributes such as hospital name, geographical coordinates (latitude and longitude), total bed capacity, available beds, and ED status. Models of individual beds are also included to monitor their availability in real time. The digital twin models of the beds and EDs are developed using the DTDL, which is a JSON-LD-based used to describe digital twin models of smart devices, assets, spaces, and environments [13].
2. Azure IoT Hub: This acts as a bridge between IoT devices and the cloud, receiving data from sensors deployed in hospitals. The data, such as bed occupancy status, is forwarded to Azure Digital Twins for processing and visualization [28].
3. Azure Function: A serverless computing component processes incoming IoT data, performs calculations such as travel time estimation, handles emergency application requests, and updates the digital twin models [29].
4. Flutter mobile application: Flutter is an open-source framework that facilitates rapid application development across multiple platforms from a unified codebase. This feature makes it particularly suitable for the creation of

visually appealing and responsive systems, such as applications for real-time emergency response. The user-oriented application enables patients or emergency responders to initiate requests for assistance by delivering a ranked list of hospitals based on proximity, bed availability, and other relevant factors [30].

5. OpenStreetMap: This is an open-source cost-efficient platform that offers global coverage. It is integrated to calculate distances and estimated travel times between the patient's location and nearby hospitals. Its cost-efficiency and integration with open-source tools provide flexibility and reliability, particularly in specialized applications like emergency response [31].

B. Digital Twin Model Design

To ensure a comprehensive representation, two primary digital twin models were developed using DTDL, as follows.

1) Bed Model

- ID: Each bed is assigned a unique identifier, which for simplicity begins with the prefix "bed," followed by the hospital identifier, and then the specific identifier of the bed. An instance of a bed's identifier can be represented as "bed-12345-0."
- Availability: A Boolean property that indicates whether the bed is currently available.

2) Emergency Department Model

Name: Represents the name of the hospital.

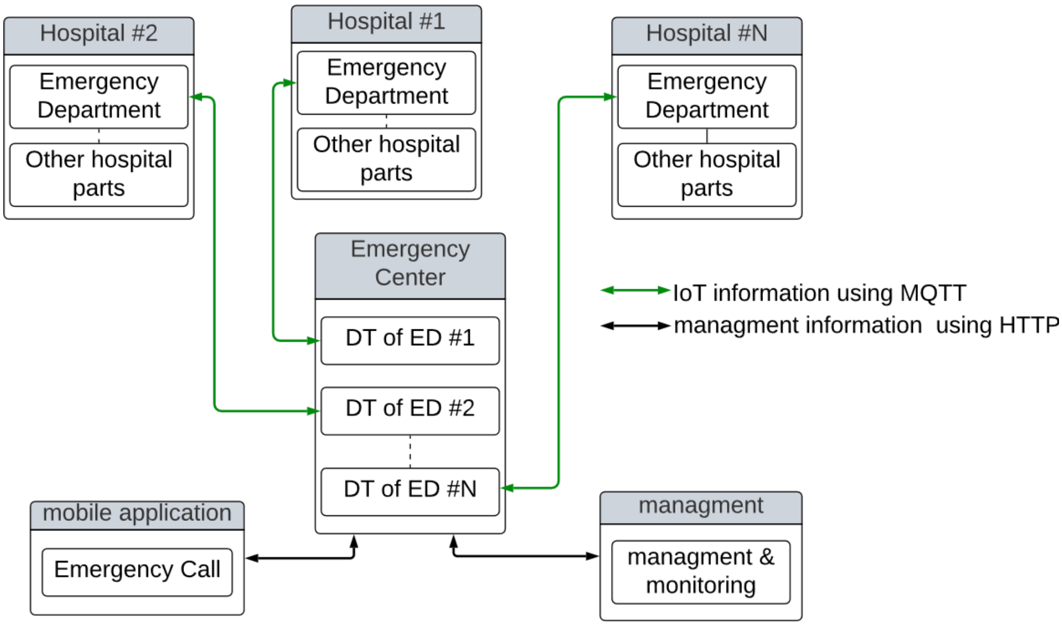


Fig. 1. Proposed system overview.

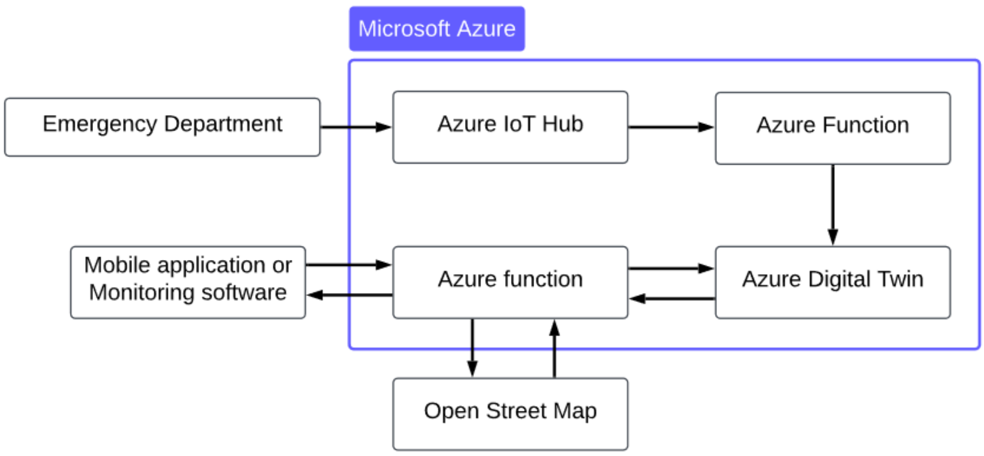


Fig. 2. Proposed system components.

- Location: Contains the hospital's geographic coordinates, including latitude and longitude values.
- Total beds: The total number of beds available in the ED.
- Available beds: Dynamically calculated to reflect the current number of unoccupied beds based on bed availability updates.
- Has bed: This is a relational attribute that facilitates the connection of a specific bed to a specific ED. This relationship simplifies the identification of which bed instance is associated with each ED instance, thereby aiding in the determination of the number of available beds within the ED.

C. *IoT Data Integration*

IoT devices play a key role in the system by providing real-time updates. The process is as follows.

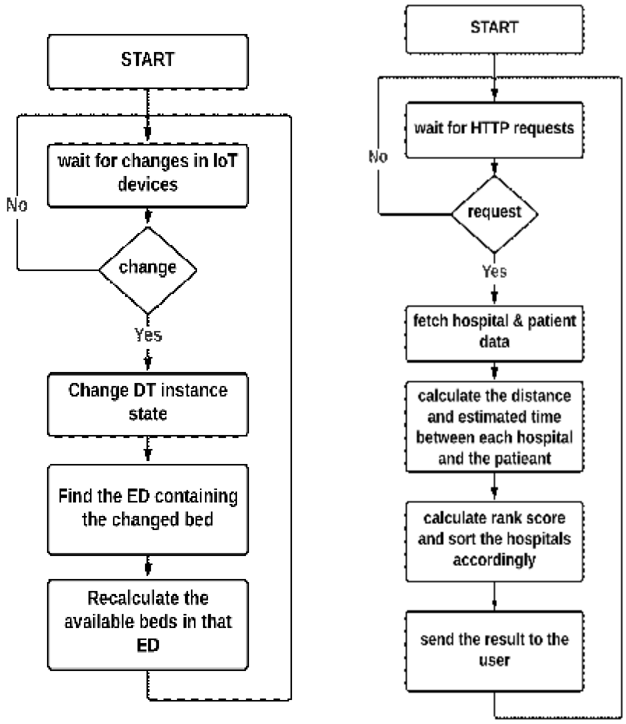
1) *Data Collection*

Sensors monitor bed occupancy and other environmental readings and send data to the IoT hub. Our work used Python code to simulate bed availability status, with each bed connected to a device instance in the IoT hub via a unique connection string. When the bed availability changes, the bed availability details are sent to the Azure IoT Hub, triggering the first Azure function.

2) *Data Processing*

Data processing within our system is divided based on the type of trigger that initiates the function. Two Azure Functions have been implemented:

- An event-triggered function: This function is activated by an event, specifically detecting changes in bed occupancy sensors. It is responsible for updating the digital twin instance based on the sensor readings, as well as changing the number of available beds within the ED instance utilizing the previously established 'hasbed' relationship. It verifies which ED owns this relationship and, after obtaining the ED ID, calculates the number of available beds linked by the same relationship. Figure 3(a) illustrates the flowchart of the event-triggered function.
- HTTP-triggered function. This function is activated whenever an HTTP request is made to its URL, which is provided by the Azure platform upon deployment of the function. The function is designed to manage HTTP requests from mobile application users that provide emergency case information, particularly their location. Utilizing these data, along with the information from digital twin instances of nearby hospitals, the function computes the distance and estimated arrival time between the emergency case and the ED locations using OpenStreetMap. It then uses this information, along with the data on available beds from the ED instance, to calculate a rank score using (1). The rank score is used to rank the hospitals, after which a list of hospitals with the highest rank scores is sent back to the HTTP request source. Figure 3(b) illustrates the flowchart of the HTTP-triggered function.



(a) (b)

Fig. 3. Azure Functions flowcharts: (a) event-triggered function, (b) HTTP-triggered function.

Both functions were implemented and tested locally using C# before being deployed to the Azure framework. Using the Azure portal, each function is then linked to its triggering type.

3) *Twin Updates*

Since the details of the digital twin instances are initiated from the dataset, the real-time updates are performed using the event-triggered function. The processed data, update the corresponding digital twin models, ensuring real-time synchronization.

D. *System Workflow*

The system is designed to assist in emergency situations by recommending the most appropriate hospital. Figure 4 shows the complete system architecture; as well as the interaction and data flow between the system components. Locally, Python was used to configure the IoT devices and digital twins, as well as to simulate the bed status. Azure Digital Twins Explorer was employed to monitor the state of the digital twins. The system workflow includes the following steps.

First, a patient or responder uses the Flutter app to share their location by pressing an emergency button in the app; a screenshot of the app is shown in Figure 5. The first screenshot shows the home page with an emergency button and at the top is the user's geographical information (if GPS is enabled). The other screenshot shows a list of hospital recommendations based on the following process: The app automatically initiates an HTTP request with the user's location to the Azure platform, which activates the HTTP-triggered function. The system then retrieves information about nearby hospitals, including bed availability and distance from digital twin instances. Using OpenStreetMap, the system then calculates the travel distance and estimated time to each hospital. A sample of the data calculated using OpenStreetMap is shown in Table II. The hospitals are ranked based on a ranking score calculated using (1):

If $avail\_beds > 0$, then

$$Rank\ Score = \frac{k_1}{arrival\_time} + k_2 \cdot ln(avail\_beds) \qquad (1)$$

otherwise $Rank\ Score = 0$

where $avail\_beds$ is the number of currently unoccupied beds in the hospital's ED, $arrival\_time$ is the estimated travel time from the patient's location to the hospital, as calculated by OpenStreetMap, and k_1, k_2 are weighting factors that can be adjusted based on priorities, such as favoring closer hospitals or those with higher capacity. Equation (1) ensures that hospitals with more available beds and closer proximity are ranked higher. The weighting factors can be fine-tuned according to the specific needs of the emergency response system.

The mobile application provides a list of ranked hospitals to help the user make an informed decision. Figure 5 shows a screenshot of the user interface of the developed Android application. As can be seen, the hospitals are arranged according to the $Rank\ Score$, and the application offers a functional button to direct the user to the Google Maps application with the location of the desired hospital. The values used to calculate the $Rank\ Score$ are presented in Table II. In this table, the $arrival\_time$ is calculated using the OpenStreetMap service, whereas the number of available beds should be obtained from the real-time status of the ED's digital twin, but for testing purposes these data were simulated using Python. These values are then inserted into (1) to calculate the $Rank\ Score$. For example, Almeda hospital has an $arrival\_time$ of 18.77 min and 10 available beds. By inserting these values into (1), we get a $Rank\ Score$ of 28.35. The values of k_1 and k_2 were set to 100 and 10, respectively. By repeating this operation for all hospitals, we obtain the results in Table II.

E. Connection Information and Communication Protocols

The system ensures the use of secure and efficient communication protocols to facilitate data transmission between IoT devices, the IoT hub, and user applications. The protocol used to transmit data from IoT devices to the IoT hub is the Message Queuing Telemetry Transport (MQTT) protocol. This lightweight and efficient protocol is specifically designed for real-time communication, making it suitable for resource-constrained IoT devices. Shared Access Signature (SAS) tokens were used to authenticate connections for security, limiting access to authorized devices. The telemetry data that is being sent from the devices to the IoT hub, such as bed occupancy is JSON formatted [32].

The Hypertext Transfer Protocol Secure (HTTPS) protocol is used for communication between the user application and Azure services. HTTPS provides robust encryption for secure data transmission [33]. Azure Active Directory (Azure AD) is used for user authentication, ensuring secure access to sensitive information. Using RESTful APIs, the application retrieves hospital rankings and transmits emergency requests to the backend system [34]. After processing the data, Advanced Message Queuing Protocol (AMQP) is used to transmit the data from the IoT hub to the Azure Digital Twins platform. This protocol supports reliable message delivery, making it ideal for cloud-to-cloud communication. The data are then used to update the state of the digital twins in near real-time, ensuring consistent and accurate system behavior [35].

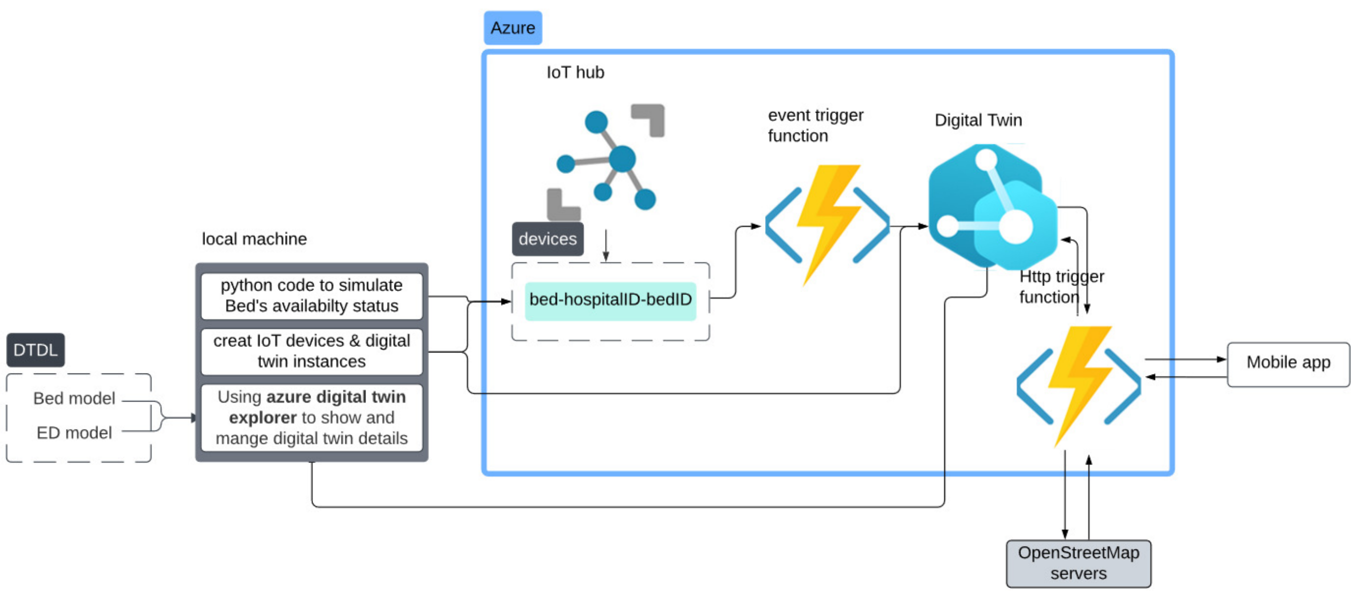


Fig. 4. Proposed system architecture.

These protocols were selected to provide an optimal balance of efficiency, security, and reliability to ensure smooth data flow across all system components.

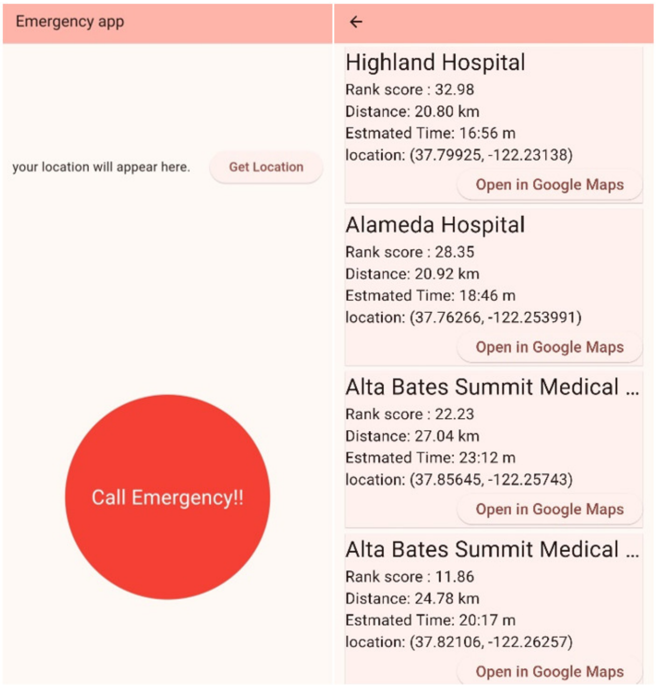


Fig. 5. The results on the Android application.

TABLE II. THE SAMPLE VALUES USED TO CALCULATE THE RANK SCORE IN FIGURE 5

Hospital name	*arrival_time* (min)	*avail_beds*	*Rank Score*
Alameda hospital	18.77	10	28.35
Alta Bates summit medical center – Alta Bates campus	23.2	6	22.23
UC San Francisco Benioff children's hospital Oakland	22.28	0	0
Highland hospital	16.94	15	32.98
Alta Bates summit medical center	20.28	2	11.86

F. Real-Time Monitoring

Azure Digital Twins Explorer was used to visualize and monitor system performance. It provides live updates on hospital status and resource availability, enabling administrators to make data-driven decisions effectively. Figure 6 illustrates the digital twins of the hospitals and beds, along with their respective information. For example, in Figure 6(a), Highland hospital is shown with its corresponding information such as ID, name, available beds, and location. Figure 6(b) shoes a digital twin of a bed in the hospital with its corresponding information such as ID and state.

G. Scalability and Adaptability

The system is designed to scale effortlessly. Adding new hospitals, beds, or IoT devices requires minimal effort, making it suitable for larger regions or additional use cases. The modular architecture also supports future enhancements and integrations.

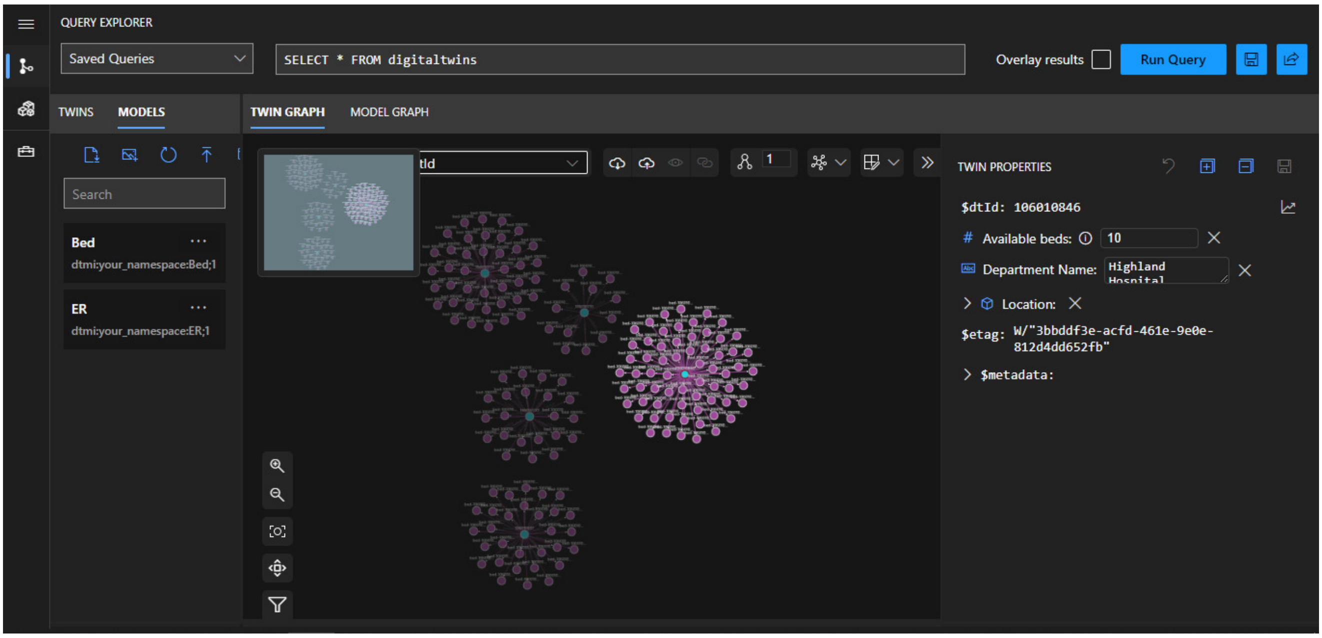
H. System Validation

To ensure the validity of the system, it was tested with multiple scenarios, including varying bed occupancy and emergency calls from different locations, and the rank score behavior was tested with multiple values. The phone application was also tested with Android emulators and real physical Android phones. The results showed accurate real-time updates and proper hospital rankings, demonstrating the reliability and efficiency of the system. However, formal user feedback from healthcare professionals has not yet been collected.

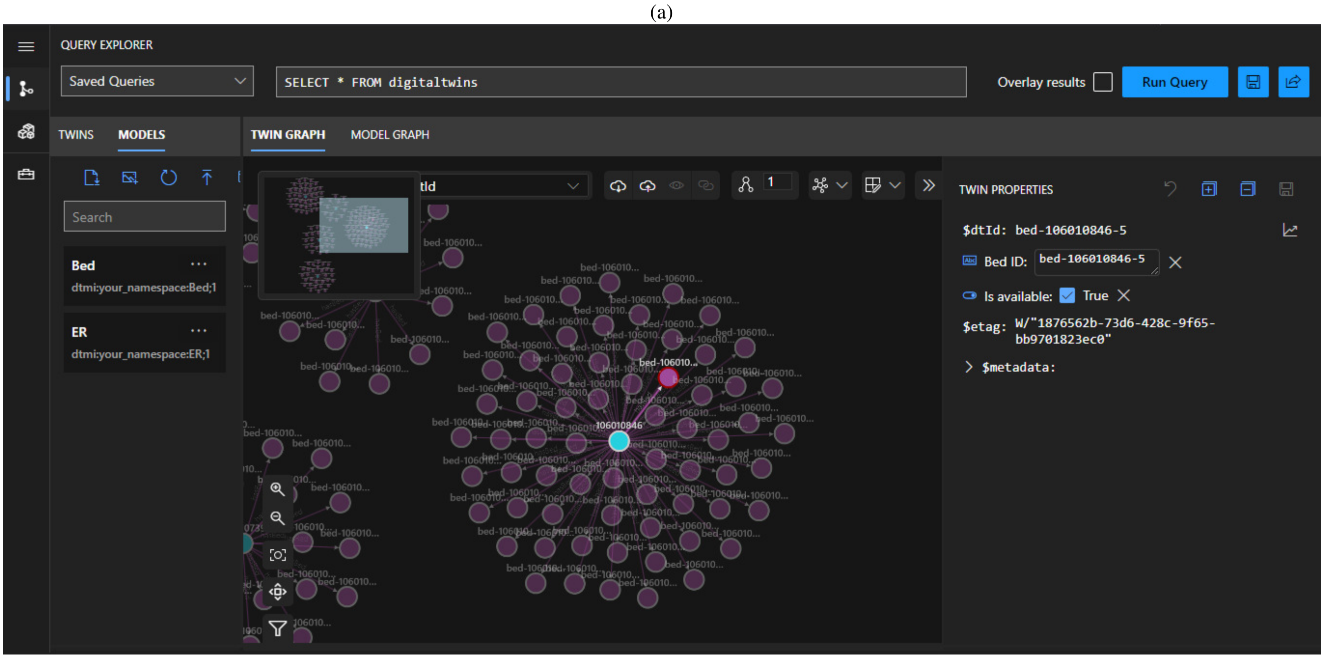
V. CONCLUSION AND FUTURE WORK

This project used Azure Digital Twins, Azure IoT Hub, and other tools to develop a decision support system for managing Emergency Departments (EDs) across hospitals. It provides real-time, ranked patient recommendations based on hospital location and resource availability. The digital twin model uses the necessary data from EDs, such as name, location, and bed availability. IoT devices send the data to the Azure IoT Hub in real time, and then the Azure Functions process updates the model. OpenStreetMap enables distance and arrival time calculations, while a Flutter app delivers ranked hospital recommendations by utilizing another Azure Function to perform the rank calculations. Unlike other works that focus on IoT monitoring or general hospital management systems, our work leverages the integration of real-time IoT data with digital twin technology to create a dynamic, real-time decision support system. The key innovation is the dynamic ranking mechanism that considers bed availability and estimated arrival time to provide the optimal hospital choice for the patient.

Future work could include further system improvements by incorporating additional parameters into the digital twin, such as staff availability, specialized equipment, or real-time traffic data, to enhance decision-making accuracy. Another option could be to develop dedicated software to monitor and manage the digital twin with detailed information about the hospitals that can be used by authorized entities, such as the hospital management team or the 911 call center. In addition, integrating machine learning models into the system could provide predictive insights to improve system performance. On the application itself, other enhancements could be implemented, such as a notification system to inform the hospital staff of incoming cases to minimize preparation time, a hospital specialty indicator for specialized hospitals, and the addition of an emergency case description option that can be used to guide the user to better options or inform the hospital staff to better understand the emergency case.



(a)



(b)

Fig. 6. The digital twins visualized in Azure Digital Twins Explorer: (a) ED digital twin, (b) bed digital twin.

ACKNOWLEDGMENT

The authors would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) in Baghdad, Iraq for their support in the present work.

REFERENCES

[1] H. A. Abu-Alsaad and R. R. K. Al-Taie, "Smart Parking System Using IoT," in *2024 16th International Conference on Electronics, Computers and Artificial Intelligence*, Iasi, Romania, 2024, pp. 1–7, https://doi.org/10.1109/ECAI61503.2024.10607419.

[2] S. M. Mahmoud, H. A. Alabbasi, and T. E. Abdulabbas, "Monitoring and detecting outliers for elder's life activities in a smart home: A case study," in *2017 E-Health and Bioengineering Conference*, Sinaia, Romania, 2017, pp. 458–461, https://doi.org/10.1109/EHB.2017.7995460.

[3] A. F. Abbas and M. Z. Abdullah, "Design and Implementation of Tracking a user's Behavior in a Smart Home," *IOP Conference Series: Materials Science and Engineering*, vol. 1094, no. 1, Feb. 2021, Art. no. 012008, https://doi.org/10.1088/1757-899X/1094/1/012008.

[4] A. F. Abbas and M. Z. Abdullah, "Design and Implementation of a Smart Home Gas Detection based on Mobile Network System," *Journal of Engineering and Sustainable Development*, vol. 25, no. 2, pp. 9–16, Mar. 2021, https://doi.org/10.31272/jeasd.25.2.2.

[5] H. A. Abu-Alsaad, "CNN-Based Smart Parking System," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 11, pp. 155–170, Jun. 2023, https://doi.org/10.3991/ijim.v17i11.37033.

[6] C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," *Agricultural Systems*, vol. 189, Apr. 2021, Art. no. 103046, https://doi.org/10.1016/j.agsy.2020.103046.

[7] B. F. Alshammari and M. T. Chughtai, "IoT Gas Leakage Detector and Warning Generator," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6142–6146, Aug. 2020, https://doi.org/10.48084/etasr.3712.

[8] E. Wang, P. Tayebi, and Y.-T. Song, "Cloud-Based Digital Twins' Storage in Emergency Healthcare," *International Journal of Networked and Distributed Computing*, vol. 11, no. 2, pp. 75–87, Dec. 2023, https://doi.org/10.1007/s44227-023-00011-y.

[9] R. H. Hussan and N. H. Hussan, "A Review of Enhancing Government Offices Experience and Operational Efficiency of Knowledge Management Systems depending on Information Technologies," *Mustansiriyah Journal of Pure and Applied Sciences*, vol. 1, no. 3, pp. 134–147, Oct. 2023, https://doi.org/10.47831/mjpas.v1i3.56.

[10] G. N. Keshava Murthy, "Smart Emergency Management with IoT for Integration of Emergency Centers and Patients," *International Journal of Recent Technology and Engineering*, vol. 8, no. 4, pp. 3979–3983, Nov. 2019, https://doi.org/10.35940/ijrte.D8506.118419.

[11] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital Twin: Origin to Future," *Applied System Innovation*, vol. 4, no. 2, Jun. 2021, Art. no. 36, https://doi.org/10.3390/asi4020036.

[12] H. K. Omar, M. Frikha, and A. K. Jumaa, "Improving Big Data Recommendation System Performance using NLP techniques with multi attributes," *Informatica*, vol. 48, no. 5, pp. 63–70, Feb. 2024, https://doi.org/10.31449/inf.v48i5.5255.

[13] "Azure Digital Twins Documentation - Tutorials, API Reference." Microsoft Learn. https://learn.microsoft.com/en-us/azure/digital-twins/.

[14] "AWS IoT TwinMaker Documentation." AWS. https://docs.aws.amazon.com/iot-twinmaker/.

[15] "IoT platform product architecture on Google Cloud | Cloud Architecture Center." Google Cloud. https://cloud.google.com/architecture/connected-devices/iot-platform-product-architecture.

[16] R. Issa, M. S.Hamad, and M. Abdel-Geliel, "Digital Twin of Wind Turbine Based on Microsoft® Azure IoT Platform," in *2023 IEEE Conference on Power Electronics and Renewable Energy*, Luxor, Egypt, 2023, pp. 1–8, https://doi.org/10.1109/CPERE56564.2023.10119576.

[17] J. Corral-Acero *et al.*, "The 'Digital Twin' to enable the vision of precision cardiology," *European Heart Journal*, vol. 41, no. 48, pp. 4556–4564, Dec. 2020, https://doi.org/10.1093/eurheartj/ehaa159.

[18] J. Chen *et al.*, "A Revolution of Personalized Healthcare: Enabling Human Digital Twin With Mobile AIGC," *IEEE Network*, vol. 38, no. 6, pp. 234–242, Nov. 2024, https://doi.org/10.1109/MNET.2024.3366560.

[19] Mrs. C. Ramya, S. Elamathi, D. Nithya, K. Priyadarshini, and S. Suganthini, "IoT-Powered Digital Twin Solution for Real-Time Health Monitoring," *International Journal of Engineering Research & Technology*, vol. 12, no. 2, Feb. 2023, https://doi.org/10.17577/IJERTCONV12IS02037.

[20] M. Al-khafajiy, H. Kolivand, T. Baker, D. Tully, and A. Waraich, "Smart hospital emergency system," *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 20087–20111, Jul. 2019, https://doi.org/10.1007/s11042-019-7274-4.

[21] K. Papachristou, P. F. Katsakiori, P. Papadimitroulas, L. Strigari, and G. C. Kagadis, "Digital Twins' Advancements and Applications in Healthcare, Towards Precision Medicine," *Journal of Personalized Medicine*, vol. 14, no. 11, Nov. 2024, Art. no. 1101, https://doi.org/10.3390/jpm14111101.

[22] A. Vallée, "Digital twin for healthcare systems," *Frontiers in Digital Health*, vol. 5, Sep. 2023, Art. no. 1253050, https://doi.org/10.3389/fdgth.2023.1253050.

[23] R. S. A. Corpuz, J. Usa, J. C. Obaña, J. N. Borbe, D. A. Salazar, and T. M. Valeza, "A hospital bed availability monitoring system using the Internet of Things," *International Journal on Information Technologies and Security*, vol. 16, no. 3, pp. 13–26, Sep. 2024, https://doi.org/10.59035/NKFQ4271.

[24] S. Han, S. He, and H. C. Oh, "Data-Driven Inpatient Bed Assignment Using the P Model." arXiv, Nov. 16, 2021, https://doi.org/10.48550/arXiv.2111.08269.

[25] M. C. Lam, M. Ayob, J. Y. Lee, N. Abdullah, F. A. Hamzah, and S. S. M. Zahir, "Mobile-based Hospital Bed Management with Near Field Communication Technology : A Case Study," *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5706–5712, Jun. 2020, https://doi.org/10.48084/etasr.3527.

[26] S. Ayouni, M. Maddeh, S. Al-Otaibi, M. B. Alazzam, N. M. Alturki, and F. Hajjej, "Development of a Smart Hospital Bed Based on Deep Learning to Monitor Patient Conditions," *Journal of Disability Research*, vol. 2, no. 2, pp. 25–36, Jul. 2023, https://doi.org/10.57197/JDR-2023-0017.

[27] "Emergency Department Volume and Capacity." Department of Health Care Access and Information, [Online]. Available: https://catalog.data.gov/dataset/emergency-department-volume-and-capacity.

[28] "Azure IoT Hub Documentation." Microsoft Learn. https://learn.microsoft.com/en-us/azure/iot-hub/.

[29] "Azure Functions documentation." Microsoft Learn. https://learn.microsoft.com/en-us/azure/azure-functions/.

[30] "Flutter documentation." Flutter Docs. https://docs.flutter.dev.

[31] "OSRM API Documentation." OSRM Project. https://project-osrm.org/docs/v5.5.1/api/?language=Python#general-options.

[32] P. S. Akshatha, S. M. D. Kumar, and K. R. Venugopal, "MQTT Implementations, Open Issues, and Challenges: A Detailed Comparison and Survey," *International Journal of Sensors, Wireless Communications and Control*, vol. 12, no. 8, pp. 553–576, Dec. 2022, https://doi.org/10.2174/2210327913666221216152446.

[33] M. Trevisan, D. Giordano, I. Drago, and A. S. Khatouni, "Measuring HTTP/3: Adoption and Performance," in *2021 19th Mediterranean Communication and Computer Networking Conference*, Ibiza, Spain, 2021, pp. 1–8, https://doi.org/10.1109/MedComNet52149.2021.9501274.

[34] A. M. Khan, "Proposing and Deployment of Attractive Azure AD Honeypot With Varying Security Measures To Evaluate Their Performance Against Real Attacks," M.S. thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, Netherlands, 2021.

[35] C. B. Gemirter, Ç. Şenturca, and Ş. Baydere, "A Comparative Evaluation of AMQP, MQTT and HTTP Protocols Using Real-Time Public Smart City Data," in *2021 6th International Conference on Computer Science and Engineering*, Ankara, Turkey, 2021, pp. 542–547, https://doi.org/10.1109/UBMK52708.2021.9559032.