Int j simul model 24 (2025) 4, 671-682
Original scientific paper

A MACHINE LEARNING-ENABLED DIGITAL TWIN FOR AN ORTHOPAEDIC CLINIC: A PROOF OF CONCEPT

Ozen, N. S.[*]; Farsi, M.[**]; Erkoyuncu, J. A.[**]; Koyuncu, M.[*] & Gray, J.[***]
[*] Cukurova University, Faculty of Engineering, Department of Industrial Engineering, Adana, Turkey
[**] Cranfield University, Centre for Digital and Design Engineering, Bedford, United Kingdom
[***] Luton & Dunstable University Hospital, Luton LU4 0DZ, United Kingdom
E-Mail: selin.ozen@toros.edu.tr, j.a.erkoyuncu@cranfield.ac.uk, maryam.farsi@cranfield.ac.uk, melihkoyuncu@yahoo.com, jim@dashmed.co.uk

Abstract
The emergence of Digital Twins (DTs) creates opportunities for ad-hoc solutions in complex systems. This paper proposes a DT for the Luton & Dunstable Orthopaedic Clinic in the UK to support decision-making. In this study, the interactive operations learn from actual patient data and offer predicted feedback over time, using machine learning algorithms such as Random Forest, LightGBM, and XGBoost. The system simulation model is built in AnyLogic using a Discrete Event Simulation (DES) technique and tested with data from the hospital's DASHclinic app. The best-performing model, XGBoost, for predicting patient waiting times, achieved a mean absolute error (*MAE*) of 13.67 minutes. Based on predicted wait times, a feedback mechanism decides whether to notify patients about their expected waiting duration. The simulation records actual waiting times for upcoming patients, allowing prediction validation. Real-world validation yielded an *MAE* of 31.40 minutes, demonstrating promising accuracy for a clinical setting. The DT's added value lies in providing individual feedback to patients, enabling a real-time feedback loop with potential for optimisation of appointment scheduling.
(Received in September 2025, accepted in October 2025. This paper was with the authors 1 week for 1 revision.)

Keywords: Digital Twin, Machine Learning, Healthcare, Discrete Event Simulation, System Simulation

1. INTRODUCTION

Researchers and practitioners from a broad range of scientific and engineering fields have recently shown a great interest in DT technologies [1]. A DT is a technology that connects the physical and virtual worlds by using models, sensors, data, and software. According to Barricelli et al. [2], a DT is a virtual version of a real-world object or process, which can be either living or non-living. Examples of DTs include virtual machines, processes, people, hospitals, or even entire cities. A DT typically comprises a virtual representation of a physical process or object, a detailed digital model that captures its characteristics, and a digital thread that enables seamless data exchange between the physical and digital environments. It creates a virtual representation of a real system to monitor and analyse its performance, in line with the Industry 4.0 strategy [3, 4]. Specifically, healthcare organisations often struggle to allocate resources such as hospital beds, staff, and equipment, leading to inefficiencies and delays in patient care [5]. These challenges include long patient waiting times, unpredictable service durations, and a lack of proactive response mechanisms when delays occur. Benevento et al. [6] accordingly reported that hospital administrators can better allocate emergency department resources and monitor process performance in real time with accurate and timely estimates of expected patient waiting times. However, the study [6] does not utilise a DES or similar modelling approach to replicate system dynamics, which limits its scope to predictive analytics rather than a full DT framework that would enable real-time system adaptation and feedback. To address these issues, DT allows healthcare systems to test and evaluate operational changes in a virtual environment before applying them in real life, enabling safer and more informed

decision-making. According to Vallée [7], a hospital's DT can provide data-driven insights on staffing, capacity, and operations to support decision-making and preparation for upcoming issues. The efficacy of traditional simulation methods is constrained by their lack of real-time integration and adaptive feedback, which limits their ability to reflect evolving system performance during operation. Traditional simulation models in hospitals rely on fixed-interval data collection (e.g., stopwatches) and are heavily dependent on historical data which limits their ability to respond to real-time changes. Given the stochastic nature of healthcare environments and the increasing demand for real-time responsiveness, DT technologies have become essential tools far beyond what traditional simulations can offer. The main difference between DT and traditional simulation lies in the data link, which allows the exchange of information between digital and physical objects. This exchange called as a feedback loop, continually informs medical staff by refreshing the model with new data, allowing for real-time monitoring and valuable predictive insights that enhance decision-making. This ongoing feedback loop provides a more adaptive and responsive approach to modelling compared to traditional models, which are typically limited and do not have real-time updates [8].

Building on this foundation, machine learning (ML) algorithms have gained significant interest due to their capability to detect complex patterns in large datasets and support data-driven insights. ML techniques have been successfully applied in various domains, specifically in healthcare, including disease recognition, resource allocation, demand forecasting and risk assessment. In line with this, Aldoseri et al. [9] highlighted that increasing the predictive ability and personalisation of models, the combination of artificial intelligence (AI) and ML would make it possible to analyse large amounts of complex data more effectively. One of the critical areas where these predictive capabilities can offer tangible value is in forecasting patient waiting times and enhancing the management of patient flow in healthcare settings. The duration and predictability of waiting times significantly influence patient satisfaction. Moreover, the significant variability in patients' waiting times challenges effective planning for both care providers and patients [10]. Uncertain waiting times cause patients to worry about whether they will receive the care they need, which negatively impacts their overall experience. Providing clear wait time information, on the other hand, increases patient satisfaction. Patients can manage the process more easily when they know how long they will wait [11]. Accordingly, ML algorithms are increasingly utilised for predictive tasks such as predicting patient waiting times by uncovering hidden patterns in complex healthcare data. However, the advanced integration of simulation and ML-based techniques into real-time decision support systems is still lacking, despite the growing interest in these approaches in the healthcare industry. Current outpatient systems are not equipped to predict patient-specific delays or adjust operations dynamically. This results in long and uncertain waiting times, inefficient resource allocation, and an overall lack of responsiveness to emerging bottlenecks during the day.

Outpatient orthopaedic clinics often struggle with unpredictable waiting times due to variable patient pathways and limited resources. To address this challenge, this study proposes a DT framework that integrates DES, ML, and an automated feedback loop. A virtual replica of an outpatient clinic was developed, with a specific focus on multi-pathway decision logic to represent patient-specific routes. The DT can train a predictive model using historical patient flow data with individual-level pathways, forecasting patient-specific waiting times for upcoming patients, validating predictions against actual outcomes and continually updating itself with the new data. Accordingly, this research seeks to answer: How can ML empower DTs to deliver proactive and data-driven decision-making support in outpatient healthcare?

This research study is structured as follows: Section 2 reviews the state of the art and identifies research gaps. Section 3 describes the methodology employed including system architecture, simulation design and feedback loop mechanism. Section 4 and Section 5 present results, discussion and conclusion respectively.

2. STATE OF THE ART

In this study, a comprehensive literature review was conducted using multiple academic databases to investigate DTs in healthcare. Particularly, Google Scholar was accessed extensively to capture a wide range of relevant articles and conference proceedings. In addition, Wiley Online Library, Science Direct, Scopus, Web of Science and the National Institutes of Health (NIH)'s PubMed were systematically searched to include specialised publications and in the scope of healthcare and system simulation. The literature search focused on key subject areas including digital twins in healthcare, patient waiting time prediction, discrete event simulation-based modelling, and the application of ML techniques in outpatient service delivery. Once the initial documents were retrieved, duplicates were removed, and titles and abstracts were screened for relevance. Articles that aligned with our research focus were then selected for full-text review to further refine the selection.

The DT concept is still evolving, with various models and frameworks proposed in the literature, particularly in distinct categories within healthcare. One such category involves the virtual representation of a patient's diseases, as explored by Zhang et al. [12]. Another category includes treatments, highlighted by research from Li et al. [13]. In hospital and healthcare management, topics like planning, facility use, and patient flow have been widely explored, mostly through experimental DT applications or discussions of their potential contributions. Focusing on these studies, in Ireland, a DT study was conducted with Siemens Healthineers for process improvement to address increased waiting times and high patient demand in the radiology department. In the study, a DT was created for the radiology department, which was fed with real data, and various predictions were made using the DT. According to the results, savings of 9,500 euros could be achieved [14]. Another study, carried out by Karakra et al. [15], proposed a new methodology named HospiT'Win, which utilises IoT devices connected to DES using a DT platform to create patient paths in a hospital based on real-time-like monitoring and prediction. In the study, the DT is composed of two modules: a monitoring DT (DTM) and a prediction DT (DTP). The primary purpose of the DTP model is to provide a proactive view of the hospital every hour and make proactive decisions based on this view. The use of DTs to improve crisis management in emergency departments was the focus of Moyaux et al. [16]. In addition to predicting the department's capabilities in various "what-if" scenarios, they explored how to enable active monitoring and tracking of resources in emergency departments for more intelligent resource utilisation. While these studies demonstrated the potential of DTs in hospital planning and monitoring, they lacked integration with ML to leverage its predictive capability and to deliver patient-specific real-time notifications. Beyond healthcare, DES has been acknowledged as a powerful tool in a number of fields, including inventory management [17] and supply chain [18]. Although its wide range of applications demonstrates its adaptability as a modelling technique, its integration with ML within a DT framework is still rather limited, especially in the field of healthcare.

In recent years, AI, Deep Learning (DL) [19], and ML has been increasingly applied in healthcare to improve prediction, diagnosis, and operational decision-making. We can broadly categorise the applications of ML algorithms in the healthcare literature into three main areas. Firstly, ML algorithms have been widely used for clinical diagnosis and disease prediction (e.g., diabetes) [20]. Secondly, the literature has focused on forecasting patient behaviours and improving scheduling accuracy using advanced predictive modelling techniques. For example, Aldhoayan and Alobaidi [21] developed ML models (RF, XGBoost (XGB), Logistic Regression, Gradient Boosting (GB)) to forecast patient tardiness. Thirdly, a more specific area of research focuses on predicting patient waiting times in clinical settings. Li et al. [22] focused on implementing four ML algorithms (RF, LR, GB decision tree, and K-nearest neighbour) for waiting time of patients in four department categories. RF performed the best for one of the

four departments with a *MAE* of 5.03 minutes. Similarly, Joseph et al.'s [23] study utilised two ML algorithms, XGB and RF, in order to predict patient waiting times by extracting Electronic Health Records data for the general medicine department's patients in India. The best performed algorithm indicated as XGB. Although earlier studies have demonstrated the utility of ML in predicting operational metrics in healthcare, models remained standalone, lacking integration with dynamic feedback systems or simulation that could support real-time.

3. METHODOLOGY

This paper addresses the concept of an adaptive DT by proposing a four-stage approach within the context of DT architecture: (1) data preparation and preprocessing, (2) DES modelling, (3) integration of machine learning (predictive model development), (4) feedback and continuous learning mechanism. The study operates under several key assumptions: first, there are no walk-in patients, meaning the appointment schedule is never disrupted by unexpected arrivals. Secondly, healthcare staff do not work overtime and always adhere to their assigned shifts. Thirdly, doctors are primarily responsible for the examination of patients, meaning they are not required to perform other tasks. Lastly, it is assumed that patients receive consecutive treatments with a scheduled lunch break of one hour, with the exception during scheduled sessions. Overall, the DT primarily aims to improve predictability in a healthcare setting, the feedback provided to patients is a resultant outcome of this process.

3.1 Data collection and preparation

A detailed insight into the clinic's workflow was initially gained through an on-site visit, which included interviews with the lead physician and direct observation of routine operations. Information about the resources included the resources' shifts, and the number of rooms available for the particular activity. To further investigate and prepare the model, data was gathered from the DASHclinic app used by the clinics as a substitute for an IoT device. The app employs codes to classify patient activities: D stands for doctor, P for plaster, X for x-ray, and N for nurse. Statuses show if a process is pending (0), queued (1), active (2), or completed (3). To ensure data accuracy, only completed operations were analysed and data cleaning techniques were applied to account for any delays or missed updates. The data collected was pre-processed to make it ready for the simulation input. This preprocessing includes pathway analysis, imputation, data cleaning, decoding, sorting, and merging. The patients' appointments were decoded and merged based on their appointment date and time, and a connection between 16 CSV files was processed to guarantee that all of the patients' same-day visits were combined into a single row to show their whole visit. In other words, the data were grouped by patient ID and the same date in order to create a coherent route for every patient. This produced a comprehensive pathway that showed the journey each patient walked from registration to departure for a certain day. After preprocessing, the final dataset contained 7,303 patient records. Due to delays in updating patient entries and exits in the app by the hospital staff, unusually long activity durations were observed. These outliers were identified based on the duration thresholds provided during the interviews. If the patient's pathway was incomplete, these records were removed from the dataset. In cases where the pathway was complete, but the activity durations did not fall within acceptable clinical ranges, specifically, 0–15 minutes for D, 5–15 for X, 15–50 for P, and 5–12 for N, the values for those stations were set to 0. Subsequently, these missing or invalid values were imputed using average durations for each station: 7 mins for D, 7 mins for X, 24 mins for P, 5 mins for N, 10 mins for D2, 9 mins for X2, and 32 mins for P2. Consequently, the latest input dataset includes individual patient information such as appointment date and time, patient ID, consultant name, doctor name index, pathway length, and the time required at each station (D, X, P, N) and their second visit (D2,

X2, P2, N2). In this context, the variables *timeRequired_D2*, *X2*, *P2*, and *N2* represent the time spent during the patient's second visit to the corresponding station. It also contains clinic-specific details such as clinic name, and clinic ID. Since the data included time logs specific to each patient at each clinical station, it was possible to incorporate personalised durations for all completed stages, instead of relying on generalised probability distributions. Therefore, a more accurate depiction of the system is made possible by capturing the variability in appointment durations through individual durations, which offer greater granularity and reduce the uncertainty introduced by wide-range probability distributions. That is, each patient in the simulation spends the exact duration at each station as specified in the dataset, based on time logs obtained from the DASHclinic app. Subsequently, the KPI's were monitored, and the individual waiting time of each patient was recorded and saved to a second dataset. This dataset is later used for training ML algorithms.

3.2 Discrete event simulation

A computer-based technique called DES is frequently used to model patient pathways, optimise resource usage, and improve system efficiency in medical settings [24]. Compared to other methods like System Dynamics or Agent-Based Modelling, DES is better equipped to model individual patient journeys, queuing behaviour, and stochastic service times typical in outpatient clinics. It serves as a suitable modelling approach for effectively capturing the dynamic and stochastic nature of healthcare systems. In our case, the clinic environment involves multiple patient flows and variable processing times. The simulation model is built upon the pre-processed dataset derived from the DASHclinic app, as described in the data preparation section. The main characteristics of the system (e.g., doctors' schedules, patient pathways, etc.) were defined to mirror the physical system and create a virtual replica of the clinic. In the simulation model, each patient is represented as an individual agent with a structured set of parameters and variables that capture both static attributes and dynamic behaviours. Static parameters include *patientID* as a unique identifier, *arrivalTime* indicating the scheduled appointment time when the patient is expected to enter the system, and pathway, a string defining the sequence of stations the patient will follow (e.g., "D-X-P" for Doctor → X-ray → Plaster). The patient's pathway is also decomposed into a string format (*pathwayStations*) to allow dynamic routing during the simulation. To simulate this routing mechanism, a condition is defined within the SelectOutput block of each station using Java code, as shown in Table I.

Table I: Java code snippets for dynamic patient routing in the simulation.

Block / Context	Java Code	Description
SelectOutput block (e.g., X-ray)	agent.pathwayIndex < agent.pathwayStations.length && agent.pathwayStations[agent.pathwayIndex].equals("X")	Checks if the next station in the patient's pathway is the current block
On-enter logic in each station	if (agent.stationIndex+1 < agent.pathwayStations.length) { agent.stationIndex++; agent.currentStation = agent.pathwayStations[agent.stationIndex]; }	Updates the patient's current station and advances the index along the pathway

This enables the patient to proceed step by step along their individual route until the entire pathway is completed. To reflect realistic scenarios in outpatient settings, the model accounts for situations where patients may revisit the same station. For this purpose, variables such as *timeRequired_D* and *timeRequired_D2* represent the durations for the first and second visits to the doctor station, respectively. Similar variables exist for other stations (e.g., X, N, P). During

the simulation run, dynamic variables such as *currentStation*, and *stationIndex* are updated to monitor the patient's current location and progress through the pathway. Additional variables like *waitingTime* and *enterQueueTime* are used to compute key performance indicators (KPIs), including total waiting time and time spent in queues at each station. The agent includes doctor-specific attributes such as *doctorName_index*, which ensures that each patient is assigned to a specific doctor based on the appointment data. Each doctor's working shifts were extracted from the appointment records in the app. When the patients come to their first station in their pathways, they are waiting for their assigned doctor's availability. This design enables personalised resource allocation. In the digital space, the clinic's operational states, expressed through KPIs such as patient queue sizes, individual resource utilisation, average waiting time and length of stay (*LOS*), were visualised within the AnyLogic Personal Learning Edition 8.9.3 environment. The DES was designed for the clinic illustrated in Fig. 1. Source 1 represents historical patients used for simulation and model training, while Source 2 simulates upcoming patients as if they have not yet arrived. As a result, this setup enables the prediction and validation process by comparing estimated waiting times with actual durations during the second phase of the simulation.

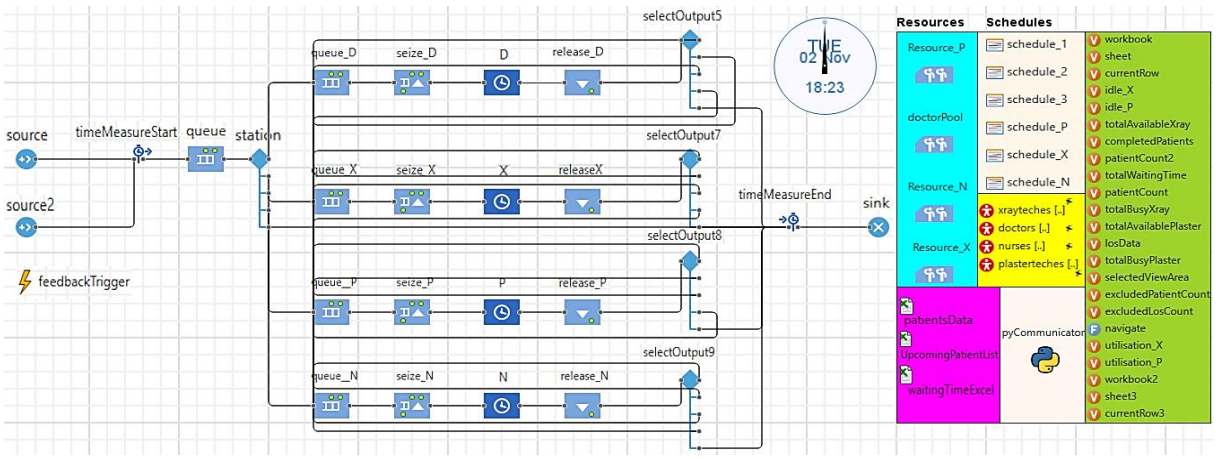


Figure 1: DES design of the clinic.

3.3 Digital twin

The proposed DT architecture, illustrated in Fig. 2, is defined across three spaces and involves a multi-step procedure. On the left, the physical space includes the Luton & Dunstable Orthopaedic Clinic, its entities (doctors, nurses, etc.), resources, appointment schedule and the built-in app, DASHclinic. The DASHclinic app is used as a substitute for an IoT device within the DT concept. Each entity has access to the app via a tablet. At every patient arrival or departure at the stations, the app is updated by the responsible staff member. Thus, the difference between a patient's arrival and departure times reflects the duration spent at each station. The edge layer serves in the middle, serves as a bridge, utilising communication technologies like Wi-Fi to facilitate seamless data flow between the physical and digital spaces. The data collected from the physical space is first processed through local preprocessing services to convert it into a usable format. This is followed by removing incomplete records and activity durations outside the clinical ranges (defined in Section 3.1) to ensure high-quality data transfer to the digital space. In the digital space, both DES and ML modules utilise historical and real-time data for simulation and prediction purposes. This environment supports DT monitoring, bottleneck detection, and predictive messaging to patients, enabled through dashboards that track queue sizes, and resource utilisation. The data extracted from the app was used as input for the DES model built in AnyLogic software and includes patient information such as required skillset, arrival time, and activity times (i.e., durations spent with doctors).

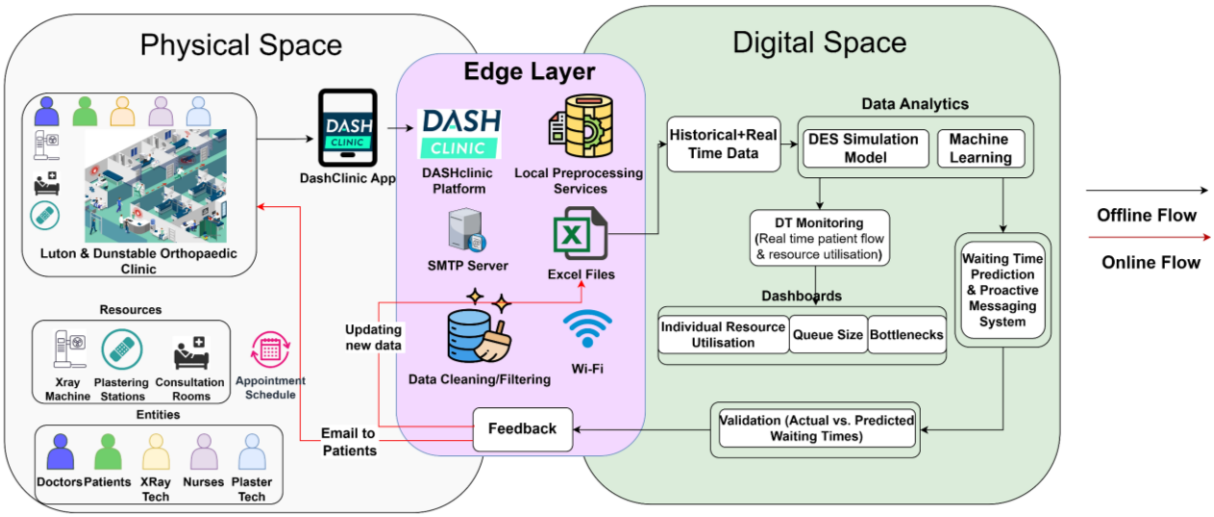


Figure 2: Digital twin conceptual architecture.

Following this, the same dataset used in the DES simulation is also employed to train ML algorithms, RF, LightGBM and XGB, to predict patient-specific waiting times for upcoming patients. This integration allows the system to make individual-level predictions before patients enter the system. While DES simulates the existing system dynamics based on predefined data, it cannot anticipate future system states or patient-specific delays. ML addresses this gap by learning from historical patterns and contextual variables (e.g., time, weather) which enables proactive feedback such as early delay alerts. A separate training dataset is also created by duplicating the original input data and appending the actual individual waiting times obtained from the simulation, so that the training dataset includes all features alongside the true labels required for ML algorithms. This dataset is generated after all patients from Source 1 have been processed, incorporating the actual waiting times recorded during the first phase. This enriched dataset is prepared through a Python script triggered by an 'Event' block (via PyCommunicator) in AnyLogic at 08:00 AM on the morning of the final day of the simulation, when all relevant historical data from previous days has been accumulated and is available for training the predictive model. This script performs merging, feature engineering (e.g., time of day, weather), model training, and algorithm selection based on the lowest *MAE* value. Subsequently, using the best-performing algorithm, the individual waiting times of upcoming patients (Source 2) are predicted and saved in a file. Following the completion of Source 2, a second Python script begins running through the PyCommunicator element to carry out a number of tasks. By comparing the expected and actual waiting times using *MAE* and root mean square error (*RMSE*) metrics, it first verifies the model's predictive performance. Second, it updates and enriches the training dataset by combining the current training records with real patient data, such as waiting times and contextual information such as time and weather. Third, it updates new patient records collected during the simulation's second phase in the main dataset. The Edge Layer facilitates the transmission of predicted outcomes and initiates patient notifications, forming the system's feedback loop as shown in Fig. 3. Lastly, the second script is used to automatically send dummy notification emails with a SMTP server to those whose predicted waiting time exceeded the threshold which is 50 minutes. Patients are informed of potential delays prior to entering the system, which allows a feedback mechanism that operates in real time. This step allows patients to adjust their plans or expectations without having to wait for the entire simulation loop to finish. Consequently, this continuous update process enhances the model's predictive performance and data quality. As a result, the DT is not only replicating system behaviour but also adapt dynamically as new data is collected which leads to more accurate decision-making over time.

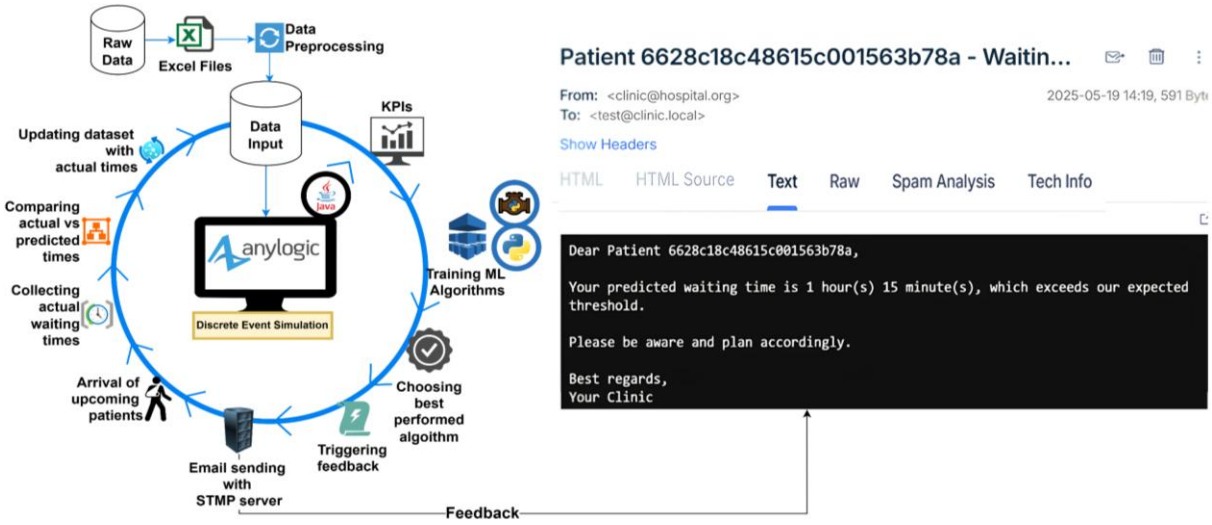


Figure 3: Workflow of the DT mechanism and dummy email snapshot.

3.4 Machine learning algorithms

As evidenced in the literature [25], RFs, GB trees, and neural networks have shown remarkable performance in predicting both binary outcomes and high-variance continuous variables within the healthcare setting. Accordingly, RF, XGB, and LightGBM were employed in this study due to their ability to handle mixed-type healthcare datasets with both categorical and numerical variables, as well as their robustness in small-to-medium-sized datasets, particularly in capturing complex, non-linear relationships and handling missing data. RF Regression is an ensemble learning method that builds multiple decision trees using bootstrap sampling and combines their average outputs to improve prediction accuracy [26]. On the other hand, GB algorithms utilise gradient descent to minimise the loss function during training by correcting mistakes in weak learners, typically decision trees, through weighted averaging. Among these algorithms, XGBoost is an optimised distributed GB library that increases efficiency by horizontally growing decision trees. It compiles predictions as an ensemble learning method. Moreover, by using a leaf-wise splitting technique, LightGBM improves accuracy by minimising loss more efficiently. Despite its remarkable accuracy, LightGBM's remarkable speed justifies being known as "Light" [27]. Performance metrics are used to identify the best-performing models based on evaluation results, measuring the performance of ML algorithms. In this study, the dataset was split into two, with 85 % for training and 15 % for testing and *RMSE*, *MAE*, and R^2 were selected as the key performance metrics. All models were trained with default hyperparameters due to technical limitations; hyperparameter tuning caused system slowdowns and freezing when integrated with AnyLogic.

MAE is a statistical metric used to quantify the average magnitude of the discrepancy between the predicted and actual values; that is, $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'|$. Similarly, R^2 quantifies the amount of variation in the dependent variable that is explicable by the independent variables. R^2 is defined as 1 minus the sum of squared residuals (*SSR*) divided by the total sum of squares (*TSS*), or $R^2 = 1 - \frac{SSR}{TSS}$. Likewise, *RMSE* is defined as the standard deviation of the difference between the true values and the predicted values, which can be considered as the standard deviation of the prediction errors: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2}$ [28]. During preprocessing for ML algorithms, some features were added from the '*dateTime*' variable to enrich the training dataset. In order to enrich the training data, weather-related features were added to the training dataset. These included '*timePeriod*', '*dayOfWeek*', '*hourOfDay*', '*month*', '*season*', '*day_of_month*', '*number_of_patients_at_same_time*'. Additionally, historical

weather conditions were added using Meteostat library in Python, contributing *temperature_C*, *humidity* and *precipitation_mm* as numerical features. To guarantee compatibility with tree-based regression models, label encoding was used to encode categorical variables mentioned in Section 3, such as *pathway*, *clinic name*, *timePeriod*, etc.

4. RESULTS

Along with a noteworthy R^2 value for a healthcare setting of 0.8375, XGB had the best results among the tested models with the lowest *MAE* of 13.67 and *RMSE* of 22.11 as shown in Table II. A *MAE* of 13.67 minutes indicates that predictions are, on average, within ±13 minutes of actual waiting times, despite the dataset containing noisy and inconsistent records, such as incomplete entries and extreme outliers in activity times. This level of accuracy can be considered clinically acceptable for providing early feedback to patients and supporting operational decisions in outpatient settings. LightGBM had the lowest *RMSE* (21.97), yet its *MAE* (14.32) exceeded RF and XGB. Although RF performed reasonably well (*MAE* = 13.91, $R^2 = 0.8185$), both GB models surpassed it. The results show that XGB offers more balanced and accurate predictive performance, supporting the dependability and explainability of the model.

Table II: Performance of ML algorithms.

(mins)	RF	XGB	LGBM
MAE	13.91	13.67	14.32
RMSE	23.37	22.11	21.97
R^2	0.8185	0.8375	0.8395

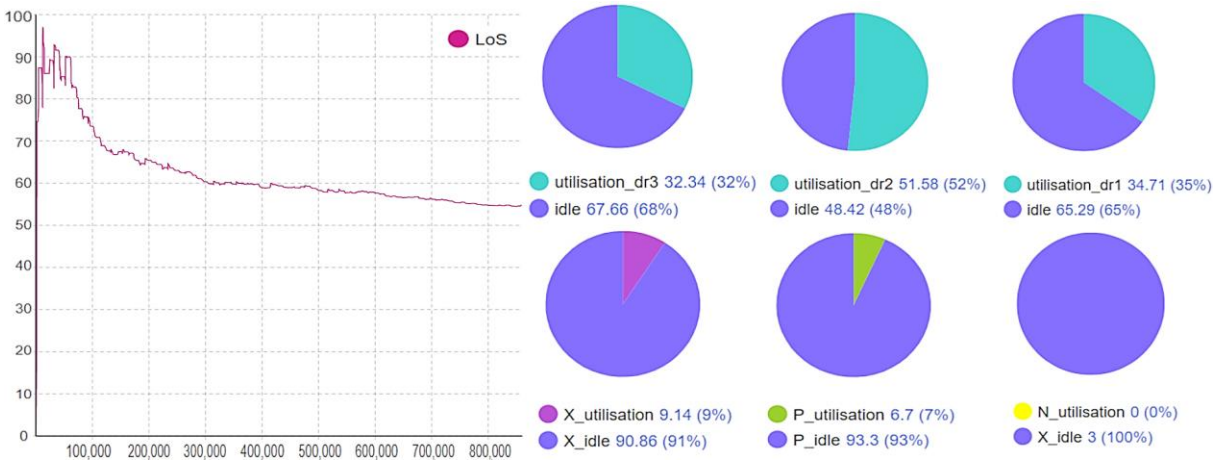


Figure 4: *LOS* and utilisation of resources.

The clinic simulation demonstrated individual doctor resource utilisation, with Dr1 at approximately 32 %, Dr2 at 52 %, and Dr3 at 35 % as shown in Fig. 4. These rates were calculated based on each doctor's actual clinic shifts. These utilisation rates suggest that Dr2 experienced a higher patient load compared to the others, possibly due to uneven patient distribution or more consistent and accurate data entries associated with Dr2. This pattern may also be attributed to Dr2 being scheduled more frequently across busier clinic days, as reflected in the temporal density of the patient data. Nurse utilisation appeared as zero due to the absence of tablet inputs in the DASHclinic app, which hindered full mean imputation during preprocessing. This also reflects the low presence of N in patient pathways, highlighting a data quality gap. X-ray and plaster technicians showed utilisation rates of 9 % and 7 %, respectively. The low utilisation can be due to insufficient training and data entry. The simulation also revealed queue sizes per station as demonstrated in Fig. 5; the lowest were associated with

nurses, reflecting their limited presence and missing data. The highest queue was observed at station D, with maximum queue sizes of four at P and eight at X. This suggests that station D represents a key bottleneck in the system, where variability in consultation durations and uneven patient distribution often result in longer queues at the doctor stations. The queue at X is likely due to X-ray being a single-resource point. In addition, the simulation estimated an average patient waiting time of 38.25 minutes. This value serves as a baseline reference for system performance, and future scenarios or optimisation efforts could aim to reduce this benchmark through improved scheduling or staffing strategies. The best-performing model, XGBoost, achieved an *MAE* of 31.40 and an *RMSE* of 41.69 minutes during real-time validation with upcoming patients. Considering the complexity and high variability of real-world hospital environments, where numerous unpredictable factors affect waiting times, a deviation of approximately 30 minutes can be considered acceptable and useful for decision support purposes. This reinforces the predictive value of the ML model under real clinical conditions.

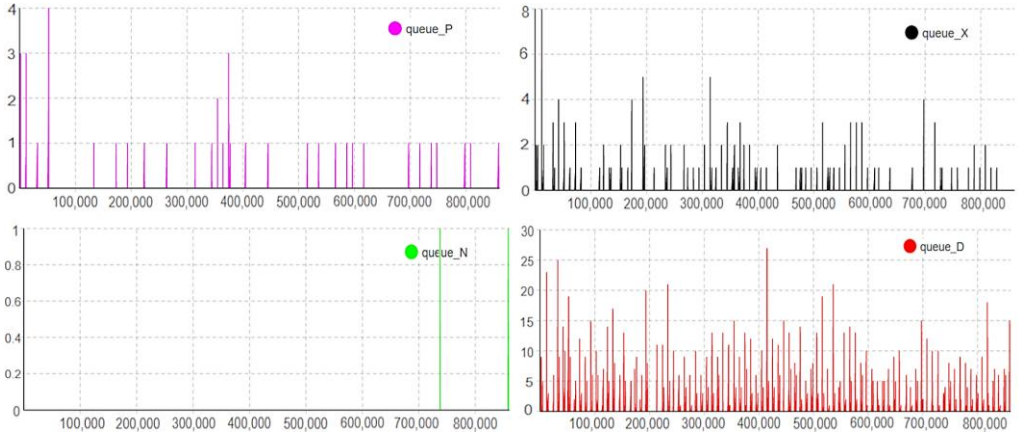


Figure 5: Queue size of the stations.

5. DISCUSSION AND CONCLUSION

This study addressed a research question by developing and evaluating a DT framework integrating ML predictions with DES. The proposed DT enables bidirectional information flow by providing patients with previously unknown waiting time predictions, validated against actual clinic data using performance metrics. Reflecting on the current literature, several studies have explored DES to model patient flows and ML techniques to predict healthcare outcomes. However, these approaches are often developed as standalone methods without cohesive integration under the concept of DT. Unlike previous DT studies in healthcare [15] and earlier works on waiting time prediction [22, 23], our approach uniquely integrates DES with a continuously updated ML model in a feedback-enabled DT capable of providing clinic-level decision support. DT implementation in healthcare faces several challenges: absence of synchronised data flows, lack of continuous model updating, limited interoperability between hospital systems and simulations, lack of bidirectional interaction, and difficulty validating DT models. This involves not only using ML to predict simulation inputs (e.g., waiting times) but also enabling feedback from simulation to update ML continuously. The key contribution of this study is the integration of ML and DES within a DT framework using AnyLogic and Python. This research contributes to knowledge by demonstrating that the combined use of ML and DES within a DT can deliver clinically relevant predictions, and identify operational bottlenecks, added value lies in patient-specific waiting time predictions and a feedback loop that improves ML via data updates and triggers automated notifications. KPIs from the DT, as shown in Section 4 (Table II, Fig. 4, Fig. 5), demonstrated both the predictive accuracy of the ML model (*MAE* = 13.67 min) and clinic-level operational insights, providing real-time

feedback. Dr2's higher utilisation rate (52 %) compared to Dr1 (32 %) and Dr3 (35 %) indicates an uneven workload distribution. One notable insight is the unexpectedly low staff utilisation, likely due to inconsistent or delayed data entries, mainly missing tablet inputs. This mostly affected nurses but also influenced doctors and technicians. As in Fig. 5, doctor bottlenecks suggest planning and data entry issues, while the X-ray bottleneck reflects a capacity constraint. The study is limited by the lack of real-time optimisation and the absence operational feedback. For future work, further investigation is needed to explore how a DT framework can contribute to more efficient appointment scheduling. Human-related inconsistencies in clinic may also affect input data quality and, in turn, reduce the accuracy of ML predictions. To address this, future implementations may benefit from integrating IoT devices to track patient movements and enhance data reliability. In addition, integrating check-in status into the app might help predict no-show probability, improving the DT concept to better reflect the real world. The communication channel used in this study (e-mail) can be extended to other forms such as SMS or mobile applications. This study shows that a data-driven DT can simulate outpatient workflows, predict patient waiting times, and provide foresight and feedback.

ACKNOWLEDGEMENT

This study was supported by the TÜBİTAK 2214-A International Research Fellowship Programme for PhD Students.

REFERENCES

[1] Latsou, C.; Ariansyah, D.; Salome, L.; Erkoyuncu, J. A.; Sibson, J.; Dunville, J. (2024). A unified framework for digital twin development in manufacturing, *Advanced Engineering Informatics*, Vol. 62, Part A, Paper 102567, 24 pages, doi:10.1016/j.aei.2024.102567

[2] Barricelli, B. R.; Casiraghi, E.; Fogli, D. (2019). A survey on digital twin: definitions, characteristics, applications, and design implications, *IEEE Access*, Vol. 7, 167653-167671, doi:10.1109/ACCESS.2019.2953499

[3] Wang, G.; Zhang, G.; Guo, X.; Zhang, Y. (2021). Digital twin-driven service model and optimal allocation of manufacturing resources in shared manufacturing, *Journal of Manufacturing Systems*, Vol. 59, 165-179, doi:10.1016/j.jmsy.2021.02.008

[4] Karanina, E. V.; Karaulov, V. M.; Konysheva, A. V. (2024). The concept of resilience-diagnosis of the regional ecosystem on the basis of the digital twin model, *International Journal for Quality Research*, Vol. 18, No. 2, 613-626, doi:10.24874/IJQR18.02-19

[5] Hassani, H.; Huang, X.; MacFeely, S. (2022). Impactful digital twin in the healthcare revolution, *Big Data and Cognitive Computing*, Vol. 6, No. 3, Paper 83, 17 pages, doi:10.3390/bdcc6030083

[6] Benevento, E.; Aloini, D.; Squicciarini, N. (2023). Towards a real-time prediction of waiting times in emergency departments: a comparative analysis of machine learning techniques, *International Journal of Forecasting*, Vol. 39, No. 1, 192-208, doi:10.1016/j.ijforecast.2021.10.006

[7] Vallée, A. (2023). Digital twin for healthcare systems, *Frontiers in Digital Health*, Vol. 5, Paper 1253050, 6 pages, doi:10.3389/fdgth.2023.1253050

[8] Hashmi, R.; Liu, H.; Yavari, A. (2024). Digital twins for enhancing efficiency and assuring safety in renewable energy systems: a systematic literature review, *Energies*, Vol. 17, No. 11, Paper 2456, 34 pages, doi:10.3390/en17112456

[9] Aldoseri, A.; Al-Khalifa, K. N.; Hamouda, A. M. (2023). Re-thinking data strategy and integration for artificial intelligence: concepts, opportunities, and challenges, *Applied Sciences*, Vol. 13, No. 12, Paper 7082, 33 pages, doi:10.3390/app13127082

[10] Rastpour, A.; McGregor, C. (2022). Predicting patient wait times by using highly deidentified data in mental health care: enhanced machine learning approach, *JMIR Mental Health*, Vol. 9, No. 8, Paper e38428, 11 pages, doi:10.2196/38428

[11] Joseph, J.; Senith, S.; Kirubaraj, A. A.; Ramson, J. S. R. (2022). Machine learning for prediction of wait times in outpatient clinic, *Procedia Computer Science*, Vol. 215, 230-239, doi:10.1016/j.procs.2022.12.026

[12] Zhang, Y.; Qin, G.; Aguilar, B.; Rappaport, N.; Yurkovich, J. T.; Pflieger, L.; Huang, S.; Hood, L.; Shmulevich, I. (2024). A framework towards digital twins for type 2 diabetes, *Frontiers in Digital Health*, Vol. 6, Paper 1336050, 10 pages, doi:10.3389/fdgth.2024.1336050

[13] Li, X.; Loscalzo, J.; Firoj Mahmud, A. K. M.; Mansour Aly, D.; Rzhetsky, A.; Zitnik, M.; Benson, M. (2025). Digital twins as global learning health and disease models for preventive and personalized medicine, *Genome Medicine*, Vol. 17, Paper 11, 14 pages, doi:10.1186/s13073-025-01435-7

[14] Siemens Healthineers. Case Study – Mater Private Hospital, Dublin, Ireland, from *https://cdn0.scrvt.com/39b415fb07de4d9656c7b516d8e2d907/58ca7e40bd68181a/03caef1c1035/siemens-healthineers_case_study_mater_private_2-06766281.pdf*, accessed on 21-08-2022

[15] Karakra, A.; Fontanili, F.; Lamine, E.; Lamothe, J. (2022). A discrete event simulation-based methodology for building a digital twin of patient pathways in the hospital for near real-time monitoring and predictive simulation, *Digital Twin*, Vol. 2, No. 1, 24 pages, doi:10.12688/digitaltwin.17454.1

[16] Moyaux, T.; Liu, Y.; Bouleux, G.; Cheutet, V. (2023). An agent-based architecture of the digital twin for an emergency department, *Sustainability*, Vol. 15, No. 4, Paper 3412, 13 pages, doi:10.3390/su15043412

[17] Pekarcikova, M.; Trebuna, P.; Matiscsak, M.; Kopec, J. (2024). Inventory management supported by Tecnomatix Plant Simulation tool, *International Journal of Simulation Modelling*, Vol. 23, No. 2, 251-262, doi:10.2507/IJSIMM23-2-682

[18] He, S. H. (2024). Coordination of production planning in multi-echelon supply chains: a simulation approach, *International Journal of Simulation Modelling*, Vol. 23, No. 4, 728-739, doi:10.2507/IJSIMM23-4-CO20

[19] Al-Asmari, A. S. (2022). Applications of deep learning to improve the quality of healthcare outcomes, *International Journal for Quality Research*, Vol. 16, No. 1, 77-92, doi:10.24874/IJQR16.01-05

[20] Arockiasamy, J. M. I. (2025). Proactive healthcare analytics: early detection of diabetes with SDOH insights and machine learning, *European Journal of Computer Science and Information Technology*, Vol. 13, No. 2, 64-74, doi:10.37745/ejcsit.2013/vol13n26474

[21] Aldhoayan, M. D.; Alobaidi, R. M. (2023). The use of machine learning to predict late arrivals at the adult outpatient department, *Cureus*, Vol. 15, No. 6, Paper e39886, 6 pages, doi:10.7759/cureus.39886

[22] Li, X.; Liu, W.; Kong, W.; Zhao, W.; Wang, H.; Tian, D.; Jiao, J.; Yu, Z.; Liu, S. (2023). Prediction of outpatient waiting time: using machine learning in a tertiary children's hospital, *Translational Pediatrics*, Vol. 12, No. 11, 2030-2043, doi:10.21037/tp-23-58

[23] Joseph, A.; Hijal, T.; Kildea, J.; Hendren, L.; Herrera, D. (2017). Predicting waiting times in radiation oncology using machine learning, *Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1024-1029, doi:10.1109/ICMLA.2017.00-16

[24] Çubukçuoğlu, C.; Nourian, P.; Sariyildiz, I. S. (2020). A discrete event simulation procedure for validating programs of requirements: the case of hospital space planning, *SoftwareX*, Vol. 12, Paper 100539, 8 pages, doi:10.1016/j.softx.2020.100539

[25] Golmohammadi, D.; Zhao, L.; Dreyfus, D. (2023). Using machine learning techniques to reduce uncertainty for outpatient appointment scheduling practices in outpatient clinics, *Omega*, Vol. 120, Paper 102907, 17 pages, doi:10.1016/j.omega.2023.102907

[26] Huo, J.; Shi, T.; Chang, J. (2016). Comparison of Random Forest and SVM for electrical short-term load forecast with different data sources, *Proceedings of the 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 1077-1080, doi:10.1109/ICSESS.2016.7883252

[27] Kavitha, R.; Dharshini, V. R. S.; Priyadharshini, M. (2024). Performance comparison of XGBoost and LightGBM gradient boosting algorithms in predicting cervical cancer risk, *Proceedings of the 2024 International Conference on Computing and Data Science (ICCDS)*, 6 pages, doi:10.1109/ICCDS60734.2024.10560387

[28] Tatachar, A. V. (2021). Comparative assessment of regression models based on model evaluation metrics, *International Research Journal of Engineering and Technology (IRJET)*, Vol. 8, No. 9, 853-860



https://dspace.org/

A machine learning-enabled digital twin for an orthopaedic clinic: a proof of concept

Ozen, N. Selin; Farsi, Maryam; Erkoyuncu, John Ahmet; Koyuncu, Melih; Gray, Jim

2025-12
DAAAM International
https://doi.org/10.2507/ijsimm24-4-747
Ozen NS, Farsi M, Erkoyuncu JA, et al., (2025) A machine learning-enabled digital twin for an orthopaedic clinic: a proof of concept. International Journal of Simulation Modelling, Volume 24, Issue 4, December 2025, pp. 671-682

Downloaded from a DSpace repository. https://dspace.org/