Software Testing, Verification and Reliability

WILEY

RESEARCH ARTICLE OPEN ACCESS

Configuration Testing of an Artificial Pancreas System Using a Digital Twin: An Evaluative Case Study

Richard Somers[1] | Neil Walkinshaw[1] | Robert Mark Hierons[1] | Jackie Elliott[2] | Ahmed Iqbal[2] | Emma Walkinshaw[2]

[1]Computer Science Department, University of Sheffield, Sheffield, UK | [2]Sheffield Teaching Hospitals, Diabetes and Endocrine CentreNorthern General Hospital, Sheffield, UK

Correspondence: Richard Somers (rsomers1@sheffield.ac.uk)

Received: 15 May 2024 | **Revised:** 19 November 2024 | **Accepted:** 10 January 2025

Funding: Neil Walkinshaw and Robert Mark Hierons are funded by the UKRI, United Kingdom, CITCOM Grant EP/T030526/1. Hierons is funded by the UKRI, United Kingdom, Grants EP/R025134/2 and EP/V026801/2.

Keywords: configuration | digital twins | insulin pumps | medical devices | modelling | software testing

ABSTRACT

The recent growth in popularity of wearable medical devices has improved the quality of life of people with medical conditions. Testing such devices may require users to configure these systems using physical trials, putting themselves in potentially dangerous scenarios. Misconfiguration of such devices has caused disease misdiagnoses and incorrect drug prescriptions. Digital twins have been proposed as an opportunity to reduce such risks of testing system configurations in simulated environments, decoupling the user from the system under test. In this paper, we perform an evaluative case study to assess the use of a digital twin for configuration testing of an artificial pancreas system (APS) control algorithm. These systems regulate the blood glucose levels in people with type 1 diabetes mellitus, and so misconfigurations can cause severe hypoglycaemia or hyperglycaemia, which can be life-threatening. We tested the OpenAPS control algorithm against 156 people's clinical data. We found that our digital twin provided an accurate simulation environment to perform configuration testing and accurately predict blood glucose–insulin behaviour. We evaluated different APS configurations, identifying a potentially unsafe configuration without the risks associated with a physical trial. We identified the challenges associated with modelling clinical data, which could lead to misinterpretations in configuration testing and the reduction of test reliability when modelling stochastic body dynamics.

1 | Introduction

In recent years, the use of wearable medical devices has rapidly grown in popularity [1]. These devices allow for health monitoring, chronic disease management, diagnosis, treatment and rehabilitation [2]. Personalized configurations of medical devices allow a user to unobtrusively manage medical conditions with treatments that are specific to them [3].

Misconfiguration of medical devices can lead to users being put in potentially dangerous scenarios. Recent works have found that this can cause misdiagnoses of diseases [4] and drugs being overprescribed [5]. One such example involved the configuration of an insulin pump being misinterpreted. This misinterpretation led to 100 times the expected insulin dosage being administered to a user [5].

Misconfiguration of software systems is an issue affecting systems across several domains [6–9]. Configuration testing is a software testing approach that aims to remedy this challenge by evaluating specific system configurations during software testing [10]. This presents a way of identifying incorrectly behaving configurations before releasing them into production.

Configuration testing, however, is challenging with medical devices due to there being a human-in-the-loop. Potentially

https://doi.org/10.1002/stvr.70000

dangerous configurations would put a user at risk and physical testing, especially with a human-in-the-loop, may require extensive set-up times [11, 12]. As with the example of the misconfigured insulin pump [5], this could expose users to potentially life-threatening system configurations.

Digital twins [13] present a potential solution to the challenge of configuration testing. Digital twins simulate a physical entity, traditionally a cyber-physical system, to enhance the system's behaviour through monitoring, evaluation and prediction [14]. These abilities allow for the assessment and prediction of behaviours in a simulated environment. As a result, digital twins reduce the need for physical testing [15].

Digital twins have been proposed to optimize medical devices [12] by providing personalized simulation. Simulated environments present an opportunity to perform configuration testing without the human-in-the-loop. This has been suggested as a means to reduce the risk of potentially dangerous scenarios when configuring such devices [16]. Digital twins have been proposed to test medical devices using the following steps: obtaining clinical data, generating a digital twin and fitting it to represent this data, validating the digital twin's model and, finally, using the digital twin to evaluate human responses to clinical interventions [12].

Digital twins appear to present a useful basis for testing hard-to-test systems such as cyber-physical systems. However, they also give rise to several practical questions. Depending on the domain, it can be inherently difficult to ensure that a digital twin is an accurate representation of the system [12, 17]. This then gives rise to the question of how one can determine whether a digital twin can support the testing of software systems they are interacting with. At the time of writing, there are no published research or experience reports that can provide this practical insight.

In this paper, we perform an evaluative case study to assess a digital twin's ability to aid in the configuration testing of medical devices. To accomplish this, we adapt an existing, explainable model of the blood glucose–insulin dynamics from a healthy pancreas which we can be personalized to observational data. This enabled us to implement a digital twin which can be personalized to represent a person with type 1 diabetes mellitus (T1DM) using an artificial pancreas system (APS). We produce a fitting strategy for our digital twin and train it against the largest open-source T1DM dataset, which has 156 users. By fitting the model, we perform an evaluative case study in which we assess the model's ability to represent personalized T1DM dynamics, identify the extent to which digital twin predictions are accurate and perform configuration testing of a widely used APS without requiring clinical trials. Through our study, we make the following contributions:

- We perform an evaluative case study by implementing a proposed methodology, investigating the extent to which a digital twin can be used to simulate a human-in-the-loop during configuration testing, uncovering the challenges of fitting a complex model to represent the behaviour of blood glucose–insulin dynamics using clinical data and highlighting the requirements for more controlled data sources for future implementations.
- To facilitate this, we implement a digital twin by adapting an existing blood glucose–insulin model to be used in the configuration testing of a widely used, open-source APS. We modify the model to represent a body with T1DM and develop a feedback loop between the model and the APS.
- We assess the capability of our digital twin to perform configuration testing of an APS. As a result, we identify potentially dangerous behaviours from misconfigured blood glucose targets without requiring clinical trials. Unsafe behaviours could be observed and explained in a simulated environment while being personalized to the user and yet decoupled from the human-in-the-loop.
- We demonstrate the potential threats to the technique associated with using observational clinical data. Inconsistent manual carbohydrate recording and sensor error recording nonphysiologically possible values required extensive data cleaning and resulted in very little of the data being suitable for fitting the model. We identify the need for techniques which can deal with such uncontrolled data sources or more curated data sources.
- We provide a comprehensive replication package of our case study including the derived digital twin, fitting strategy and the evaluation scripts used to drive the evaluative case study.

From these contributions, we uncover the advantages of employing digital twins for configuring medical devices, while also mapping out the essential steps needed to alleviate challenges within this domain. We are able to demonstrate how different configurations can be trialled in a simulated environment, isolated from the user to examine potentially dangerous system behaviour. We also found that only 2.37% of the data in the largest APS dataset is usable for our implementation, highlighting the challenge of working with clinical data and the requirement for models which can accommodate it.

The remainder of this paper is set out as follows: Section 2 presents the necessary background required for this study. Section 3 describes the adaptation of an existing blood glucose–insulin model for the implementation of our digital twin, and Section 4 outlines the rationale and methodology for our case study. Sections 5 and 6 answer the research questions (RQs) and discuss their potential impact on medical device testing. Section 7 presents the related works, and Section 8 concludes the paper.

2 | Background

In this section, we present the motivating context for our study: the difficulties related to testing configurations of APSs and the potential consequences of misconfiguration. We describe the importance of configuration testing and present digital twins in healthcare as a potential solution to the difficulties of performing this testing approach on APSs. We also explore a proposed digital twin–based optimization approach that we use to test medical devices as the basis for our case study.

2.1 | APSs

APSs provide a modern approach to managing T1DM. They achieve this through the use of a continuous glucose monitor (CGM) sensor, an insulin pump and a control algorithm [18]. Figure 1 illustrates how these systems are used to create a feedback loop between the body and insulin pump. As a result, this system imitates a healthy pancreas.

Having a human-in-the-loop, however, makes testing configurations of APSs very challenging. Typically, APS systems are configured by trained professionals, informed by knowledge obtained by clinical trials [19]. However, if set up incorrectly or if the control algorithm is faulty, an APS can cause blood glucose levels to leave the safe glycaemic range [20] by injecting too much or too little insulin. This could induce hypoglycaemia or hyperglycaemia which can cause life-altering conditions, such as stroke and heart disease, and if severe, can be life-threatening [21, 22].

2.1.1 | OpenAPS

Recently, there has been an increase in the use of 'do-it-yourself' (DIY) APS implementations [23, 24]. One such implementation, OpenAPS [25], provides an open-source control algorithm called oref0.[1] The control algorithm can be downloaded, compiled and executed to create a feedback loop between CGMs and insulin pumps [26]. oref0 has multiple different versions with different features [27] which improve functionality through updates to the APS control algorithm and provide additional tools for research and ease of use.

oref0 presents a highly configurable environment, where users require personalized configurations. The oref0 documentation presents 13 commonly changed parameters, from 47 parameters in total. The documentation also presents how a single person requires multiple different configuration profiles for different times of day and activities [28]. For example, exercise requires a specific configuration profile that has six additional exercise-specific parameters.

The *blood glucose target* of oref0 is an example of such a configuration, which defines the value that the APS attempts to keep the user's blood glucose at. The target may vary with different activities, such as sleep and exercise, as well as different people. Testing different configurations of the *blood glucose target* may produce undesirable and potentially dangerous behaviours if misconfigured. Performing this testing with a human-in-the-loop would enact these behaviours on the user.

The documentation for the iOS implementation of OpenAPS, Loop [29], contains the following quote: 'You can count on your fingers the number of doctors in the US who are capable of properly adjusting settings for Loop. You can probably count on your fingers and toes the number worldwide who can successfully help you with Loop settings'. This is an indicator of how much prior knowledge is required to correctly configure these systems. If an APS is misconfigured, it can have severe consequences [21, 22].

OpenAPS has a thriving community across GitHub issues [30], Gitter [31], Facebook groups [32] and a Google group [33]. At the time of viewing, one of the main forums for APS configuration with over 30,000 users [32] contained multiple daily posts where users were asking for advice with configuration, struggling with code compilation and having difficulties integrating with other applications and hardware. From this, we identified that the configurability of oref0 is a challenge. As found in Section 2.1, misconfiguration could lead to potentially life-threatening scenarios.

These configuration difficulties are not only faced by DIY implementations of APSs but are more widely faced by medical devices [2]. Closed-source APS devices have recently posed a challenge for system configuration. A recent example presented a clinically approved device parsing configuration inputs incorrectly [5]. In some cases, this caused potentially dangerous amounts of insulin to be administered.

2.2 | Configuration Testing

Incorrect system configuration is one of the leading causes of system failures [10]. Due to the sheer complexity of troubleshooting

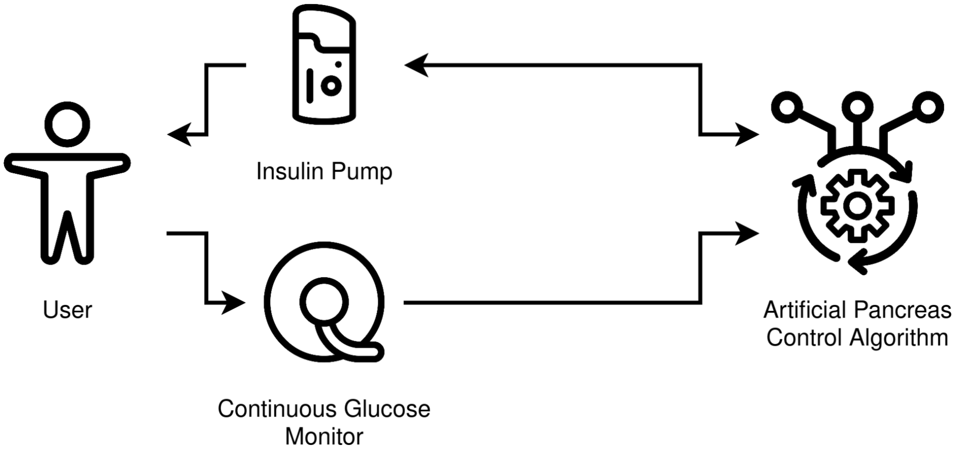

Insulin Pump
User
Continuous Glucose Monitor
Artificial Pancreas Control Algorithm

FIGURE 1 | The interaction between an APS and its user. Blood glucose data are measured by the CGM and sent to the APS control algorithm. The algorithm then, along with historical insulin data, suggests insulin prescription. The insulin pump prescribes the insulin based on the algorithm's output.

configuration errors [34], ensuring the reliability of system configurations can be a challenge. Misconfiguration has led to large-scale software failures across services [6, 7] and led to greater vulnerability to cyber-attack [8, 9].

Configuration testing is an active area of software testing [35]. It is not a test generation technique but a procedure for ensuring that new configuration values lead to correct system behaviour [36]. When a system configuration is changed, unit and integration tests should be executed in an isolated development environment with the new configuration before the new configuration is applied to the production system. Configurations that enable erroneous system behaviour can be identified before they are deployed, reducing the potential risks associated with system misconfiguration.

Configuration testing does not attempt to 'cover' the configuration space, as is already done by combinatorial testing techniques [37]. Instead, it tests system executions with selected configurations expected for system deployment [35].

In the context of an APS, we identified in Section 2.1.1 that the blood glucose target is a configuration that can result in dangerous scenarios if misconfigured. An APS can be executed with expected configurations of the blood glucose target in order to test the resulting behaviour.

However, applying configuration testing to a medical device, such as an APS, can be challenging. The human-in-the-loop is necessary for the device to function correctly. Unfortunately, this makes testing configurations which may lead to incorrect behaviour potentially dangerous to the user. We require a way of decoupling the human-in-the-loop from the system to safely test configurations of these systems and assess their behaviour.

In Section 5.3, we apply configuration testing to an APS as part of our evaluative case study. From this, we aim to investigate whether configuration testing can be used to identify safe and unsafe behaviours of an APS from existing configurations.

2.3 | Digital Twins

A digital twin is a simulation model that runs in parallel to a physical entity. Digital twins change with their physical counterpart [38]. As a result, they present a virtual replica of the system on which the behaviour from interventions can be predicted. Insights can be gained from the simulation to enhance the behaviour of the physical twin in the real world [13, 14]. Examples of this enhancement include real-time visualization [39] and physical degradation prediction [40].

Ensuring confidence in digital twin model predictions is paramount to their trustworthiness. Using an explainable model allows for a more informed confidence in predictions regarding the physical system as domain experts are able to fully understand the causes of potentially faulty model behaviour. Digital twins have achieved this through physics-driven and explainable AI approaches [13, 41]. Prior work, however, has found that these predictive capabilities are not widely used, especially when validating system behaviour [15].

2.3.1 | Healthcare Applications

The emergence of digital twins in healthcare presents an opportunity for APSs. Typically, APSs exist in an environment with a human-in-the-loop. This can make testing the configurations of the control algorithms a challenge due to different treatment reactions from individuals [12]. The ability to perform configuration testing in the personalized and simulated environment of a digital twin could reduce the risks of hypoglycaemia and hyperglycaemia from incorrect configuration.

Digital twins have been proposed to provide personalized medicine by enabling added safety through simulation and improved explainability of treatments [17, 42]. Computational modelling [43, 44] and digital twins [12, 45] are an emerging technology in healthcare as a part of Healthcare 4.0 [46, 47]. Digital twins present an ability to adapt and trace clinical interventions [48] as well as decoupling the human-in-the-loop through simulation [16]. Such decoupling presents an opportunity for medical device configuration testing. The system itself can be evaluated across different configurations without putting the user in potentially dangerous scenarios.

2.3.2 | Testing in Healthcare

Corral-Acero et al. [12] propose an approach for the optimization of clinical devices through the use of a personalized explainable model that capitalizes on the predictive power of digital twins. This allows the model to predict human responses to clinical intervention without putting the user in potentially dangerous scenarios. We summarize the approach outlined by Corral-Acero et al. as follows:

1. Obtain clinical data, medical images or other context specific information.
2. Generate a mechanistic or statistical model to replicate the mechanics of the body dynamics being modelled.
3. Calibrate and optimize this model based on the user's data.
4. Validate the model's accuracy.
5. Present human responses to clinical interventions based on model predictions

These steps present a theoretical framework for optimizing clinical devices based off digital twin predictions. We use this approach in the context of configuration testing for wearable medical devices. In our implementation, we aim to use predictions from the digital twin to perform configuration testing in an environment isolated from the user. This aims to provide insight into the behaviour of medical device configurations without the potential risk of physical trials.

2.4 | Modelling Blood Glucose–Insulin Dynamics

For our evaluative case study in a future section, we require an explainable model that models the blood glucose–insulin dynamics of a person. For this, we present work by Contreras et al. [49] in which they define a model for representing the blood glucose–insulin dynamics within a healthy body that is *not* affected by T1DM. This model uses differential equations to provide an explainable, physics-driven representation of blood glucose–insulin dynamics. The model is supported by an extensive sensitivity analysis [49] and has been used to inform methodologies in recent works [50]. This model is presented in Equation (1).

$$
\begin{aligned}
\frac{dS}{dt} &= -k_{js}S,\\
\frac{dJ}{dt} &= k_{js}S - k_{gj}J - k_{jl}J,\\
\frac{dL}{dt} &= k_{jl}\varphi(t) - k_{gl}L(t),\ \varphi(t) = \begin{cases} 0, & \text{if} \, t < \tau \\ J(t-\tau), & \text{if} \, t \geq \tau \end{cases},\\
\frac{dG}{dt} &= -(k_{xg} + k_{xgi}I)G + G_{prod} + \eta(k_{gj}J + k_{gl}L), \qquad (1)\\
\frac{dI}{dt} &= k_{xi}I_b\left(\frac{\beta^{\gamma}+1}{\beta^{\gamma}\left(\frac{G_b}{G}^{\gamma}+1\right)} - \frac{I}{I_b}\right),\\
G_{prod} &= \frac{k_{\lambda}}{\frac{k_{\lambda}}{G_{prod0}} + (G - G_b)}.
\end{aligned}
$$

The differential equations of the model represent the process of carbohydrates being consumed and the subsequent insulin–glucose dynamics. The differential equations are used to calculate the amount of carbohydrates in the stomach (S), ilium (L) and jejunum (J), as well as the blood glucose level (G) and insulin on board (I). This allows for representation of carbohydrates as they move from the stomach and are absorbed across the Ilium and Jejunum in the small intestine. These equations show how carbohydrate absorption, insulin production and hepatic glucose production (G_{prod}) regulate blood glucose levels. The rate of these dynamics are represented by the remaining constants including kinetic constants (k_{js}, k_{xi}, etc.), steady states (G_b, I_b), a time delay (τ) and insulin production scales (β, γ).

This model provides explainability by representing physiological behaviour with transparent mathematical equations. Figure 2a demonstrates a scenario in which this model is used to represent the glucose–insulin dynamics for a person with normal insulin sensitivity consuming carbohydrates. Intuitively, a person with a lower insulin sensitivity should have a higher blood glucose level over time. We can simulate this by changing the insulin sensitivity constant in the model, the result of which is represented in Figure 2b.

However, this model does not represent the blood glucose–insulin dynamics of a person with T1DM and is not a digital twin. In the following section, we take this model and adapt it to represent the dynamics of T1DM. We then interface it with oref0 to generate a digital twin that can be used to predict user responses to oref0 interventions.

2.5 | Summary

In this section, we highlighted the difficulties associated with configuring an APS. Human interaction makes physical testing of configurations time consuming and potentially unethical for dangerous behaviours. Digital twins have been proposed to enable such testing, removing the human from the loop and running the APS in a simulated environment. However, such an approach has yet to be implemented, raising the question as to whether a digital twin would alleviate the challenges associated with testing APS configurations.

For the remainder of this paper, we perform an evaluative case study in order to assess the applicability of the methodology proposed by Corral-Acero et al. [12], described in Section 2.3.2. We use these steps to develop a digital twin of a person using an APS, use a large open-source T1DM dataset to calibrate the digital twin and then perform configuration testing using real configurations used by users.

3 | A Digital Twin for Testing an APS

In Section 2, we discussed the challenge of testing and configuring an APS. Digital twins have shown promise in using predictions to decouple testing from the human-in-the-loop in other domains. For blood glucose–insulin dynamics, only models are currently available. To enable configuration testing in a safe environment, we require a digital twin that is capable of predicting user responses to clinical interventions. The key task is to provide an environment where new configurations can be trialled, without interacting directly with the user.

We use the five steps in the framework set out by Corral-Acero et al. [12], presented in Section 2.3.2, to guide this implementation. We first adapt the model outlined in Section 2.4 to represent people with T1DM (Step 2). We then devise a strategy to fit the large number of model constants. This allows the model to simulate real-world blood glucose–insulin dynamics (Steps 3 and 4). Using this model, we complete the digital twin by interfacing it with the oref0 APS algorithm. As a result, we can observe how a person would be affected by different configurations of an APS without requiring clinical trials (Step 5). Step 1 of the framework, gathering data, is explored in Section 4.2.

Figure 3 illustrates our approach. The model of a person with T1DM represents the blood glucose dynamics of a user by fitting its parameters based on their blood glucose, carbohydrate and insulin history. The user then provides an initial blood glucose, carbohydrate and insulin value, henceforth referred to as a *scenario*, for which they would like to observe APS interaction over time with the model. The model periodically sends its current state to the APS and simulates any suggested insulin interventions. A user can explore different APS configurations, observing the impact of potentially dangerous scenarios within a simulated environment.

3.1 | Adapting the Contreras Model to Represent People With T1DM

To accurately represent the blood glucose–insulin of a person with T1DM, we first adapted the model proposed by Contreras et al. [49], presented in Section 2.4, to represent the physiology of T1DM. This can be seen as Step 2 of the Corral-Acero et al. [12] approach, generating a mechanistic model. We use this section to describe the required changes to the model so it can then be fit to real-world data in the following section.

For our evaluative case study, we focus on the implementation of a mechanistic model to predict and enable personalized medicine. Such models allow for prediction of unseen outcomes due to their encapsulation of human physiology [51]. Corral-Acero et al. [12] also propose the use of statistical models in digital twins. However, they are proposed to be used for when the underlying behaviours are not well understood. Future work could be performed to evaluate the introduction of a statistical model to complement the mechanistic model, but we find this out of scope for this case study.

The adapted model makes the simplifying assumption that a person with T1DM does not produce any insulin. We illustrate

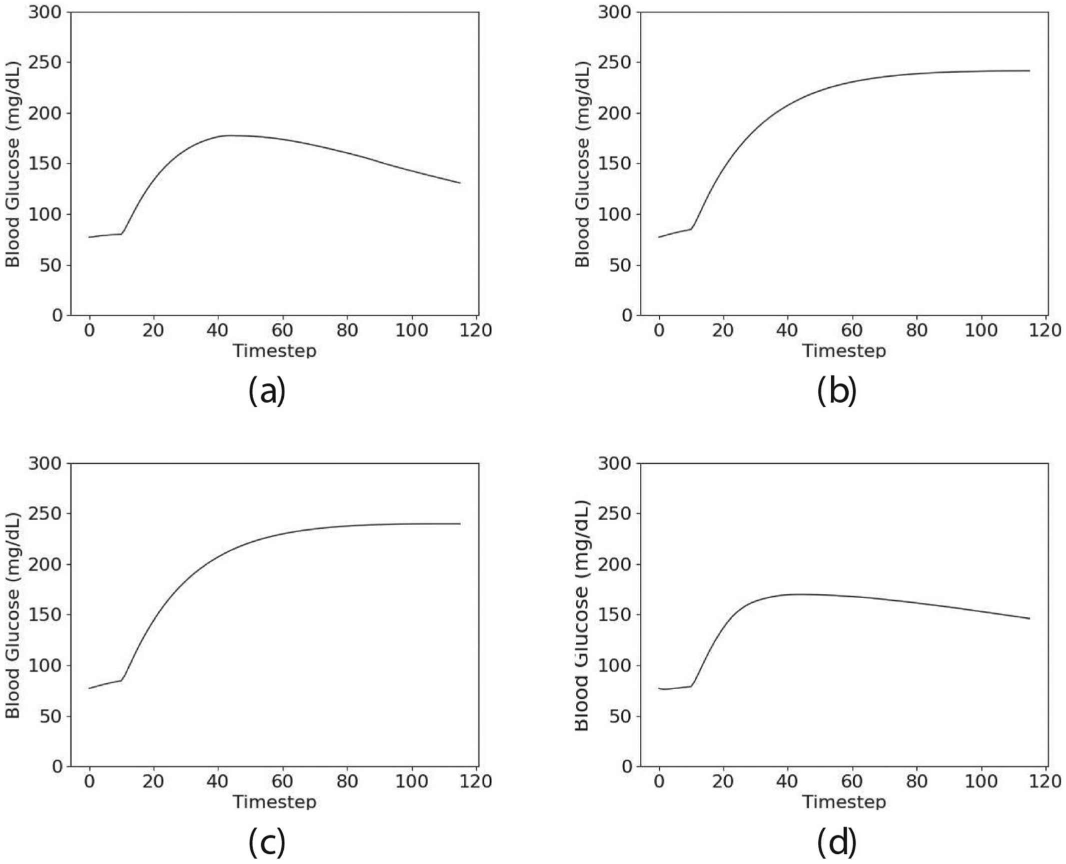


FIGURE 2 | Model outputs presenting blood glucose dynamics over time for an initial blood glucose, insulin and carbohydrate value (a scenario). (a) Blood glucose dynamics as defined by Corral-Acero et al. [12] for a healthy pancreas. (b) presents the same scenario with suppressed insulin sensitivity. (c) presents the same scenario with 1% insulin secretion. (d) presents the same scenario with suppressed insulin sensitivity when interacting with oref0.

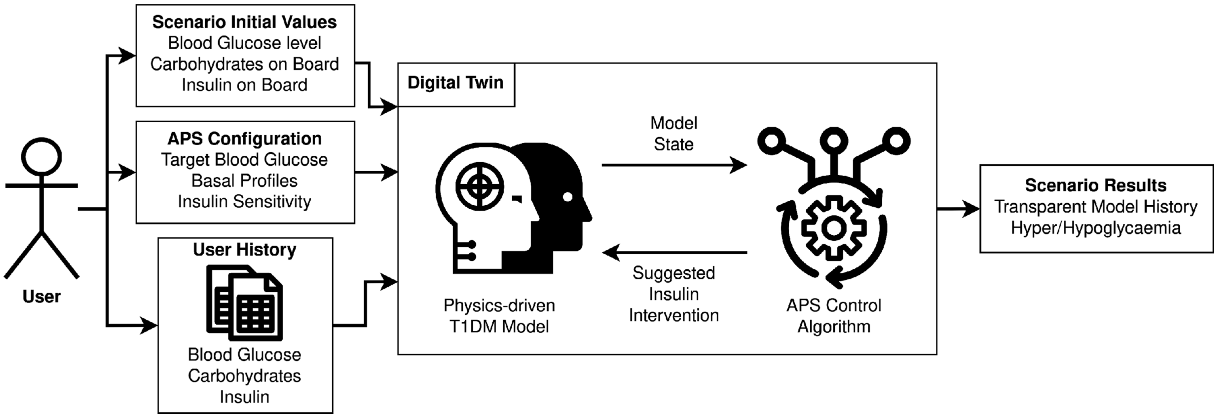


FIGURE 3 | The process of a digital twin of a person with T1DM using an APS simulating user provides scenarios for different APS configurations. A model is generated based on the user's historical data representing their blood glucose–insulin dynamics. This model communicates with an APS control algorithm, using the user supplied APS configurations, to simulate blood glucose dynamics for user generated scenarios.

this change in Equation (2). We set the insulin production term $\left(\frac{\beta^\gamma+1}{\beta^\gamma(\frac{G_b}{\tilde{G}}^\gamma+1)}\right)$ of this equation to zero. This allows the insulin steady state (I_b) to cancel out, resulting in the simplification of insulin dynamics to the rate of insulin degradation (k_{xi}).

$$\frac{dI}{dt} = k_{xi}\xcancel{I_b}\left(\xcancel{\frac{\beta^\gamma+1}{\beta\gamma(\frac{G_b}{\tilde{G}}^\gamma+1)}} - \frac{I}{\xcancel{I_b}}\right) = -k_{xi}I \tag{2}$$

In practice, some people with T1DM do produce a small amount of insulin and are known as 'microsecretors'. Januszewski et al. [52] measured the concentration of insulin-producing beta cells in people with T1DM compared with a control group. They found that, on average, 55.3% of people with T1DM secrete insulin, but at a level less than 1% of the control. Figure 2c presents how there is no noticeable difference between 1% insulin secretion and no insulin absorption, as in Figure 2b. As such, secretion would not noticeably impact the adapted model's blood glucose–insulin dynamics.

$$\begin{aligned}
\frac{dS}{dt} &= -k_{js}S,\\
\frac{dJ}{dt} &= k_{js}S - k_{gj}J - k_{jl}J,\\
\frac{dL}{dt} &= k_{jl}\varphi(t) - k_{gl}L(t),\ \varphi(t) = \begin{cases} 0, & \text{if} t < \tau \\ J(t-\tau), & \text{if} t \geq \tau \end{cases},\\
\frac{dG}{dt} &= -(k_{xg} + k_{xgi}I)G + G_{prod} + \eta(k_{gj}J + k_{gl}L),\\
\frac{dI}{dt} &= -k_{xi}I,\\
G_{prod} &= \frac{k_\lambda(G_b - G)}{k_\mu + (G_b - G)} + G_{prod0}.
\end{aligned} \tag{3}$$

As with the original model by Contreras et al. [49], we define a person's internal mechanics by the differential equations presented in Equation (3).[2] As a person's internal mechanics are specific to them, we define the constants for these equations constants in Equation (4) as the set $K_x(T)$, where x is a specific person. This set is bounded by the time period T because the internal mechanics change over time based on factors, such as the time of day and exercise. The meaning of each constant can be found in Table A1 in Appendix A. This presents a mechanistic approach (Step 2 of Corral-Acero et al. [12]) to calculate a person's blood glucose–insulin dynamics based on a set of personalized constants.

$$\begin{aligned}
K_x(T) = \ & \{k_{js}, k_{gj}, k_{jl}, k_{gl}, k_{xg}, k_{xgi}, k_{xi},\\
& \tau, \eta, k_\lambda, k_\mu, G_{prod0}\}.
\end{aligned} \tag{4}$$

3.2 | Fitting the Model to a Person With T1DM

Because blood glucose–insulin dynamics vary from person to person, the model will need to be tailored to an individual. To achieve this, a large number of constants ($K_x(T)$) must be set to their correct values using historical blood glucose–insulin data. Contreras et al. [49] provide error equations but does not explicitly show how they can be used for parameter fitting. This equates to Steps 3 and 4 of the approach outlined by Corral-Acero et al. [12].

First, we take a trace from our data source which will be used as the training data for fitting. This trace represents the desired behaviour our model should represent after training. From this trace, we can extract the interventions applied to the human body. These include how much and when the user has eaten, if insulin was manually injected, and the insulin injected by oref0. Such interventions can then be applied at the relevant time steps to the model during training. For example, if the user injected insulin after consuming a meal, the model would be trained using both of these extracted interventions because the blood glucose–insulin dynamics would represent the resulting behaviour.

We then use metaheuristic search, executing the model with different constant values, to find the values of $K_x(T)$ which best represent our desired behaviour. From this, we have fit our model, allowing for the blood glucose–insulin dynamics of the data source trace to be represented. The fitting process is available in our replication package (Section 4.4).

We use a genetic algorithm to minimize the error between the training dataset and the model output for the parameters $K_x(T)$. Genetic algorithms are metaheuristic optimization algorithms that use chromosomes to represent different solutions [53]. For this paper, we used the genetic algorithm PyGAD [54] as it provided an open-source implementation of a highly configurable optimization algorithm. The configuration of our genetic algorithm is presented in Table 1. In our case, each chromosome is a configuration of $K_x(T)$, and the relative fitness of each chromosome is defined by the error of the resulting model compared to the training data. Low error chromosomes are assigned a high fitness and stochastically combined and mutated to search the solution space to optimize the configuration of $K_x(T)$. Other metaheuristic algorithms were trialled for this evaluative case study, and the results to those preliminary tests are available in Appendix B.

To determine the accuracy of each chromosome, we calculate the root mean squared error (RMSE), presented in Equation (5). This provides an error value based on the difference between the model outputs ($z_m(t)$) and the observational training data ($z_o(t)$) across a given time period (T). RMSE provides an error that is always positive and does not increase with the number of model timesteps.

TABLE 1 | The configuration of PyGAD used when fitting our model. This was derived from preliminary experimentation.

Generations	1500
Solutions per generation	30
Parent selection	Elitism
Number of mating parents	4
Mutation percentage	20%

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=0}^{T}(z_o(t) - z_m(t))^2}. \tag{5}$$

Differential equation models can introduce oscillations that are not possible in physiological systems [49]. To reduce unnatural oscillatory behaviour, we measure the oscillation error of the model using the Equation (6). This value is scaled by an oscillation constant (k_{osc}) to reduce its impact on the overall error, allowing for physiologically correct oscillations when they existed in the data.

$$e_{osc} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(z_m(t) - z_m(t-1))^2}. \tag{6}$$

Our final error function for model optimization is the sum of RMSE and the scaled oscillation error:

$$e_m(T) = RMSE + k_{osc}\, e_{osc}. \tag{7}$$

Using $e_m(T)$ as the error function of a genetic algorithm, we generate configurations for $K_x(T)$ that best represent the observational training data. The training dataset used in the fitting of our model was the OpenAPS Data Commons, presented in Section 4.2.1. We extracted blood glucose, insulin and carbohydrate values from these data to create a time series from which to train our model.

This technique produces models that represent a person's historical blood glucose, carbohydrate and insulin levels, as illustrated in Figure 3.

3.3 | Digital Twin of a Person Using an APS

Now, we have generated a fitted model, we interface this with an APS control algorithm. As a result, we create a digital twin of a person with T1DM using an APS. We illustrate this as the interaction between the model and control algorithm in Figure 3. This interaction allows a user to simulate interactions between their model and the APS to observe how APS interventions would affect their blood glucose levels over time. This equates to Step 5 of the approach outlined by Corral-Acero et al. [12].

We incorporate the oref0 control algorithm [25], as described in Section 2.1, as part of our digital twin to predict blood glucose dynamics when interacting with an APS. oref0 is a JavaScript program made up of utility scripts for creating an APS pipeline invoked by shell scripts. Figure 4 presents a sequence diagram representing the pipeline used by our digital twin to invoke oref0 and the data required at each step. The constant exchange of information between the model and control algorithm creates a feedback loop which models the APS' effect on the person's blood glucose dynamics over time. This mimics the data flows from CGMs and insulin pumps to an APS control algorithm. The code for this exchange of data can be found within our replication package (Section 4.4). Following Figure 4, our digital twin interfaces with oref0 as follows:

1. The scenario runner fetches the current state of the blood glucose insulin model and then invokes an oref0 pipeline wrapper which manages the interaction with oref0 scripts. This passes in the model's current and past states into the wrapper, as reflected in Figure 3, and any oref0 configurations from the user.
2. The wrapper invokes the *oref0-calculate-iob* script passing in the data seen in Figure 4. This invokes oref0 to infer the insulin on board based on the state of the model for future calculations.
3. The wrapper then invokes the *oref0-meal* script to generate a file containing all the carbohydrate on board data required for future calculations. Figure 4 presents the data required for this step.
4. The wrapper calls a final script, *oref0-determine-basal*, passing in all the gathered and calculated data required. This script generates a suggestion for the insulin requirements of the user. The wrapper interprets this output and returns a value of insulin which should be 'injected' into the model.
5. The scenario runner adds the suggested insulin to the model and then updates the model for the next five timesteps (5 min) to simulate an insulin pump injecting insulin over time. This process is repeated every five timesteps until the simulation concludes.

The transparency of the blood glucose model makes simulations explainable by allowing the internal state to be mathematically calculated for any timestep. Effects of an APS control algorithm intervention can be trialled in a controllable environment and traced back through the model without the need for physical trials which may be expensive or hazardous to the user.

An APS should aim to mimic a healthy pancreas as closely as possible. To test this, we refer back to Figure 2a of a person with a healthy pancreas. Figure 2d presents the same scenario replacing the human with our digital twin. We observe a negligible difference between the healthy and APS supported blood glucose curves, presenting oref0's ability to stabilize blood glucose levels.

4 | Case Study Design

In this section, we present our case study to explore the suitability of using a digital twin for configuration testing of an APS. We also attempt to highlight any limitations and difficulties which may threaten future implementations. We follow the case study protocol outlined in Runeson et al. [55] and the ACM empirical case study standards [56]. First, we reiterate the motivations for the case study to justify our case selection. Then, we present the RQs which will drive our evaluation. From this, we are able to identify any data required. Finally, we outline any further measures required to answer the RQs.

As presented in Section 2.1, APSs present a challenging software environment within which to find correct software configurations, due to the human-in-the-loop. Incorrect configuration of the control algorithm can be potentially dangerous

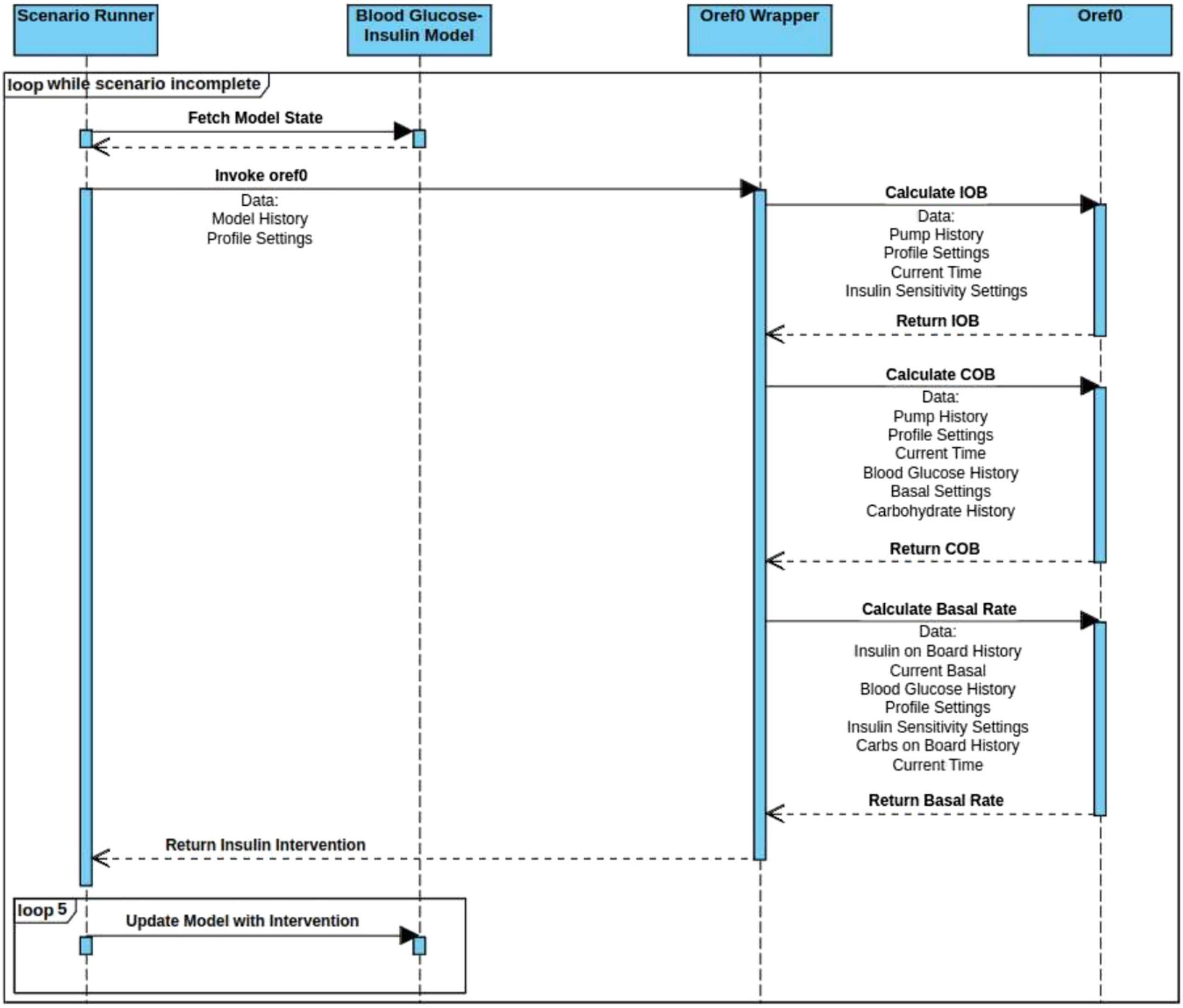


FIGURE 4 | A sequence diagram presenting how our model interacts with oref0 and the data required at each step. First, the insulin on board is calculated, then carbohydrates on board is calculated and finally the suggested insulin basal rate is calculated. These data are then fed back into the model to simulate an insulin pump supplying insulin.

for the user. Digital twins are a promising approach to reducing risk when configuring human-interfacing devices but this technique has yet to be implemented.

We chose to evaluate our case study on the configuration testing of oref0. Section 2.1.1 presents how highly configurable the control algorithm is and the challenges and potential consequences associated with misconfiguration. oref0 also presents open-source data availability in terms of its source code and the OpenAPS Data Commons, outlined in Section 4.2.1. The remainder of this study presents a Healthcare 4.0 inspired case study to identify the benefits and challenges associated with configuration testing for an APS control algorithm using a digital twin.

4.1 | RQs

In this section, we present the RQs for our case study. Our aim is to evaluate the use of a digital twin in configuration testing of an APS control algorithm and its ability to successfully simulate the environment in which it is used.

RQ1. To what extent can our digital twin provide a simulated environment for the configuration testing of an APS?

To ensure correct evaluation of APS control algorithm configurations, the environment in which it is being simulated must also be accurate. Here, we investigate whether the derived digital twin model is able to accurately represent the blood glucose dynamics of a person with T1DM. This will allow us to understand the effectiveness of adapting the model described by Contreras et al. [49] for representing T1DM and any difficulties encountered with regards to configuration testing. We can also evaluate the applicability of adapting the model set out by Contreras et al. [49] as opposed to other techniques. We use this RQ to evaluate Steps 3 and 4 of the approach outlined by Corral-Acero et al. [12], model calibration and model validation.

RQ2. To what extent can our digital twin make predictions to enable configuration testing for an APS?

To ensure our observations of the predicted configurations of an APS control algorithm, the predictions about its execution environment must also be accurate. From this RQ, we investigate to what extent the adapted model's fixed parameters can accurately extrapolate unseen blood glucose behaviour of a person with T1DM and identify any challenges that occur when extrapolating blood glucose–insulin dynamics. We use this RQ to further evaluate Step 4 of the approach outlined by Corral-Acero et al. [12], model validation.

RQ3. Can digital twins be used to test the blood glucose target configuration of an APS system, without relying on humans-in-the-loop?

After developing the simulated environment, we evaluate the behaviours and reliability of an APS control algorithm. We aim to observe the effectiveness of the APS control algorithm across multiple scenarios and configurations. We evaluate values of oref0's blood glucose target as, in Section 2.1.1, we present it as a setting which could produce potentially dangerous behaviour if misconfigured. We aim to investigate how the use of an explainable model as part of a digital twin allows for the tracing of medical interventions from different system configurations. This RQ evaluates Step 5 of the approach outlined by Corral-Acero et al. [12], testing system configurations against predictions from the calibrated model.

4.2 | Data Cleaning

In this section, we describe the data cleaning process used to find the candidate traces for this study. We first outline the data source for our case study, the OpenAPS Data Commons. We then present the steps required to process this data for use in our case study. Our data cleaning process equates to Step 1 of the approach outlined by Corral-Acero et al. [12], presented in Section 2.3.2.

4.2.1 | OpenAPS Data

To enable the digital twin to make personalized predictions to the user during configuration testing, we use a historical clinical dataset. The OpenAPS Data Commons[3] is the largest openly available APS dataset. It contains approximately 10 million data points across 156 volunteers [24]. This dataset contains volunteered data from people using implementations of oref0 in the form of CGM, insulin pump and control algorithm outputs. Section 4.2 outlines how we use these data to provide blood glucose, insulin and carbohydrate histories in our case study.

However, we must first ensure the data retrieved from the OpenAPS Data Commons is compatible with our digital twin. Due to the intrinsic nature of medical data, the dataset may be missing values, have inconsistent formatting or suffer from sensor error [57, 58]. This is especially important for data that is volunteered and not obtained through clinical trials as there may be less control in the data collection. The OpenAPS Data Commons, as with other medical datasets, suffer from these issues, requiring us to first preprocess the dataset.

4.2.2 | Data Inclusion and Exclusion

To make the OpenAPS Data Commons dataset compatible with the model fitting procedure, outlined in Section 3.2, the data must first be processed. The data must be transformed into blood glucose, insulin and carbohydrate time series with any traces with nonphysiological behaviours or measurement errors excluded. We define our process for data cleaning as follows:

1. Exclude all traces that do not include blood glucose, insulin and carbohydrate data (which are required by the model). *This allowed us to exclude datasets which did not include all required historical data for training the digital twin.*
2. Exclude all traces that are not in a consistent format or contain null values. *This ensured our digital twin could correctly learn the blood glucose–insulin dynamics of a person with T1DM as a complete trace was required.*
3. Split up each trace into time periods of 2 h with no carbohydrates on board before a single carbohydrates consumption. *The original data were in long continuous time series. The training procedure outlined in the original model by Contreras et al.* [49] *required 2 h of training data after a single carbohydrate consumption. Two hours of no carbohydrate consumption was required before each trace as this could cause blood glucose levels to increase at a delayed time.*
4. Exclude all traces that contain any large nonphysiological increases or decreases in blood glucose. *This was used to remove traces that contain behaviours that should not be physiologically possible. This includes blood glucose changes over 50 mg/dL (2.8 mmol/L) in a 5-min timestep or no absorption in carbohydrates over the trace. Such traces exhibit nonphysiological behaviour and are therefore invalid. Such traces may exist in the data due to sensor error, human error in data collection or just invalid formatting as traces in the OpenAPS Data Commons were volunteered by users.*

Using these criteria, we processed appropriate traces from the OpenAPS Data Commons. We started with 156 original traces and ended with 930 time period traces after applying our procedure. These made up the candidate traces used throughout the case study. Our final traces were stored as 4-h trace files with data points every 5 min.

The original dataset contained 9,423,805 min (18 years) of data points. After applying our data cleaning process, we found that 223,200 min (2.37%) of the data was suitable. This was due to model training requiring carbohydrate consumption, nonphysiological behaviour in the data or simply human error in capturing the data. The incredibly low acceptance of data highlights the difficulty of obtaining high-fidelity data, especially in a real-time environment. In Section 6, we discuss how this is a potential threat to configuration testing of medical devices and suggest how we could potentially improve the data inclusion rates for future evaluations.

4.3 | Measures

The effectiveness of an APS is typically evaluated by its user's time in range (TIR). Battelino et al. [20] define TIR as the 'percentage of readings and time per day within target glucose range'. This metric shows how effective an APS is at reducing hyperglycaemia (ensuring blood glucose stays below 180 mg/dL or 10.0 mmol/L) and not inducing hypoglycaemia (ensuring blood glucose stays above 70 mg/dL or 3.9 mmol/L).

Blood glucose monitoring devices also use Clarke error grids [59] to evaluate their accuracy and to determine the clinical significance of device outputs [60]. The results of the error grid are split into five zones: clinically accurate (A), clinically acceptable (B), overcorrection (C), failure to detect (D) and erroneous (E) [61]. From this, a trained professional can assess the reliability of the blood glucose monitoring device.

4.4 | Reproducibility

To ensure the reproducibility of this case study, we have created a reproducibility package[4] including the digital twin model of a person with T1DM, a script for applying our data cleaning procedure to the OpenAPS Data Commons and scripts for each RQ. We cannot redistribute OpenAPS data used within this study due to our data management agreement with OpenAPS Data Commons. Our work was completed using version '$n = 183$' of the OpenAPS Data Commons.

5 | Case Study Evaluation

In this section, we present the evaluation of our case study. We present the methodology used for each RQ and then present their findings. We further discuss these findings and their relation to the case study in Section 6.

5.1 | RQ1—Model Validity

This RQ evaluates the ability for a mechanistic model as part of a digital twin to represent mechanics of body dynamics. We equate this to an application and evaluation of Steps 3 and 4 of the testing procedure outlined in Section 2.3.2.

5.1.1 | Methodology

For this RQ, we took the 930 clinical traces, generated from Section 4.2, to personalize the adapted model. These traces represent real blood glucose–insulin dynamics across 156 different people. Because the fitting process outlined in Section 3.2 is nondeterministic, we fitted each trace 10 times. We used the RMSE metric (Equation 5) to quantify the difference between the observed data for each of the 930 traces and their corresponding fitted model outputs. We use this metric to quantify the accuracy of a digital twin and, therefore, its ability to represent blood glucose–insulin behaviours.

We also defined model accuracy based on Clarke error grids [59], mentioned in Section 4.3. We use the 930 candidate traces as reference values to identify the zones generated by the adapted model. From this, we were able to quantify the number of traces in the OpenAPS Data Commons which a digital twin's simulation would lead to clinically accurate treatment.

We then identified traces with high and low RMSE values. Incorrect behaviours in the model were identified and the transparency of the internal variables used to find the causes of such behaviour.

To further evaluate the applicability of our adapted model, we also perform this methodology using other modelling techniques. More specifically, we compare our approach to a data-driven explainable approach (a symbolic regressor) and a purely data-driven nonexplainable approach (a neural network). We replicated the methodology using the symbolic regressor and neural network to produce a distribution of RMSEs for each approach. The default settings were used for each modelling technique, except increasing the neural network's convergence iterations, which is explained in the findings.

A symbolic regressor is a model that represents the training data as a set of equations [62], similar to our model. However, unlike our approach, these equations are derived purely from the data, removing the need for an understanding of the underlying physiology. Symbolic regressors are also explainable because their resulting equations can be traced back, allowing a user to understand the causes of information flow within the model.

A neural network is another data-driven approach that mimics the flow of information through neurons and synapses [63]. Similarly to the symbolic regressor, this approach learns its structure from the data so minimal knowledge of the underlying physiology is required. However, neural networks are not explainable. It is very difficult for a domain expert to understand the information flow through a neural network, making finding the causes of blood glucose behaviours difficult.

5.1.2 | Findings

Figure 5a presents the distribution of RMSEs across each of the 930 traces; 82.5% of the traces produce a median RMSE less than 20. We define an RMSE less than 20 as accurate in Section 4.3 as predicted values inside the safe blood glucose range would not result in severe hypoglycaemia or severe hyperglycaemia. From these results, we show that the adapted model is able to accurately represent the blood glucose dynamics of a person with T1DM.

We observe that the standard deviation of RMSE values for a single trace generally increases as the average RMSE increases. Six hundred forty (68.8%) of the traces had a low standard deviation of less than 10 (half of our accuracy value), which largely applied to those values with lower median RMSE.

Figure 5b presents a Clarke error grid for all 930 traces; 92.02% of the model outputs are within Zone A of the error grid and are therefore clinically accurate. From this, we observed how the adapted model was able to clinically accurately represent 92.02% of the traces in the OpenAPS Data Commons. These points would lead to clinically correct treatment decisions [60].

Although these results present the accuracy of our approach, we also assessed the applicability of our adapted model compared to other modelling approaches. Figure 6 presents the different RMSE distributions across the 930 traces for each different blood glucose–insulin modelling method. Our adapted model has the lowest RMSE, with the symbolic regressor being less accurate and the neural network being the most inaccurate.

The data-driven techniques appear to struggle due to the nature of clinical data. These approaches lack the domain knowledge that makes up the 'template' for how blood glucose–insulin dynamics should behave. The symbolic regressor learned a simplification of the dynamics from the dataset, not taking into account factors such as the time delay of absorption across the ilium and jejunum, as described in Section 2.4. The neural network struggled to untangle the complex dataset, failing to converge for several traces even with five times the default number of convergence iterations.

Similar to our mechanistic approach, a symbolic regressor could be used to trace back through model traces. This explainability would allow a user to understand how different insulin interventions may have affected blood glucose levels. However, because there is no structure to the derived equations, a user may struggle to work out why a symbolic regressor has learnt specific behaviour. For example, in Section 5.1, we identified that the insulin sensitivity of a model was incorrect from the k_{xgi} constant. Because the symbolic regressor does not have such a structure with specific constants for specific behaviours, understanding why specific behaviours are exhibited becomes more difficult.

The neural network is not an explainable model. Unlike the other two models, there is no practical way to examine why a neural network's inputs result in its outputs. As a result, a user would not be able to reason about the learned behaviour or trace back insulin interventions through the model to understand their cause.

The lack of explainability also made it difficult to determine the exact reason for the neural network's bad performance. Typically, factors such as overfitting can be accounted for through splitting the data into test and training datasets [64]. However, our clinical dataset presented a couple of challenges regarding this. Each trace represented blood glucose–insulin dynamics independent from the other traces. As a result, we could not use some traces for training and others for testing. Also, due to blood glucose–insulin dynamics changing over time, our traces were made up of 120 data points (every 5 min across 2 h). The short traces made each data point necessary for training, meaning removing some for testing would have resulted in less accurate training.

To better understand the causes of accurate and inaccurate model dynamics, we can use the explainability of the adapted model. For this, we examine the RMSE values of model outputs (grey dotted lines) with respect to their observational data trace (red solid lines), shown in Figure 7.

Figure 7 presents three examples of how the adapted model can accurately represent blood glucose dynamics and how that accuracy can vary. (a) presents a trace for which the model was able to represent the blood glucose dynamics across all model attempts. This represents a trace which resulted in a low RMSE. For (b), the majority of the 10 attempts to fit the model accurately

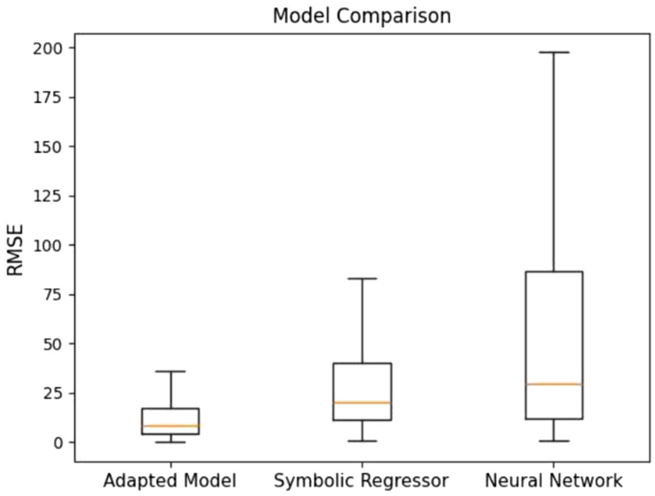
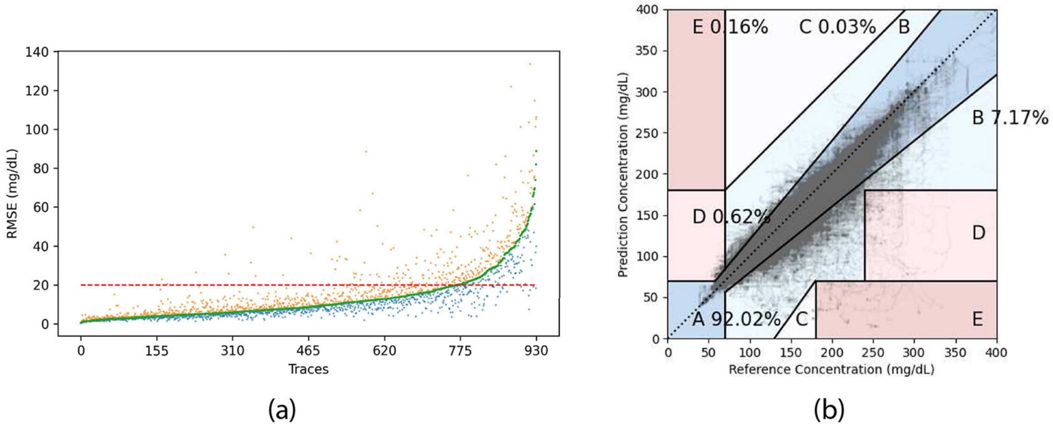


FIGURE 6 | Box plot presenting the distribution of RMSEs across all 930 traces for our adapted model, a symbolic regressor and a neutral network. Outliers have been removed.



(a)

(b)

FIGURE 5 | (a) RMSE values across the 10 runs of each trace. The median (green), first quartile (blue) and third quartile (orange) values are plotted. The red dashed line represents the maximum RMSE value for an accurate model. (b) A Clarke error grid showing model outputs across all 930 candidate traces. Trace blood glucose values are used as the reference, and model outputs are used as the prediction.

captured the blood glucose dynamics of the person with T1DM. One attempt, however, resulted in a large RMSE. We were able to trace this back through the model to find that the value for k_{xgi} was significantly larger than the other attempts. This caused the blood glucose level to decrease at a much greater rate than the observational data. (c), however, resulted in a large RMSE for all model fitting attempts. The model appears to over generalize the blood glucose dynamics, resulting in model outputs which do not follow the trace.

Summary: These results show how the stochastic elements of the fitting algorithm can allow for successful navigation of the solution space. However, nondeterminism combined with the complexity of the solution space can sometimes produce suboptimal solutions. Fitting the model a single time may not be optimal so the model should instead be fitted multiple times and the most accurate values of $K_x(T)$ used to simulate a user's blood glucose dynamics. In our case, we fitted the model 10 times as it presented accurate solutions for traces where not all model outputs had a low RMSE while not being too computationally expensive. This ensured that an accurate environment was simulated by the digital twin when performing configuration testing. Further work could be performed to understand the optimal fitting strategies and iterations required for this kind of problem.

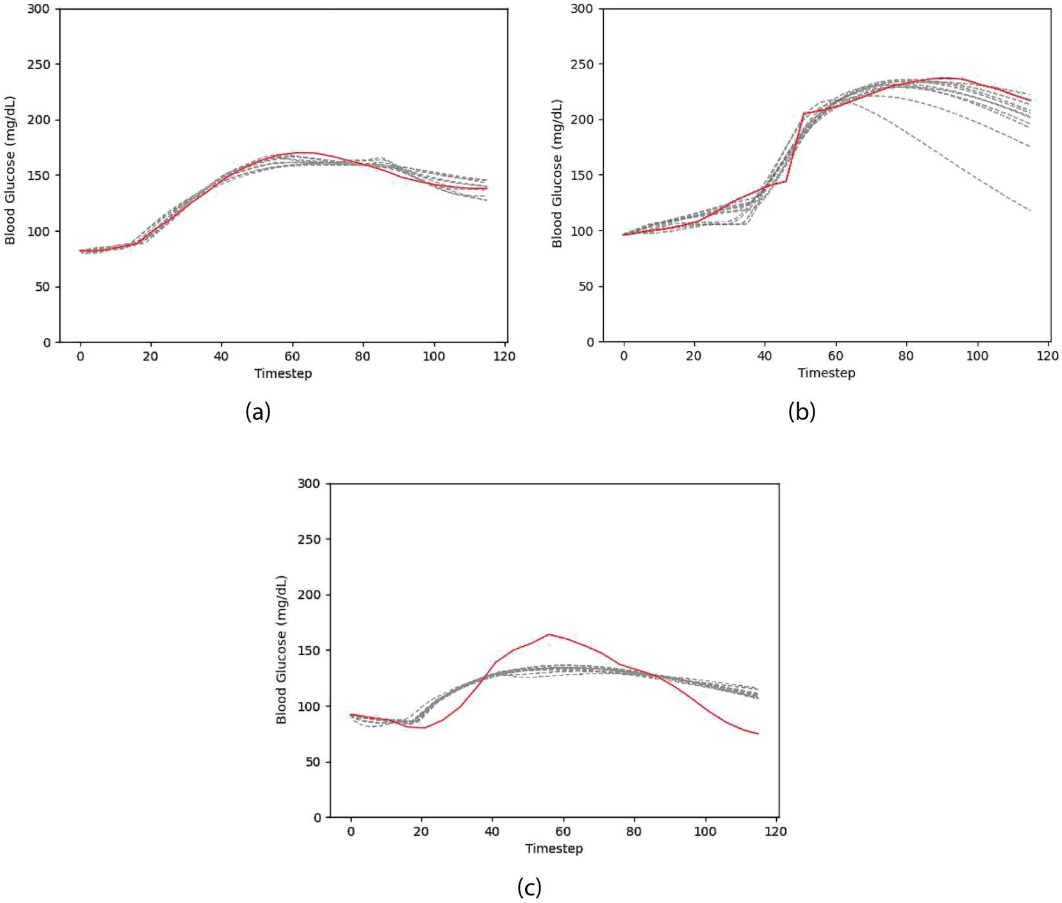
Our results also found both of the data-driven approaches less suitable for modelling blood glucose–insulin dynamics than our adapted, domain knowledge-driven model. The underlying domain knowledge of blood glucose–insulin physiology of our approach simplified the task of learning personalized dynamics. The purely data-driven approaches did not have this underlying knowledge to provide a structure, resulting in less accurate models.

5.2 | RQ2—Prediction Accuracy

This RQ presents the ability for a digital twin to predict unknown body dynamics in order to aid in configuration testing. We use this RQ to further evaluate the application of Step 4 in the testing procedure outlined in Section 2.3.2.

5.2.1 | Methodology

For this RQ, we only used traces in the lowest 50% of RMSE values from RQ1, providing they had an RMSE≤20 and are therefore shown to be accurate. Using this model that was only trained on the first 2 h of each candidate trace, we executed the model with additional timesteps in order to use the



(a)

(b)

(c)

FIGURE 7 | Data traces showing fitted blood glucose dynamics. Observational data are shown in red, each of the 10 model attempts at learning are shown in grey. (a) presents a trace that all model attempts were able to represent, (b) presents a trace that some model attempts were able to represent and (c) presents a trace that the model struggled to represent.

remaining 2 h of data points of each trace to evaluate model prediction.

For each of the traces, we increased the additional timesteps from 0 to 120 min in steps of 20 min, measuring the RMSE at each. These additional timesteps are a continuation of the trained observational data. From this, we observed the accuracy of the model during extrapolation, providing insight into the reliability of the static set $K_x(T)$.

We used the RMSE values to find traces in which error did and did not increase with extrapolation. Using the transparency of the adapted model, we traced through these interesting extrapolation behaviours in order to find the causes of accurate and inaccurate extrapolation.

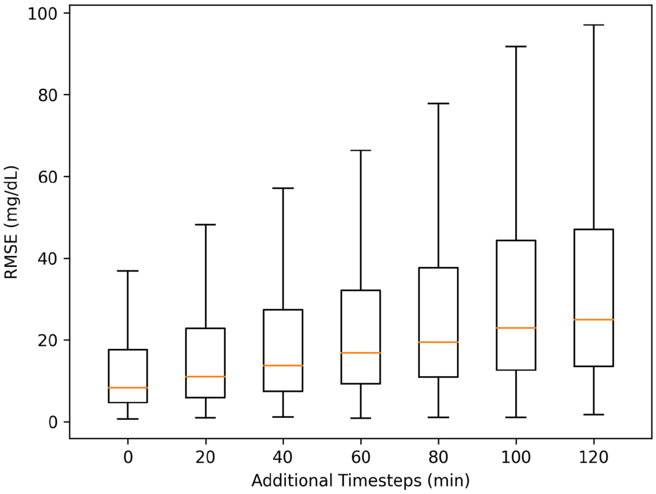


FIGURE 8 | Box plot presenting the distribution of errors as the level of extrapolation is increased across the 465 most accurate traces.

5.2.2 | Findings

For this RQ, we explored how blood glucose dynamics for a person with T1DM change over time. We used this question to understand to what extent our digital twin can predict blood glucose behaviour as the internal dynamics of the person change over time.

To answer this question, we used the RMSE metric to measure model accuracy when adding additional observed timesteps outside of the time bound captured by the training data. Figure 8 presents the distribution of RMSE across model outputs when extrapolating. From no extrapolation to 120 min of extrapolation, the average RMSE increased from a value of 8.3–24.9. From this, we observed an increase in RMSE as the number of additional timesteps is increased. Section 4.3 presents how a prediction with an RMSE of 20 is considered accurate as it would ensure any incorrect predictions outside the safe blood glucose range are not severe. From this, Figure 8 presents how the adapted model can accurately predict 80 min of blood glucose dynamics outside the training data.

To further explore blood glucose prediction, we identified models which originally had low RMSE values but increased in RMSE after extrapolation. Figure 9a,b shows a single trace with 120 timesteps in Figure 9a and 240 timesteps in Figure 9b. We use this example to observe the difference between no extrapolation and 120 min of extrapolation. Figure 9a presents a set of accurate model outputs where the blood glucose dynamics were

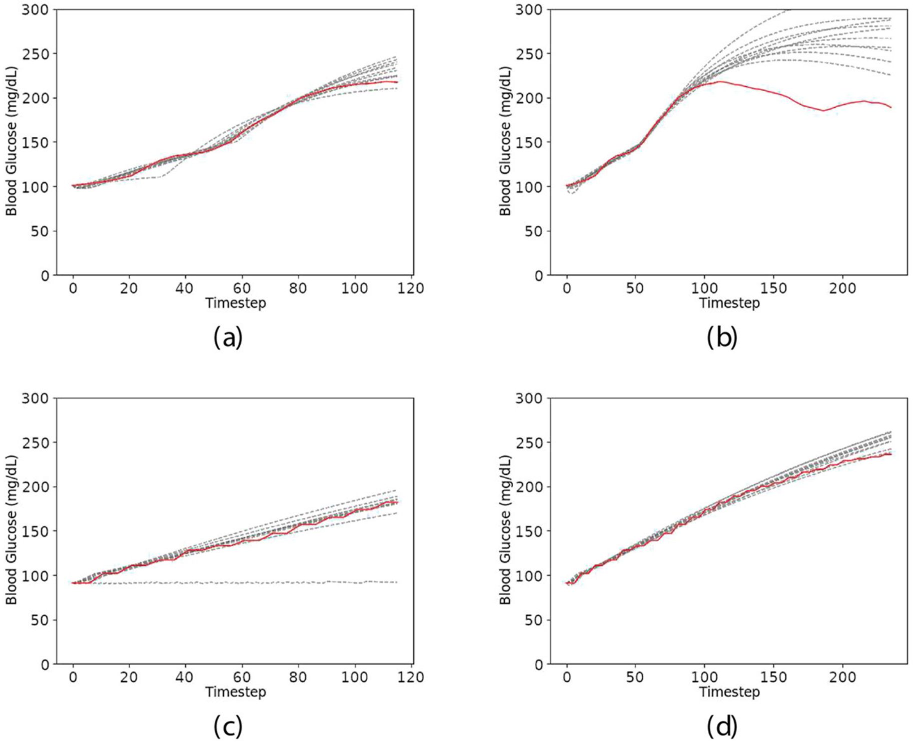


FIGURE 9 | Traces showing reliability of extrapolation. (a) and (b) present a trace showing an increase in RMSE due to extrapolation. (c) and (d) present a trace showing no increase in RMSE due to extrapolation.

correctly captured by the model. This led to a low RMSE in RQ1. Figure 9b shows behaviour being extrapolated which no longer followed the observed data. The trend of the behaviour was captured, but the nuances were not.

By splitting up this trace and only training on the second half, we were able to observe a completely different set of $K_x(T)$ to the original trace. This represented a change in blood glucose behaviour. The transparency of the adapted model allows for these sets of constants to be compared.

Some traces, however, did not increase their RMSE through extrapolation. Figure 9c,d presents a set of traces in which the extrapolated behaviour matched that of the trace data. Similar to Figure 9a,b, the original trace for RQ1 had a low RMSE. Because the blood glucose dynamics did not fluctuate during the extrapolation, this resulted in a low RMSE for Figure 9d.

Training the model on both the first and second halves of this trace produced very similar $K_x(T)$ values. Because this set did not change much, the adapted model was able to accurately predict the untrained blood glucose behaviour. From this, we observed that the adapted model's prediction is more accurate given stable blood glucose dynamics.

Summary: From these results, we suggest that the adapted model can be used to predict blood glucose dynamics up to 80 min outside of the time bounded by the training data. For continuously accurate representation, the values of $K_x(T)$ should be relearnt outside this boundary to ensure accurate predictions. This would ensure predictions made by the digital twin would produce accurate and reliable behaviour for configuration testing.

5.3 | RQ3—Configuration Testing

This RQ aims to evaluate whether our digital twin can facilitate configuration testing of the blood glucose target setting of oref0 without clinical trials. We use this to evaluate the application of Step 5 of the testing procedure outlined in Section 2.3.2.

5.3.1 | Methodology

We interfaced the model of a person with T1DM with oref0 to observe APS behaviour. To apply configuration testing, we need to select a small number of expected configurations, as described in Section 2.2. With this in mind, we identified three different blood glucose targets for oref0 which were widely used within the OpenAPS Data Commons: 100 (5.6 mmol/L), 120 (6.7 mmol/L) and 140 mg/dL (7.8 mmol/L). Section 2.1.1 identifies how the misconfiguration of blood glucose targets can result in dangerous scenarios. For this RQ, we test these expected configurations with all other oref0 configurations unchanged.

For each of the 930 candidate traces, we fitted the model to represent the blood glucose dynamics for that given scenario, as seen in RQ1. Given these dynamics and the initial carbohydrate, blood glucose and insulin values found in the trace, we simulated the effects of using the different oref0 configurations for that scenario. Each configuration of oref0 was executed 10 times, allowing for any nondeterministic elements. Because the body does not react immediately to the interventions of oref0, the full 4-h traces, as used in RQ2, were required to capture the resulting behaviour.

For this analysis, we used the TIR metric, outlined in Section 4.3, to present the efficacy of each oref0 configurations for a given scenario. Battelino et al. [20] defined TIR as a percentage of time spent in a blood glucose range of 70–180 mg/dL (3.9–10.0 mmol/L). From this, we performed configuration testing on oref0 and evaluated each configuration based on its resulting TIR.

For this RQ, we used oref0 0.7.1 which is not the same as that found in the OpenAPS Data Commons. To compare the changes to the control algorithm over time, we calculate the TIR of the original OpenAPS Data Commons traces to observe improvements in the control algorithm.

Now that the control algorithm is interfaced with the adapted model, we found it important to evaluate the temporal expense of this process. The motivation was that this analysis could provide insights regarding the applicability of our approach for configuration testing when applied to real-world applications. We performed a temporal analysis to enable this evaluation.

We measured the time expense required for the different stages of the configuration testing process. This was split up into fitting the model parameters, executing the model without oref0 and executing the model with oref0. For each of the 930 traces, the time required for each of these stages was calculated, resulting in a distribution of temporal expenses.

5.3.2 | Findings

Table 2 presents the TIR across different oref0 configurations averaged across each candidate trace. No intervention presents the TIR when no insulin is provided to the user, OpenAPS (2014–2021) provides the TIR for each trace using data from the original dataset and oref0 (v0.7.1) provides the TIR observed at different target blood glucose configurations.

TABLE 2 | Mean time in range (TIR) for different interventions across each 930 traces to observe oref0 effectiveness.

Intervention	BG target (mg/dL)	Mean TIR (%)
No intervention	N/A	63.08
OpenAPS (2014–2021)	Various	73.57
oref0 (v0.7.1)	100	87.69
oref0 (v0.7.1)	120	89.06
oref0 (v0.7.1)	140	88.10

‘No intervention’ provided the lowest TIR across each implementation. This was not surprising as people with T1DM struggle to naturally regulate their blood glucose levels [52]. The high mean TIR was observed to be due to most of the traces starting within range, increasing the TIR for uncontrolled blood glucose.

‘OpenAPS (2014–2021)’ presented how controlling blood glucose levels increases TIR. This represents the increase in quality of life found by Litchman et al. [23] when using OpenAPS. Comparing this to no interventions is an unfair comparison, so we use this metric to compare older versions of oref0, present in the OpenAPS Data Commons, with the latest version at time of writing.

‘oref0 (v0.7.1)’ provided a significant increase in mean TIR compared to older versions of oref0. From this, we were able to show an increase in user safety when using more modern APS control algorithms. The difference in average TIR between the different oref0 blood glucose configurations, however, was not noticeable. To explore this further, Figure 10 presents two scenarios that show the behaviour of oref0 interventions across configurations. For both of these scenarios, a large amount of carbohydrate (≥60 g) was consumed causing uncontrolled blood glucose levels to rise out of the safe blood glucose range, defined in Section 4.3. We used the explainability of the adapted model to explore how oref0 adapts to these scenarios.

Figure 10a presents a scenario in which oref0 successfully controlled the user's blood glucose within a safe range. From this, we observed how the different configurations resulted in different resulting blood glucose behaviours. Depending on the scenario, such as exercise, different targets and glucose behaviours are required. In this scenario, we see how the target of 140 mg/dL sometimes produced undesirable behaviour, allowing the blood glucose to rise above a safe level. Our digital twin allowed for these configurations and their variability to be trialled in a safe environment.

Figure 10b presents a scenario in which the TIR was low for all configurations. We can see from the graph that there is minimal difference in system behaviour between the configurations. By investigating the model execution at each timestep, we found that the APS algorithm was acting correctly but the model's fitted blood glucose dynamics were inaccurate. A lack of useful insulin data in the observational trace led to the model consistently learning a very low insulin sensitivity value (k_{xgi}). This resulted in the oref0 interventions having very little effect on blood glucose levels and the insights gained from configuration testing being misleading.

To further evaluate the applicability of configuration testing, we performed a temporal analysis of each stage of the testing procedure. Figure 11 presents the different time expense distributions for each stage of the configuration testing process across the 930 traces. We do note that the times found in this evaluation are those required for 4 h of oref0 execution.

These results illustrate how the training of the model is the most computationally expensive task of the procedure. Due to the large number of constants required when fitting the model, as described in Section 3.2, this is not surprising. However, the results of RQ1 have shown the efficacy of this procedure in ensuring accurate models. Also, compared to the 4 h of oref0 execution, the time required is still minimal. However, because this is the most computationally expensive part of the configuration testing process, we discuss potential solutions to this in Section 8.1.

Figure 11 shows a large disparity between executing the model with and without oref0. The model execution alone took on average 0.002 s, whereas also executing oref0 took on average 38.7 s.

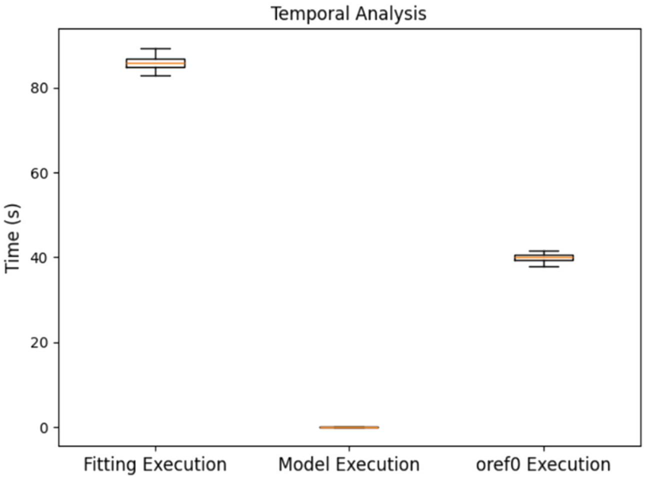


FIGURE 11 | Distribution of times taken for fitting the model, executing the model and executing oref0.

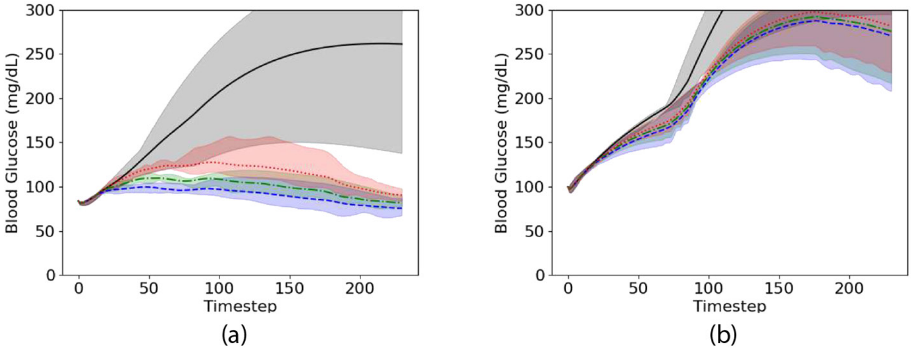


FIGURE 10 | Trace showing the effectiveness of oref0 against uncontrolled blood glucose. No intervention (black solid), oref0 targeting 100 (blue dashed), 120 (green dash dotted) and 140 (red dotted). The mean lines are plotted for 10 runs with the ranges shown.

This difference demonstrates the computational efficiency of using a differential equations model compared to the rest of the testing procedure. The difference also demonstrated the computational expense of running an APS. We observed how the expense of running the APS is comparable to fitting the model itself.

Summary: From this RQ, we were able to evaluate the TIR of different settings of an APS control algorithm through configuration testing. Configurations could be trialled for different people to ensure an APS control algorithm is set up correctly, without the need for clinical trials. We also observed the limitations of using real-world data, which will be explored further in Section 6.

We observed that fitting the model was the most computationally expensive part of the testing procedure. We also demonstrated the efficiency of our model and the extent to which interacting with oref0 increased the temporal expense required for configuration testing.

6 | Discussion

In this section, we discuss the finding of our case study to evaluate the applicability of using a digital twin to perform configuration testing an APS. We approach this discussion from the perspective of strengths, weaknesses, opportunities and threats.

6.1 | Strengths

Medical devices present a challenge for configuration as humans-in-the-loop can result in dangerous scenarios when the system is misconfigured. During our case study, we were able to develop a digital twin to alleviate this difficulty and perform configuration testing of an APS control algorithm without putting users at risk.

The process outlined by Corral-Acero et al. [12], presented in Section 2.3.2, provided a framework for enabling the configuration testing of medical devices without requiring clinical trials. Our case study has shown that by following this process, we were able to develop a digital twin to represent individuals and interface it with an APS control algorithm to observe different APS behaviours for different system configurations across a multitude of people. From this, we were able to perform configuration testing for the oref0 control algorithm for scenarios which were potentially dangerous, without putting any users at risk. We propose that the process outlined by Corral-Acero et al. [12] presented a compelling framework for other medical device configuration testing challenges.

6.2 | Weaknesses

We did, however, encounter some difficulties when applying configuration testing to an APS through the use of a digital twin. Such a complex physiological phenomena as blood glucose–insulin dynamics required an equally intricate model. RQ1 found that the complex solution space of the adapted model sometimes presented a challenge when fitting it to data.

We further encountered this in RQ2 where the digital twin sometimes struggled to predict more dynamic blood glucose–insulin behaviours. We suggest that this may be a sign of model overfitting as our explainable model could not adapt its dynamics. More adaptive models, further explored in Section 8.1, present a potential solution to this challenge.

Unreliable digital twin predictions could lead to the observation of incorrect software behaviour during configuration testing. These weaknesses come from using a static model as opposed to a less explainable but more dynamics modelling technique.

6.3 | Opportunities

The above weakness provides an opportunity for evaluating different modelling techniques for body dynamics. Future work could investigate the use of models that can better represent dynamics behaviours, such as machine learning–based modelling [65], and how this could potentially improve the reliability of digital twin–based configuration testing in this context. Such an approach, however, would have reduced the explainability of the model and introduced the social challenge, presented by Corral-Acero et al., of users finding it difficult to trust a 'black-box' for clinical decisions [12]. We expand on this in Section 8.1.

Curated datasets from future work would allow for more usable data. The OpenAPS Data Commons, however, demonstrates the unique challenge of the APS's clinical setting as such curated data does not typically exist. As a result, we found that only 2.37% of the data being suitable as a significant challenge concerning future evaluations. In which case, we suggest that more open-source datasets are made available from future clinical trials designed specifically for learning blood glucose–insulin dynamics. This would allow for potentially more coverage of T1DM behaviours. However curated datasets may reduce the authenticity of applying such techniques to a real user's data due to the intrinsic nature of clinical datasets, as described in Section 4.2.1.

6.4 | Threats

We found that the greatest threat posed to configuration testing using a digital twin was the use of clinical data. In Section 4.2.1, we present the OpenAPS Data Commons as our data source. At the time of writing, this was the largest and most well-documented dataset for people with T1DM [24]. Regardless of this, a large quantity of rejected traces led to only a 2.37% acceptance rate, as presented in Section 4.2. This was due to the data containing inconsistent or nonphysiological behaviour, human error in data collection or not containing a recorded carbohydrate consumption that was required for training the model. These are real characteristics of clinical data and reduce the reliability of applying configuration testing using past observational data.

A potential solution for increasing the number of accepted traces would be to interpolate missing values in the data. As a result, more of the data would be compatible with training the digital twin. Therefore, more behaviours could be trained, allowing for configuration testing of more contexts. However, this could introduce a potential threat. Interpolated values could result in behaviour which is not representative of the person's real glucose–insulin dynamics. Subsequent configuration testing may be impacted by using training data and, as a result, may be less reliable.

Even after rejecting 97.63% of the individual data points, some accepted traces did not contain enough granularity in the data, leading to misinterpretation of blood glucose–insulin dynamics. This meant other human factors in the data may have still introduced erroneous data into the model. This would produce potentially incorrect predictions from the model, which could lead to inaccurate assumptions about APS behaviours during configuration testing.

For digital twins to accurately predict medical interventions when performing configuration testing, more accurate and consistent data sources or techniques to account for uncontrolled data sources are required. Being able to account for the clinical data would allow for more models to be generated and an increase in their accuracy. By alleviating the challenges associated with clinical data, the applicability for the configuration testing of medical devices would greatly improve.

7 | Related Work

In this section, we explore works which are related to our paper. We describe how we build on existing work that provides the context for our work. We describe how digital twins have been used in healthcare, how digital twins have been used in testing and applications of configuration testing.

7.1 | Digital Twins in Healthcare

Digital twins have been proposed to be implemented across the medical domain to better inform life-saving decisions without costly clinical trials. Kiagias et al. [66] present a simulation-based approach that predicts treatment responses for people with tuberculosis. They show how in silico experiments speed up the ability for medical interventions to be delivered, without compromising safety and effectiveness. Our work capitalizes on such notions by simulating the medical interventions of system in a safe environment to ensure the system is behaving correctly for a given configuration.

The Cardiac Physiome Project [67, 68] argues the importance of simulation-based medicine for a better understanding of cardiac tissues, while outlining challenges surrounding the representation of tissue behaviour due to model resolution and approximation. This presents an ongoing challenge for digital twins in the medical domain. Corral-Acero et al. [12] propose a solution to this through a digital twin for precision cardiology. This would allow simulations to be patient specific, influence clinical decisions and be updated with both population and individual data over time.

Corral-Acero et al. do, however, highlight an important social challenge with the adoption of such technology, as clinicians may find it difficult to trust a 'black-box' for clinical decisions [12]. We use this notion to further motivate the use of an explainable model when developing our digital twin in Section 3.

7.2 | Digital Twins in Testing

Digital twins have presented a way of indirectly testing physically interacting software [15]. As a result, domain specific models have been used to enhance software testing. Peng et al. [69] use a mathematical model to represent the intended behaviour of a lithium-ion battery. As a result, the real system can be compared to the digital twin to identify any divergences resulting from battery degradation. The use of a mathematical model allows for the encapsulation of the physical properties of the battery. The digital twin can, therefore, represent unseen scenarios without relying on training data.

This notion motivates Corral-Acero et al.'s [12] use of a mechanistic model to represent the body's physiology [51]. In this paper, we use a mechanistic model consisting of differential equations derived from the knowledge of biological processes [49]. As a result, our digital twin is underpinned by the knowledge of blood glucose–insulin dynamics, allowing for the inference of behaviour outside of trained data.

Our case study used digital twin predictions to test software configurations. Prior works have performed prediction-based software testing using a digital twin [70, 71]. However, the predictive capabilities of digital twins are underutilized in software testing [15]. In this study, we found the strength of using a digital twin's predictions in a medical setting to simulate and evaluate potentially dangerous scenarios and configurations. An expansion of this to other safety-critical domains may present opportunities for more proactive software testing without the associated risks.

7.3 | Configuration Testing Applications

Section 2.2 described the process and importance of configuration testing. Prior work has focused on the impact of configuration testing on large industrial systems [10, 35]. Our work instead applies configuration testing to medical devices. From our evaluative case study, we were able to highlight the importance of configuration testing in their domain and uncover the limitations associated with using clinical data.

As well as testing for individual configurations, prior work has aimed to present techniques that cover the configuration space [72]. This is particularly important for highly configurable systems such as oref0 [28]. Because this is an evaluative case study on applying configuration testing, we find this out of scope of this paper. However, future work into covering the configuration space of systems like oref0 should be performed.

8 | Conclusion

With the recent spread of medical devices, is it important to ensure they are correctly configured so that users can correctly manage health conditions without risk. Misconfiguration poses a particular challenge in this domain due to the human-in-the-loop. This creates the potential for dangerous behaviour being enacted on the user if the system behaves incorrectly.

We investigated the impact of digital twins on configuration testing of medical devices through an evaluative case study on an open-source APS. We evaluated a proposed technique [12] for configuration testing using a digital twin. We identified usable data from the largest open-source dataset of T1DM blood glucose–insulin dynamics to fit our model. We adapted an existing model into a digital twin to represent blood glucose–insulin dynamics of a person with T1DM and validated the personalization and predictive capabilities by fitting it to our dataset. From this, we performed configuration testing on the APS control algorithm oref0. We tested system behaviours under different configurations resulting in the successful identification of a blood glucose target that was not suitable for a user. This configuration testing was performed without requiring clinical trials or putting any users at risk.

From this case study, we found the proposed framework to be applicable in the domain of an APS, allowing for accurate digital twin–based blood glucose–insulin predictions to enable configuration testing of an open-source APS. We were able to evaluate simulations and predictions of the digital twin to validate the environment they provided for configuration testing. We also observed different APS configurations in this simulated environment, resulting in the identification of behaviours that could be unsafe to a user. This was performed without requiring physical trials.

We also identified difficulties associated with configuration testing using a digital twin. Model misrepresentation of a user's blood glucose–insulin dynamics led to distorted system behaviour being presented. We alleviated this challenge by fitting the digital twin multiple times and confining its predictions to a 2-h time window due to changing glucose–insulin dynamics over time. Using real clinical data, also, presented a substantial threat to the approach. Human factors and data inconsistency led to a large proportion of the data being impractical, resulting in only a 2.37% of the initial data being usable.

8.1 | Future Work

Digital twins present a promising approach to the configuration testing of medical devices but further work is needed before their implementation. We investigated the configuration testing of an APS using a digital twin to provide an evaluation of this technique's applicability in this domain. In future studies, we propose applying digital twin–based configuration testing to more case studies of medical devices. Such studies have been proposed [17, 73] but not in a practical setting.

Furthermore, work should be conducted to help alleviate the challenges identified regarding the use of clinical data and the complexity of the human body's dynamics. Investigation into different modelling techniques which can account for uncontrolled data and uncertainty in biological behaviour could help alleviate these difficulties. For example, techniques, such as that presented by Edington et al. [74], present a potential solution by generating a digital twin comprised of multiple models. A combination of data-driven and physics-driven models allows for a better representation of systems whose internal dynamics change over time.

Further evaluation of our model would present more evidence in its real-world applicability. We performed a temporal analysis to identify which aspects of the testing procedure were the most expensive. A further sensitivity analysis may provide information required to streamline the most temporally expensive segment of testing, fitting the model constants. Future stability analysis and resource-based evaluation would also present evidence as to whether such a testing procedure could be applied in real time.

Acknowledgements

The authors would like to thank the OpenAPS Data Commons volunteers for allowing access to their data and for their communication during this study. Neil Walkinshaw and Robert Mark Hierons are funded by the UKRI, United Kingdom, CITCOM Grant EP/T030526/1. Hierons is funded by the UKRI, United Kingdom, Grants EP/R025134/2 and EP/V026801/2. For the purpose of open access, the author has applied a 'Creative Commons Attribution (CC BY)' licence to any Author Accepted Manuscript version arising.

Data Availability Statement

The data that support the findings of this study are available from OpenAPS Data Commons. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from https://openaps.org/outcomes/data-commons/ with the permission of OpenAPS Data Commons.

Endnotes

[1] https://github.com/openaps/oref0.

[2] We also present a model correction to G_{prod} in Equation (A1) in Appendix A.

[3] https://openaps.org/outcomes/data-commons/.

[4] https://github.com/CITCOM-project/APSDigitalTwin.

References

1. K. Guk, G. Han, J. Lim, et al., "Evolution of Wearable Devices With Real-Time Disease Monitoring for Personalized Healthcare," *Nanomaterials* 9, no. 6 (2019): 813, https://doi.org/10.3390/nano9060813.

2. L. Lu, J. Zhang, Y. Xie, et al., "Wearable Health Devices in Health Care: Narrative Systematic Review," *JMIR mHealth uHealth* 8, no. 11 (2020): e18907, https://doi.org/10.2196/18907.

3. B. Lin, "Wearable Smart Devices for P4 Medicine in Heart Disease: Ready for Medical Cyber-Physical Systems?,"*OMICS* 23, no. 5 (2019): 291–292 en.

4. A. Roque, J. Francisco-Pascual, J. F. Andrés-Cordón, et al., "A Protection Against Infection but a Risk of Misdiagnosis? False Positive Uptake in an Implanted Cardiac Device," *Journal of Nuclear Cardiology* 30, no. 6 (2023): 2846–2849, https://doi.org/10.1007/s12350-023-03331-2.

5. D. Mann, "Omnipod 5 Insulin System Reports Decimal Point Glitch," (2023), https://www.diabetes.co.uk/2023/news/dec/alert-omnipod-5-insulin-system-raises-dosage-concerns.html.

6. C. Tang, T. Kooburat, P. Venkatachalam, et al., “Holistic Configuration Management at Facebook,” in *Proceedings of the 25th Symposium on Operating Systems Principles*, SOSP '15, (New York, NY, USA: Association for Computing Machinery, 2015): 328–343, https://doi.org/10.1145/2815400.2815401.

7. D. Oppenheimer, A. Ganapathi, and D. Patterson, “Why Do Internet Services Fail, and What Can Be Done About It?” (2003).

8. S. Srinivasa, J. M. Pedersen, and E. Vasilomanolakis, “Open for Hire: Attack Trends and Misconfiguration Pitfalls of IoT Devices,” in *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, (New York, NY, USA: Association for Computing Machinery, 2021): 195–215, https://doi.org/10.1145/3487552.3487833.

9. M. Khera, “Think Like a Hacker: Insights on the Latest Attack Vectors (and Security Controls) for Medical Device Applications,” *Journal of Diabetes Science and Technology* 11, no. 2 (2017): 207–212, PMID: 27920270; https://doi.org/10.1177/1932296816677576.

10. Z. Yin, X. Ma, J. Zheng, Y. Zhou, L. N. Bairavasundaram, and S. Pasupathy, “An Empirical Study on Configuration Errors in Commercial and Open Source Systems,” in *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, (New York, NY, USA: Association for Computing Machinery, 2011): 159–172, https://doi.org/10.1145/2043556.2043572.

11. W. ElMaraghy, H. ElMaraghy, T. Tomiyama, and L. Monostori, “Complexity in Engineering Design and Manufacturing,” *CIRP Annals* 61, no. 2 (2012): 793–814, https://doi.org/10.1016/j.cirp.2012.05.001.

12. J. Corral-Acero, F. Margara, M. Marciniak, et al., “The ‘Digital Twin’ to Enable the Vision of Precision Cardiology,” *European Heart Journal* 41, no. 48 (2020): 4556–4564, https://academic.oup.com/eurheartj/article-pdf/41/48/4556/46616698/ehaa159.pdf.

13. J. Eyre, S. Hyde, D. Walker, et al., *Untangling the Requirements of a Digital Twin* (Sheffield: Advanced Manufacturing Research Centre, 2021), https://www.amrc.co.uk/pages/digital-twin-report.

14. J. A. Douthwaite, B. Lesage, M. Gleirscher, et al., “A Modular Digital Twinning Framework for Safety Assurance of Collaborative Robotics,” *Frontiers in Robotics and AI* 8 (2021): 758099.

15. R. J. Somers, J. A. Douthwaite, D. J. Wagg, N. Walkinshaw, and R. M. Hierons, “Digital-Twin-Based Testing for Cyber–Physical Systems: A Systematic Literature Review,” *Information and Software Technology* 156 (2023): 107145, https://doi.org/10.1016/j.infsof.2022.107145.

16. T. Erol, A. F. Mendi, and D. Doğan, “The Digital Twin Revolution in Healthcare,” in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, (Istanbul, Turkey: IEEE, 2020): 1–7, https://doi.org/10.1109/ISMSIT50672.2020.9255249.

17. G. Coorey, G. A. Figtree, D. F. Fletcher, and J. Redfern, “The Health Digital Twin: Advancing Precision Cardiovascular Medicine,” *Nature Reviews Cardiology* 18, no. 12 (2021): 803–804.

18. H. Thabit and R. Hovorka, “Coming of Age: The Artificial Pancreas for Type 1 Diabetes,” *Diabetologia* 59, no. 9 (2016): 1795–1805 en.

19. M. N. Teferra, “ISO 14971-Medical Device Risk Management Standard,” *International Journal of Latest Research in Engineering and Technology (IJLRET)* 3, no. 3 (2017): 83–87.

20. T. Battelino, T. Danne, R. M. Bergenstal, et al., “Clinical Targets for Continuous Glucose Monitoring Data Interpretation: Recommendations From the International Consensus on Time in Range,” *Diabetes Care* 42, no. 8 (2019): 1593–1603, https://doi.org/10.2337/dci19-0028.

21. N. Kagansky, S. Levy, and H. Knobler, “The Role of Hyperglycemia in Acute Stroke,” *Archives of Neurology* 58, no. 8 (2001): 1209–1212.

22. E. R. Seaquist, J. Anderson, B. Childs, et al., “Hypoglycemia and Diabetes: A Report of a Workgroup of the American Diabetes Association and the Endocrine Society,” *Diabetes Care* 36, no. 5 (2013): 1384–1395.

23. M. L. Litchman, H. R. Walker, C. Fitzgerald, M. Gomez Hoyos, D. Lewis, and P. M. Gee, “Patient-Driven Diabetes Technologies: Sentiment and Personas of the #WeAreNotWaiting and #OpenAPS Movements,” *Journal of diabetes science and technology* 14, no. 6 (2020): 990–999.

24. A. Shahid and D. M. Lewis, “Large-Scale Data Analysis for Glucose Variability Outcomes With Open-Source Automated Insulin Delivery Systems,” *Nutrients* 14, no. 9 (2022): 1906.

25. OpenAPS, “OpenAPS.org—#WeAreNotWaiting to Reduce the Burden of Type 1 Diabetes,” https://openaps.org/.

26. M. J. Burnside, D. M. Lewis, H. R. Crocket, et al., “Open-Source Automated Insulin Delivery in Type 1 Diabetes,” *New England Journal of Medicine* 387, no. 10 (2022): 869–881, PMID: 36069869; https://doi.org/10.1056/NEJMoa2203913.

27. D. Lewis, “openaps/oref0 Releases—GitHub,” https://github.com/openaps/oref0/releases.

28. OpenAPS, “Understanding Your Preferences and Safety Settings,” (2017), https://openaps.readthedocs.io/en/latest/docs/While%20You%20Wait%20For%20Gear/preferences-and-safety-settings.html#oref1-related-preferences.

29. “Loop and Learn,” https://www.loopnlearn.org/starting-loop/.

30. “Issues openaps/oref0,” https://github.com/openaps/oref0/issues.

31. “OpenAPS Gitter,” https://app.gitter.im/#/room/#nightscout_intend-to-bolus:gitter.im.

32. OpenAPS, “Facebook TheLoopedGroup,” https://www.facebook.com/groups/TheLoopedGroup.

33. OpenAPS, “OpenAPS Dev,” Google, https://groups.google.com/g/openaps-dev?pli=1.

34. T. Xu and Y. Zhou, “Systems Approaches to Tackling Configuration Errors: A Survey,” *ACM Computing Surveys* 47, no. 4 (2015): 1–41, https://doi.org/10.1145/2791577.

35. T. Xu and O. Legunsen, “Configuration Testing: Testing Configuration Values as Code and With Code,” (2019), arXiv preprint arXiv:1905.12195.

36. R. Cheng, L. Zhang, D. Marinov, and T. Xu, “Test-Case Prioritization for Configuration Testing,” in *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2021, (New York, NY, USA: Association for Computing Machinery, 2021): 452–465, https://doi.org/10.1145/3460319.3464810.

37. M. B. Cohen, J. Snyder, and G. Rothermel, “Testing Across Configurations: Implications for Combinatorial Testing,” *ACM SIGSOFT Software Engineering Notes* 31, no. 6 (2006): 1–9, https://doi.org/10.1145/1218776.1218785.

38. M. Liu, S. Fang, H. Dong, and C. Xu, “Review of Digital Twin About Concepts, Technologies, and Industrial Applications,” *Journal of Manufacturing Systems* 58 (2021): 346–361.

39. Z. Zhu, C. Liu, and X. Xu, “Visualisation of the Digital Twin Data in Manufacturing by Using Augmented Reality,” *Procedia CIRP* 81 (2019): 898–903, https://doi.org/10.1016/j.procir.2019.03.223.

40. Y. Peng, S. Zhao, and H. Wang, “A Digital Twin Based Estimation Method for Health Indicators of DC–DC Converters,” *IEEE Transactions on Power Electronics* 36, no. 2 (2021): 2105–2118, https://doi.org/10.1109/TPEL.2020.3009600.

41. R. Sahal, S. H. Alsamhi, and K. N. Brown, “Personal Digital Twin: A Close Look Into the Present and a Step Towards the Future of Personalised Healthcare Industry,” *Sensors (Basel, Switzerland)* 22, no. 15 (2022): 5918.

42. B. Björnsson, C. Borrebaeck, N. Elander, et al., “Digital Twins to Personalize Medicine,” *Genome Medicine* 12, no. 1 (2019): 4.

43. S. A. Niederer, J. Lumens, and N. A. Trayanova, “Computational Models in Cardiology,” *Nature Reviews Cardiology* 16, no. 2 (2019): 100–111, https://doi.org/10.1038/s41569-018-0104-y.

44. J. Huang, L. Lin, F. Yu, et al., "Parkinson's Severity Diagnosis Explainable Model Based on 3D Multi-Head Attention Residual Network," *Computers in Biology and Medicine* 170 (2024): 107959.

45. S. Khan, A. Alzaabi, T. Ratnarajah, and T. Arslan, "Novel Statistical Time Series Data Augmentation and Machine Learning Based Classification of Unobtrusive Respiration Data for Respiration Digital Twin Model," *Computers in Biology and Medicine* 168 (2024): 107825, https://doi.org/10.1016/j.compbiomed.2023.107825.

46. J. Al-Jaroodi, N. Mohamed, and E. Abukhousa, "Health 4.0: On the Way to Realizing the Healthcare of the Future," *IEEE Access* 8 (2020): 211189–211210, https://doi.org/10.1109/ACCESS.2020.3038858.

47. G. L. Tortorella, F. S. Fogliatto, A. Mac Cawley Vergara, R. Vassolo, and R. Sawhney, "Healthcare 4.0: Trends, Challenges and Research Directions," *Production Planning & Control* 31, no. 15 (2020): 1245–1260.

48. M. N. Kamel Boulos and P. Zhang, "Digital Twins: From Personalised Medicine to Precision Public Health," *Journal of Personalized Medicine* 11, no. 8 (2021): 745.

49. S. Contreras, D. Medina-Ortiz, C. Conca, and A. Olivera-Nappa, "A Novel Synthetic Model of the Glucose-Insulin System for Patient-Wise Inference of Physiological Parameters From Small-Size OGTT Data," *Frontiers in Bioengineering and Biotechnology* 8 (2020): 195.

50. C. Furió-Novejarque, R. Sanz, T. K. Ritschel, et al., "Modeling the Effect of Glucagon on Endogenous Glucose Production in Type 1 Diabetes: On the Role of Glucagon Receptor Dynamics," *Computers in Biology and Medicine* 154 (2023): 106605.

51. M. R. Davies, K. Wang, G. R. Mirams, et al., "Recent Developments in Using Mechanistic Cardiac Modelling for Drug Safety Evaluation," *Drug Discovery Today* 21, no. 6 (2016): 924–938.

52. A. S. Januszewski, Y. H. Cho, M. V. Joglekar, et al., "Insulin Micro-Secretion in Type 1 Diabetes and Related MicroRNA Profiles," *Scientific Reports* 11, no. 1 (2021): 11727.

53. S. Katoch, S. S. Chauhan, and V. Kumar, "A Review on Genetic Algorithm: Past, Present, and Future," *Multimedia Tools and Applications* 80, no. 5 (2021): 8091–8126.

54. A. F. Gad, "PyGAD: An Intuitive Genetic Algorithm Python Library," *Multimedia Tools and Applications* 83 (2024): 58029–58042.

55. P. Runeson, M. Höst, A. Rainer, and B. Regnell, "Design of the Case Study," in *Case Study Research in Software Engineering* (John Wiley & Sons Ltd, 2012): 23–45.

56. P. Ralph, S. Baltes, D. Bianculli, et al., "ACM SIGSOFT empirical standards, CoRR abs/2010.03525," (2020), arXiv:2010.03525.

57. C. H. Lee and H.-J. Yoon, "Medical Big Data: Promise and Challenges," *Kidney Research and Clinical Practice* 36, no. 1 (2017): 3–11 en.

58. A. Facchinetti, S. Del Favero, G. Sparacino, J. R. Castle, W. K. Ward, and C. Cobelli, "Modeling the Glucose Sensor Error," *IEEE Transactions on Biomedical Engineering* 61, no. 3 (2014): 620–629, https://doi.org/10.1109/TBME.2013.2284023.

59. W. L. Clarke, "The Original Clarke Error Grid Analysis (EGA)," *Diabetes Technology & Therapeutics* 7, no. 5 (2005): 776–779.

60. A. Maran, C. Crepaldi, A. Tiengo, et al., "Continuous Subcutaneous Glucose Monitoring in Diabetic Patients: A Multicenter Analysis," *Diabetes Care* 25, no. 2 (2002): 347–352, https://doi.org/10.2337/diacare.25.2.347.

61. B. P. Kovatchev, L. A. Gonder-Frederick, D. J. Cox, and W. L. Clarke, "Evaluating the Accuracy of Continuous Glucose-Monitoring Sensors: Continuous Glucose–Error Grid Analysis Illustrated by TheraSense Freestyle Navigator Data," *Diabetes Care* 27, no. 8 (2004): 1922–1928, https://doi.org/10.2337/diacare.27.8.1922.

62. D. Angelis, F. Sofos, and T. E. Karakasidis, "Artificial Intelligence in Physical Sciences: Symbolic Regression Trends and Perspectives," *Archives of Computational Methods in Engineering* 30, no. 6 (2023): 3845–3865.

63. O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-Art in Artificial Neural Network Applications: A Survey," *Heliyon* 4, no. 11 (2018).

64. J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, "A Critical Look at the Current Train/Test Split in Machine Learning," (2021), arXiv preprint arXiv:2106.04525.

65. D. J. Wagg, K. Worden, R. J. Barthorpe, and P. Gardner, "Digital Twins: State-of-the-Art and Future Directions for Modeling and Simulation in Engineering Dynamics Applications," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering* 6, no. 3 (2020): 030901, https://doi.org/10.1115/1.4046739.

66. D. Kiagias, G. Russo, G. Sgroi, F. Pappalardo, and M. A. Juárez, "Bayesian Augmented Clinical Trials in TB Therapeutic Vaccination," *Frontiers in Medical Technology* 3 (2021): 719380.

67. R. H. Clayton, O. Bernus, E. M. Cherry, et al., "Models of Cardiac Tissue Electrophysiology: Progress, Challenges and Open Questions," *Progress in Biophysics and Molecular Biology* 104, no. 1 (2011): 22–48, https://doi.org/10.1016/j.pbiomolbio.2010.05.008.

68. G. R. Mirams, P. Pathmanathan, R. A. Gray, P. Challenor, and R. H. Clayton, "Uncertainty and Variability in Computational and Mathematical Models of Cardiac Physiology," *Journal of Physiology* 594, no. 23 (2016): 6833–6847, https://doi.org/10.1113/JP271671.

69. Y. Peng, X. Zhang, Y. Song, and D. Liu, "A Low Cost Flexible Digital Twin Platform for Spacecraft Lithium-Ion Battery Pack Degradation Assessment," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (Auckland, New Zealand: IEEE, 2019): 1–6.

70. K. Xia, C. Sacco, M. Kirkpatrick, et al., "A Digital Twin to Train Deep Reinforcement Learning Agent for Smart Manufacturing Plants: Environment, Interfaces and Intelligence," *Journal of Manufacturing Systems* 58 (2021): 210–230, https://doi.org/10.1016/j.jmsy.2020.06.012.

71. E. Cioroaica, F. Di Giandomenico, T. Kuhn, et al., "Towards Runtime Monitoring for Malicious Behaviors Detection in Smart Ecosystems," in *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, (Berlin, Germany: IEEE, 2019): 200–203, https://doi.org/10.1109/ISSREW.2019.00072.

72. V. Rothberg, C. Dietrich, A. Ziegler, and D. Lohmann, "Towards Scalable Configuration Testing in Variable Software," *ACM SIGPLAN Notices* 52, no. 3 (2016): 156–167, https://doi.org/10.1145/3093335.2993252.

73. C. Curreli, V. Di Salvatore, G. Russo, F. Pappalardo, and M. Viceconti, "A Credibility Assessment Plan for an In Silico Model That Predicts the Dose–Response Relationship of New Tuberculosis Treatments," *Annals of Biomedical Engineering* 51, no. 1 (2023): 200–210, https://doi.org/10.1007/s10439-022-03078-w.

74. L. Edington, N. Dervilis, A. Ben Abdessalem, and D. Wagg, "A Time-Evolving Digital Twin Tool for Engineering Dynamics Applications," *Mechanical Systems and Signal Processing* 188 (2023): 109971.

75. Y. Liu, "Overview of Some Theoretical Approaches for Derivation of the Monod Equation," *Applied Microbiology and Biotechnology* 73, no. 6 (2007): 1241–1250.

76. P. M. Jedrzejewski, I. J. Del Val, A. Constantinou, et al., "Towards Controlling the Glycoform: A Model Framework Linking Extracellular Metabolites to Antibody Glycosylation," *International Journal of Molecular Sciences* 15, no. 3 (2014): 4492–4522.

77. K. Sharabi, C. D. J. Tavares, A. K. Rines, and P. Puigserver, "Molecular Pathophysiology of Hepatic Glucose Production," *Molecular Aspects of Medicine* 46 (2015): 21–33.

Appendix A

Model Modification

The model implementation by Contreras et al. [48] produced negative hepatic blood glucose (G_{prod}) at blood glucose levels less than the steady-state blood glucose level (G_b). We modified the original model by introducing a Monod equation [75] for G_{prod} with respect to $-G$. This is inline with prior work that has used Monod equations to represent hepatic glucose production [76] as this approach a simple yet effective representation of this behaviour [77]. Equation (A1) introduces the hepatic glucose growth rate (k_μ), the hepatic glucose limit (k_λ) and the hepatic glucose production (G_{prod0}) at the glucose steady state (G_b).

$$G_{prod} = \frac{k_\lambda(G_b - G)}{k_\mu + (G_b - G)} + G_{prod0}. \tag{A1}$$

TABLE A1 | The meaning of each constant from $K_x(T)$ representing the blood glucose–insulin dynamics of a person.

Constant	Meaning
k_{js}	Rate of carbohydrate movement from stomach to jejunum
k_{gj}	Rate of carbohydrate absorption from jejunum to blood
k_{jl}	Rate of carbohydrate movement from jejunum to ilium
k_{gl}	Rate of carbohydrate absorption from ilium to blood
k_{xg}	Rate of blood glucose basal uptake
k_{xgi}	Rate of blood glucose insulin sensitivity uptake
k_{xi}	Rate of insulin degradation
τ	Time taken for ilium to receive glucose from jejunum
η	Bioavailability of glucose absorbed in the intestine
k_λ	Hepatic glucose production limit
k_μ	Hepatic glucose production growth rate
G_{prod0}	Hepatic glucose production at the steady state

Appendix B

Fitting Algorithm Choice

When deciding the technique for fitting the digital twin, we evaluated four different metaheuristic algorithms. Each algorithm was fitted using the same 10 randomly selected datasets and executed 10 times to account for randomness. Figure A1 presents the fitness values for each metaheuristic algorithm. It can be seen that the generic algorithm resulted in the highest average fitness across traces. We use this to justify our choice of fitting algorithm. The code used to generate this figure can be found in *fitness_checker.py* in our replication package.

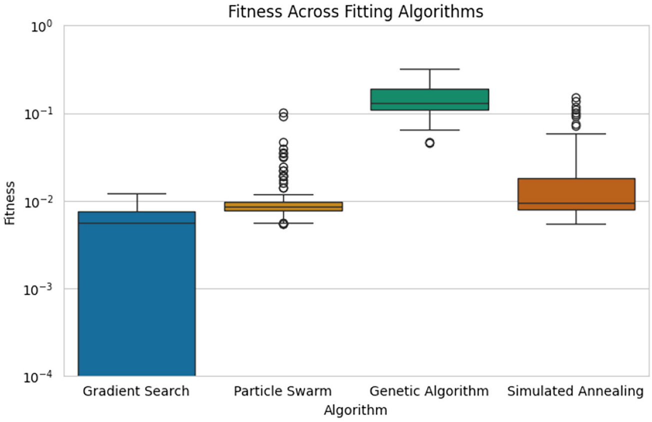


FIGURE A1 | The fitness values when fitting the digital twin across four different fitting algorithms. *Note:* The scale of fitness is logarithmic.