Towards Developing Safety Assurance Cases for Learning-Enabled Medical Cyber-Physical Systems

Maryam Bagheri*, Josephine Lamp, Xugui Zhou, Lu Feng and Homa Alemzadeh

University of Virginia, Charlottesville, VA, USA, {ntb5gu, jl4rj, xugui, lu.feng, alemzadeh}@virginia.edu

Abstract

Machine Learning (ML) technologies have been increasingly adopted in Medical Cyber-Physical Systems (MCPS) to enable smart healthcare. Assuring the safety and effectiveness of learning-enabled MCPS is challenging, as such systems must account for diverse patient profiles and physiological dynamics and handle operational uncertainties. In this paper, we develop a safety assurance case for ML controllers in learning-enabled MCPS, with an emphasis on establishing confidence in the ML-based predictions. We present the safety assurance case in detail for Artificial Pancreas Systems (APS) as a representative application of learning-enabled MCPS, and provide a detailed analysis by implementing a deep neural network for the prediction in APS. We check the sufficiency of the ML data and analyze the correctness of the ML-based prediction using formal verification. Finally, we outline open research problems based on our experience in this paper.

Keywords

Machine Learning, Safety Assurance Case, Medical Cyber-Physical Systems, Artificial Pancreas System

1. Introduction

Medical Cyber-Physical Systems (MCPS) integrate connected software and hardware components with sensors and actuators to monitor and control patient physiology. Machine Learning (ML) technologies have been increasingly used in MCPS, often deployed in the estimation and prediction components, to make data-driven decisions based on sensor or patient input and guide control actions [1]. Ensuring the successful deployment of ML within MCPS can be challenging, as the ML components must be able to handle the intricacies of patient physiology, time lags between the impact of a control action and sensor measurements, uncertainties in the operational environment that may affect the patient's physiology, and variability in patient profiles which may result in differing impacts of the control actions. Moreover, due to physiological complexities and the limited availability of realistic patient profiles or datasets, ML techniques may use synthetic data or virtual patient models for training. The mismatch between the training data and the real-world data seen in deployment may result in erroneous, biased, or incomplete output predictions [2]. Failure of the ML component due to any of the challenges noted above could result in irreparable harm to patients. As such, the use of ML within MCPS should be assured by evidence that these components are safe and reliable [2].

Assurance cases (AC) are structured arguments, supported by evidence, to justify claims for an application in a given environment [3]. Recent efforts in safety analysis and assurance have confirmed that AC are valuable for assessing and demonstrating trust [4]. For example, AC have been deployed for unmanned aircraft systems [5] and medical devices [6]. The U.S. FDA has also issued a guideline [7] suggesting medical manufacturers provide AC with pre-market submissions. Among the standards published by various organizations (e.g., ISO 26262 [8], ISO/PAS 21448 [9], ANSI/UL 4600 [10], and the FDA guideline for Artificial Pancreas Systems (APS) [11]), ANSI/UL 4600 is the only one offering evaluations of ML technology for the safety of autonomous vehicles.

Among more AC-centric studies, Hawkins et al. [12] provide a guideline on the assurance of ML in autonomous systems, including a safety case pattern for each stage of the ML life cycle. Kaur et al. [13] proposed a modular assurance case pattern based on assume/guarantee reasoning for ML-enabled CPS, where the safety of the ML component and the rest of the system are assessed separately. However, the ML lifecycle is not dealt with in this pattern. A few studies have also developed AC for concrete learning-enabled use cases in the automotive domain [14, 15, 16]. Within the healthcare domain, [17] is the only work that presents an assurance case pattern to justify the use of ML. Even so, none of [12]-[17] instantiate the process activities and generate evidence for a concrete application in the medical domain. This paper tackles this gap by presenting detailed AC to assure the safety and effectiveness of a general framework of APS [18], suitable for all types of APS. We select APS as a representative of learning-enabled MCPS as it contains a collection of typical components of many ML-enabled MCPS, including data-driven estimation algorithms and embedded controllers for insulin dosage calculation.

SafeAI: The AAAI's Workshop on Artificial Intelligence Safety, Feb 13–14, 2023, Washington, D.C., US

*Corresponding author.

0000-0001-9576-2478 (M. Bagheri); 0000-0002-4982-7768 (J. Lamp); 0000-0002-3663-7447 (X. Zhou); 0000-0002-4651-8441 (L. Feng); 0000-0001-5279-842X (H. Alemzadeh)

Expanding on the patterns proposed in [12, 13], we develop a safety assurance case for the ML-enabled controller of MCPS, consisting of a controller algorithm and an ML prediction algorithm, where our emphasis is greatly on the ML-based prediction. By proving trust in ML prediction, the safety of the controller algorithm can be assessed in isolation. Considering the patient as an essential element of the control loop, we discuss the AC elements that should be instantiated based on an individual patient profile or a population of patients. The instantiation is due to different physiologies of different patients, which may affect control actions, the ML controller's expected behavior, and the claims satisfaction. This is the first time that the patient profiles are included in AC for ML controllers and MCPS. In support of claims in AC for APS, we implement a deep neural network for blood glucose prediction in APS. We then present an analysis characterizing the sufficiency of the training data for the ML controller and the ML development process. We also utilize APS domain knowledge to specify a set of properties based on a patient's metabolism (e.g., insulin senstivity, carbohydrate absorption profile) and use formal verification to check them against the ML prediction component. We are unaware of any work verifying the ML components in APS except [19], which unlike us, compares the output ranges of two identical networks given a slight change in their input ranges.

Contributions. The major contributions of this paper are summarized as follows:

- We present preliminary results on developing a safety assurance case template for ML controllers in MCPS, which includes patient profiles in its element descriptions.
- We present a detailed safety assurance case for APS that is supported by a thorough analysis of ML-based glucose prediction module.
- We define properties based on the body's metabolism and check them against the ML prediction component using formal verification.

In the end, we discuss open research problems in developing safety assurance cases for learning-enabled MCPS.

2. Artificial Pancreas Systems

Type 1 Diabetes (T1D) is a chronic disease in which a patient's pancreas produces little to no insulin. Patients with T1D must constantly monitor their blood glucose (BG) levels and inject insulin to regulate their concentrations of BG. Artificial Pancreas Systems (APS) are closed-loop insulin delivery systems that relieve the burden of T1D on patients by regulating a patientâĂŹs BG level, using input from various sensors such as continuous glucose monitors (CGM). Figure 1 depicts the typical structure of APS, consisting of a CGM sensor to continuously monitor BG values, an APS controller that calculates the correct insulin dosages based on the CGM and user input, and an insulin pump that delivers insulin dosages.

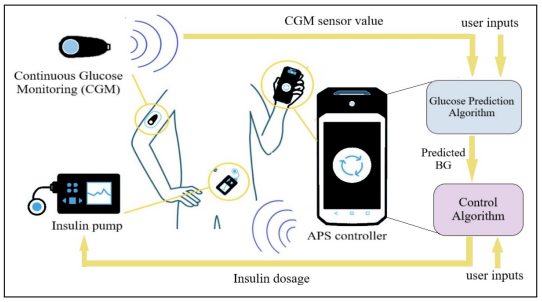


Figure 1.: The structure of APS (modified from [20]). The APS controller consists of a prediction algorithm to predict the BG values and a controller algorithm to adjust the insulin dosage.

The APS controller consists of a data-driven glucose prediction algorithm to predict future BG values and a control algorithm to adjust the insulin dosage based on the predicted BG values. Recently, researchers have begun exploring the use of neural networks [21, 22] and reinforcement learning [23] for the design of the APS controllers (e.g., use of machine learning for glucose prediction). The primary objective of the controller is to provide safe and efficient glycemic control by infusing an appropriate amount of insulin to keep the patient's BG within the proper range (between 70 and 180 mg/dL) and avoid hypoglycemias and hyperglcemias. To provide such safe control, the controller needs to account for the complexities of glucose metabolism and deal with unpredicted meal intake, exercise, stress, or illness, rapid changes in BG concentration, and time lags between BG measurement and insulin impact.

As of the date of this writing, there are four commercially available APS that have received FDA approval and/or the Conformité Européenne (CE) mark: Medtronic MiniMed 670/770/780G [24], Tandem Control-IQ [25], Omnipod 5 [26], and CamAPS FX [27]. These systems use some form of data-driven learning algorithm, i.e., Tandem Control-IQ uses a simple linear regression algorithm [28] to predict BG values 30 minutes in the future and then a PID algorithm to adjust the insulin dosage based on the predicted BG values. To ensure the adoption and use of such systems, patients need to be confident in the underlying ML technology embedded in the controllers.

3. Overall Safety Assurance Case

Typical CPS consist of embedded software and hardware components controlling the plant through interconnected sensors and actuators. MCPS are a distinct class of CPS with the patient as the plant, aiming to monitor and control multiple aspects of the patient's physiology. With the patient in the control loop, the MCPS should either be tai-

lored for specific physiological parameters of the patient or should cover a population of patients. This adaptation is even more critical as a part of the design process in learning-enabled MCPS that employ a learning-enabled controller, a controller relying on machine learning to perform perception or prediction tasks. So, in the regulatory process for checking the safety of MCPS, it would make sense to instantiate the safety assurance case for individual patient profiles or populations. To emphasize this, we mark the context elements in safety assurance cases with the uppercase letter P in a half circle to denote the decision points where the context needs to be initialized for an individual patient profile or the population.

The top-level goal in safety AC of learning-enabled MCPS is to ensure that "x as a case of learning-enabled MCPS is safe and effective". Confidence in this claim is obtained by ensuring the safety and effectiveness of all constituent components, including sensors, actuators, and their interactions. For this paper, we discuss only the safety and effectiveness of the *learning-enabled controller*, assuming that the safety and effectiveness of other components have been adequately examined. We first present a safety assurance case template for a learning-enabled controller in MCPS, shown in Figure 2. We then use APS as an instance of MCPS and explain how instantiating the template results in a general safety assurance case for APS. We use goal structuring notation (GSN) [29], with a slight of notation abuse, to show our AC.

3.1. Safety Assurance Case Template for learning-enabled controllers in MCPS

The root goal G0 in Figure 2 asserts that the learning-enabled controller c is safe and effective while the device is used in treating the patient. The environment and the system within which the learning-enabled controller is used are described in context C0-1, and the requirements assigned to the learning-enabled controller are explained in context C0-2. We use assume/guarantee reasoning to justify goal G0. This is because the learning-enabled controller c is a combination of two main algorithms that perform in sequence: an algorithm that performs the ML tasks and delivers its output to the control algorithm, and the control algorithm that selects the control action and initiates it in the system. Assuming that the results provided by the ML tasks are correct, the control algorithm should guarantee the safe and effective treatment of the patient. Hereafter, we use the term controller to refer to the control algorithm of the learning-enabled controller. We consider a separate component for each algorithm. As reflected in goal G1-2, the safety and effectiveness of the ML component are justified in isolation. The controller's input is an interface with the ML component, containing the results received from the ML component.

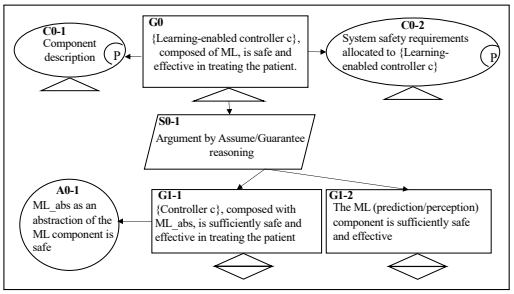


Figure 2.: A safety assurance case template for a learning-enabled controller in MCPS.

This interface shows an abstraction of the ML component, denoted as ML_abs in goal G1-1. Assuming the outputs of the ML component are safe, as presented in assumption A0-1, goal G1-1 claims that the controller component combined with ML_abs is safe and effective.

3.2. Instantiating Learning-Enabled Controller Assurance Case for APS

Regardless of the system type and the technology underlying the APS, the main components of APS remain the same, i.e., they all consist of a learning-enabled controller. Additionally, APS should satisfy a set of safety and performance properties common and desirable to regulatory agencies. For instance, requirements specified by the FDA [11] include high accuracy of CGM readings, safe insulin dosages, usable design, and so on. The most significant requirement is that APS must not increase the incidence and severity of hypoglycemic and hyperglycemic events. These reasons induce a general safety assurance case to be developed for all APS, like [30] which presents AC for a generic infusion pump device. Thus, the proposed template in Figure 2 can be employed for APS, where goals G1-1 and G1-2 are modified as follows.

• G1-1: Assuming that the BG predictions are accurate, the insulin dosage management component is sufficiently safe and effective for treating patients.

• G1-2: The ML glucose prediction component is sufficiently safe and effective.

Context Elements. Context C0-1 describes inputs to the APS controller (i.e., history of the CGM values and insulin injected and prediction horizon), outputs of the controller (i.e., the amount of insulin to be injected), the component's role in the system, and the environmental phenomena (i.e., uncertain meal intake, daily activity).

Context C0-2 includes all requirements of the learning-enabled controller. Table 1 shows a set of these requirements, which we have extracted from the diabetes treatment literature (e.g., [31]). The main requirement in Table 1 is RQ.C.1, which is further refined into its following requirements. Although this is not an extensive list of requirements, it represents some of the most impor-

Table 1.
Requirements for the learning-enabled APS controller

RQ.C.1	Accurately calculate dose of basal and bolus insulin
RQ.C.1.1	Determine the output every T minutes (e.g., T=5 in MiniMed)
RQ.C.1.2	Stop dosing if a maximum amount has been delivered by the pump
RQ.C.1.3	Suspend dosing if the actual or predicted CGM readings fall below a threshold
RQ.C.1.4	Interrupt in a safe way if trustworthy control is not guaranteed
RQ.C.1.5	BG should not remain below 10th-percentile threshold for more than α_1 minutes
RQ.C.1.6	BG should not remain above 90th-percentile threshold for more than α_2 minutes following a bolus injection
RQ.C.1.7	BG should not remain above 90th-percentile threshold for more than α_3 minutes
RQ.C.1.8	The BG value is always greater than 70 and less than 180
RQ.C.1.9	The controller infuses additional insulin while the blood glucose level is below a target level
RQ.C.1.10	The morning wake up blood glucose level can not exceed β

Table 2.
Performance and robustness requirements (that are independent of ML technology) for the ML glucose prediction component

Performance	
ML-RQ1	Accurately predict the BG values T minutes in the future
ML-RQ1.1	BG's rate of change has to be bound by established physiological norms
ML-RQ1.2	Meal intake has a direct effect on the BG value
ML-RQ1.3	Exercise has an inverse effect on the BG value
ML-RQ1.4	Within t minutes of a bolus, there should be an accompanying change in BG of more than α
ML-RQ1.5	The glucose level starts to rise at a specific time after a meal's onset
ML-RQ1.6	There is a delay between the injection of insulin and the disposal of glucose
ML-RQ1.7	The blood concentration of insulin reaches its maximum after a particular time
ML-RQ1.8	Insulin has an inverse effect on the BG value
Robustness	
ML-RQ2	Perform as required for different patients of different ages/sexes
ML-RQ3	Perform as required in the presence of external factors such as meals and exercises.

tant requirements for such a system. A few examples of peripheral requirements are related to the controller's platform, such as security, reliability, and usability. For example, the smartphone used in some of the APS is a platform. The goal G1-1 is also defined in the same context as C0-2 since it claims the safety and effectiveness of the APS controller when BG predictions are reliable. We assume that the controller requirements are adequately examined in consultation with domain experts and medical professionals and do not discuss them in this paper.

Patient or Population. The contexts C0-1 and C0-2 can be defined based on an individual patient profile or a population of patients. A clear example of this decision is that different patients have varying insulin sensitivity levels, and their physiologies may be affected differently by meal volume or activity. In addition to the training datasets and the control algorithms themselves, other components such as thresholds and target values that affect the requirements, can be defined differently based on individual patients or a population of patients.

In the next section, we develop a general argument for goal G1-2, claiming that the ML glucose prediction component is sufficiently safe and effective.

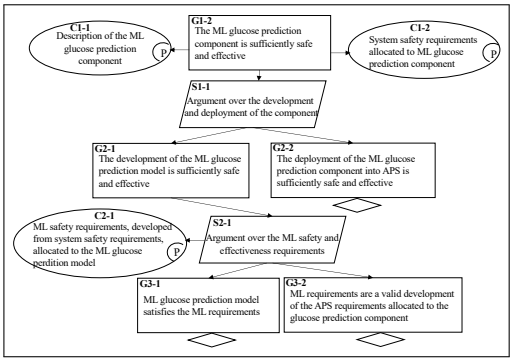


Figure 3.: A general safety assurance case for the ML glucose prediction component of APS.

4. Safety Assurance Case for the Glucose Prediction Component

Different parts of the safety assurance case developed for the glucose prediction component are shown in Figures 3, 4, and 5. We describe each part in a separate section and provide concrete pieces of evidence in Section 5.

4.1. Sufficiency of the ML Development

The argument to justify claim G1-2 is shown in Figure 3. This claim is supported by contexts C1-1 and C1-2. Context C1-1 describes the ML prediction component, its expected inputs and outputs (e.g., CGM, insulin, and meal

values), along with their possible sources and targets (e.g., various CGM or pump devices). Besides, it is necessary to determine whether the component is specialized based on a patient profile or a population of patients.

We categorize the requirements allocated to the ML glucose prediction component into performance and robustness requirements and enumerate them in Table 2. These requirements are independent of ML technology and are defined in context C1-2. ML-RQ1 is the primary performance requirement refined into ML-RQ1.1 through ML-RQ1.8 based on the patient's physiology. Prediction results are accurate when the learning component has learned the physiological dynamics of the patients, and hence ML-RQ1.1 to ML-RQ1.8 are satisfied. Although we have extracted these requirements from the APS literature, based on our knowledge, it is the first time that physiologically-inspired requirements are assigned to an ML component. The thresholds and target values in these requirements depend on the patient's or population's profiles. Robustness requirements ML-RQ2 and ML-RQ3 refer to variations in the input space of the component. For instance, ML-RQ2 ensures a variety of patient profiles are considered in a population-based setting.

To justify claim G1-2, the approach of [12] splits the argument based on the development and deployment of the ML component. Goal G2-1 claims that the development of the ML model predicting the BG values is sufficiently safe and effective, and goal G2-2 claims that the integration of the ML component into the system is sufficiently safe and effective. The G2-2 justification involves techniques such as runtime assessment that are beyond the scope of this paper. We leave G2-2 undeveloped and emphasize G2-1. The first step to support claim G2-1 is to develop ML requirements using the concepts amenable to the ML implementation. The performance requirement ML-RQ1 in Table 2 can be measured by the accuracy or mean prediction error of the ML algorithm. Thus, ML-RQ1 is defined as " ML component should predict the glucose value with the mean prediction error of less than *thres* mg/dL", where *thres* is determined by human experts or compared to the most reliable existing method for BG prediction. ML-RQ1.1 to ML-RQ1.8 are meaningful to the ML model when defined over inputs and outputs of the ML model. We specify these requirements in Section 5. These ML requirements are expressed in context C2-1.

The development of the ML component refers to the process of designing and training the ML model. So, claim G2-1 is supported by sub-claims G3-1 and G3-2 through strategy S2-1. Goal G3-1 claims that the ML model satisfies the ML requirements. A complete argumentation must demonstrate that the ML requirements are a valid development of the APS requirements allocated to the glucose prediction component, as expressed in claim G3-2. In our case, there is an exact mapping between requirements of Table 2 and the ML requirements

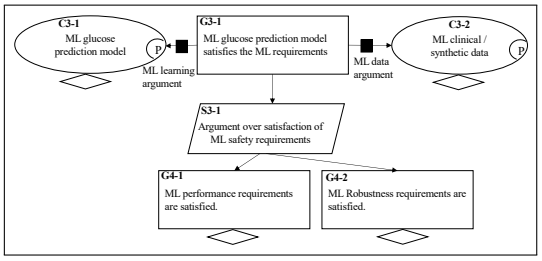


Figure 4.: Argument to ensure the sufficiency of the ML glucose prediction model.

(Section 5), so G3-2 is justified. The G3-1 justification is described in the next section.

4.2. Sufficiency of the ML Model

The argument to ensure the sufficiency of the ML glucose prediction model is shown in Figure 4. The claim G3-1 is made in context C3-1 of the ML model created and context C3-2 of the ML data, respectively. ML data can include data from only an individual patient or a population of patients, and the ML model and its hyperparameters are tuned based on collected data. The goal G3-1 is supported by goals G4-1 and G4-2 claiming that the ML model satisfies the performance and robustness requirements. Further assurance is also needed regarding the ML process and ML data used for development. The ML learning and ML data arguments provide arguments and evidence for the safety and effectiveness of the ML process and ML data. We provide an assurance case for the sufficiency of ML data in Section 4.3 and concrete evidence for both arguments in Section 5. The links with the ML learning and data arguments are established using assurance claim points [12] (black squares), representing the points at which further assurance is required.

4.3. Sufficiency of the ML Data

The argument to ensure the sufficiency of the ML data is shown in Figure 5. Claim G4-3 justifies that the data collected meet desiderata, including relevance, completeness, balance, and accuracy [12], thus, the assurance that the model trained on such data satisfies ML requirements increases. The first step to check the data against the desiderata is to provide a list of ML data requirements for each desideratum. The sub-claim G5-1 assures that the list has sufficient ML data requirements, and the sub-claim G5-2 checks whether the data meet the ML data requirements. We enumerate the ML data requirements in Table 3. Section 5 provides concrete evidence to support G5-2. In the following, we describe data requirements and their effects on the satisfaction of the performance and robustness requirements in support of G5-1.

The requirements DR.R1 and DR.R2 concern the position of the CGM sensor on the patient's body and the format of the data captured by the sensor, respectively. A CGM sensor is worn on specific body areas. It should be

Table 3.
Data Requirements in the ML lifecycle of an APS.

Relevance	
DR.R1	Each data sample shall assume sensor positioning which is representative of that used on the patients
DR.R2	The format of each data sample shall be representative of that captured using sensors deployed on the body
DR.R3	The type of each data sample (insulin) shall be representative of that used
DR.R4	Each data sample shall represent the diabetes type for which the system is developed
DR.R5	Each data sample shall represent the sex, age, and ethnicity of the persons for which the system is developed
Completeness	
DR.C1	The data samples shall include examples with a sufficient range of meal carbs, different intraday meal intakes, and exercise
DR.C2	The data samples shall include examples with different sensor positioning
DR.C3	The data samples shall include examples with different ages and weights within the allowed ranges
DR.C4	The data samples shall include patients with frequent hypoglycemic, hyperglycemic, and ketoacidosis problems
DR.C5	The data samples shall include the profile of patients during the day and night and illness
Accuracy	
DR.A1	Each data sample shall assume sensor positioning which is representative of that used on the patients
DR.A2	CGM sensor readings and pump infusions must be correctly recorded
DR.A3	The total insulin delivered must be within the limit in each data sample
Balance	
DR.B1	The datasets shall have a comparable number of samples for features

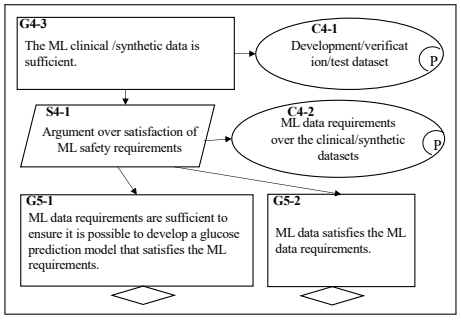


Figure 5.: Argument to ensure the sufficiency of the ML data.

placed around a fattier area of the body, i.e., the upper arm or abdomen for the adult and the abdomen or buttocks for kids. DR.R1 also relates to the accuracy desideratum (DR.A1), as the sensor position affects the accuracy of the sensor readings and, consequently, the accuracy of the BG prediction. The requirement DR.R3 refers to the type of insulin, i.e., rapid-acting, regular-acting, intermediate-acting, or long-acting, and even the brand of insulin. The APS controller may support a specific type of insulin, as different types have different absorption mechanisms. The APS designed for adults may not be allowed to be used for kids or vice versa. As DR.R4 explains, a similar argument can be expressed for other characteristics such as gender, insulin type, etc., and relates to the relevance desideratum. Sex, age, and insulin type may affect the satisfaction of physiological and robustness properties.

The APS controller should be able to safely adjust the insulin dosage in the face of uncertain events such as intraday meal intakes, exercise, and different values of meal carbohydrates. To support this, as explained by DR.C1, the datasets should include a sufficient range of examples in which the appropriate features refer to the mentioned events. If the system is supposed to work for different positions of CGM sensor installments, as explained by DR.C2, sufficient examples regarding each position should be presented in the datasets. A similar requirement can be specified for weight and age. For instance, consider that the system is designed to work for people aged 14 to 60 who weigh between 20 and 120 pounds. The datasets should not only include sufficient samples with all allowed ages and weights, but also include samples with the combination of these features (DR.C3). DR.C4 is specified to ensure that the data samples include patients with frequent hypoglycemic, hyperglycemic, and ketoacidosis events. As specified by DR.C5, the data samples shall include the profile of patients during the day and night and even in sickness. Nighttime sleep and sickness impact metabolic regulation and endocrine release by the pancreas. Considering all the requirements above is crucial to satisfying physiological properties and ensuring robustness.

From the accuracy perspective, the CGM readings and the pump infusions not affected by a system failure must be correctly recorded, and the total amount of insulin delivered for each person be within the limit, as explained by DR.A2 and DR.A3, respectively. The only data requirement regarding the balance desideratum is that the number of samples for features should be comparable (DR.B1). For instance, the number of samples representing kids and adults should be comparable if the system is supposed to work for both categories of kids and adults. Notably, the dataset should include data from patients of different ages, sexes, weights, etc., if the context is defined for a population of patients. Hence, context C4-2 is annotated with P. Similarly, the size of the development, test, and verification datasets change according to the data collected and the model learned, which should be reflected in C4-1. So, C4-1 is also annotated with P.

Table 4.
ML Performance requirements. We use BG_i, In_i, and M_i to denote BG, insulin, and meal intake, where $T^I = 12$ and $T^O = 6$. The superscript I indicates the input and O indicates the output of the network. We use Δ, β_1, β_2, β_3, β_4, β_5, ρ_1, ρ_2, α to denote the thresholds in requirements. $\Rightarrow$ denotes implication.

ML Performance Properties	
ML-RQ1.1	$\bigwedge_{i=0}^{T^I-2} \lvert BG_{i+1}^I - BG_i^I \rvert \leq \Delta \Rightarrow \bigwedge_{j=0}^{T^O-2} \lvert BG_{j+1}^O - BG_j^O \rvert \leq \Delta$
ML-RQ1.2	$\bigvee_{i=0}^{T^I-1} M_i^I \geq \beta_1 \Rightarrow \bigvee_{j=0}^{T^O-1} BG_j^O \geq \rho_1$
ML-RQ1.3	No available data
ML-RQ1.4	$In_0^I \geq \beta_2 \Rightarrow \bigvee_{j=1}^{T^O-1} \lvert BG_j^O - BG_0^O \rvert \geq \alpha$
ML-RQ1.5	$\bigvee_{i=0}^{T^I-1} M_i^I \geq \beta_3 \Rightarrow \bigvee_{j=1}^{T^O-1} \lvert BG_j^O - BG_0^O \rvert > 0$
ML-RQ1.6	$In_0^I \geq \beta_4 \Rightarrow 70 \leq BG_{T^O-1}^O \leq 180 \wedge \bigwedge_{j=0}^{j=T^O-2} (BG_j^O \leq 70 \vee BG_j^O \geq 180)$
ML-RQ1.7	$In_0^I \geq \beta_4 \Rightarrow 70 \leq BG_{T^O-1}^O \leq 180 \wedge \bigwedge_{j=0}^{j=T^O-2} (BG_j^O \leq 70 \vee BG_j^O \geq 180)$
ML-RQ1.8	$\bigvee_{i=0}^{T^I-1} In_i^I \geq \beta_5 \Rightarrow \bigvee_{j=0}^{T^O-1} BG_j^O \leq \rho_2$

5. Concrete Evidence

In this section, we provide concrete evidence in support of ML learning argument, and claims G4-1, G4-2, and G5-2. We used the Simglucose simulator [32, 33] to generate synthetic data for T1D patients and trained a Feed-Forward Neural Network (FFNN) to predict BG values. The model, contexts, and all properties in our experiments are based on a population of patients. We performed our experiments on Ubuntu 20.04 with Intel Core i7, CPU 3.60GHz ÃÛ 8, and 15.6 GiB memory.

ML Data (*Context C3-2*). Simglucose is a Python implementation of the FDA-approved UVA-Padova Simulator that employs a glucose-insulin meal model to simulate 30 virtual patients (ten adolescents, ten adults, and ten children). Using Simglucose, we emulated all patients for 40 days and nights, where the BG and insulin values are provided every 5 minutes. Simglucose implements a basic basal-bolus controller and generates random meals for each patient, where the amount and the time of each meal are random numbers from pre-specified intervals. Each patient's data includes 11,521 entries, and each entry includes a set of features from which we use only BG, insulin, and meal data. We removed data of 4 adolescents, 1 adult, and 5 children from the dataset, since their data included negative BG values.

ML Model (*Context C3-1*). Our FFNN has three dense layers with 8, 8, and 6 neurons in each layer, respectively. It has 36 inputs, including BG, insulin, and meal intake of the patient for an hour (12 timesteps with 5 min intervals) and predicts BG values 30 minutes into the future (6 timesteps). We scale the inputs between 0 and 1. More details on the model are available at [34].

Evidence for ML Learning Argument. This argument grounds on the sufficiency of the iterative process to design and train the model. This process selects the model structure and appropriate values for the model parameters. We used the same number of neurons proposed in [21, 19]. We tested the network with different neurons in each hidden layer and compared them using the root mean squared error (RMSE), which was very similar for those networks. We chose eight neurons in the first and second layers of the network, as network size affects verification complexity. To make sure that the model does not overfit on data, we plotted training loss versus validation loss. We observed that validation loss decreases over the increasing number of epochs but, like training loss, becomes nearly fixed after a few epochs.

Evidence for G4-1 and G4-2. We need to ensure that the ML model meets each ML performance and robustness requirement. We used test-based verification to check ML-RQ1. We split the data into training and test data with a proportion of 80% to 20%, respectively (context C4-1), and calculated RMSE. We consider ML-RQ1 is satisfied if RMSE is less than a threshold (i.e., 12 mg/dL [21]). The RMSE in our experiments is 3.03 mg/dL. We also used formal verification to check ML-RQ1.1 to ML-RQ1.8. We employed the DNNV framework [35], using which we compared the performance of different NN verifiers and selected Nnenum [36]. The properties are specified using inputs and outputs of the network by constraining their ranges of values. Table 4 shows the mapping between the performance requirements allocated to the ML component (Table 2) and the requirements amenable to the ML implementation. We describe ML-RQ1.1 and ML-RQ1.2 as an example. In ML-RQ1.1, the difference between two consecutive BG values in the input and output is limited by Δ. In ML-RQ1.2, we assume that if meal intake is larger than a value (β_1), BG will be greater than a value (ρ_1). We use an OR condition to indicate the timestep in which the meal is consumed is not relevant. The meal intake should be sufficiently large to assure us about its effect on the BG value.

To verify the properties, we first determined ranges of values based on the minimum and maximum values of the corresponding variables in the dataset. We also chose the thresholds based on our knowledge of the literature (e.g., we set Δ, the constraint for max glucose rise/drop over 5 min, to 40 based on [19]). As a result, all properties were violated. This confirms that learning

Table 5.
The properties checked on the FFNN of the glucose prediction. The third and forth columns indicate whether the property is satisfied over networks with 8,8,6 and 128,64,6 neurons. We use BG_i, In_i, and M_i to denote BG, insulin, and meal intake, where $i \in [1, 12]$, $\alpha = 0.006525$, and $\beta = [0, 1]$. The superscript I indicates the input and O indicates the output. * denotes that the property was satisfied using another verifier (Marabou) as Nnenum raised error, and † shows that the property was satisfied after 16 days (using Marabou). The verification time for other requirements was fast enough.

Property	Constraints	(8,8,6)	(128,64,8)
ML-RQ1.1	$BG_i^I \in [130, 180], In_i^I \in \beta, M_i^I \in \beta, \Delta = 20$	Satisfied	Nnenum Error*
ML-RQ1.1	$BG_i^I \in [109, 180], In_i^I \in \beta, M_i^I \in \beta, \Delta = 20$	Satisfied	Nnenum Error †
ML-RQ1.8 ($In_1^I = 5 \Rightarrow BG_6^O \leq 230$)	$BG_i^I \in [212, 230], In_{i\neq 1}^I = \alpha, M_i^I = 0$	Violated	Satisfied
ML-RQ1.8 ($In_{12}^I = 5 \Rightarrow BG_6^O \leq 230$)	$BG_i^I \in [211, 220], In_{i\neq 12}^I = \alpha, M_i^I = 0$	Satisfied	Satisfied
ML-RQ1.8 ($In_{12}^I = 5 \Rightarrow BG_6^O < 220$)	$BG_i^I \in [212, 222], In_{i\neq 12}^I = \alpha, M_i^I = 0$	Satisfied	Violated
ML-RQ1.2 ($M_{12}^I = 20 \Rightarrow BG_6^O > 210$)	$BG_i^I \in [180, 180], In_i^I = \alpha, M_{i\neq 12}^I = 0$	Violated	Satisfied
ML-RQ1.2 ($M_{12}^I = 20 \Rightarrow BG_6^O > 200$)	$BG_i^I \in [180, 180], In_i^I = \alpha, M_{i\neq 12}^I = 0$	Satisfied	Satisfied
ML-RQ1.2 ($M_{12}^I = 20 \Rightarrow BG_6^O > 200$)	$BG_i^I \in [180, 183], In_i^I = \alpha, M_{i\neq 12}^I = 0$	Violated	Satisfied

complex body physiology in the presence of uncertain meal intake is difficult, and having high precision does not necessarily show the algorithm's correctness. Selecting the thresholds also needs consulting with physicians and domain experts. So, we considered specific forms of the properties and selected the thresholds with try and test. We tried to increase the likelihood of property satisfaction by tightening ranges and thresholds. A part of our experiments are shown in Table 5. Besides, we checked the properties on a network with 128 and 64 neurons in layers one and two. We observed that a property satisfied on the first network is not necessarily satisfied on the other, and vice versa. These experiments confirm the need to instantiate AC according to the patient profiles or the population. Because the network structure as well as the thresholds and ranges of values may change based on data available for a patient or a population of patients.

The robustness can be checked by measuring RMSE, given data of a virtual patient as the test data. The data does not include the exercise information (ML-RQ3).

Evidence for G5-2. Since we use synthetic data, the requirements DR.R1 (DR.A1), DR.R3, and DR.C2 in Table 3 are not applied to our dataset. Synthetic data generation was conducted via the Dexcom sensor, and this can serve as evidence to support DR.R2. Simglucose is a simulator to generate data of virtual patients with T1D, so DR.R4 is met. If the controller is used for all diabetic patients, DR.R5 and DR.C3 are violated since the data is generated for three subject groups of patients, excluding the elderly group and patients weighing more than 118 kg. We are also uncertain about sex and ethnicity. Simglucose generates random intraday meal intake. Thus, DR.C1 is partially met because the data does not include exercise information. Over the whole data, 0.14% of the data samples are hyperglycemic, 0.11% are hypoglycemic, and 99.75% are in the glycemic range. Thus, DR.C4 is met, but the data balance, DR.B1, is violated. We are not certain about DR.C5. This requirement is met if the data generation model considers illness. Although the data is generated by the simulator, DR.A2 is satisfied because the simulator models both the sensor and pump. There are not equal numbers for three subject groups in the dataset, which is another reason for the DR.B1 violation.

6. Open Research Problems

Herein, based on our experience in developing safety assurance cases for learning-enabled MCPS, we outline several interesting open research problems.

- We observed that the network structure influences the satisfaction or violation of a property. Undoubtedly, the training data, using which the weights in the network are calculated, also has an impact. How we can trace the violation of a property back to its origin?
- The difficulty of learning the patient's physiology solely from the training data may explain why several physiologically-based properties are violated. How can we enforce the ML model to satisfy the properties while it develops over the data?
- RNN is a very commonly used network for time series data. However, we are unaware of any RNN verifiers that can assess a broad range of properties (not just robustness) and are not specialized for specific applications. Also, the current FFNN verifiers do not support properties with complex structures like ours. How can we develop an RNN verifier functional for various properties?

In addition, addressing the following questions can improve the assurance case.

- How to develop adaptive safety AC for online learning models, e.g., where the datasets and consequently the learned model change during the system operation?
- How to develop quantitative measures to evaluate the confidence in a dynamic assurance case, via aggregating the uncertainty introduced by different evidence (e.g., from model training, testing, and verification) and reasoning about the sufficiency for assurance?
- How to build automated tool support for the development and review of safety AC for ML-enabled MCPS?

7. Conclusion

In this paper, we presented a safety assurance case template for APS as a representative of learning-enabled MCPS. We focused on ensuring the safety and effectiveness of the ML-based APS controller. We first extracted the primary performance and robustness requirements allocated to the APS controller. Then we enumerated the requirements on the dataset and provided concrete evidence regarding ML and data requirements. In the future, we plan to continue this line of research and investigate the open problems listed in Section 6.

Acknowledgment

This work was supported in part by the National Science Foundation (NSF) grants CCF-1942836, CCF-2131511, and CNS-2146295 and by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development.

References

[1] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, K. Zhang, The practical implementation of artificial intelligence technologies in medicine, Nature Medicine 25 (2019) 30–36.

[2] R. Ashmore, R. Calinescu, C. Paterson, Assuring the machine learning lifecycle: Desiderata, methods, and challenges, ACM Comput. Surv. 54 (2021).

[3] R. Bloomfield, P. Bishop, Safety and assurance cases: Past, present and possible future – an adelard perspective, in: Making Systems Safer, 2010, pp. 51–67.

[4] E. Asaadi, E. Denney, J. Menzies, G. J. Pai, D. Petroff, Dynamic assurance cases: A pathway to trusted autonomy, Computer 53 (2020) 35–46.

[5] R. Clothier, E. Denney, G. J. Pai, Making a risk informed safety case for small unmanned aircraft system operations, in: 17th AIAA Aviation Technology, Integration, and Operations Conference, 2017, p. 3275.

[6] L. Feng, A. L. King, S. Chen, A. Ayoub, J. Park, N. Bezzo, O. Sokolsky, I. Lee, A Safety Argument Strategy for PCA Closed-Loop Systems: A Preliminary Proposal, in: 5th Workshop on Medical Cyber-Physical Systems, 2014.

[7] Infusion pumps total product life cycle, guidance for industry and FDA staff, 2014.

[8] ISO 26262: Road vehicles âĂŤ functional safety, 2018.

[9] ISO/PAS 21448: Road vehicles âĂŤ safety of the intended functionality, 2019.

[10] ANSI/UL 4600: Standard for safety for the evaluation of autonomous products, 2022.

[11] The content of Investigational Device Exemption (IDE) and Premarket Approval (PMA) Applications for Artificial Pancreas Device Systems, 2012.

[12] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, I. Habli, Guidance on the assurance of machine learning in autonomous systems (AMLAS), 2021.

[13] R. Kaur, R. Ivanov, M. Cleaveland, O. Sokolsky, I. Lee, Assurance case patterns for cyber-physical systems with deep neural networks, in: A. Casimiro, F. Ortmeier, E. Schoitsch, F. Bitsch, P. Ferreira (Eds.), SAFECOMP Workshops, 2020, pp. 82–97.

[14] S. Burton, I. Kurzidem, A. Schwaiger, P. Schleiss, M. Unterreiner, T. Graeber, P. Becker, Safety assurance of machine learning forÂăchassis control functions, in: Computer Safety, Reliability, and Security, 2021.

[15] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, I. Habli, Assuring the safety of machine learning for pedestrian detection at crossings, in: A. Casimiro, F. Ortmeier, F. Bitsch, P. Ferreira (Eds.), Computer Safety, Reliability, and Security, 2020, pp. 197–212.

[16] S. Burton, L. Gauerhof, B. B. Sethy, I. Habli, R. Hawkins, Confidence arguments for evidence of performance in machine learning for highly automated driving functions, in: Computer Safety, Reliability, and Security, 2019.

[17] C. Picardi, R. Hawkins, C. Paterson, I. Habli, A pattern for arguing the assurance of machine learning in medical diagnosis systems, in: A. Romanovsky, E. Troubitsyna, F. Bitsch (Eds.), Computer Safety, Reliability, and Security, 2019, pp. 165–179.

[18] S. Kapil, R. Saini, S. Wangnoo, S. Dhir, Artificial pancreas system for type 1 diabetesâĂŤchallenges and advancements, Exploratory Research and Hypothesis in Medicine 5 (2020).

[19] M. Narasimhamurthy, T. Kushner, S. Dutta, S. Sankaranarayanan, Verifying conformance of neural network models: Invited paper, in: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019.

[20] mobihealthnews, 2021. URL: https://www.mobihealthnews.com/news/roche-inks-deal-diabeloop-integrate-automated-insulin-delivery.

[21] S. Dutta, T. Kushner, S. Sankaranarayanan, Robust data-driven control of artificial pancreas systems using neural networks, in: M. Češka, D. Šafránek (Eds.), Computational Methods in Systems Biology, 2018, pp. 183–202.

[22] M. Zhang, K. B. Flores, H. T. Tran, Deep learning and regression approaches to forecasting blood glucose levels for type 1 diabetes, Biomedical Signal Processing and Control 69 (2021).

[23] S. Lee, J. Kim, S. W. Park, S.-M. Jin, S.-M. Park, Toward a fully automated artificial pancreas system

using a bioinspired reinforcement learning design: In silico validation, IEEE Journal of Biomedical and Health Informatics 25 (2021).
[24] FDA approval for Medtronic MiniMed, 2022. URL: https://www.accessdata.fda.gov/cdrh_docs/pdf16/P160017S076b.pdf.
[25] B. Kovatchev, P. Cheng, S. M. Anderson, J. E. Pinsker, F. Boscari, B. A. Buckingham, F. J. Doyle III, K. K. Hood, S. A. Brown, M. D. Breton, et al., Feasibility of long-term closed-loop control: a multicenter 6-month trial of 24/7 automated insulin delivery, Diabetes technology & therapeutics 19 (2017) 18–24.
[26] J. L. Sherr, B. A. Buckingham, G. P. Forlenza, A. Galderisi, L. Ekhlaspour, R. P. Wadwa, L. Carria, L. Hsu, C. Berget, T. A. Peyser, et al., Safety and performance of the omnipod hybrid closed-loop system in adults, adolescents, and children with type 1 diabetes over 5 days under free-living conditions, Diabetes technology & therapeutics 22 (2020) 174–184.
[27] N. S. Chen, C. K. Boughton, S. Hartnell, J. Fuchs, J. M. Allen, M. E. Willinska, A. Thankamony, C. de Beaufort, F. M. Campbell, E. Fröhlich-Reiterer, et al., User engagement with the CamAPS FX hybrid closed-loop app according to age and user characteristics, Diabetes care 44 (2021) e148–e150.
[28] T. D. Care, Basal-IQ, 2022. URL: https://www.tandemdiabetes.com/providers/products/basal-iq.
[29] Goal Structuring Notation Community Standard Version 2, 2018. URL: https://scsc.uk/r141B:1?t=1.
[30] 2022. URL: https://rtg.cis.upenn.edu/gip/.
[31] F. Cameron, G. Fainekos, D. M. Maahs, S. Sankaranarayanan, Towards a verified artificial pancreas: Challenges and solutions for runtime verification, in: E. Bartocci, R. Majumdar (Eds.), Runtime Verification, 2015, pp. 3–17.
[32] X. Zhou, M. Kouzel, H. Ren, H. Alemzadeh, Design and validation of an open-source closed-loop testbed for artificial pancreas systems, in: the IEEE/ACM international conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2022.
[33] J. Xie, Simglucose v0.2.1 (2018) [online], 2022. URL: https://github.com/jxx123/simglucose.
[34] 2022. URL: https://arxiv.org/abs/2211.15413.
[35] D. Shriver, S. Elbaum, M. B. Dwyer, Dnnv: A framework for deep neural network verification, in: Computer Aided Verification: 33rd International Conference, 2021, p. 137âĂŞ150.
[36] S. Bak, nnenum: Verification of relu neural networks with optimized abstraction refinement, in: NASA Formal Methods (NFM), 2021.

Table 6.
Root Mean Squared Error (RMSE) for different combinations of neurons in layers one and two

Neurons in layer 1 and 2	RMSE
(8,8)	3.03
(8,10)	3.09
(8,20)	3.11
(8,64)	3.13
(8,128)	3.27
(8,200)	3.19
(10,8)	3.11
(20,8)	3.14
(64,8)	3.19
(128,8)	3.19
(200,8)	3.15

A. Appendix

A.1. Glucose Prediction Model

we denote the structure of our FFNN along with its inputs and outputs in Figure 6. We use timesteps to indicate the order in which the values are organized in the input and output sequences. The t min in the input denotes t minutes into the past, and t min in the output indicates t minutes into the future. We use MinMaxScalar to scale the inputs before feeding them to the network. We use the relu activation function and the adam optimizer to compile the model.

We show the training loss versus validation loss for our FFNN of the glucose prediction in Figure 7(a). The test data are compared with the predicted data in Figure 7(b). Our FFNN includes 8 and 8 neurons in the first and second layers, respectively. We provide RMSE for different combinations of neurons in two layers in Table 6.

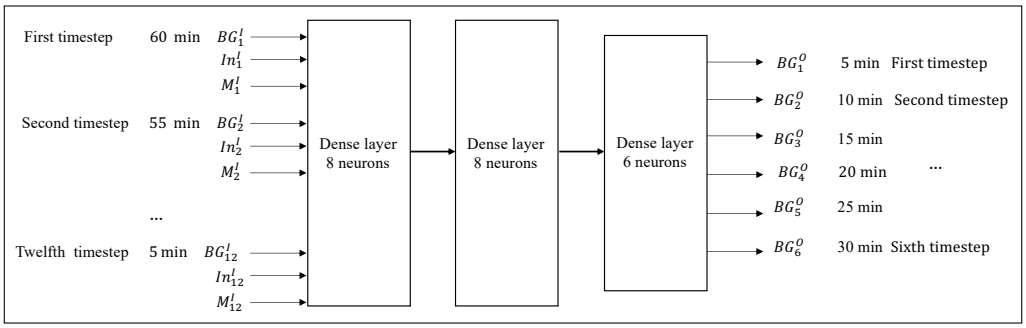
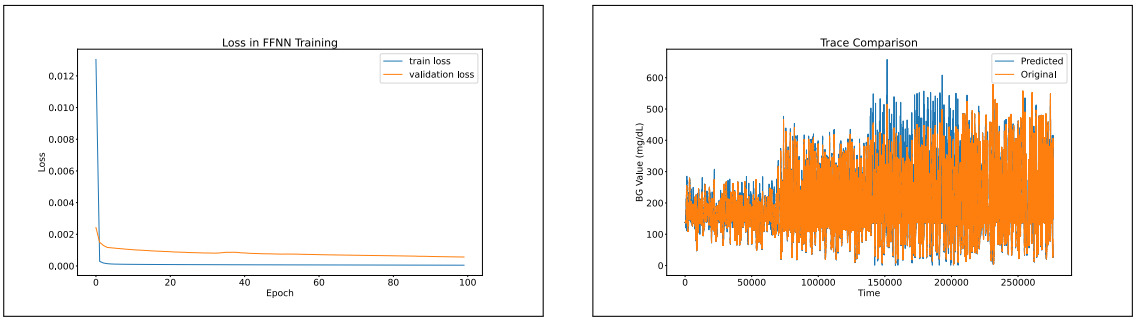


Figure 6.: The structure of our FFNN for glucose prediction with its inputs and outputs. The timesteps indicate the order in which the values are organized in the input and output sequences. The t min in the input denotes t minutes into the past, and t min in the output denotes t minutes into the future. In denotes the insulin and M denotes the meal.



(a) (b)

Figure 7.: (a) Training loss versus validation loss for the FFNN with 8, 8, and 6 neurons in the first, second, and third layer, respectively, (b) The original and predicted BG values