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

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

Electrical and Computer Engineering, University of Virginia

SafeAI 2023, Feb 13, Washington, D.C., US @AAAI-23
ML-enabled Medical CPS

Machine Learning (ML)

- In the estimation and prediction components
- To make data-driven decisions based on sensor or patient input and guide control actions
Safety of ML-enabled Medical CPS

Challenges

- ML component must handle
  - Intricacies of patient physiology
  - Uncertainties in the operational environment
  - Variability in patient profiles

- Mismatch between the training data and the real-world data
  - Erroneous, biased, or incomplete output predictions
Safety of ML-enabled Medical CPS

Challenges

- ML component must handle
  - Intricacies of patient physiology
  - Uncertainties in the operational environment
  - Variability in patient profiles

- Mismatch between the training data and the real-world data
  - Erroneous, biased, or incomplete output predictions
Assurance Cases (AC)

Structured arguments, supported by evidence, to justify claims

- Literature on using AC for safety assurance
  - ML lifecycle is not dealt with
  - No concrete learning-enabled use cases
Assurance Cases (AC)

Structured arguments, supported by evidence, to justify claims

- Literature on using AC for safety assurance
  - ML lifecycle is not dealt with
  - No concrete learning-enabled use cases

Our paper presents detailed AC to assure the safety and effectiveness of a general framework of Artificial Pancreas Systems (APS), suitable for all types of APS.
Artificial Pancreas Systems (APS)

Commercially available APS have the same structure and behavior.
Contributions

- Preliminary results on developing a safety assurance case template for ML controllers in MCPS
  - Including patient profiles in its element descriptions.

- Detailed safety assurance case for APS that is supported by a thorough analysis of its ML model

- Defining properties based on the body’s metabolism and checking them against the ML prediction component using formal verification
Safety Assurance Case Template

- Using **assume/guarantee** reasoning

- Instantiating based on **individual patient profiles or populations**
Safety Assurance Case Template

- Using **assume/guarantee** reasoning

- Instantiating based on **individual patient profiles or populations**

---

**C0-1** Component description

**G0**

{Learning-enabled controller c}, composed of ML, is safe and effective in treating the patient.

**C0-2**

System safety requirements allocated to {Learning-enabled controller c}

**S0-1**

Argument by Assume/Guarantee reasoning

**A0-1**

ML_abs as an abstraction of the ML component is safe

**G1-1**

{Controller c}, composed with ML_abs, is sufficiently safe and effective in treating the patient

**G1-2**

The ML (prediction/perception) component is sufficiently safe and effective

---

**Context**
Safety Assurance Case Template

- Using **assume/guarantee** reasoning

- Instantiating based on **individual patient** profiles or populations
Learning-enabled Controller Assurance Case for APS

Assuming that the BG predictions are accurate, the insulin dosage management component is sufficiently safe and effective for treating patients. The ML glucose prediction component is sufficiently safe and effective.
Learning-enabled Controller Assurance Case for APS

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

The ML glucose prediction component is sufficiently safe and effective.
Safety Assurance Case for the Glucose Prediction Component

**C1-1** Description of the ML glucose prediction component

**G1-2** The ML glucose prediction component is sufficiently safe and effective

**S1-1** Argument over the development and deployment of the component

**C2-1** ML safety requirements, developed from system safety requirements, allocated to the ML glucose prediction model

**G2-1** The development of the ML glucose prediction model is sufficiently safe and effective

**G2-2** The deployment of the ML glucose prediction component into APS is sufficiently safe and effective

**S2-1** Argument over the ML safety and effectiveness requirements

**C2-2** System safety requirements allocated to ML glucose prediction component

**C3-1** ML glucose prediction model satisfies the ML requirements

**C3-2** ML requirements are a valid development of the APS requirements allocated to the glucose prediction component
Safety Assurance Case for the Glucose Prediction Component

C1-1
Description of the ML glucose prediction component

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

C1-2
System safety requirements allocated to ML glucose prediction component

S1-1
Argument over the development and deployment of the component

G2-1
The development of the ML glucose prediction model is sufficiently safe and effective

G2-2
The deployment of the ML glucose prediction component into APS is sufficiently safe and effective

C2-1
ML safety requirements, developed from system safety requirements, allocated to the ML glucose prediction model

S2-1
Argument over the ML safety and effectiveness requirements

G3-1
ML glucose prediction model satisfies the ML requirements

G3-2
ML requirements are a valid development of the APS requirements allocated to the glucose prediction component
Safety Assurance Case for the Glucose Prediction Component

C1-1 Description of the ML glucose prediction component

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

S1-1 Argument over the development and deployment of the component

G2-1 The development of the ML glucose prediction model is sufficiently safe and effective

G2-2 The deployment of the ML glucose prediction component into APS is sufficiently safe and effective

S2-1 Argument over the ML safety and effectiveness requirements

C2-1 ML safety requirements, developed from system safety requirements, allocated to the ML glucose prediction model

G3-1 ML glucose prediction model satisfies the ML requirements

G3-2 ML requirements are a valid development of the APS requirements allocated to the glucose prediction component

C1-2 System safety requirements allocated to ML glucose prediction component
Safety Assurance Case for the Glucose Prediction Component

- **C1-1** Description of the ML glucose prediction component
  - **G1-2** The ML glucose prediction component is sufficiently safe and effective
  - **C1-2** System safety requirements allocated to ML glucose prediction component

- **S1-1** Argument over the development and deployment of the component
  - **G2-1** The development of the ML glucose prediction model is sufficiently safe and effective
  - **G2-2** The deployment of the ML glucose prediction component into APS is sufficiently safe and effective

- **C2-1** ML safety requirements, developed from system safety requirements, allocated to the ML glucose prediction model
  - **S2-1** Argument over the ML safety and effectiveness requirements
  - **G3-1** ML glucose prediction model satisfies the ML requirements
  - **G3-2** ML requirements are a valid development of the APS requirements allocated to the glucose prediction component
## System Safety Requirements for ML Glucose Prediction Component

### Performance

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-RQ1</td>
<td>Accurately predict the BG values $T$ minutes in the future</td>
</tr>
<tr>
<td>ML-RQ1.1</td>
<td>BG’s rate of change has to be bound by established physiological norms</td>
</tr>
<tr>
<td>ML-RQ1.2</td>
<td>Meal intake has a direct effect on the BG value</td>
</tr>
<tr>
<td>ML-RQ1.3</td>
<td>Exercise has an inverse effect on the BG value</td>
</tr>
<tr>
<td>ML-RQ1.4</td>
<td>Within $t$ minutes of a bolus, there should be an accompanying change in BG of more than $\alpha$</td>
</tr>
<tr>
<td>ML-RQ1.5</td>
<td>The glucose level starts to rise at a specific time after a meal’s onset</td>
</tr>
<tr>
<td>ML-RQ1.6</td>
<td>There is a delay between the injection of insulin and the disposal of glucose</td>
</tr>
<tr>
<td>ML-RQ1.7</td>
<td>The blood concentration of insulin reaches its maximum after a particular time</td>
</tr>
<tr>
<td>ML-RQ1.8</td>
<td>Insulin has an inverse effect on the BG value</td>
</tr>
</tbody>
</table>

### Robustness

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-RQ2</td>
<td>Perform as required for different patients of different ages/sexes</td>
</tr>
<tr>
<td>ML-RQ3</td>
<td>Perform as required in the presence of external factors such as meals and exercises.</td>
</tr>
</tbody>
</table>
Argument to Ensure the Sufficiency of the ML Model

C3-1
ML glucose prediction model

ML learning argument

G3-1
ML glucose prediction model satisfies the ML requirements

ML data argument

C3-2
ML clinical / synthetic data

S3-1
Argument over satisfaction of ML safety requirements

G4-1
ML performance requirements are satisfied.

G4-2
ML Robustness requirements are satisfied.
Argument to Ensure the Sufficiency of the ML Model

- C3-1: ML glucose prediction model
- ML learning argument
- G3-1: ML glucose prediction model satisfies the ML requirements
- S3-1: Argument over satisfaction of ML safety requirements
- G4-1: ML performance requirements are satisfied.
- G4-2: ML Robustness requirements are satisfied.
- C3-2: ML clinical/synthetic data
- ML data argument
Argument to Ensure the Sufficiency of the ML Data

G4-3
The ML clinical/synthetic data is sufficient.

S4-1
Argument over satisfaction of ML safety requirements

G5-1
ML data requirements are sufficient to ensure it is possible to develop a glucose prediction model that satisfies the ML requirements.

C4-1
Development/verification/test dataset

C4-2
ML data requirements over the clinical/synthetic datasets

G5-2
ML data satisfies the ML data requirements.
## Data Requirements in the ML Lifecycle of APS

### Relevance

<table>
<thead>
<tr>
<th>DR.R1</th>
<th>Each data sample shall assume sensor positioning which is representative of that used on the patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR.R2</td>
<td>The format of each data sample shall be representative of that captured using sensors deployed on the body</td>
</tr>
<tr>
<td>DR.R3</td>
<td>The type of each data sample (insulin) shall be representative of that used</td>
</tr>
<tr>
<td>DR.R4</td>
<td>Each data sample shall represent the diabetes type for which the system is developed</td>
</tr>
<tr>
<td>DR.R5</td>
<td>Each data sample shall represent the sex, age, and ethnicity of the persons for which the system is developed</td>
</tr>
</tbody>
</table>

### Completeness

<table>
<thead>
<tr>
<th>DR.C1</th>
<th>The data samples shall include examples with a sufficient range of meal carbs, different intraday meal intakes, and exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR.C2</td>
<td>The data samples shall include examples with different sensor positioning</td>
</tr>
<tr>
<td>DR.C3</td>
<td>The data samples shall include examples with different ages and weights within the allowed ranges</td>
</tr>
<tr>
<td>DR.C4</td>
<td>The data samples shall include patients with frequent hypoglycemic, hyperglycemic, and ketoacidosis problems</td>
</tr>
<tr>
<td>DR.C5</td>
<td>The data samples shall include the profile of patients during the day and night and illness</td>
</tr>
</tbody>
</table>
**Data Requirements in the ML Lifecycle of APS**

### Accuracy

<table>
<thead>
<tr>
<th>DR.A1</th>
<th>Each data sample shall assume sensor positioning which is representative of that used on the patients</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR.A2</td>
<td>CGM sensor readings and pump infusions must be correctly recorded</td>
</tr>
<tr>
<td>DR.A3</td>
<td>The total insulin delivered must be within the limit in each data sample</td>
</tr>
</tbody>
</table>

### Balance

| DR.B1 | The datasets shall have a comparable number of samples for features                           |
ML Safety Requirements

- Feed Forward Neural Network for the BG prediction
- Defining requirements as the **constraints over the inputs and outputs** of the network

<table>
<thead>
<tr>
<th>ML Performance Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ML-RQ1.1</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.2</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.3</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.4</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.5</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.6</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.7</strong></td>
</tr>
<tr>
<td><strong>ML-RQ1.8</strong></td>
</tr>
</tbody>
</table>
# ML Safety Requirements

- Checking the properties using DNNV framework and **Nnenum ML verifier**

<table>
<thead>
<tr>
<th>Property</th>
<th>Constraints</th>
<th>Constraints Range</th>
<th>Result 1</th>
<th>Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-RQ1.1</td>
<td>$BG_i^l \in [130, 180], In_i^l \in \beta, M_i^l \in \beta, \Delta = 20$</td>
<td>(8,8,6)</td>
<td>Satisfied</td>
<td>Nnenum Error*</td>
</tr>
<tr>
<td>ML-RQ1.1</td>
<td>$BG_i^l \in [109, 180], In_i^l \in \beta, M_i^l \in \beta, \Delta = 20$</td>
<td>(128,64,8)</td>
<td>Satisfied</td>
<td>Nnenum Error</td>
</tr>
<tr>
<td>ML-RQ1.8 ($In_i^l = 5 \Rightarrow BG_6^O \leq 230$)</td>
<td>$BG_i^l \in [212, 230], In_i_{i\neq 1} = \alpha, M_i^l = 0$</td>
<td>Violated</td>
<td>Satisfied</td>
<td></td>
</tr>
<tr>
<td>ML-RQ1.8 ($In_i^l = 5 \Rightarrow BG_6^O \leq 230$)</td>
<td>$BG_i^l \in [211, 220], In_i_{i\neq 12} = \alpha, M_i^l = 0$</td>
<td>Satisfied</td>
<td>Satisfied</td>
<td></td>
</tr>
<tr>
<td>ML-RQ1.8 ($In_i^l = 5 \Rightarrow BG_6^O &lt; 220$)</td>
<td>$BG_i^l \in [212, 222], In_i_{i\neq 12} = \alpha, M_i^l = 0$</td>
<td>Satisfied</td>
<td>Violated</td>
<td></td>
</tr>
<tr>
<td>ML-RQ1.2 ($M_{i2} = 20 \Rightarrow BG_6^O &gt; 210$)</td>
<td>$BG_i^l \in [180, 180], In_i^l = \alpha, M_i^l_{i\neq 12} = 0$</td>
<td>Violated</td>
<td>Satisfied</td>
<td></td>
</tr>
<tr>
<td>ML-RQ1.2 ($M_{i2} = 20 \Rightarrow BG_6^O &gt; 200$)</td>
<td>$BG_i^l \in [180, 183], In_i^l = \alpha, M_i^l_{i\neq 12} = 0$</td>
<td>Satisfied</td>
<td>Satisfied</td>
<td></td>
</tr>
<tr>
<td>ML-RQ1.2 ($M_{i2} = 20 \Rightarrow BG_6^O &gt; 200$)</td>
<td>$BG_i^l \in [180, 183], In_i^l = \alpha, M_i^l_{i\neq 12} = 0$</td>
<td>Violated</td>
<td>Satisfied</td>
<td></td>
</tr>
</tbody>
</table>
Open Research Problems

- How can we trace the violation of a property back to its origin?
- How can we enforce the ML model to satisfy the properties while it develops over the data?
- How to develop adaptive safety AC for online learning models?
- How to develop quantitative measures to evaluate the confidence in a dynamic assurance case?
- How to build automated tool support for the development and review of safety AC for ML-enabled MCPS?
Thank you!
Dataset

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)
- All patients for 40 days and nights, where the BG and insulin values are provided every 5 minutes
- Each patient’s data includes 11,521 entries
First timestep 60 min $BG_1^I$, $In_1^I$, $M_1^I$

Second timestep 55 min $BG_2^I$, $In_2^I$, $M_2^I$

Twelfth timestep 5 min $BG_{12}^I$, $In_{12}^I$, $M_{12}^I$

Dense layer 8 neurons

Dense layer 8 neurons

Dense layer 6 neurons

$BG_1^O$ 5 min First timestep

$BG_2^O$ 10 min Second timestep

$BG_3^O$ 15 min

$BG_4^O$ 20 min

$BG_5^O$ 25 min

$BG_6^O$ 30 min Sixth timestep
# Requirements for Learning-enabled APS Controller

<table>
<thead>
<tr>
<th>RQ.C.1</th>
<th>Accurately calculate dose of basal and bolus insulin</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ.C.1.1</td>
<td>Determine the output every T minutes (e.g., T=5 in MiniMed)</td>
</tr>
<tr>
<td>RQ.C.1.2</td>
<td>Stop dosing if a maximum amount has been delivered by the pump</td>
</tr>
<tr>
<td>RQ.C.1.3</td>
<td>Suspend dosing if the actual or predicted CGM readings fall below a threshold</td>
</tr>
<tr>
<td>RQ.C.1.4</td>
<td>Interrupt in a safe way if trustworthy control is not guaranteed</td>
</tr>
<tr>
<td>RQ.C.1.5</td>
<td>BG should not remain below 10th-percentile threshold for more than $\alpha_1$ minutes</td>
</tr>
<tr>
<td>RQ.C.1.6</td>
<td>BG should not remain above 90th-percentile threshold for more than $\alpha_2$ minutes following a bolus injection</td>
</tr>
<tr>
<td>RQ.C.1.7</td>
<td>BG should not remain above 90th-percentile threshold for more than $\alpha_3$ minutes</td>
</tr>
<tr>
<td>RQ.C.1.8</td>
<td>The BG value is always greater than 70 and less than 180</td>
</tr>
<tr>
<td>RQ.C.1.9</td>
<td>The controller infuses additional insulin while the blood glucose level is below a target level</td>
</tr>
<tr>
<td>RQ.C.1.10</td>
<td>The morning wake up blood glucose level can not exceed $\beta$</td>
</tr>
</tbody>
</table>
The properties checked on the FFNN of the glucose prediction. The third and fourth 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.

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