Toward Operational Safety Verification Via Hybrid Automata Mining Using I/O Traces of AI-Enabled CPS

Imane Lamrani, Ayan Banerjee, and Sandeep K.S. Gupta
IMPACT lab, CISDE,
Arizona State University,
Tempe, AZ
{ilamrani, abanerj3, sandeep.gupta}@asu.edu

Abstract

AI enabled cyber-physical systems such as artificial pancreas suffer from the "no oracle problem". The system is subjected to inputs and scenarios which are not observed during training time and hence the expected outputs are not known. Hence, popular model-based verification techniques that characterize behavior of a control system before deployment using predictive models may be inaccurate and may result in incorrect safety analysis results. In this research, we propose an operational safety verification technique through hybrid system mining from input/output traces of deployed AI-enabled cyber-physical systems. The hybrid automaton model enables formal verification of safety despite the "no oracle problem". We apply our technique to the artificial pancreas control system utilizing data from an outpatient study on an artificial pancreas system. We demonstrate that our technique successfully infers accurate hybrid automata representation of these systems in the field and can be used to perform safety analysis to ascertain safety of the system in presence of inputs and scenarios for which the expected output of the system is unknown. We identify an evaluation scenario under which there exists a clear safety violation.

1 Introduction

The increasing use of artificial intelligence (AI) and machine learning (ML) in safety-critical cyber-physical systems (CPS) and their recent cases of fatal failures have renewed the discussion on the certification problem and has brought with it a pressing need for developing rigorous safety verification techniques. However, operational components interaction circumstances, inclusion of human-in-the-loop, and environmental changes in AI-enabled CPS make formal safety verification a very challenging task. Traditional approaches of safety verification involve testing and simulation and are no longer sufficient to assess safety in the case of AI-enabled CPS, wherein exhaustive safety verification is necessary. In contrast, formal methods such as model checking were developed to overcome the limitations of traditional safety verification techniques. However, these methods may fall short for AI-Enabled CPS, where complete formal verification models are often unavailable. As a result, the AI-enabled CPS operation in the real world tends to diverge from the safety assured design of the system. AI-enabled CPSs such as artificial pancreas (AP) or autonomous cars are using machine learning to make several critical decisions. As an example, let us consider the glucose predictive system of the Medtronic 670G AP (closed loop blood glucose control system). The system uses the predictive model to predict low blood glucose levels. If the blood glucose level is predicted to be low in the future, the infusion pump shuts off. The purpose is to avoid impending hypoglycemia which can be a fatal consequence. However, if the prediction is wrong, it can lead to hyperglycemia, which has long term negative consequences on the body. The model prediction is dependent on the physiological parameters of the human user such as insulin sensitivity. These parameters are dependent on the human behavior such as physical activity, meal patterns, and mental states. Such parameter variations cannot be replicated while designing the control systems. In this paper, we propose a novel approach presented in Figure 1 to solve the given problem of model-based safety verification of AI enabled cyber-physical control systems with limited oracle. Our approach initially considers a hybrid system representation of the control system that describes the expected operation for which the system was tested, validated, and verified using controlled experimental studies. We then describe a methodology to mine a hybrid system representation of the AI-enabled control system from input/output (I/O) timeseries data. If the mined hybrid system is same as the initial hybrid model defined in the documentation provided by the manufacturer, then there is no change in the safety conclusion. However, if the mined hybrid system differs from the initially expressed one (w.r.t to the number of modes, flow dynamics, modes transitions, guard conditions, or reset conditions), then there might be a significant change in the safety conclusions. In such cases, we consider the reachability analysis of the newly mined hybrid system to evaluate the safety of the control system (Alur et al. 1995). If the newly mined hybrid system is unsafe, then potentially a root cause analysis algorithm can be invoked. We do not discuss this in our paper, but is surely a future endeavor. In addition, safety critical CPS should meet government regulatory requirements before marketing. Due to production pressure and conflicting goals and tradeoffs, or-

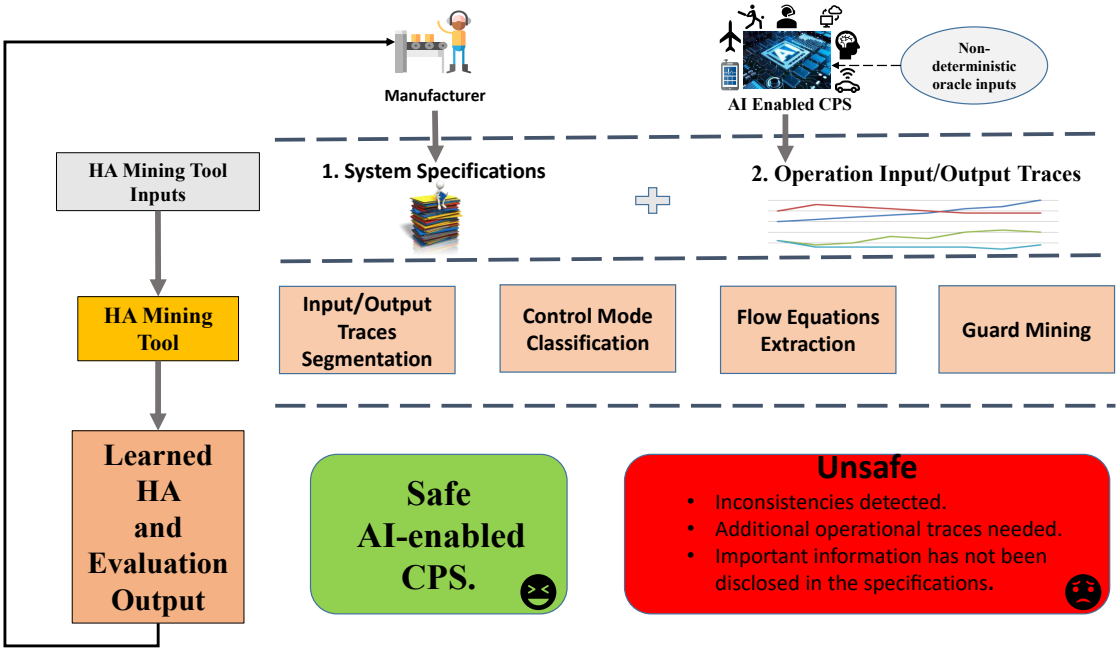

Figure 1: Overall scheme of the proposed safety verification technique.

ganizations tend to migrate to a state of heightened risk by relaxing safeguards and controls (Leveson 2011). For example, the Volkswagen's defeat device that allowed vehicles to improperly meet US standards during regulatory testing (Contag et al. 2017). This dissonance between *"what the system is designed to do"*, *"what the operator thinks the system is doing"*, and *"what the system is actually doing"* is an important problem (McDermid, Jia, and Habli 2019). It was reported in the final report No. KNKT.18.10.35.04 to be one of the compounding factors leading to Boeing Max 8 fatal crashes [1]. This problem becomes even more important and highly challenging for Industry 4.0 systems with AI-enabled or self-adaptive components, where the system is flexible, fully autonomous, and adapts its model as it encounters new situations. This motivates a need for rigorous operational safety verification techniques to help monitor and maintain the system safe operation in the real world. This paper addresses this issue by combining information theory, formal methods, and ML to obtain a proactive operational safety verification technique that detects intentional or unintentional deviations from the safety assured design of the AI-enabled CPS once deployed to the field. The paper is organized as follows: Section 2 discusses competing works towards solving the discussed problem, Section 3 discusses definitions and preliminaries related to AI-enabled system model, Section 4 provides details about the proposed technique, Section 5 evaluates the effectiveness of the proposed technique for extracting hybrid automata, and finally Section 6 to conclude the paper.

[1]https://www.flightradar24.com/blog/wp-content/uploads/2019/10/JT610-PK-LQP-Final-Report.pdf

2 Related Work

2.1 Engineering Safety-Critical Systems

Verifying the safety and correct operation of AI-enabled CPS relies on verifying the correct interaction between the software and the physical environment (Leveson 2011). Leveson uses system-theoretic processes to identify safety constraints which help in designing or re-designing safer systems. These techniques are usually applied in the design phase of complex safety-critical systems. They may also be used in accidents root-cause analysis. Formal design and verification of safety-critical CPS with artificial intelligence (AI) and machine learning (ML) components is becoming an important topic in the field of AI/ML-based systems (Dreossi et al. 2019; Zhu et al. 2019). Dreossi et al. propose a toolkit VERIFAI that focuses on simulation-based safety analysis of AI-based systems where a simulatable abstract model of the system is used. In this work, we want to ascertain that the safety verification results learned using traces collected from real-world operation of the AI-enabled CPS are consistent with formal or simulated safety verification results. On the other hand, reachability analysis is a formal safety verification technique that has been extensively studied in the literature for time-invariant systems (Alur et al. 1995; Frehse et al. 2011; Fan et al. 2016). It determines the set of states that the system may visit when starting from a bounded set of initial conditions. If no unsafe state is reachable, the system can be deemed to be safe. In this work, we propose a safety verification scheme that aims at identifying deviations of safety verified AI-enabled CPS in the field (during the operational phase) in a proactive manner. The proposed scheme is based on hybrid automaton

mining using real-time data collected from the operation of an AI-enabled CPS in the field. We apply reachability analysis over a learned hybrid automaton to verify the safety of the operational AI-enabled CPS.

2.2 Mining Hybrid Automata

Several previous work have proposed algorithms and frameworks for learning hybrid automata. Minopoli and Frehse present a tool for translating a simulink model to a hybrid automaton (Minopoli and Frehse 2016). Lyde and Might propose an approach for analyzing control code using abstract interpretation and inferring a hybrid automaton from an abstract state transition system (Lyde and Might 2013). However, our work differs in that we learn hybrid automata models automatically using system operational I/O traces.

Mining HA from I/O traces: Medhat et al. proposed a framework for mining mealy automata from black-box systems using only execution traces. This framework is limited to systems that exhibit input changes in the form of step functions and these changes are assumed to have an instantaneous effect in the output trace, which is not often observed in practice (Medhat et al. 2015). In addition, the authors only consider guard conditions as time-based transitions and thus guards on output values cannot be modelled using their proposed framework. Balakrishnan et.al presented an algorithm to determine a maximum-likelihood hybrid system model using only continuous output of the system (Balakrishnan et al. 2004), but this work assumes that guard conditions are independent of the continuous state which limits the class of hybrid automata that can be learned using the proposed technique. Blackmore et. al extended this work by including autonomous mode transitions which are conditioned on the continuous state, but their approach assumes that the guard conditions are given (Blackmore et al. 2007). Our proposed hybrid automata mining technique derives the guard conditions through clustering of the continuous states. Ly and Lipson presented an approach that uses clustered symbolic regressions and a machine learning algorithm to infer non-linear symbolic expressions that model the behavior of a dynamical system from unlabeled time-series data (Ly and Lipson 2012). The authors also propose a transition modeling algorithm that searches for non-linear symbolic inequalities to model guard conditions. Unlike our proposed technique, their work assumes that the guard condition is strictly related to a change in the inputs of the system and that the system can not have two distinct modes with similar behavior. Moreover, the behavior of the system is defined as a strict I/O relationship, as opposed to our hybrid mining technique where behaviors are represented by differential equations. In addition, some of the related approaches require a priori knowledge of number of discrete modes (Santana et al. 2015; Ly and Lipson 2012), as opposed to our technique. Niggemann et. al share same motivation for the automated leaning of hybrid system's behavioral model and application of the learned model to detect anomalies in the overall system behavior (Niggemann et al. 2014; Niggemann and Lohweg 2015). On the other hand, HyBUTLA (Niggemann et al. 2012) infers hybrid timed probabilistic automata while our technique relaxes this timing constraint which allows it to infer hybrid automata models for a larger class of hybrid systems. CHARDA is the closest work to ours and was applied to learn hybrid behaviors of videogame characters (Summerville, Osborn, and Mateas 2017). However, our proposed technique and CHARDA differ in the fact that CHARDA is limited to systems where the derivatives of the continuous state variables are constant, which is often not observed in practice. In this work, we adapt the hybrid automata mining technique HyMn (Lamrani, Banerjee, and Gupta 2018) with a new flow extraction technique where multi-variable non linear polynomial regression analysis are employed to derive non-linear dynamics evolution. In addition, we updated the change-point detection technique used in HyMn algorithm by the RuLSIF technique discussed in Section 4.1. The output of this re-classification are unique modes of the AI enabled CPS.

2.3 Conformance Testing

Our proposed approach shares same motivation as the verification of the conformance between a running CPS and the formal specifications of its required behavior, which is referred to as *conformance testing* (Woehrle, Lampka, and Thiele 2012; Abbas 2015). Woehrle et. al presented a conformance testing method that relies on mapping the specifications of the system and its implementation generated traces to timed automata and verifying whether each generated implementation trace is included in the traces of the specifications timed automaton. However, their approach is solely limited to the class of timed automata. As opposed to this conformance notion, other works define conformance testing as a closeness measure between an implementation and the specifications model, whose computation solely relies on system traces(Abbas 2015; Araujo et al. 2018). However, even for simple linear systems, providing guarantees about the conformance degree remains a challenge. Finally, to the best of our knowledge, conformance testing methods output are limited to a pass or fail. Hence, in case of conformance failure only a debugging trace is provided to the test engineer, which helps in debugging implementation errors of a CPS. For complex systems, locating the root cause without the presence of an operational model may require extensive work.

Novelty: The HA mining algorithm presented in this paper differs from related work in the following key factors:

- It can extract control modes where the controller output is a linear combination of the continuous state variables,
- The continuous state variables follow a set of non-linear differential equations.

3 Definitions and preliminaries

3.1 System Model

An AI-enabled CPS is a system comprising a perception component, a planner/controller, and the environment

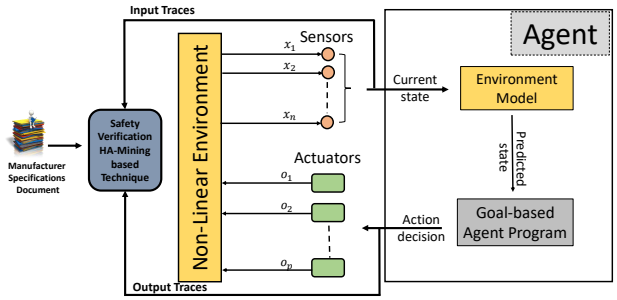

Figure 2: Non-Linear AI Enabled Cyber-Physical System.

(system under control) (Russell and Norvig 2016). As shown in Figure 2, an AI-enabled CPS interacts with the environment using a set of sensors and actuators. The environment can be expressed using a set of n continuous variables $\{x_1, x_2, \ldots, x_n\}$. The continuous variables are governed by a set of non-linear differential equations which are also modulated by p control outputs $\{o_1, o_2, \ldots, o_p\}$. The continuous variables states are provided as inputs to the agent that performs actions on the physical environment so as to achieve a given or computed goal using the current and predicted state of the environment. The current state is determined by sensors' information and predicted state of the environment is determined using the environment predictive model.

3.2 Operational Safety

Operational safety is ensuring that the operation of the AI-enabled CPS in the field does not deviates from the safe certified design of the system. It aims at detecting dissonance between the system's behavior out in the field and the safe certified behavior of the system. This dissonance between what the system is actually doing and what the operator thinks the system is doing can be due to intentional corruption scenarios (due to production pressure or goals trade-off) or to unintentional corruptions scenarios (flaws in requirements, specifications, design, or implementation of the system during its development lifecycle). For example, the Volkswagen's cheating defeat device that allowed vehicles to improperly meet US standards during emission regulatory testing (Contag et al. 2017) and the case of the Boeing 737 Max 8 aircraft crash where operators lack crucial information about the MCAS system.This dissonance could expose the system to potential dangerous situations.

3.3 Example of AI-enabled CPS: Artificial Pancreas (AP)

The AP control system is an example of an AI-enabled CPS used for automated control of blood glucose level for Type1 diabetic patients (Clarke et al. 2009). The agent uses predicted glucose level 30 minutes ahead in time and outputs the right amount of insulin infusion rate I_t for the infusion pump to maintain for the next 30 minutes. The agent's goal is to maintain the prescribed level of blood glucose and avoid occurrence of hypoglycemic/hyperglycemic events. These dangerous events happen as a result of an inaccurate infusion of insulin, e.g. if the glucose concentration G goes above $180mg/dl$, it can lead to hyperglycemia while low glucose level i.e. below $60mg/dl$ can cause hypoglycemia. The dynamics of the AP are represented by nonlinear equations 1, 2 and 3, where $\dot{X}$ represents the rate of the variation in the interstitial insulin concentration, $\dot{G}$ is the rate of change of blood glucose concentration (G) for the infused insulin concentration X and $\dot{I}$ is the variation in plasma insulin concentration (I) (Andersen and Højbjerre 2002). The AP device has three control modes:

1- basal, where the reset condition $I_t = 5$,
2- braking, where $I_t = 0.5G + 44.75$, and
3-correction bolus, where $I_t = 50$.

The differential equation expressing the blood glucose and insulin interaction are non-linear in nature.

$$\dot{X} = -k_2.X(t) + k_3.(I(t) - I_b), \quad (1)$$

$$\dot{G} = -X(t).G(t) + k_1.(G_b - G(t)), \quad (2)$$

$$\dot{I} = -k_4.I(t) + k_5.(G(t) - k_6)^+.t. \quad (3)$$

3.4 Non-Linear Hybrid Automata

AI-enabled systems are dynamical systems comprising discrete transition systems (intelligent agents) interacting with continuous dynamical systems (non-linear physical environments). A non-linear hybrid automaton is a model of closed-loop system combining discrete evolution (control mode transitions and variable updates) and continuous evolution (system dynamics that are governed by differential equations). We formally define a non-linear hybrid automaton formal model of AI-enabled CPS as follows.

Hybrid Automata: A hybrid automaton H is a tuple $< \mathcal{X}, \mathcal{M}, \mathcal{E}, \mathcal{G}, \mathcal{R}, \mathcal{F} >$ where:

- $\mathcal{X} = \{x_1 \ldots x_m\}$ is a finite set of continuous variables where $\mathcal{X} = \mathcal{I} \bigcup \mathcal{O}$. $\mathcal{I}$ is a set of internal input variables and $\mathcal{O}$ is a set of output (controlled) variables. A valuation over the set $\mathcal{X}$ of variables is a member of $\mathbb{R}^m$ such that each variable $x_i \in \mathcal{X}$ receives a real value. $\dot{\mathcal{X}} = \{\dot{x_1}, \ldots, \dot{x_m}\}$ is the set of dotted continuous variables representing the first time derivatives of the continuous variables during continuous change. $\mathcal{X}' = \{x'_1, \ldots, x'_m\}$ is the set of primed continuous variables, which represents the values of the variables at the conclusion of a control mode transition.
- $\mathcal{M} = \{m_1 \ldots m_n\}$ is a finite set of control modes.
- A set $\mathcal{F}_{m_i}$ of non-linear ordinary differential equations over $\mathcal{X} \bigcup \dot{\mathcal{X}}$ representing the dynamics evolution (flow rate) of each variable x_i for each the control mode $m_i \in \mathcal{M}$.
- $\mathcal{E}$ is a finite set of edges called *mode transitions* or *mode changes*. Every edge $e_i \in \mathcal{E}$ is defined by a conjunction of guard condition $\mathcal{G}_{m_i,m_j}$ and a reset condition $\mathcal{R}_{m_i,m_j}$, where $m_i \in \mathcal{M}$ is the source mode and $m_j \in \mathcal{M}$ is the target mode.
- The guard condition $\mathcal{G}_{m_i,m_j}$ is a linear polyhedral constraint over the variables in $\dot{\mathcal{X}}$. The transition between m_i

and m_j following the edge e_i is enabled when values of the continuous set of variables $\mathcal{X} \in \mathcal{G}_{m_i,m_j}$. $\mathcal{G}$ is the set of all guard conditions.

- The reset condition $\mathcal{R}_{m_i,m_j}$, given by a linear assignment over the variables in $\mathcal{X} \bigcup \dot{\mathcal{X}}'$, associates a variable assignment to the mode transition m_i to m_j following the edge e_i, for example $x_i' = x_i$ where x_i' represents the updated value of the variable x_i after the edge e_i has been traversed. $\mathcal{R}$ is the set of all reset conditions.

3.5 Assumptions

The proposed safety verification HA mining-based technique requires the manufacturer provides a reference safety assured (certified) hybrid automaton model of the system. However, manufacturers may not be able to provide a hybrid automaton as the reference specifications model for our safety verification technique since they may have used a different model for specifications such as i*, UML, SysML, MARTE, Agent UML. However, learning a hybrid automaton model from the specifications document or mapping a given specifications model to a hybrid automaton model is feasible (Burmester, Giese, and Oberschelp 2006; Schmitz et al. 2009; Liu et al. 2013a). For example, for safety-critical control systems such as aircrafts, a simulator is required for operators' training [2]. The simulator can be used in our technique as the reference specifications model. For artificial pancreas, UVA/Padova is an FDA approved simulator and has been largely adopted in research as replacement for preclinical trials of certain insulin treatments, including testing closed-loop control algorithms for AP (Man et al. 2014). Hence, combining the control logic of the artificial pancreas with the simulator can represent the reference specifications Simulink model of the Medtronic 670G. Also, several previous work have proposed algorithms and frameworks for mapping some formalism model to a hybrid automaton. For example, Minopoli and Frehse created a tool for translating a simulink model to a hybrid automaton (Minopoli and Frehse 2016). Lyde and Might proposed an approach for analyzing control code using abstract interpretation and inferring a hybrid automaton from an abstract state transition system (Lyde and Might 2013). Another assumption of the proposed technique is that the traces are noiseless. However, most CPS have signal processing and filtering algorithms to assure only good quality sensory data is used by the control logic. Thus, the controller of CPS uses filtered, adjusted, and calibrated data to make decisions. However, the data collected from the sensor may not processed if the pre-filtering, filtering, or calibration are not included as part of the sensor transmitter. In this case, our proposed approach assumes that the pre-filtering, filtering, or calibrating approaches are provided in the specifications document (since it is part of the controller). Hence, these modules are part of the hidden control variables and if we have a description of their specifications, then we can easily simulate processed data.

[2] https://www.bloomberg.com/news/articles/2019-11-08/delays-in-boeing-max-return-began-with-near-crash-in-simulator

3.6 Notation

$\mathbb{R}$ is the set of real numbers.
$\dot{x}$, and $\frac{dx}{dt}$ both mean differential of x with respect to t.
A polyhedral constraint over a a list of variables $\mathcal{X}$ is a finite conjunction of linear constraints over $\mathcal{X}$.
A linear constraint over a list of variables $\mathcal{X}$ is defined as an expression of the form $P(\mathcal{X}) \lozenge r$, where $P(\mathcal{X})$ is a polynomial term over $\mathcal{X}$, $r \in \mathbb{R}$, and $\lozenge \in \{\leq, <, \geq, >, =\}$.

4 Safety Verification HA-Mining Based Methodology

We propose a safety verification algorithm based on reverse engineering of a non-linear AI-enabled CPS from operational time-series traces collected from the operation of CPS out in the field and system specifications (initial HA) provided by the manufacturer. The scheme of the proposed safety verification technique is shown in Figure 1.

- The HA mining algorithm takes the following inputs -
  - The time series traces obtained from the operation of the AI-enabled CPS, and
  - Documentation that contains general information including controller frequency, requirements, and design document. We use this documentation to model the initial HA of the AI-enabled system, if not provided in the system documentation.
- It employs relative unconstrained least-squares importance fitting (RuLSIF) and density-based clustering algorithm on time-series data to derive the discrete mode transitions of the AI-enabled system.
- It employs Fisher information based analysis and Cramer Rao bound to derive the guard and reset conditions between every two control modes, as performed in our previous work (Lamrani, Banerjee, and Gupta 2018).
- For each derived mode, it employs multi-variable polynomial regression analysis MultiPolyRegress written by Ahmet Cecen in MATLAB Central to derive the physical environment flow equations.
- The output of the HA mining algorithm is a learned nonlinear hybrid automaton $HA_{Learned}$.
- It then evaluates the consistency between the newly mined HA ($HA_{Learned}$) and the initial HA provided by the manufacturer. If $HA_{Learned}$ is same as the initial hybrid model defined in the documentation provided by the manufacturer, then there is no change in the safety conclusion. This similarity is expressed as reach sets comparison, as shown in Figure 6.

4.1 RuLSIF Change-Point Detection Method

The goal of the change-point detection technique is to discover abrupt changes lying behind time-series data. Early proposed change detection approaches are not robust against different types of changes, which significantly limits the range of applications in practice. Recent efforts within this line of research introduced a new strategy which estimates the ratio of probability densities instead of directly estimating the density (Liu et al. 2013b). In this paper, we apply

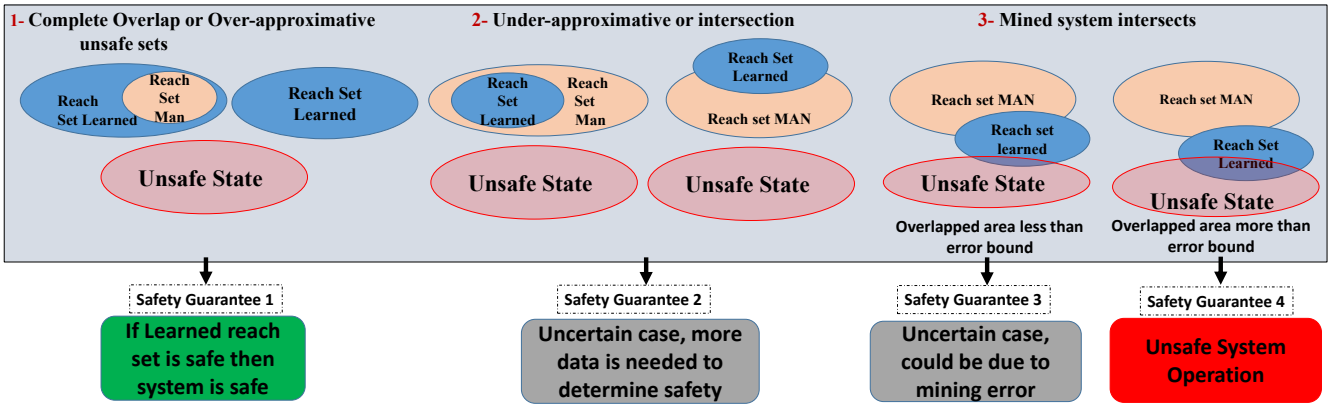

Figure 3: Safety Analysis Cases

the relative unconstrained least-squares importance fitting (RuLSIF) method, which was reported to outperform competive approaches in regard to robustness, optimality of non-parametric convergence rate, and optimality of numerical stability (Aminikhanghahi and Cook 2017). The main idea behind the RuLSIF method is to bound the density-ratio and use the α-relative Pearson (PE) divergence as a dissimilarity measure, where $0 < \alpha < 1$. Thus, the RuLSIF dissimilarity measure has the following form: $PE_\alpha[p(x)||p'(x)] = PE(p(x)||\alpha p(x) + (1-\alpha)p'(x))$.

4.2 Multivariate Non-Linear Polynomial Regression Analysis

We consider the problem of estimating a non-linear relationship among the set of continuous variables $\mathcal{X}$ from a series of observations. Multivariate polynomial regression analysis can be performed on multidimensional data to model non-linear variables that depend on more than one variable by fitting data to higher order multidimensional polynomials. For example, a second order polynomial for an equation of two variables has the following form: $y = a_1 + a_2 x_i + a_3 y_i + a_4 x_i^2 + a_5 x_i y_i + a_6 y_i^2, i = 1, ..., n$, where n represents the number of data points. In this paper, each differential equation of $x_i \in \mathcal{X}$ with respect to time is regressed on powers of the variables in $\mathcal{X}$ while fitting the data into the non-linear polynomial regression models to find the best fit curve (Cecen 2017).

4.3 Safety Guarantees

We assume that the manufacturer has proven the safety of the hybrid automaton representation of the system. This means that the reach set of the hybrid system provided by the manufacturer will not intersect with the unsafe set, as shown in Figure 3. The solution of the reachability analysis always provides a solution that is an over-approximation of the system's operating envelope (Alur et al. 1995). With respect to the learned system, there can be four distinct safety guarantee cases:

1) The reach set of the learned system is an over-approximation of the specified system and encompasses the reach set of the specified system but it does not intersect the unsafe set. In such a case, we can guarantee that the system is operating within the safety envelope.

2) The reach set of the mined system is an underapproximation of the reach set of the specified system or intersects it and mined system does not intersect the unsafe state. This is an uncertain scenario, because the deviation can be either due to a change in system operation or can be due to error in mining.

3) The reach set of the mined system intersects unsafe set but the area of intersection is within error bound of the mining technique. This case is also an uncertain case, because the intersection with unsafe set can be either due to a problem with the system operation or due to an error in the mining.

4) The reach set of the mined system intersects unsafe set and area of intersection is greater than the error bound of the mining technique. In such a scenario, we can guarantee that this is due to an unsafe operation of the system.

5 Evaluation

In this section, we consider the effectiveness of extracting a hybrid automaton for the artificial pancreas (AP) control system. Data used in this experiment are collected from our collaboration with MAYO clinic. Collected data consist of CGM readings and meal intake amounts. In order to obtain the remaining inaccessible signals, we used the UVA/Padova T1d platform to simulate traces for interstitial insulin X and plasma insulin concentration I for one T1D subject (Man et al. 2014). From I/O traces, we apply our technique to obtain the learned hybrid automaton and compare its operation to the one provided by the manufacturer (Banerjee et al. 2013) . We compare both inferred and given hybrid automata semantics to find out inconsistencies between the two models. We apply C2E2 tool to perform our reachability analysis evaluations (Fan et al. 2016).

Artificial Pancreas (AP): We first consider I/O traces from the AP control system. Here G and I are the continuous state variables and also the controller inputs, and the external insulin infusion rate I_t is the controller output.

- The first step of the HA mining algorithm is to em-

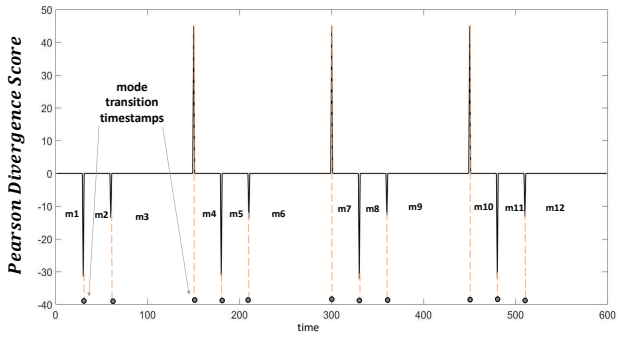

Figure 4: RuLSIF I/O segmentation example

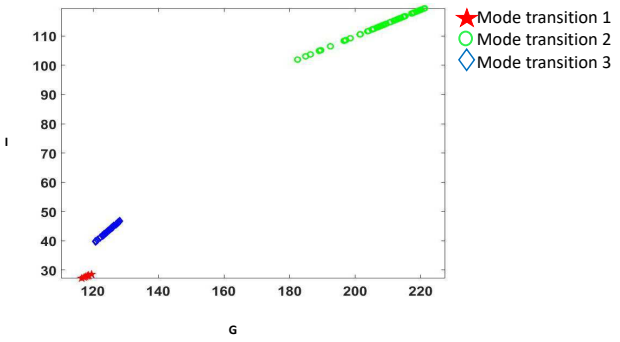

Figure 5: Density-Based Clustering of Mode Transitions.

ploy RuLSIF, as described in Section 4.1, on I/O data to find abrupt changes lying behind I/O time-series data, which represent potential control mode changes. From Figure 4, we initially consider the mode set $M = \{m_1, m_2, \ldots m_{12}\}$ 12 distinct modes.

- For the artificial pancreas, controller decisions are related to the predicted values of the continuous state variables in the next 30 minutes. At each transition timestamp, values of controller output before and after the mode transition and values of the continuous state variables in the next 30 minutes are considered. Collected timeseries data are used as features in density-based clustering algorithm to group unique control mode changes, as shown in Figure 5.

- The HA mining technique then derives the reset condition for each cluster, where each cluster represents a distinct control mode change. If the reset value is not a constant value of actuation and is varying within each data point in the cluster, we employ Fisher information and Cramer Rao bound to derive the linear relation of I_t with G, X, and I (Lamrani, Banerjee, and Gupta 2018). The analysis results in the following equation 4 for the cluster encompassing mode transitions from mode $m2$ to $m3$ depicted with blue diamonds in Figure 5.

$$I_t = 0.5G + 44.75 \tag{4}$$

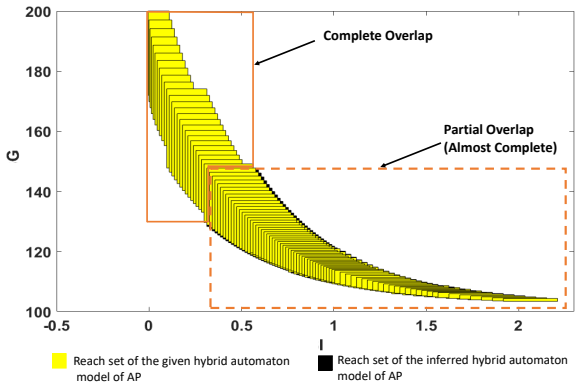

Figure 6: Comparison between the reach set of the learned HA and initial HA provided by the manufacturer.

- The HA mining technique then uses the multivariate polynomial regression MultiPolyRegress in Matlab, described in Section 4.2, to infer the flows equations for each control mode using I/O time-series data. For the variation of blood glucose we got the following equation, where terms with minor influence are canceled out:
$\dot{G} = \cancel{-4.6948^{-9}X} + \cancel{1.1483^{-12}I} + \cancel{-3.7926^{-11}IX} + -0.031684G - GX + \cancel{-7.1742^{-15}GI} + 2.9149 - \cancel{1.4819^{-14}I^2}$.
Thus, HyMn infers the following set of equations for one T1D subject:

$$\dot{G} = -X(t)G(t) - 0.03G(t) + 2.9, \tag{5}$$
$$\dot{I} = -0.23I(t) - 0.09G(t) + 17.03. \tag{6}$$

For every segment we obtained the same set of equations 5 and 6 with different reset conditions I_t, resulting in the conclusion that m_1, m_2, m_3 are unique modes and are not composite (breaking, basal, and bolus control modes).

- The next step is to determine the guard condition for each control mode change. Using an observation matrix of the continuous state variables at the time of the control mode change, rectangular or non-rectangular guards are learned as a conjunction of linear constraints over the continuous state variables, as reported in our previous work (Lamrani, Banerjee, and Gupta 2018).
- Finally, a reach set comparison is performed between the mined HA and the initial HA provided by the manufacturer as depicted in Figure 6.

6 Discussions and Conclusions

The main contribution of this work is a new scheme for the safety verification during the operational phase of an AI-enabled CPS. The proposed approach is based on hybrid automata mining from traces collected from the operation of AI-enabled systems. The safety verification uses the learned hybrid automaton and compares it with the specifications of the system given by the manufacturer to ensure that the operation of the system conforms with the design

safety properties. If the learned hybrid automaton has less control modes than the reference specifications model provided by the manufacturer, then traces representing the missing modes are not present in the available traces. With insufficient I/O traces, the learned hybrid automaton will be an underapproximative representation of the reference specifications model. In this case, the manufacturer should provide complete traces to accomplish the mining process. We apply the proposed technique to the artificial pancreas control system and demonstrated the effectiveness of this safety verification technique. The difference between the reach sets of the learned HA and the initial HA provided by the manufacturer is very minimal that it may not require root cause analysis. However, a root cause analysis using the learned HA model is surely one of our future endeavors. We identify all the possible outcomes of the proposed safety analysis and identified which cases lead to certain safety conclusions and other cases where more analysis is needed. This can provide certification agencies such as FAA or FDA with important directives regarding safety adherence of the operational system. The fidelity of the learned HA is based on numerical guarantees which consists on comparing collected I/O traces for verification (different than data used in training) to those generated using the inferred HA by calculating the root mean square error (RMSE) between the two sets of traces. It remains for future work to develop formal guarantees of the proposed method. Another direction for improvement is how to decide what is the reasonable period of time to mine again another automaton for periodic safety verification of operational AI-enabled CPS.

Operational Safety of Boeing Aircrafts:

Operational safety verification can be used to learn about the operation of MCAS of the Boeing aircraft. The MCAS system is an automatic pitch control system that gets activated when the Angle of Attack (AoA) of the aircraft is very high. When the AoA reading from the sensor is high depending on a threshold set by the manufacturer, the MCAS system gets activated. The algorithm then pushes the horizontal stabilizer trim upward at the rate of 0.27 degrees per second, to either up to 2.5 degrees or for a maximum of 9.26 seconds[3]. The proposed operational safety HA-learning based technique can be applied to learn a hybrid automaton representing two control modes MCAS enabled and MCAS disabled, the aircraft environmental model that can be expressed using differential equations representing the rate of change of AoA in each control modes, and the guard condition that enables transition from one mode to the other. This allows the Boeing operator to be aware of the operation of MCAS as well as its operational model. If the manufacturer of MCAS allows access to its specifications model, then operational safety allows conformance verification between the certified MCAS operation and its operation in the real world.

[3]https://theaircurrent.com/aviation-safety/what-is-the-boeing-737-max-maneuvering-characteristics-augmentation-system-mcas-jt610/

References

Abbas, H. Y. 2015. *Test-based falsification and conformance testing for cyber-physical systems*. Ph.D. Dissertation, Arizona State University.

Alur, R.; Courcoubetis, C.; Halbwachs, N.; Henzinger, T. A.; Ho, P.-H.; Nicollin, X.; Olivero, A.; Sifakis, J.; and Yovine, S. 1995. The algorithmic analysis of hybrid systems. *Theoretical computer science* 138(1):3–34.

Aminikhanghahi, S., and Cook, D. J. 2017. A survey of methods for time series change point detection. *Knowledge and information systems* 51(2):339–367.

Andersen, K. E., and Højbjerre, M. 2002. A bayesian approach to bergman's minimal model. *Insulin* 50(100):200.

Araujo, H.; Carvalho, G.; Mohaqeqi, M.; Mousavi, M. R.; and Sampaio, A. 2018. Sound conformance testing for cyber-physical systems: Theory and implementation. *Science of Computer Programming* 162:35–54.

Balakrishnan, H.; Hwang, I.; Jang, J. S.; and Tomlin, C. J. 2004. Inference methods for autonomous stochastic linear hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, 64–79. Springer.

Banerjee, A.; Zhang, Y.; Jones, P.; and Gupta, S. 2013. Using formal methods to improve home-use medical device safety. *Biomedical Instrumentation and Technology* 47(SPRING):43–48.

Blackmore, L.; Gil, S.; Chung, S.; and Williams, B. 2007. Model learning for switching linear systems with autonomous mode transitions. In *2007 46th IEEE Conference on Decision and Control*, 4648–4655. IEEE.

Burmester, S.; Giese, H.; and Oberschelp, O. 2006. Hybrid uml components for the design of complex self-optimizing mechatronic systems. In *Informatics in Control, Automation and Robotics I*. Springer. 281–288.

Cecen, A. 2017. *Calculation, utilization, and inference of spatial statistics in practical spatio-temporal data*. Ph.D. Dissertation, Georgia Institute of Technology.

Clarke, W. L.; Anderson, S.; Breton, M.; Patek, S.; Kashmer, L.; and Kovatchev, B. 2009. Closed-loop artificial pancreas using subcutaneous glucose sensing and insulin delivery and a model predictive control algorithm: the virginia experience.

Contag, M.; Li, G.; Pawlowski, A.; Domke, F.; Levchenko, K.; Holz, T.; and Savage, S. 2017. How they did it: An analysis of emission defeat devices in modern automobiles. In *2017 IEEE Symposium on Security and Privacy (SP)*, 231–250. IEEE.

Dreossi, T.; Fremont, D. J.; Ghosh, S.; Kim, E.; Ravanbakhsh, H.; Vazquez-Chanlatte, M.; and Seshia, S. A. 2019. Verifai: A toolkit for the formal design and analysis of artificial intelligence-based systems. In *International Conference on Computer Aided Verification*, 432–442. Springer.

Fan, C.; Qi, B.; Mitra, S.; Viswanathan, M.; and Duggirala, P. S. 2016. Automatic reachability analysis for nonlinear hybrid models with c2e2. In *International Conference on Computer Aided Verification*, 531–538. Springer.

Frehse, G.; Le Guernic, C.; Donzé, A.; Cotton, S.; Ray, R.; Lebeltel, O.; Ripado, R.; Girard, A.; Dang, T.; and Maler, O. 2011. Spaceex: Scalable verification of hybrid systems. In *International Conference on Computer Aided Verification*, 379–395. Springer.

Lamrani, I.; Banerjee, A.; and Gupta, S. K. 2018. Hymn: Mining linear hybrid automata from input output traces of cyber-physical systems. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 264–269. IEEE.

Leveson, N. 2011. *Engineering a safer world: Systems thinking applied to safety*. MIT press.

Liu, J.; Liu, Z.; He, J.; Mallet, F.; and Ding, Z. 2013a. Hybrid marte statecharts. *Frontiers of Computer Science* 7(1):95–108.

Liu, S.; Yamada, M.; Collier, N.; and Sugiyama, M. 2013b. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks* 43:72–83.

Ly, D. L., and Lipson, H. 2012. Learning symbolic representations of hybrid dynamical systems. *Journal of Machine Learning Research* 13(Dec):3585–3618.

Lyde, S., and Might, M. 2013. Extracting hybrid automata from control code. In *NASA Formal Methods Symposium*, 447–452. Springer.

Man, C. D.; Micheletto, F.; Lv, D.; Breton, M.; Kovatchev, B.; and Cobelli, C. 2014. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology* 8(1):26–34.

McDermid, J. A.; Jia, Y.; and Habli, I. 2019. Towards a framework for safety assurance of autonomous systems. In *Artificial Intelligence Safety 2019*, 1–7. CEUR Workshop Proceedings.

Medhat, R.; Ramesh, S.; Bonakdarpour, B.; and Fischmeister, S. 2015. A framework for mining hybrid automata from input/output traces. In *Proceedings of the 12th International Conference on Embedded Software*, 177–186. IEEE Press.

Minopoli, S., and Frehse, G. 2016. Sl2sx translator: from simulink to spaceex models. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, 93–98. ACM.

Niggemann, O., and Lohweg, V. 2015. On the diagnosis of cyber-physical production systems: state-of-the-art and research agenda. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 4119–4126. AAAI Press.

Niggemann, O.; Stein, B.; Vodencarevic, A.; Maier, A.; and Büning, H. K. 2012. Learning behavior models for hybrid timed systems. In *AAAI*, volume 2, 1083–1090.

Niggemann, O.; Windmann, S.; Volgmann, S.; Bunte, A.; and Stein, B. 2014. Using learned models for the root cause analysis of cyber-physical production systems. In *Int. Workshop Principles of Diagnosis (DX)*.

Russell, S. J., and Norvig, P. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Santana, P. H.; Lane, S.; Timmons, E.; Williams, B. C.; and Forster, C. 2015. Learning hybrid models with guarded transitions. In *AAAI*, 1847–1853.

Schmitz, D.; Zhang, M.; Rose, T.; Jarke, M.; Polzer, A.; Palczynski, J.; Kowalewski, S.; and Reke, M. 2009. Mapping requirement models to mathematical models in control system development. In *European Conference on Model Driven Architecture-Foundations and Applications*, 253–264. Springer.

Summerville, A.; Osborn, J.; and Mateas, M. 2017. Charda: Causal hybrid automata recovery via dynamic analysis. *arXiv preprint arXiv:1707.03336*.

Woehrle, M.; Lampka, K.; and Thiele, L. 2012. Conformance testing for cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)* 11(4):84.

Zhu, H.; Xiong, Z.; Magill, S.; and Jagannathan, S. 2019. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 686–701. ACM.