Security Analysis of RL-Based Artificial Pancreas Systems

Preston Chang
Stony Brook University
Stony Brook, NY, USA
pjchang@cs.stonybrook.edu

Veena Krish
Stony Brook University
Stony Brook, NY, USA
kveena@cs.stonybrook.edu

Amir Rahmati
Stony Brook University
Stony Brook, NY, USA
amir@cs.stonybrook.edu

Abstract

Reinforcement learning (RL) based models promise to replace time-consuming traditional model-based control methods for medical control systems. Recently, Deep RL approaches have been explored for use in autonomous systems for glycemic control, often termed Artificial Pancreas systems, which require closed-loop communication between a glucose sensor and an insulin pump. In this work, we investigate the robustness of RL4BG, a prominent deep reinforcement learning-based AP controller, to a suite of glucose sensor malfunctions. We model two classes of realistic malfunctions stemming from natural and/or adversarial factors: a Denial-of-Service failure class that simulates a worst-case sensor malfunction, and a Subtle manipulations failure class that simulates stealthier but prolonged failure. Our findings show that this new class of medical control systems may be vulnerable to anomalous inputs in safety-critical settings. These vulnerabilities motivate further work into training medical RL systems in an adversarially robust fashion.

CCS Concepts

• **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → *Adversarial learning*; • **Computer systems organization** → **Sensors and actuators**.

Keywords

Artificial Pancreas, Reinforcement Learning-based Control Systems, Adversarial Machine Learning

ACM Reference Format:
Preston Chang, Veena Krish, and Amir Rahmati. 2024. Security Analysis of RL-Based Artificial Pancreas Systems. In *Proceedings of the 2024 Workshop on Cybersecurity in Healthcare (HealthSec '24), October 14–18, 2024, Salt Lake City, UT, USA*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3689942.3694740

1 Introduction

Type 1 Diabetes, a condition where the pancreas is unable to produce insulin, affects 8.4 million individuals globally [10]. These individuals typically control their blood glucose levels via a basal-bolus method in which patients inject *basal* long-acting insulin in order to control blood sugar at rest and a rapid-acting *bolus* dose in order to combat blood sugar spikes from meals. The basal-bolus method requires careful tracking and rigorous manual adjustments throughout the day [6]. Automated closed-loop insulin delivery systems offer many advantages to current mainstream methods of control, including both improved clinical outcomes and quality of life [12]. These closed-loop systems, often referred to as Artificial Pancreas (AP) systems, are comprised of three main parts as shown in Figure 1: a continuous glucose monitor (CGM), an insulin infusion pump, and control software that determines the amount of insulin to be delivered in response to the sensed glucose measurements. The first commercially available AP, Medtronic's MiniMed 670G, was approved by the FDA in September 2016 [3]. Over the years since, five additional systems have received approval and are currently available [18].

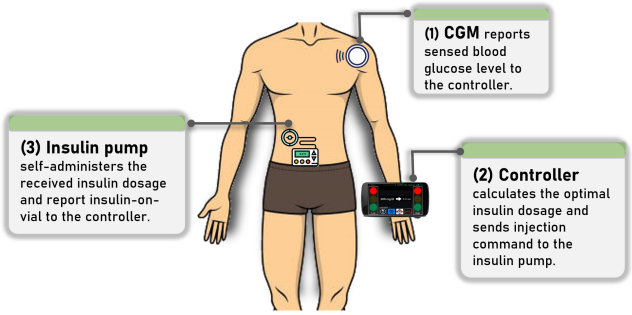

Figure 1: Artificial Pancreas System Diagram by Kim *et al.* [12] illustrating the three components of a typical AP system: (1) The CGM, (2) the Controller, and (3) the Insulin pump.

Various algorithms have been introduced to mediate between the CGM and the insulin infusion pump, ranging from older basal-bolus policies to newer ML-based methods [9]. While closed-loop policies offer tremendous benefits in terms of reducing the burden of constant monitoring, the extent to which reinforcement learning-based policies can make safe long-term decisions is still an open area of research. In this work, we focus on the potential harm that sensor failures, happening inadvertently or adversarially, can cause to such RL-based artificial pancreas systems by manipulating CGM measurements in unexpected ways. Because the safety of these models is critical to the well-being of users, it is vital to test the safety, efficacy, and robustness of these models through offensive techniques.

In this work, we take RL4BG [9], a representative ML-based approach that leverages state-of-the-art techniques in reinforcement learning, and we evaluate its reliability in response to various classes of CGM sensor malfunction. Primarily, we focus on two realistic scenarios of sensor error that may result from a natural malfunction or purposeful malware: (1) unresponsive sensors that

HealthSec '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-1238-8/24/10
https://doi.org/10.1145/3689942.3694740

cause maxed-out readings for a short period of time and (2) slight deviations to glucose measurements over time. For RL models finely tuned to specific patient models, we investigate the extent of patient harm caused by failures of various intensities and durations.

2 Background

2.1 Artificial Pancreas Systems

Individuals with T1D control their blood glucose level through frequent measurements and insulin shots, either through an injection or an insulin infusion pump, several times a day. For healthy individuals, the estimated blood glucose over 2 to 3 months is between 70 mg/dL and 126 mg/dL. The American Diabetes Association recommends that T1D patients target an estimated blood glucose level below 154 mg/dL. It is also recommended, in general, that individuals target the euglycemia range (70-180 mg/dL) and avoid both hypoglycemia (<70 mg/dL) and hyperglycemia (>180 mg/dL). Moderate hyperglycemia carries the risk of frequent urination, increased thirst, blurred vision, or feeling weak or unusually tired. In contrast, severe hyperglycemia can lead to ketoacidosis, which, at worst, can cause loss of consciousness or even death. Moderate hypoglycemia carries the risk of dizziness, headaches, or arrhythmia, while severe hypoglycemia can cause the loss of consciousness or even seizures [4, 5, 8].

An artificial Pancreas (AP) is a term used to denote a fully automated, closed-loop system comprised of three parts that work together to mimic how a healthy pancreas controls blood glucose in the body: a Continuous Glucose Monitor (CGM), an insulin infusion pump, and control software that tells the infusion pump how much insulin to deliver to the patient based on the readings of the CGM [2].

As of 2016, a majority of the devices in clinical testing used the Proportional-Derivative-Integral (PID) controller [19] for determining insulin dosages. This includes the aforementioned Medtronic's MiniMed 670G, the first commercially available artificial pancreas approved by the FDA [3]. However, there have been concerns over PID controllers' susceptibility to hypoglycemia because of delays between insulin delivery and blood glucose response. In response, there has been research into leveraging machine learning, specifically reinforcement learning approaches, to train controllers that can better recognize patterns associated with meal times, resulting in more responsive and safer policies [9].

2.2 Reinforcement Learning

Reinforcement learning (RL) is a machine learning paradigm inspired by behavioral psychology, in which an agent learns to make sequential decisions in an environment to achieve certain goals. Unlike supervised learning, in which the model is trained on labeled data pairs, or unsupervised learning, in which the model finds patterns in unlabeled data, RL deals with learning from interaction to achieve a goal.

In reinforcement learning, the agent is trained by interacting with an environment, taking actions, and observing the subsequent state transitions and rewards. The agent's goal is to learn a policy, a mapping from states to actions, that maximizes cumulative rewards over time. This is achieved through exploring various possible states and selecting actions that will lead to the highest immediate or long-term rewards based on its current knowledge. In the context of glycemic control, we consider glucose measurements as the state, dosed insulin as the action, and rely on comprehensive patient models for the state transition (*i.e.,* compute the next state in response to dosed insulin).

At each time step, the agent selects an action based on its current policy and the observed state. After taking the action, the agent receives a reward from the environment, indicating its immediate desirability. The agent then updates its policy based on this reward and its overall objective, which could be to maximize long-term rewards or to learn a specific behavior.

2.3 Simulation Environment

To simulate type 1 diabetes patients in-silico, we use the simglucose environment. simglucose is a Python implementation of the 2008 version of the FDA-approved UVa/Padova Simulator for research purposes [1]. The simglucose environment contains 30 virtual patients (10 children, 10 adolescents, and 10 adults) with different physiological profiles. The state of these patients can be simulated over the course of a user-determined time period. At each timestep of the simulation, the patient's state is passed to the controller algorithm, which would then output an insulin value. This process repeats until the simulation reaches completion. Importantly, simglucose includes the ability to define a custom reward function and a controller algorithm. By default, the simglucose simulation lasts for 10 days, or 2880 timesteps, with each timestep being 5 minutes.

2.4 RL4BG and Soft-Actor Critic

RL4BG is a RL-based algorithm for learning a policy to be used in an AP controller introduced in the paper *Deep Reinforcement Learning for Closed-Loop Blood Glucose Control* by Fox *et al.* [9]. This algorithm is an RL-based algorithm based on the soft actor-critic policy (SAC) introduced in the paper *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor* by Haarnoja *et al.* [11]. SAC comprises two parts: an actor, which is a neural network that outputs a probability distribution over actions given a state, and a critic, which is a neural network that estimates the expected return for each state-action pair. In the context of glucose control, the action taken by the controller represents a dosage of insulin administered at each execution. Training the algorithm involves iteratively updating the actor network to maximize a balance between expected return and entropy, which is a measure of the diversity and generalizability of the policy, and updating the critic network to minimize the error between predicted and actual returns [11].

The authors evaluated the performance of their trained RL4BG models by comparing them to two industry-standard controllers over 10-day simulations: (1) a Basal-Bolus (BB) controller, which is traditionally used to control type 1 diabetes by administering a longer-acting form of insulin to keep blood glucose levels stable through periods of fasting and separate injections of shorter-acting insulin to prevent rises in blood glucose levels resulting from meals, and (2) a proportional–integral–derivative (PID) controller, which adjusts the insulin response based on three different factors: the current glucose reading (proportional), the cumulative summed

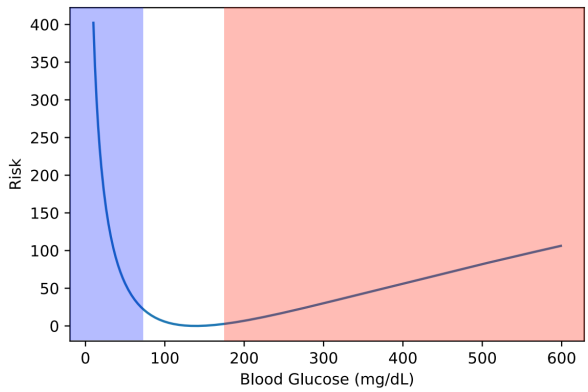

Figure 2: Magni Risk function used to train the RL4BG controller [9, 15]. The blue region represents hypoglycemia, the white represents euglycemia, and the red represents hyperglycemia. This risk function weights time spent in hypoglycemia exponentially more than time spent in other regimes.

difference between the measured and target glucose level (integral), and the rate of change of the glucose reading over time (derivative) [16]. The RL4BG models outperformed both strategies. In addition, the standard methods require patients to perform meal announcements and CHO estimations, in which they communicate to the AP system the quantity of carbohydrates that will be consumed. RL4BG achieves strong performance without meal announcement and CHO estimation, making it much more convenient for the patient and eliminating the risk of human error. The authors also trained a version of RL4BG called RL-MA, which further trains RL4BG on the automated meal boluses from the basal-bolus and PID strategies. RL-MA, which requires meal announcements, achieves even stronger performance than RL4BG at the cost of convenience. Overall, RL4BG outperformed all other non-meal announcement algorithms, and RL-MA outperformed all other meal announcement algorithms [9]. In this work, We focus on non-meal announcement scenarios as they represent the ideal case for patient convenience.

3 Sensor Failure Scenarios

The sensor failure threat model assumes that the CGM portion of the AP system is susceptible to failure, and that erroneous readings can be passed to the control software that mediates between the CGM and insulin infusion pump. These erroneous readings could result from natural failure or from an artificial, adversarial manipulation. Regardless of the vector, these errors would result in performance degradation of the system. While a well-trained controller may be able to recover from a single erroneous reading, sensor failure may persist for a period of time and cause persistent effects. Specifically, ongoing sensor failure can result in slightly degraded performance over a longer period of time; at worst, they might lead to a catastrophic failure of the system. We are interested in understanding the extent to which prolonged malfunctions can cause quantifiable patient harm.

We model two realistic types of failure: (1) a *"Denial-of-Service "* framework where we simulate sensor failure for a short period of time, and (2) *Subtle manipulations* of glucose values over a longer time frame. For each mode, we focus on a configuration that would likely result in the most harm. We represent the *Denial-of-Service* as a scenario where the CGM readings are artificially maxed out at a large value. This may result from an unresponsive sensor. For the *Subtle manipulations*, we leverage a gradient-based technique to find the most harmful (risk-maximizing) bounded perturbation of glucose values at each timestep. A well-trained controller may be able to recover from a single anomalous time step, but we investigate the duration of these types of failures to which the RL4BG controller is robust. The following subsections detail the RL4BG model training and methods for simulating sensor failures.

3.1 RL4BG Model Training

We trained RL4BG models based on 12 different simglucose patients[1] with default settings found through hyperparameter tuning. Each model was trained for 300 epochs with 5760 training steps per epoch (20 days). The best model of the 300 epochs was determined by that which maintained blood glucose above 30 mg/dL and had the least total Magni risk. [9]. In both scenarios, we simulate sensor manipulations of extremely high glucose recordings (up to 999mg/dL). While most commercial CGM sensors max out readings a lower rate [7, 17], we nevertheless investigated error from higher readings to test the brittleness of the RL4BG model, which can recieve a larger range of values. Moreover, software bugs may introduce artificially high glucose readings that would not normally be read from the CGM.

3.2 Denial-of-service model

We investigate a denial-of-service model by simulating AP behavior in response to maxing out the CGM sensor for a period of time. We set the CGM readings to the modeled sensor's maximum value for various durations. We choose this extreme scenario as one that would theoretically maximally increase patient risk, as hypoglycemia is considered more harmful than hyperglycemia during model training. This mode also allows us to investigate the possibility of inducing catastrophic failure, which is defined by the authors of RL4BG as blood glucose being less than 5 mg/dL [9]. A robust controller may be able to recover from a single anomalously high reading, but over time, we expect to see instances of catastrophic failure.

3.3 Subtle manipulation model

We additionally investigate a model of sensor failure represented by slight deviations of recorded glucose over a period of time. For this purpose, we leverage Projected Gradient Descent (PGD) [14], a framework originating from adversarial threat modeling in the computer vision domain. The PGD attack scenario assumes a powerful adversary that has full access to the weights of the trained model, but recent work has shown that the attacks succeed with estimates of these weights [13]. This full-access adversarial model is helpful for testing the robustness of trained ML models to noisy inputs. A PGD attack takes the true sample and iteratively alters it in a way that maximizes the loss function used to train the model. The perturbation is typically bounded, representing the closest noisy

[1]We used patients #001, #003, #004, #005 in the child, adolescent, and adult age cohorts.

sample to the true input that would result in the greatest error. This attack framework allows us to calculate bounded sensor noise on the CGM state history that should cause the largest deviation from the intended behavior. We explored three implementations of this framework: the *Full State*, *Current State*, and *Context-driven* approaches, as described below:

Full State Manipulation. We perturb the entire CGM history of the simulation at every time step, and each value is manipulated independently. In this sense, the stored state buffer may not be physically realizable. Each iteration of the PGD update is an independent step along the 48-dimensional vector, so manipulating one variable may not be physically consistent with that made to a neighboring value. While not necessarily physically realizable, these manipulations might result from software-based malware that can directly modify the state buffer. Moreover, this approach serves as a baseline to understand the maximal harm that might be achieved by bounded sensor manipulations and artificial sensor failures (*i.e.,* via malware).

Current State Manipulation. We perturb only the current CGM reading at each time step and do not alter historical values stored in the state buffer. The sequence of stored historical readings would look correct sequentially with respect to insulin dosage. The altered glucose readings (and resulting insulin doses) are stored in this state buffer after application, so over time, this buffer would contain physically sensible but erroneous glucose and insulin measurements.

Context-driven Manipulation. We perturb the CGM history in the controller's state buffer only at specific time steps of the simulation based on sensed insulin values. Intuitively, manipulations at "critical points" would lead to the greatest harm, so we intend to test the controller's reliability after large insulin dosages. Similar to the Current State approach, manipulated readings are not stored in the state buffer.

4 Experimental Results

For all experiments, we simulate AP behavior for given patients over 10 days. The simglucose environment is queried for the patient's state, which is represented by a 96-length buffer. These variables represent historical glucose and insulin measurements from the previous 8 hours. The patient's state array is then sent to the model every 5 minutes, which returns the amount of insulin to be injected. This amount is then evaluated within the simglucose environment to reveal the patient's next state.

At each execution of the model, we modify the glucose readings used to determine the next insulin dosage. Because the RL4BG framework retains a state history buffer, any previous modifications are retained and propagate errors over time. We investigate the extent to which manipulated readings can cause error and determine the cutoff at which the model is no longer generally robust to erroneous readings. Evaluations were repeated for each trained model over 10 random initial seeds. We evaluate the model error in terms of long-term patient harm along a suite of metrics:

Risk. The Magni Risk function, proposed in 2007 [15] is an asymmetrical mapping between blood glucose levels and a "risk" score for glycemic control. This risk weighs hypoglycemic readings higher than hyperglycemic readings, as shown in Figure 2.

Catastrophic Failure.: Binary variable indicating whether BG fell under 5 mg/dL at any time during the episode.

Minimum BG Observed.: Minimum BG value (in mg/dL) observed throughout the episode.

4.1 Denial-of-Service

We first investigated whether it was possible for the model to catastrophically fail (BG <5 mg/dL) as a result of a single model execution. We set the entire CGM history buffer to the maximum sensor reading, 9999 mg/dL, for a single execution. For each trained adult model, we tested randomly generated seeds, and in no run did the single-step perturbation at random locations result in a catastrophic failure case.

This motivated a search for the extent to which the model is robust in response to consecutive large glucose readings. We let the denial-of-service manipulation last for various durations between 1 and 86 timesteps (430 minutes or 3% of the total simulation). We found that catastrophic failure was induced in all scenarios within 14 timesteps when glucose readings were maxed out to 9999 mg/dL. Figure 3 shows these results within the "9999" series: noticeably, not all simulations failed before 10 steps.

We further investigated whether this effect can be seen with lower thresholds: whether this Denial-of-Service framework can reveal similar cases of potential failure if a device limits the maximum glucose reading that can be returned. We repeated these experiments with thresholds from 300 mg/dL to 9999 mg/dL to probe the limits of the trained model to recognize harmful readings. The remaining series in Figure 3 illustrates the tradeoff between the duration of the DoS and the intensity of the glucose reading.

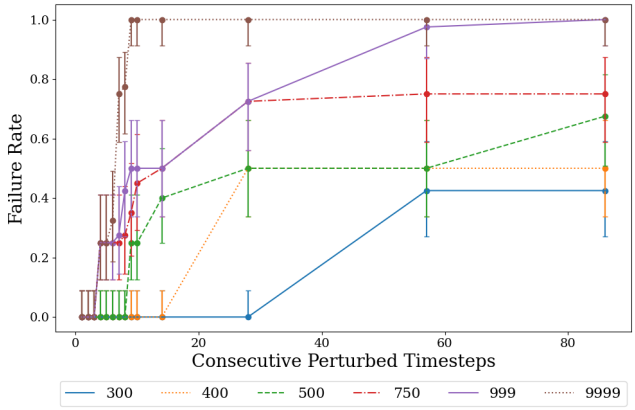

Figure 3: Average failure rate of models for each DoS limit (in mg/dL) and duration (number of consecutive timesteps that the manipulation lasted). Each color/line represents a different DoS threshold, and the Failure Rate represents the average catastrophic failure witnessed over all models and random seeds.

We observe that larger bounds generally lead to failure within a smaller period of time and that failure was seen for all configurations above 300mg/dL. Additionally, we found that we can replicate catastrophic failure in all scenarios above 400 mg/dL with simulated

sensor failure that lasts for at least 10 timesteps, which is equivalent to 50 minutes.

We also observe a discontinuous nature of the failure rate across the range of consecutive timesteps. For instance, a significant jump in failure rate was observed between 6 and 8 timesteps, generally across all experiments before a plateau. This discontinuity reveals that the model can tolerate a certain amount of extremely anomalous inputs for some time, after which failure is probable.

Moreover, all limits tested are anomalously large and physiologically unreasonable: glucose readings during training should rarely exceed 300 mg/dL, and all of these cases would require immediate emergency treatment. Yet, we see that the RL model does differentiate between 999 mg/dL and 9999 mg/dL, revealing that it's learning something in this high-glucose regime that is not physiologically meaningful and should not be useful.

4.2 Subtle manipulation

As discussed in Section 3.3, we simulated and evaluated the three gradient-based approaches at various bounded errors and compared them against a nominal, manipulation-free baseline. For each simulation, we explored various bounds of the maximal deviation of glucose at 1, 5, and 10 mg/dL, which we refer to as ϵ (consistent with the "attack budget" terminology used for the PGD attack). For example, a Full State simulation with $\epsilon = 1$ means that each glucose value in the controller's state buffer can deviate by at most $\pm$ 1mg/dL. For all gradient-based approaches, we tested the same 10 randomly generated scenario seeds as in the Denial-of-Service attacks for all patient models.

Across all experiments, we observe a noticeable trend in that the manipulations result in a "shift" of the CGM trace down over time: maximizing risk at each time step tends to lead the model towards inducing hypoglycemia. A sample trace for a single adult patient is shown in Figure 4. This shift is expected: the Magni Risk function used to train the RL model penalizes time spent in hypoglycemia greater than that spent in hyperglycemia (See Figure 2). Moreover, we also expectedly see that the Full State approach resulted in the greatest harm over the other two approaches.

The effects of sensor manipulations on Average Magni Risk, and Minimum BG observed, and time spent in hypoglycemia are shown in Figures 5, 6, and 7 for all approaches and ϵ bounds. Values shown are relative to average nominal baseline (*i.e.,* no sensor manipulation) values, averaged over all patient models and random seed repetitions. Values are shown relative to the the baseline risk, minimum glucose observed, and time spent in hypoglycemia without any manipulation.

Generally, we see that increasing the ϵ-bound results in greater harm, but that gains are diminishing. Even the slightest bound of $\epsilon = 1$mg/dL can lead to noticeable patient harm over time. We also observe that, as expected, the Full State approach is most harmful, as the entire state buffer can be modified at each step. Moreover, the insulin-contextual approach does illustrate that manipulations can be devised in a targeted manner.

Average harm by various other metrics (including catastrophic failure events, time spent in euglycemia, and time spent in hyperglycemia) for the $\epsilon = 10$mg/dL case are shown in Table 1. While only one of the Subtle manipulations approaches was seen to cause catastrophic failure, all approaches lead to increased risk of harm.

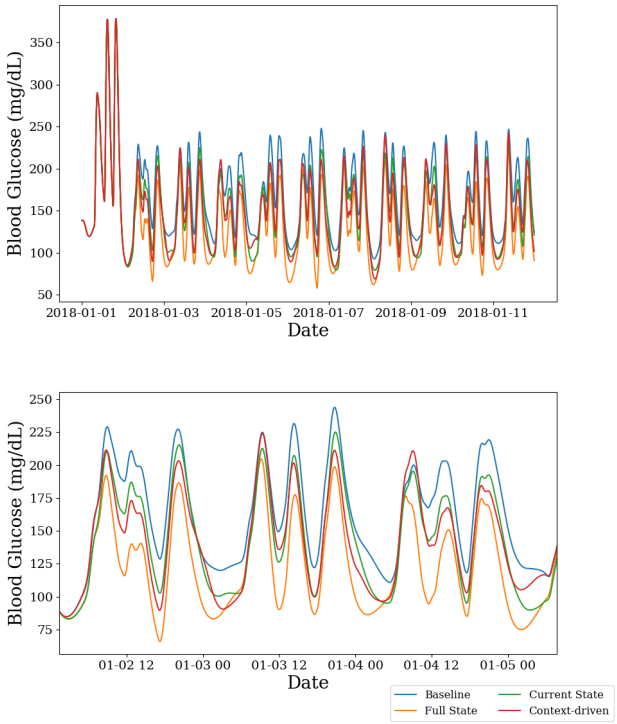

Figure 4: Averaged blood glucose trace of ϵ=10 manipulation scenarios for a single patient model over different time horizons. CGM traces are averaged over 10 random trials, and traces for the various gradient-based approaches are differentiated by color. The bottom figure displays a narrower timeband of the top trace.

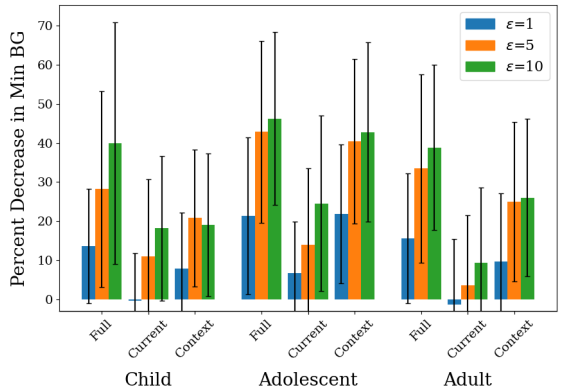

Figure 5: Percent decrease in minimum BG compared to baseline scenarios. Groups are separated by age cohort, scenario type, and epsilon bound (in mg/dL). Results are averaged among all scenarios tested in a group.

We can additionally investigate the tradeoff between the number of timesteps modified and model robustness within the context of

Table 1: Averaged Episode Statistics for Subtle manipulations strategies using ϵ=10 mg/dL, along with Basline (no modification) cases.

Cohort	Strategy with ϵ=10	Risk	Euglycemia	Hypoglycemia	Hyperglycemia	Failure	Min BG
Child	Baseline	9.0765	69.88%	2.41%	27.71%	0.0%	44.3872
	Full State	47.4434	54.32%	24.68%	21.01%	5.0%	25.2004
	Current State	13.3140	64.04%	9.55%	24.79%	0.0%	34.8493
	Context-driven	13.4155	65.84%	10.67%	23.49%	0.0%	34.8212
Adolescent	Baseline	5.9191	73.86%	1.45%	24.68%	0.0%	59.8678
	Full State	17.1106	66.22%	17.10%	16.68%	0.0%	33.0628
	Current State	9.3938	73.26%	6.39%	20.35%	0.0%	45.6501
	Context-driven	13.7540	68.99%	12.52%	18.48%	0.0%	35.0883
Adult	Baseline	7.4068	68.10%	0.82%	31.09%	0.0%	61.7442
	Full State	12.8484	66.53%	10.30%	23.17%	0.0%	54.4612
	Current State	8.6197	69.16%	3.01%	27.82%	0.0%	53.6976
	Context-driven	9.6704	67.34%	5.36%	27.30%	0.0%	44.7182

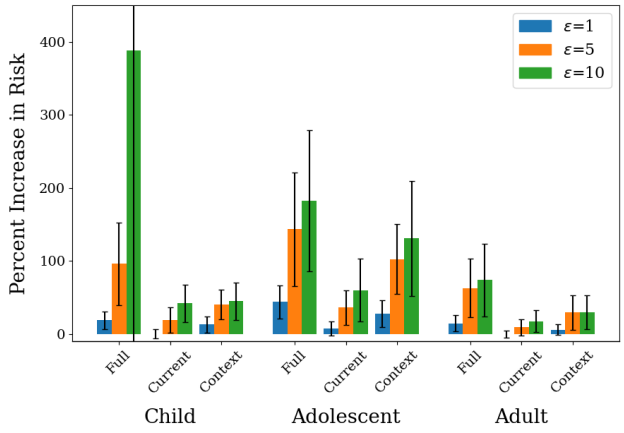

Figure 6: Percent increase in risk compared to baseline scenarios. Groups are separated by age cohort, scenario type, and epsilon bound (in mg/dL). Results are averaged among all scenarios tested in a group.

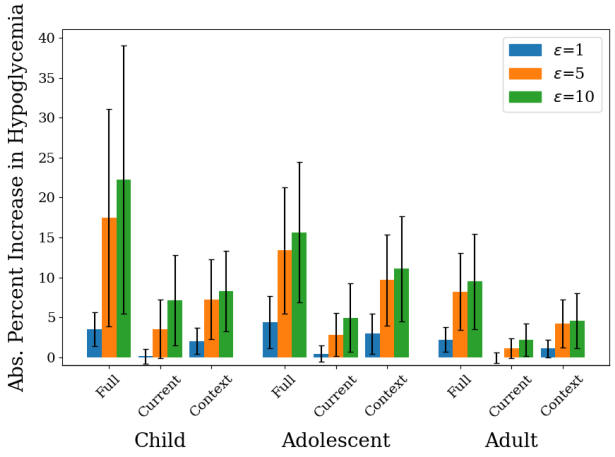

Figure 7: Absolute percent increase in time spent in hypoglycemia compared to baseline scenarios. Groups are separated by age cohort, scenario type, and epsilon bound (in mg/dL). Results are averaged among all scenarios tested in a group.

these three subtle strategies. Initial investigations reveal that the Context-driven strategy is the most successful of the three both at lower bounds and at a smaller number of timesteps modified. We found that a run of the Context-driven strategy on an adult run resulted in an increase in risk by 55% by altering around 5% of the time steps in the episode. In contrast, the Full State strategy on the same episode resulted in an increase in risk by 112% (after altering 100% of the timesteps): half of the latter's degradation in performance was achieved by altering disproportionately fewer number of time steps.

5 Comparison with PID and BB Robustness

We additionally compare the robustness of the RL4BG model with standard, existing methods of closed-loop control to understand the extent to which new vulnerabilities are introduced by adopting an RL approach. The basal-bolus (BB) method is commonly used in manual T1D treatment, while Proportional-Integral-Derivative (PID) methods are commonly used for closed-loop control. For these approaches, we use the implementation provided within simglucose environment and take advantage of the auxiliary code provided in RL4BG to run the simulations. We used default settings provided for the PID and BB simulations provided by the RL4BG authors and the same randomly generated scenario seeds as for the RL setting.

5.1 Denial-of-Service

We applied the DoS attack described in Section 4.1 to the PID and basal-bolus controllers; results are presented and compared with

the corresponding RL4BG performance in Figure 8. We sustain the DoS attack for various durations between 1 and 86 timesteps (5 to 430 minutes or up to 3% of the total simulation). As we increase the length of the DoS manipulation, we see increases in failure rate across all three methods. However, we generally see that the PID and BB controllers fail quickly within a few timesteps; sustaining the attack for longer periods of time did not lead to greater rates of failure as it did for the RL4BG model. For instance, the failure rate of the BB controller for all perturbation magnitudes did not increase from the initial rate until after 28 consecutive timesteps. We also see consistent and early plateauing of performance for BB and PID. In contrast to to the RL4BG performance, where we saw that sustaining the attack for longer continued to degrade performance (especially within the first hour, or 12 timesteps), PID and BB controllers were not as proportionally affected by longer attacks.

5.2 Transfer of subtle manipulation

We also investigated the extent to which the gradient-based subtle manipulations, derived from knowledge of the RL4BG model, could also cause failure in the PID and BB controllers. We applied the gradient-based approaches at various bounded errors derived using the RL4BG models in Section 4.2 to the PID and BB controllers. However, even at the maximal deviation of 10 mg/mL, this approach did not affect the PID and BB performance in any significant way. We believe this is because the subtler attack takes advantage of the fact that the RL4BG controller accepts a large input (a CGM state history is referenced at every execution). Errors caused by subtle changes to CGM readings propagate over time and have greater effects when the manipulated history is used repeatedly. As the PID and basal-bolus methods only rely on the single, latest CGM reading for each execution, they are more robust to subtle manipulations.

6 Discussion

This work motivates the need for comprehensive testing frameworks that model extreme sensor malfunction resulting from natural or adversarial threats. Our experiments revealed scenarios where a well-trained RL-based control model could react in harmful ways in response to unexpected glucose readings without explicit safety controls. Even with a typical safety control: a maximum CGM reading of 400mg/dL used by recent Dexcom and Medtronic devices [7, 17], we see high rates (over 40%) of hypoglycemic failure within a few hours. In addition to providing more comprehensive testing, this framework can be used to reveal trade-offs between the severity of a type of sensor malfunction and the likelihood that it would be detected, along with useful thresholds for tuning anomaly detection systems. The Subtle manipulations class of results shows that a very slight perturbation (±1mg/dL) can lead to disproportionate patient harm over time in an RL model. This type of vulnerability is typically not seen in the PID and BB standard control approaches.

Moreover, we observe differentiated effects by age cohort, which prompts increased attention to the development of age-specific models: greater risk increases over baseline rates were generally seen in the adolescent cohort, as the manipulations tended towards increasing glucose measurements, which in turn prompted large insulin boluses. We speculate that the lower variability in adolescent

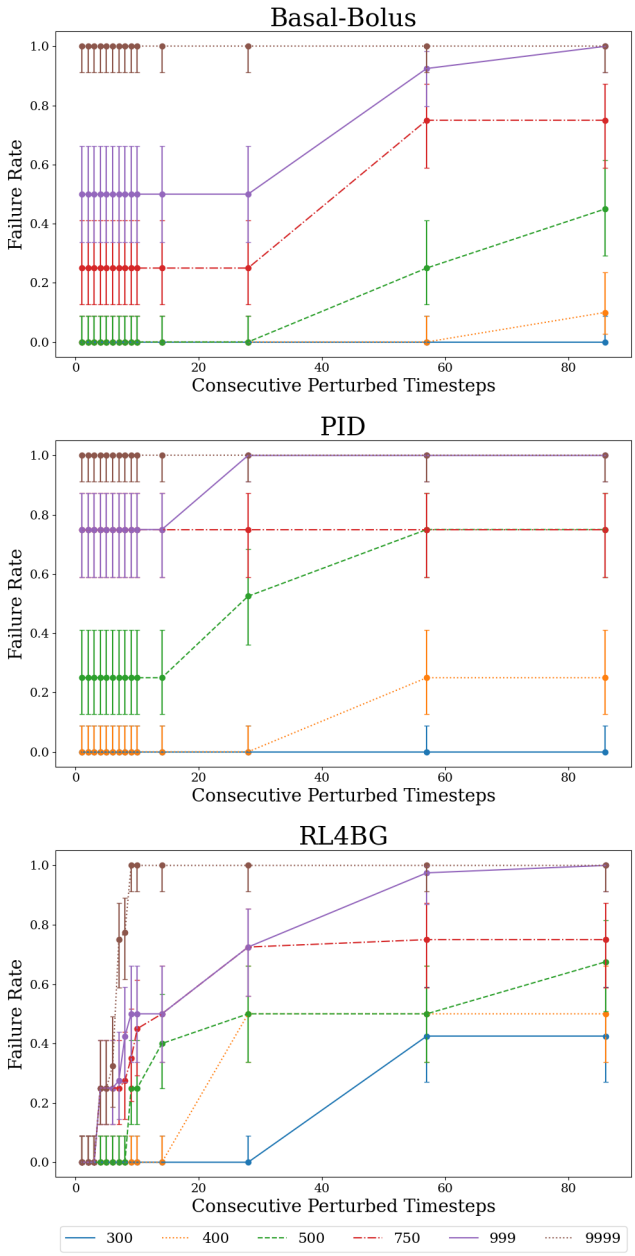

Figure 8: Average failure rate of PID/BB/RL4BG methods for each DoS limit (in mg/dL) and duration (number of consecutive timesteps that the manipulation lasted). Each color/line represents a different DoS threshold, and the Failure Rate represents the average catastrophic failure witnessed over all models and random seeds.

models contributed to this effect, but further work is needed to understand why some classes of manipulations affect cohorts in different ways.

Overall, this work prompts the need for improving the robustness of RL-based models to unexpected inputs. Anomaly detection systems might fail to catch a single slight deviated time-step, but

they could be improved by incorporating knowledge about the minimum duration of attacks needed to exacerbate harm. We anticipate that a combination of adversarial training (*i.e.,* incorporating these synthesized manipulations as part of the training process), continuous anomaly detection, and runtime boundary control decisions are needed to prevent unforeseen consequences.

7 Conclusion

In this work, we introduce two strategies for testing the extent to which RL4BG, a representative RL-based AP controller, is robust to sensor malfunctions. Our two threat models represent worst-case sensor manipulations that can result from natural or adversarial causes. Our test strategies illuminate not only cases of catastrophic failure but also prolonged periods of performance degradation of the AP system, both of which can be fatal for patients. With respect to the first strategy, we show that just 50 minutes of an unresponsive sensor can universally cause catastrophic failure across the simulated T1D patients. With respect to the second strategy, we show that the RL4BG model is not robust to subtle sensor fluctuations over time.

Our results motivate future established practices for testing the bounds of reinforcement learning-based approaches to glycemic control. Whereas standard approaches for evaluating these models with respect to random fluctuations can improve their generalizability to normal glucose inputs, an adversarial-minded suite of testing strategies is critical to test model behavior in response to completely unexpected readings resulting from sensor malfunctions. These testing strategies are vital for evaluating the safety of deep RL controllers and can be applied to future generations of closed-loop models for the AP. Moreover, this work prompts further interest in training more robust RL models. Adversarial training techniques, which incorporate erroneous inputs into the training dataset, might help improve model robustness in the face of noisy sensor readings.

Acknowledgements

We thank Bill Yurcik and the HealthSec program committee for their invaluable feedback during the peer review process. This work was supported by the Air Force Office of Scientific Research under award numbers FA9550-22-1-0450 and FA9550-22-1-0029. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of the sponsors.

References

[1] [n. d.]. Jinyu Xie. Simglucose v0.2.1 (2018) [Online]. Available: https://github.com/jxx123/simglucose. Accessed on: Feb-16-2024..
[2] [n. d.]. Artificial Pancreas. https://www.niddk.nih.gov/health-information/diabetes/overview/managing-diabetes/artificial-pancreas. (accessed Feb. 16, 2024).
[3] [n. d.]. FDA approves first automated insulin delivery device for type 1 diabetes. https://www.fda.gov/news-events/press-announcements/fda-approves-first-automated-insulin-delivery-device-type-1-diabetes. (accessed Feb. 16, 2024).
[4] [n. d.]. Hyperglycemia (High Blood Sugar). https://my.clevelandclinic.org/health/diseases/9815-hyperglycemia-high-blood-sugar. (accessed Feb. 16, 2024).
[5] [n. d.]. Low Blood Glucose (Hypoglycemia). https://www.niddk.nih.gov/health-information/diabetes/overview/preventing-problems/low-blood-glucose-hypoglycemia. (accessed Feb. 16, 2024).
[6] 2023. Insulin for type 1 diabetes. nhs.uk. https://www.nhs.uk/medicines/insulin/insulin-for-type-1-diabetes/. (accessed Feb. 16, 2024).
[7] Dexcom. [n. d.]. Dexcom G7 User Guide. https://dexcompdf.s3.us-west-2.amazonaws.com/en-us/G7-CGM-Users-Guide.pdf. (accessed Aug 30, 2024).
[8] Sandeep K. Dhaliwal. [n. d.]. Estimated average glucose (eAG). medlineplus.gov. https://medlineplus.gov/ency/patientinstructions/000966.htm. (accessed Feb. 16, 2024).
[9] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. 2020. Deep Reinforcement Learning for Closed-Loop Blood Glucose Control. arXiv:2009.09051 [cs.LG]
[10] Gabriel A Gregory, Thomas I G Robinson, Sarah E Linklater, Fei Wang, Stephen Colagiuri, Carine de Beaufort, Kim C Donaghue, International Diabetes Federation Diabetes Atlas Type 1 Diabetes in Adults Special Interest Group, Dianna J Magliano, Jayanthi Maniam, Trevor J Orchard, Priyanka Rai, and Graham D Ogle. 2022. Global incidence, prevalence, and mortality of type 1 diabetes in 2021 with projection to 2040: a modelling study. https://doi.org/10.1016/s2213-8587(22)00218-2
[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv:1801.01290 [cs.LG]
[12] Jiyoon Kim, Jongmin Oh, Daehyeon Son, Hoseok Kwon, Philip Virgil Astillo, and Ilsun You. 2023. APSec1.0: Innovative Security Protocol Design with Formal Security Analysis for the Artificial Pancreas System. *Sensors* 23, 12 (2023). https://doi.org/10.3390/s23125501
[13] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).
[14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083 [stat.ML]
[15] Lalo Magni, Davide M Raimondo, Luca Bossi, Chiara Dalla Man, Giuseppe De Nicolao, Boris Kovatchev, and Claudio Cobelli. 2007. Model predictive control of type 1 diabetes: an in silico trial.
[16] Farhanahani Mahmud, Nadir Hussien Isse, Nur Atikah Mohd Daud, and Marlia Morsin. 2017. Evaluation of PD/PID controller for insulin control on blood glucose regulation in a Type-I diabetes. *AIP Conference Proceedings* 1788, 1 (01 2017), 030072. https://doi.org/10.1063/1.4968325 arXiv:https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/1.4968325/13730996/030072_1_online.pdf
[17] Medtronic. [n. d.]. MiniMed 780G System User Guide. https://www.medtronic.com/content/dam/medtronic-wide/public/canada/products/diabetes/780g-gs3-system-user-guide.pdf. (accessed Aug 30, 2024).
[18] A. Mulvey. [n. d.]. FDA Clears a New Artificial Pancreas System. https://www.jdrf.org/blog/2023/05/22/fda-clears-new-artificial-pancreas-system/. (accessed Feb. 16, 2024).
[19] Sara Trevitt, Sue Simpson, and Annette Wood. 2016. Artificial Pancreas Device Systems for the Closed-Loop Control of Type 1 Diabetes: What Systems Are in Development?