

The PUR-1 Cyber-Physical Digital Twin

Stylianos Chatzidakis

Assistant Professor

Associate Reactor Director, PUR-1

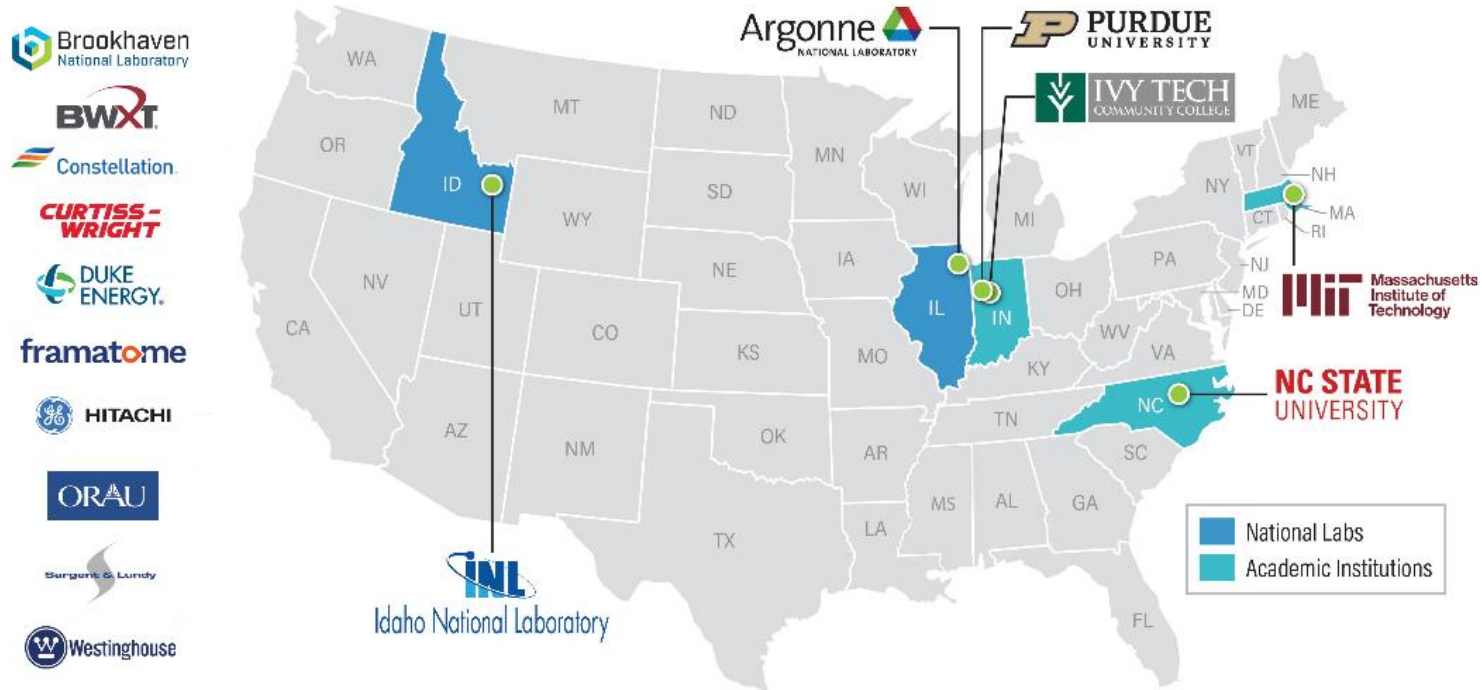
School of Nuclear Engineering

Purdue University

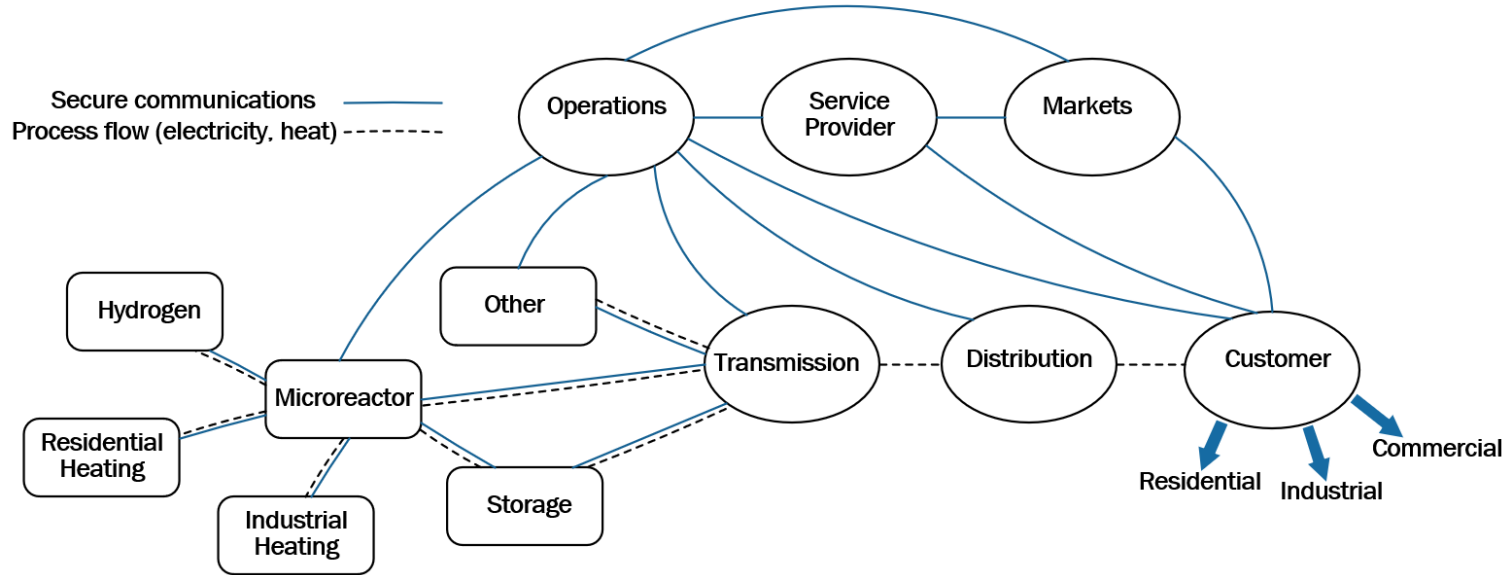
October 2026

UIUC, IL

A Consortium for Strategic Revitalization of Cyber-Physical Nuclear Infrastructure for Advanced Small Modular Reactors



New technologies...new challenges



New reactor concepts =>
Significantly different requirements
than existing fuel cycle facilities

Digitalization => New architectures
and new vulnerabilities

New technologies => Quantum computing
Adversaries now have access to new tools
with unprecedented capabilities

Digital Twin Applications

- DTs already implemented across various industries
 - Aviation, automotive, healthcare, manufacturing, smart cities, maritime industry

Aviation industry example

- DT for predictive maintenance



- Reduce costs 5-10% annually
- Decrease downtime ~20%



Digital Twin Definition

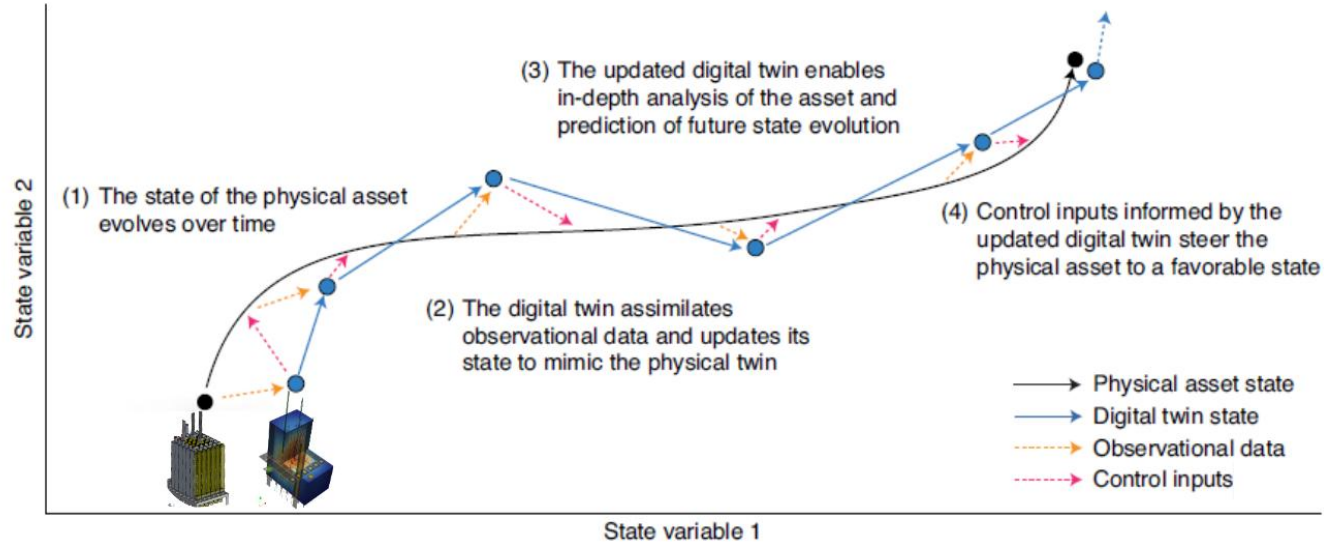
“A digital twin is a computational model (or a set of coupled computational models) that evolves over time to persistently represent the structure, behavior and context of a unique physical asset such as a component, system or process”

(AIAA Digital Engineering Integration Committee, “Digital twin: Definition & value,” *AIAA and AIA Position Paper*, 2020)

“A virtual replica of a physical asset” designed to:

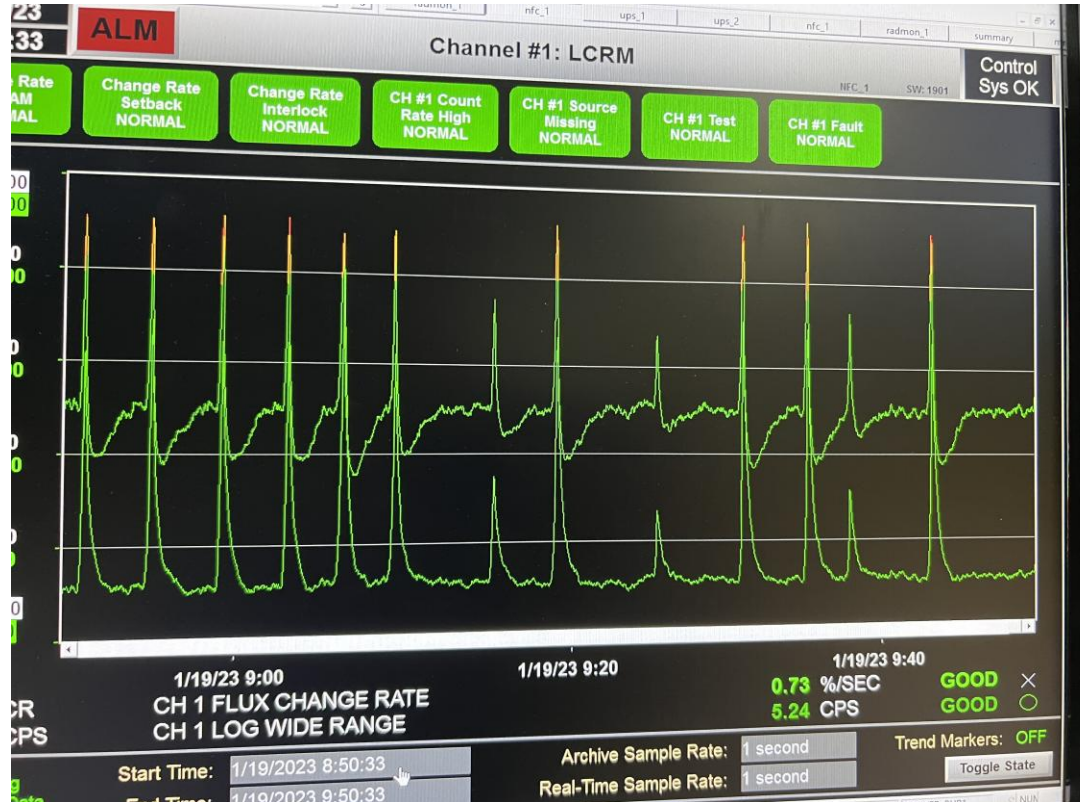
1. *maintain **continuous communication**,*
2. *same **fidelity level***
3. *complete **synchronization***

(Digital Twin Consortium definition, 2024)



What about Cybersecurity?

What about Cybersecurity?



Cybersecurity in the Modern Age



1st Cyber Age



2nd Cyber Age



3rd Cyber Age



“I don’t care what you do,
just keep the plant running!”

- CEO of a large chemical
processing plant on security



4th Cyber Age



Challenges

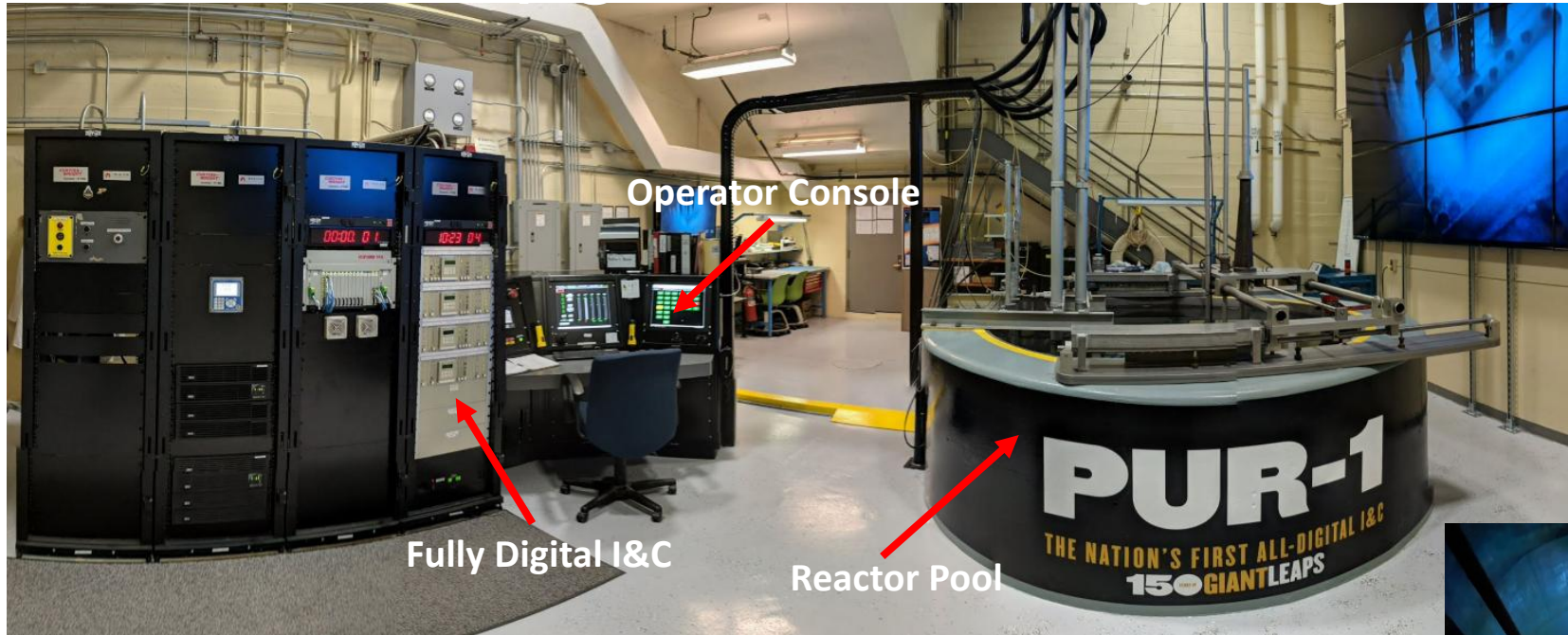
Can we design an unconditional security framework to fulfil cyber requirements?

Can we detect and characterize cyber or other events in real-time?

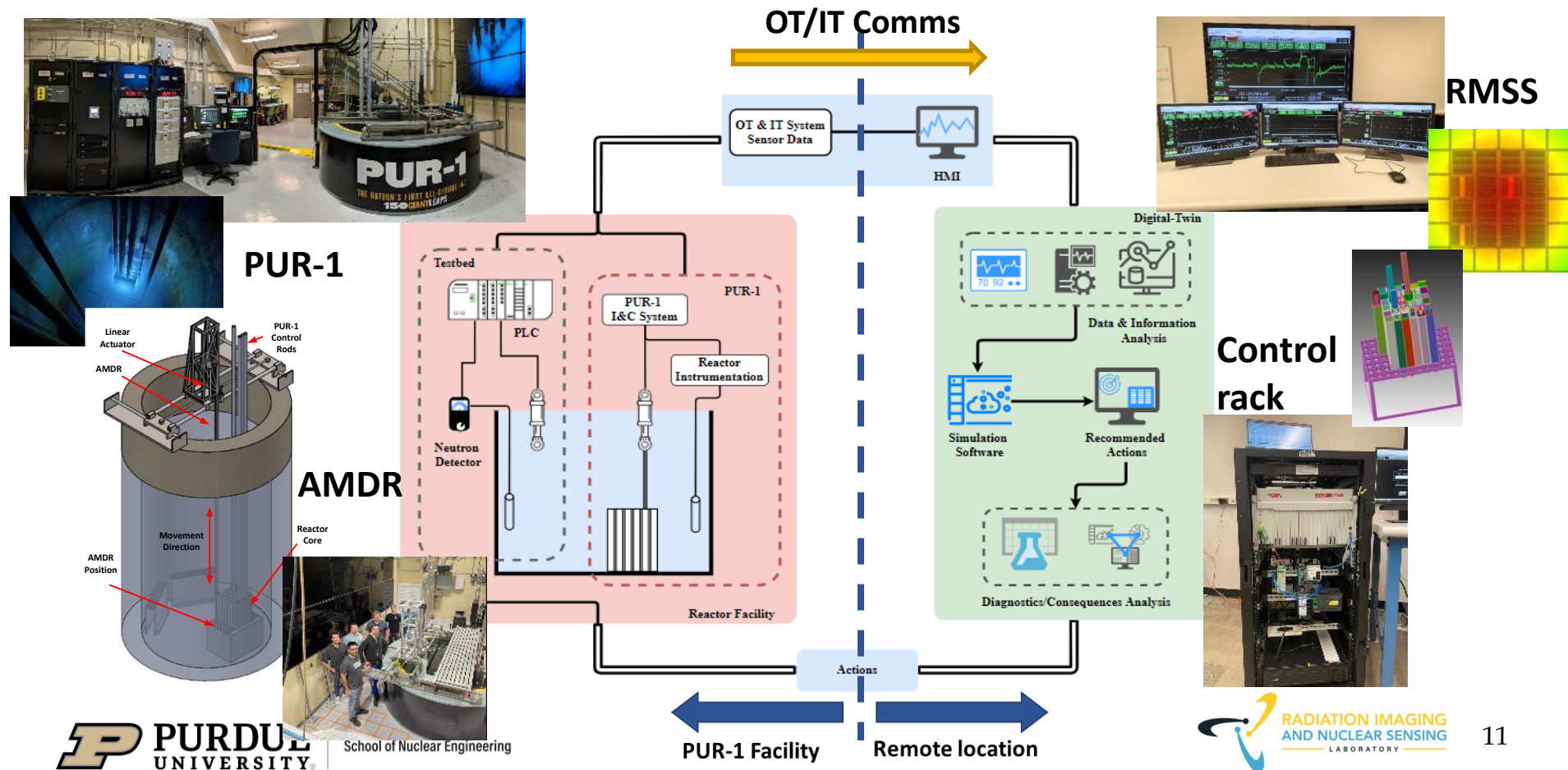
How can we license operators when 10CFR50 requires training on an existing facility?

Is (semi) autonomous control feasible?

PUR-1



Towards a Real-Time Cyber-Physical Digital Twin



Digital/Cyber Remote Station

RTP 3000 TAS N+
Nuclear grade PLC
16 CH AI/AO
32 CH DI/DO

Field
Programmable
Gate Array

Power
distribution
unit

Actuator control

IT Monitoring

R-TIME GUI

Stats:
2000 parameters
1kHz sampling

Real-time diagnostics

Siemens S7 PLC

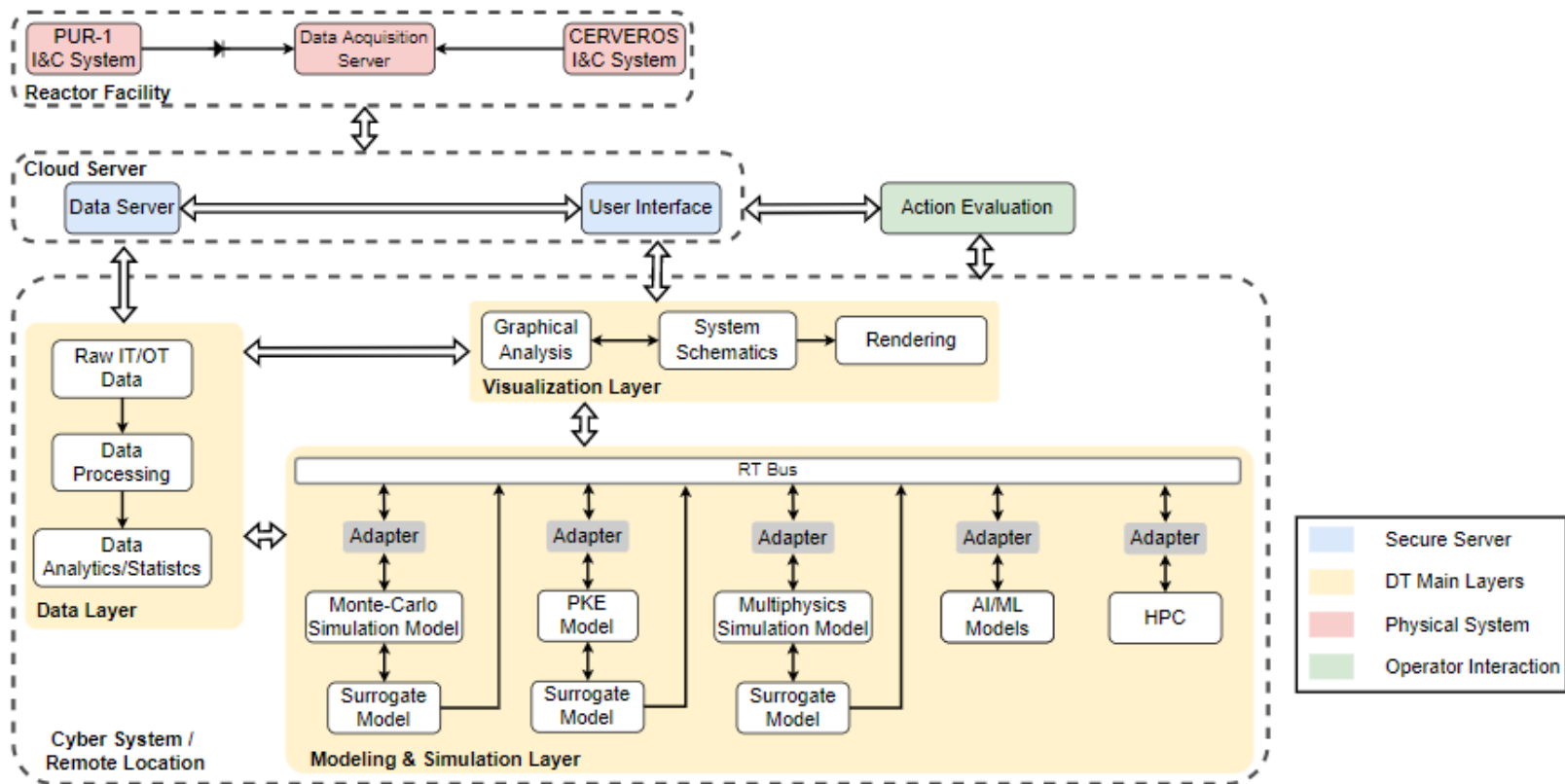
UPS APC/1500

To GPU

Digital/Cyber Remote Station

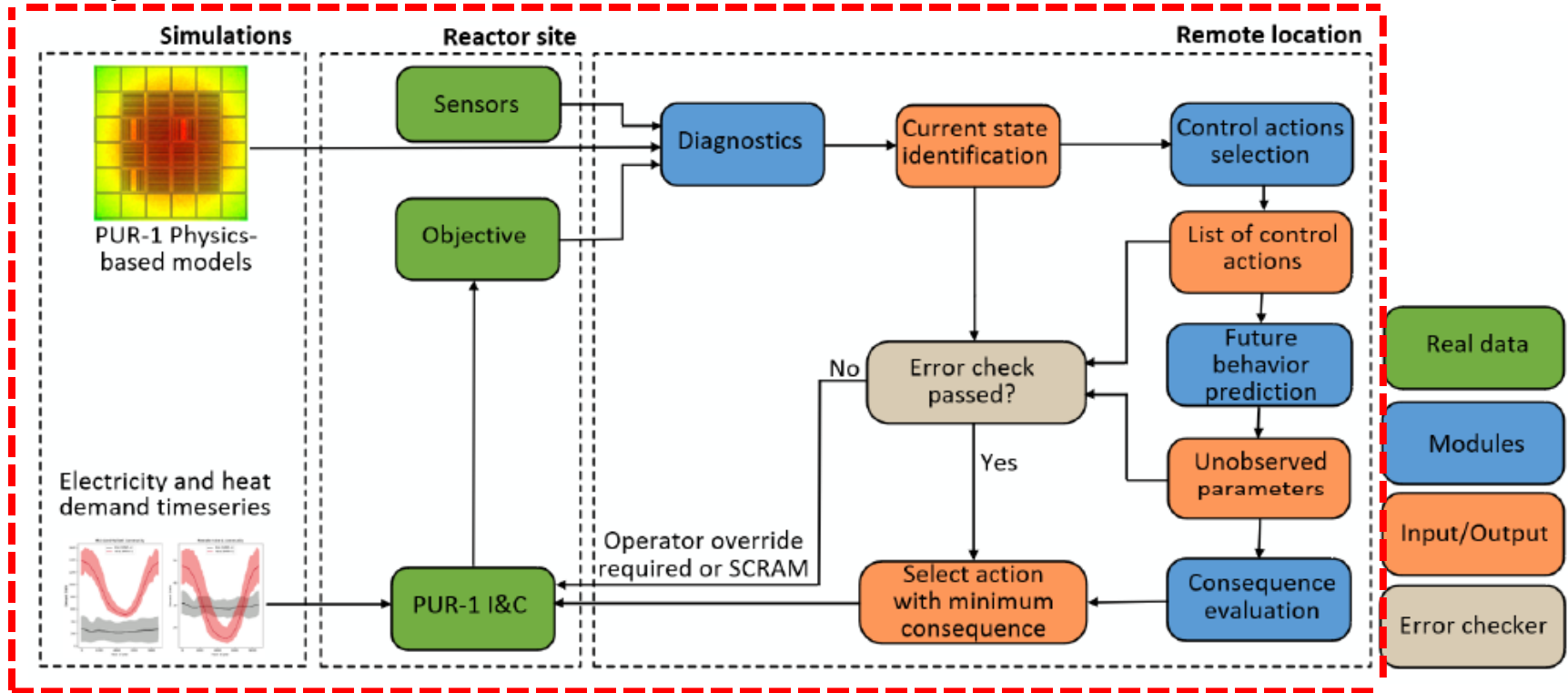


PUR-1 DT Architecture



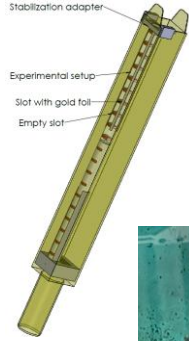
Semi-autonomous Architecture

Completed

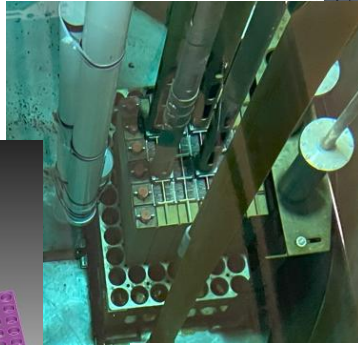
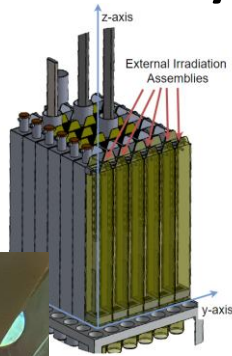
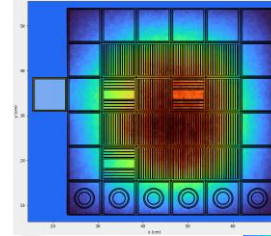
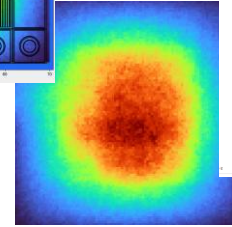
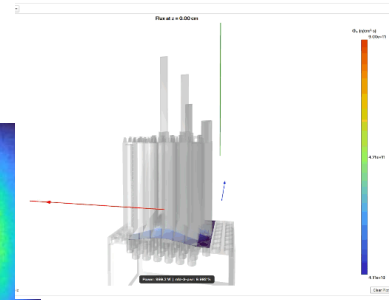


Physics-Based Models

Neutronics + PKE



PUR-1



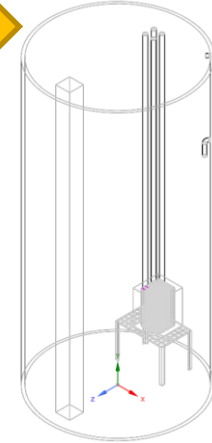
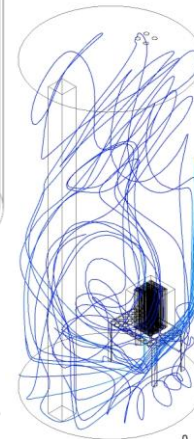
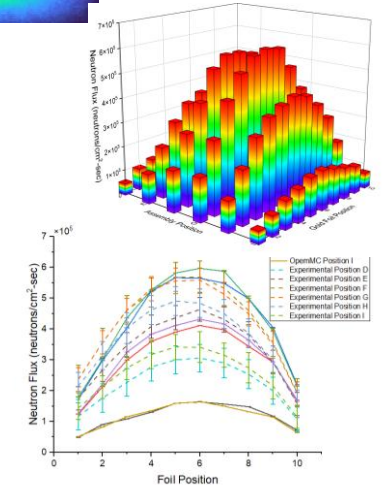
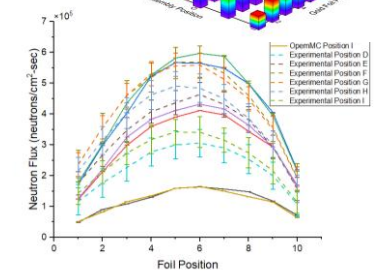
CAD Model

			C1	D1	E1	F1	G1	H1	I1	J1	
			C2	D2	E2 (12)	F2 (13)	G2 (13)	H2 (13)	I2	J2	
			SRC	D3	E3 (SS2)	F3 (12)	G3 (SS1)	H3 (13)	I3	J3	
				C4	D4	E4 (13)	F4 (13)	G4 (12)	H4 (13)	I4	J4
				C5	D5	E5 (RR)	F5 (13)	G5 (FC)	H5 (12)	I5	J5
				C6	D6	E6	F6	G6	H6	I6	J6
				C7	D7	E7	F7	G7	H7	I7	J7

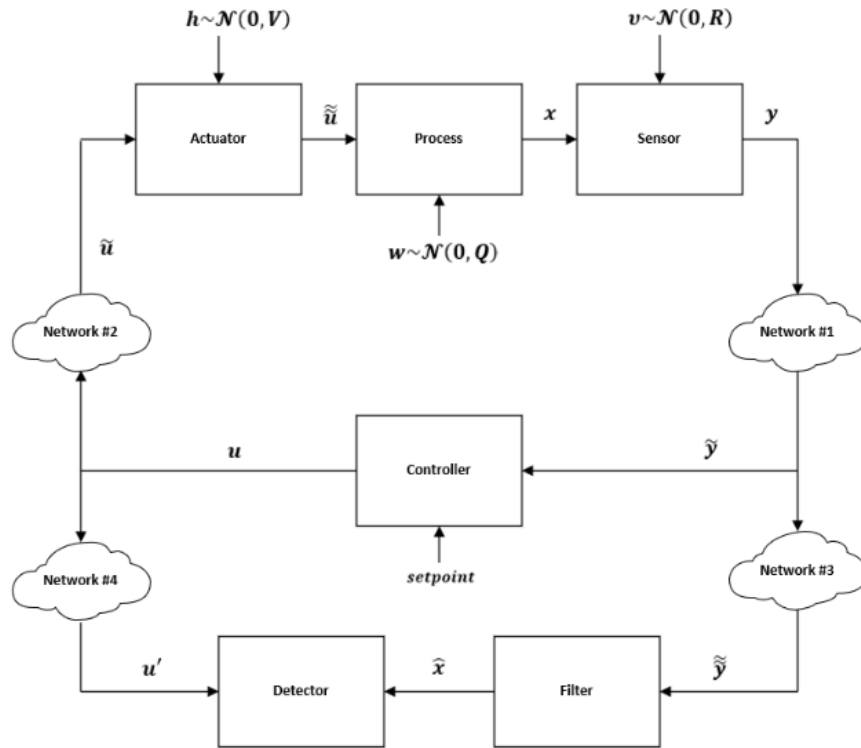
- Fuel Assembly
- Control Assembly
- Graphite Assembly
- Irradiation Assembly
- Neutron Source
- Fission Chamber
- External Irradiation Assembly

Benchmarking data

Thermal-hydraulics



Network Communications Model



Sensor

$$y_k = Cx(t_k) + v_k, v_k \sim \mathcal{N}(0, R)$$

Network

$$\tilde{y}_k = \begin{cases} y_{k-d_k} + h_k^{(net)}, & w.p. 1 - p_l \\ \tilde{y}_{k-1}, & w.p. p_l \text{ (hold)}, \end{cases}$$

Latency

$$t_{net} = t_{ser} + t_{prop} + t_{queue} + \sum t_{dev}$$

Controller

$$u_k = K_p e_k + K_i \sum_{j=0}^k e_j + K_d \frac{e_k - e_{k-1}}{\Delta t}$$

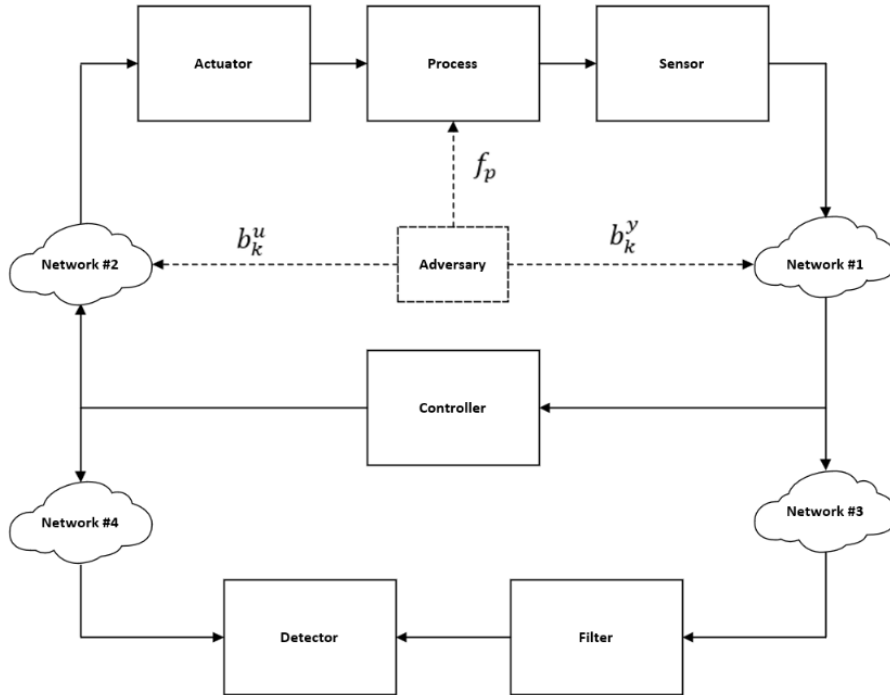
Actuator

$$r_{k+1} = r_k e^{-T_s/\tau_{act}} + \tilde{u}_k^{cmd} (1 - e^{-T_s/\tau_{act}})$$

Command

$$u_k^{cmd} = \text{sat}(r_{k-1} + u_k, r_{min}, r_{max})$$

Network Communications w/ Adversary Model



Fault vector

$$f_k = [f_k^p \ f_k^a \ f_k^c \ f_k^s]^T$$

$$y_k = Cx(t_k) + v_k + F_k^s f_k^s, v_k \sim \mathcal{N}(0, R)$$

Attack vector

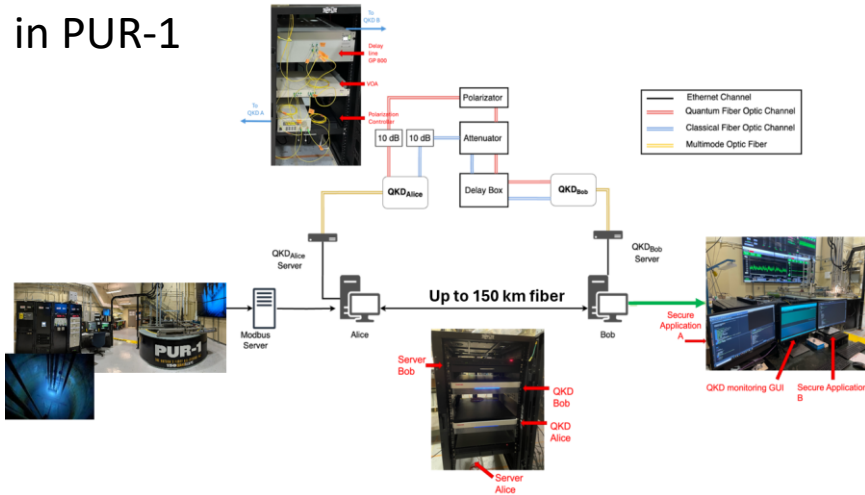
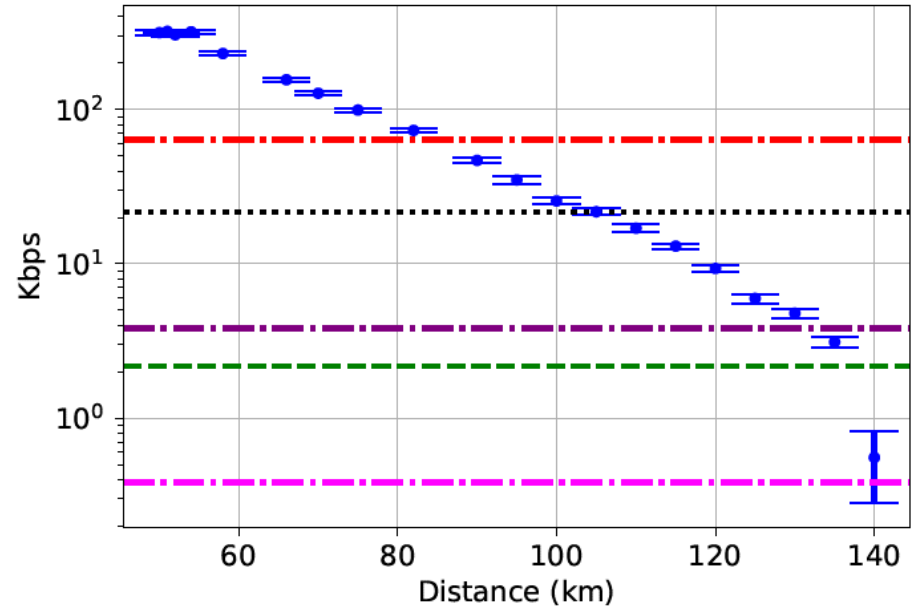
$$a_k = [b_k^y \ b_k^u]^T$$

$$\tilde{y}_k = y_k + \Gamma_k^y b_k^y$$

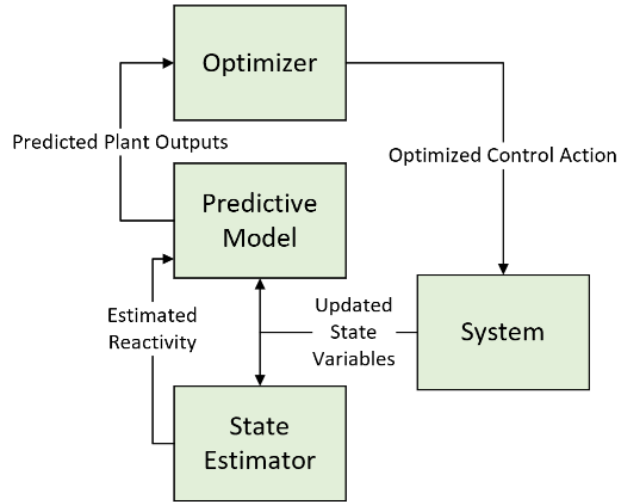
$$\tilde{u}_k = u_k + \Gamma_k^u b_k^u$$

What about Cybersecurity?

QKD Implementation in PUR-1



Model Predictive Reactor Control - Load Following



$$\min_{u_0:N_c} \sum_{k=0}^{N_p} (|\hat{n}_k - n_{set}| + c_1(u_k - u_{k-1})^2 + c_2(\hat{n}_k - \hat{n}_{k-1})^2)$$

$$\text{subject to: } \hat{x}_0 = x_0,$$

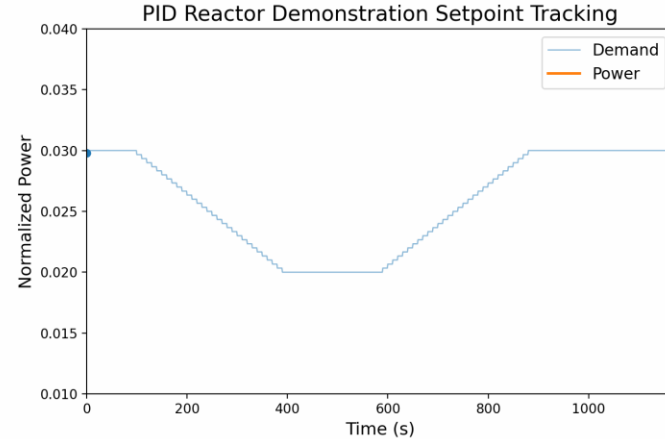
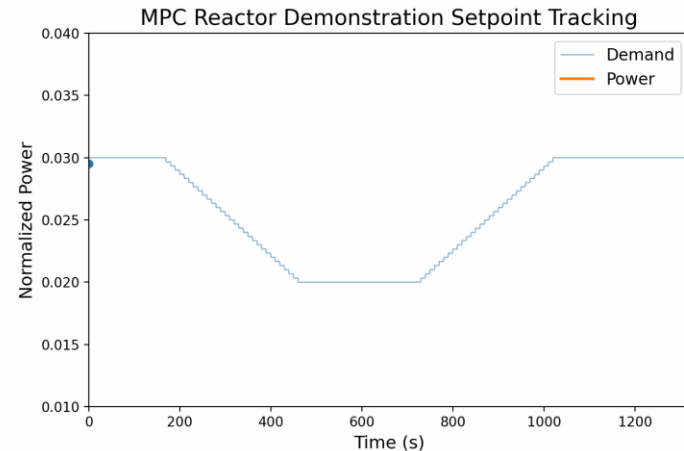
$$n_{lb} \leq \hat{n}_k \leq n_{ub} \forall k = 0, \dots, N_p$$

$$u_{lb} \leq u_k \leq u_{ub} \forall k = 0, \dots, N_p$$

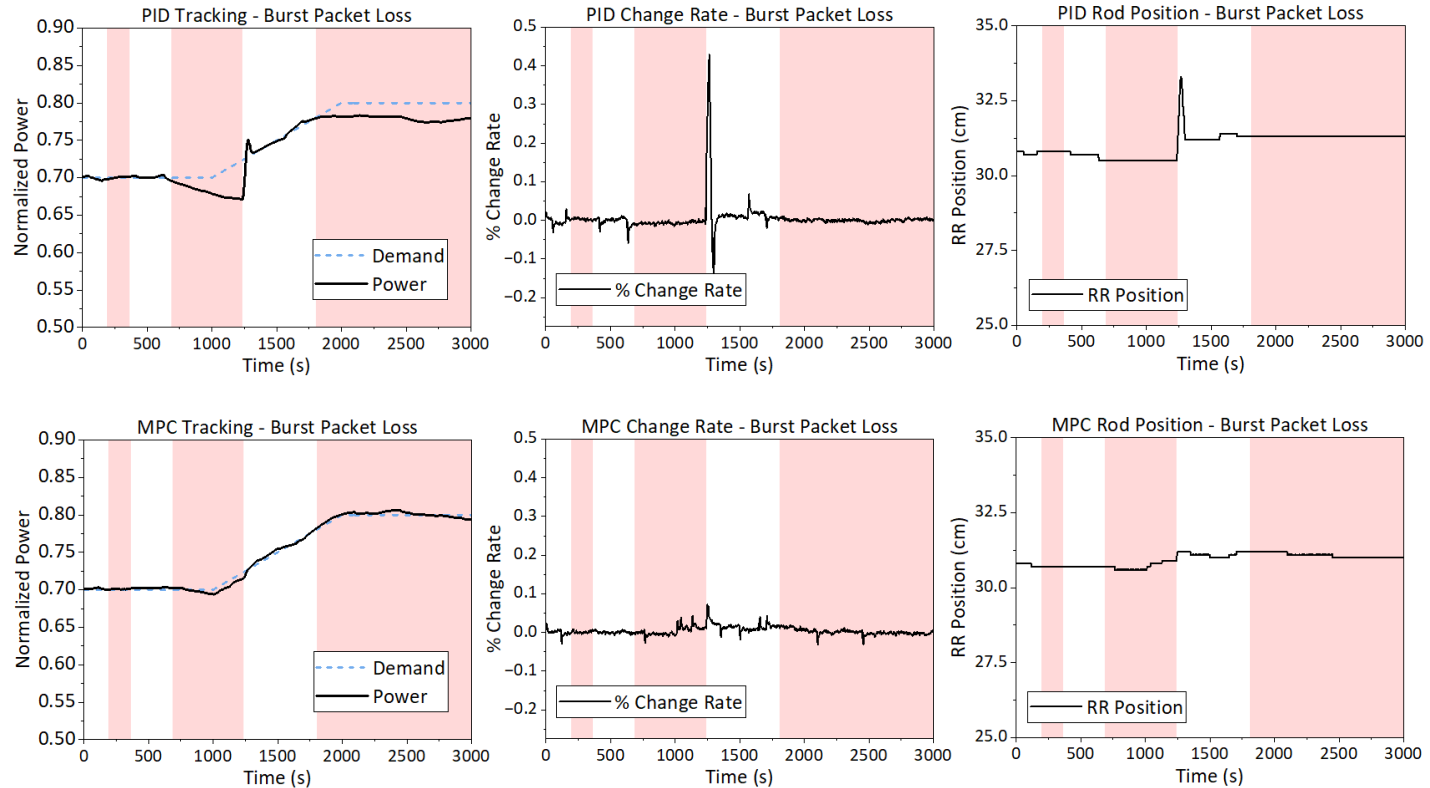
$$|u_k - u_{k-1}| \leq \Delta u_{ub}$$

$$\hat{n}_k - \hat{n}_{k-1} \leq \Delta n_{ub}$$

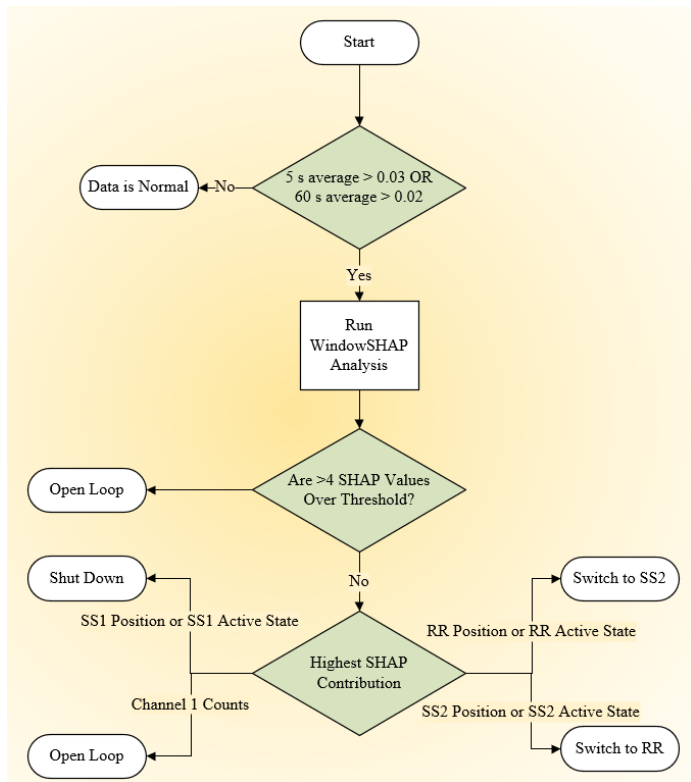
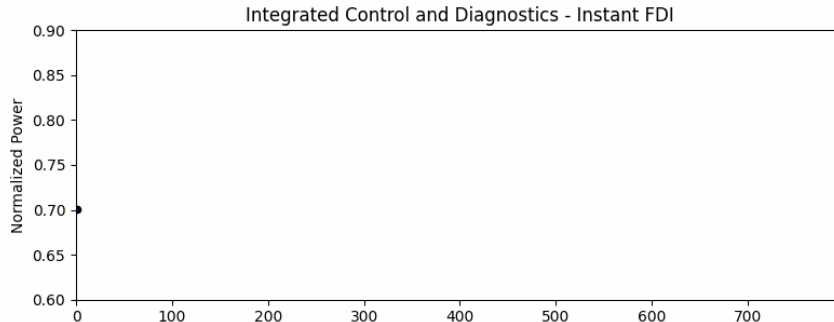
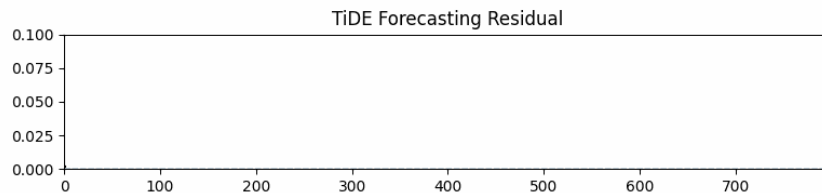
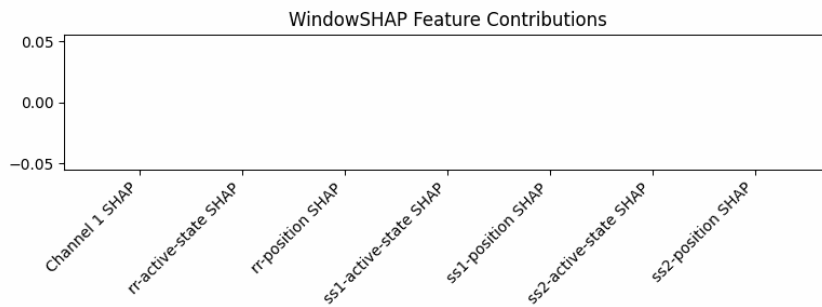
$$\hat{x}_{k+1} = f(\hat{x}_k, u_k), \forall k = 0, \dots, N_p$$



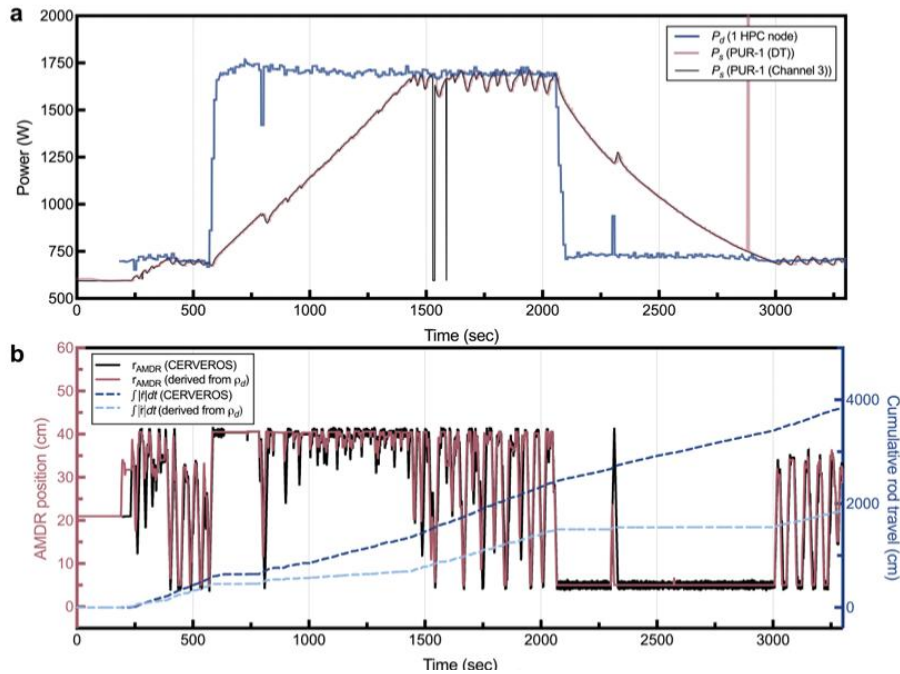
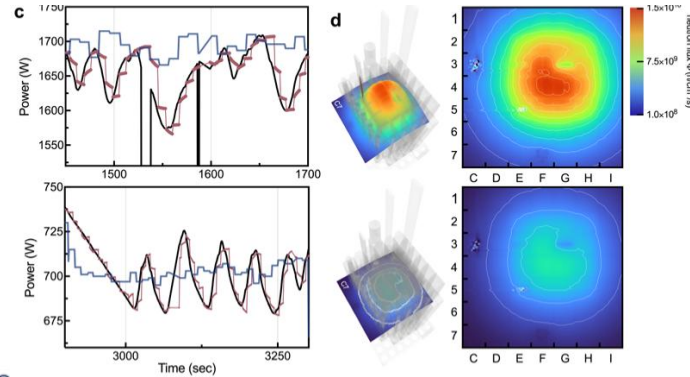
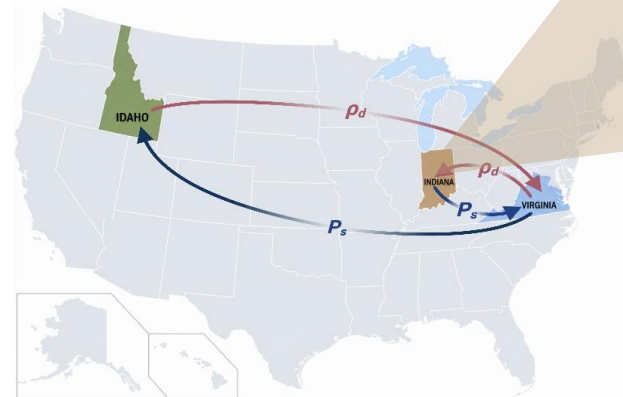
Model Predictive Reactor Control – Packet Loss



Integrated Diagnostics w/ Localization - FDI

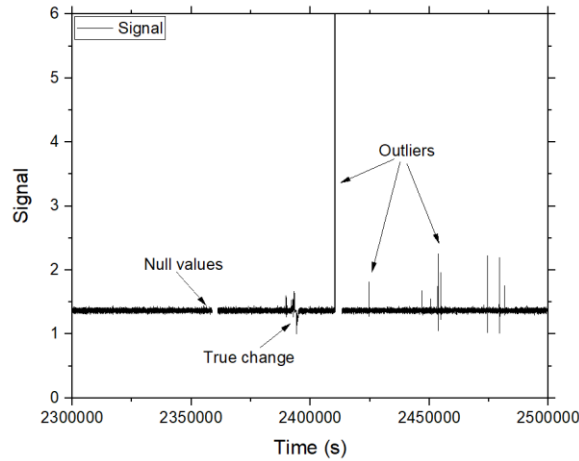
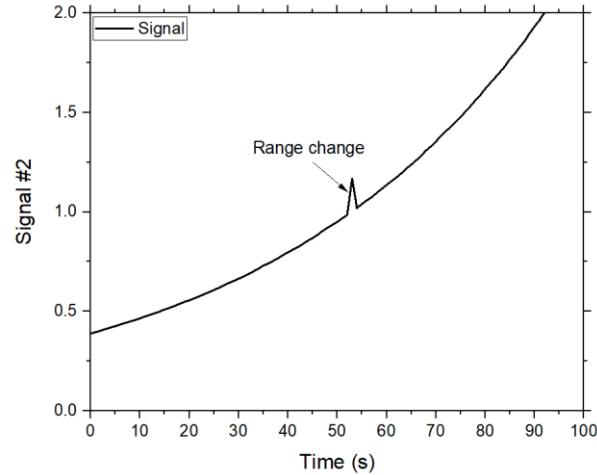
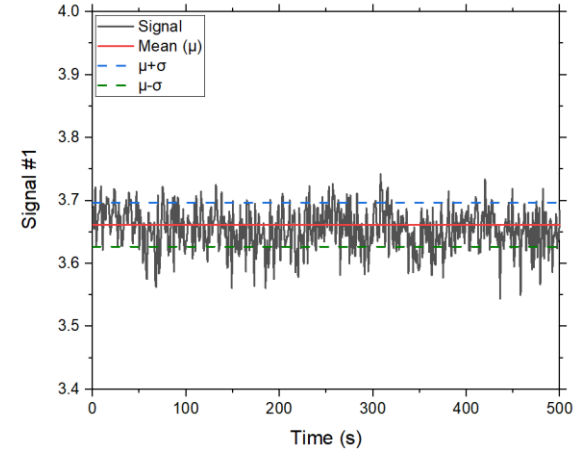


Real-Time Load Following of AI GPU Training

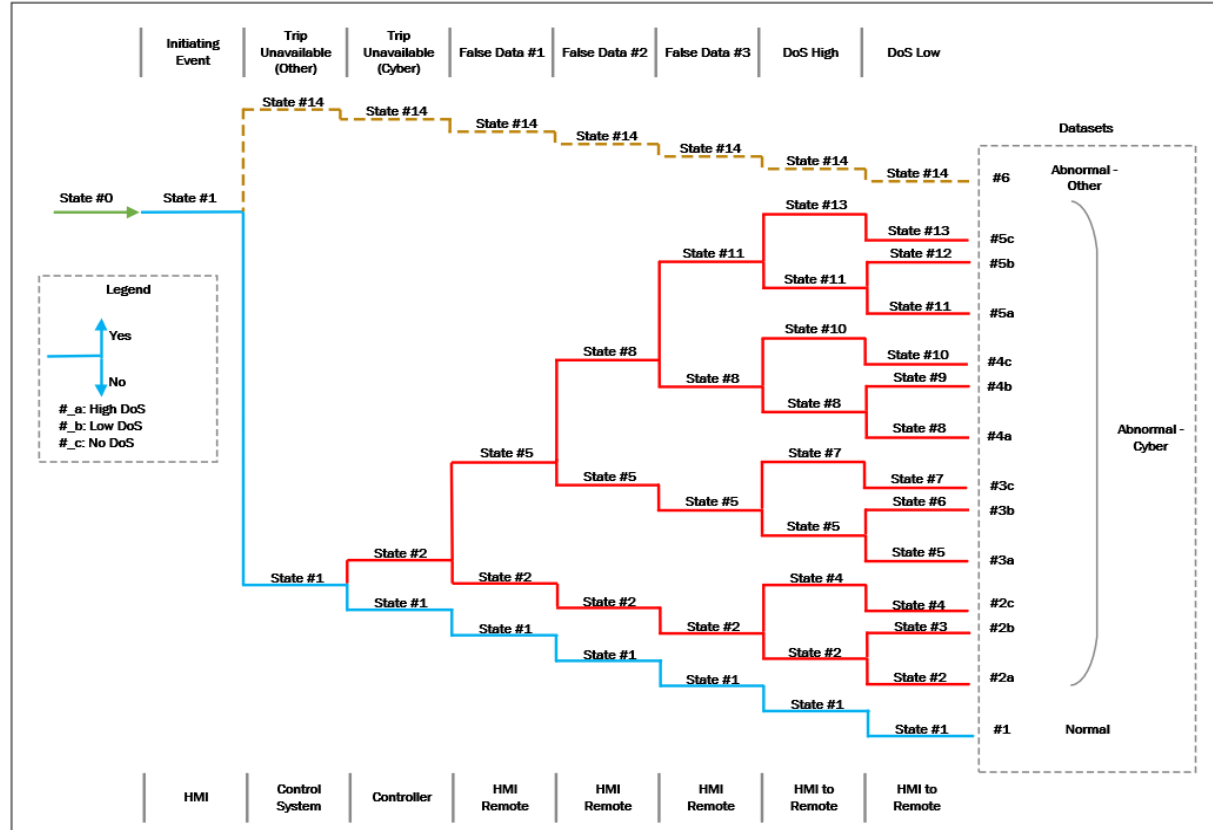


*INL/LDRD funded project. Team: Matthew Anderson, Jonah Lau, Michael Mendes, Vasileios Theos, Stylianos Chatzidakis, April Novak, Matthew Sgambati, Tim Grunloh

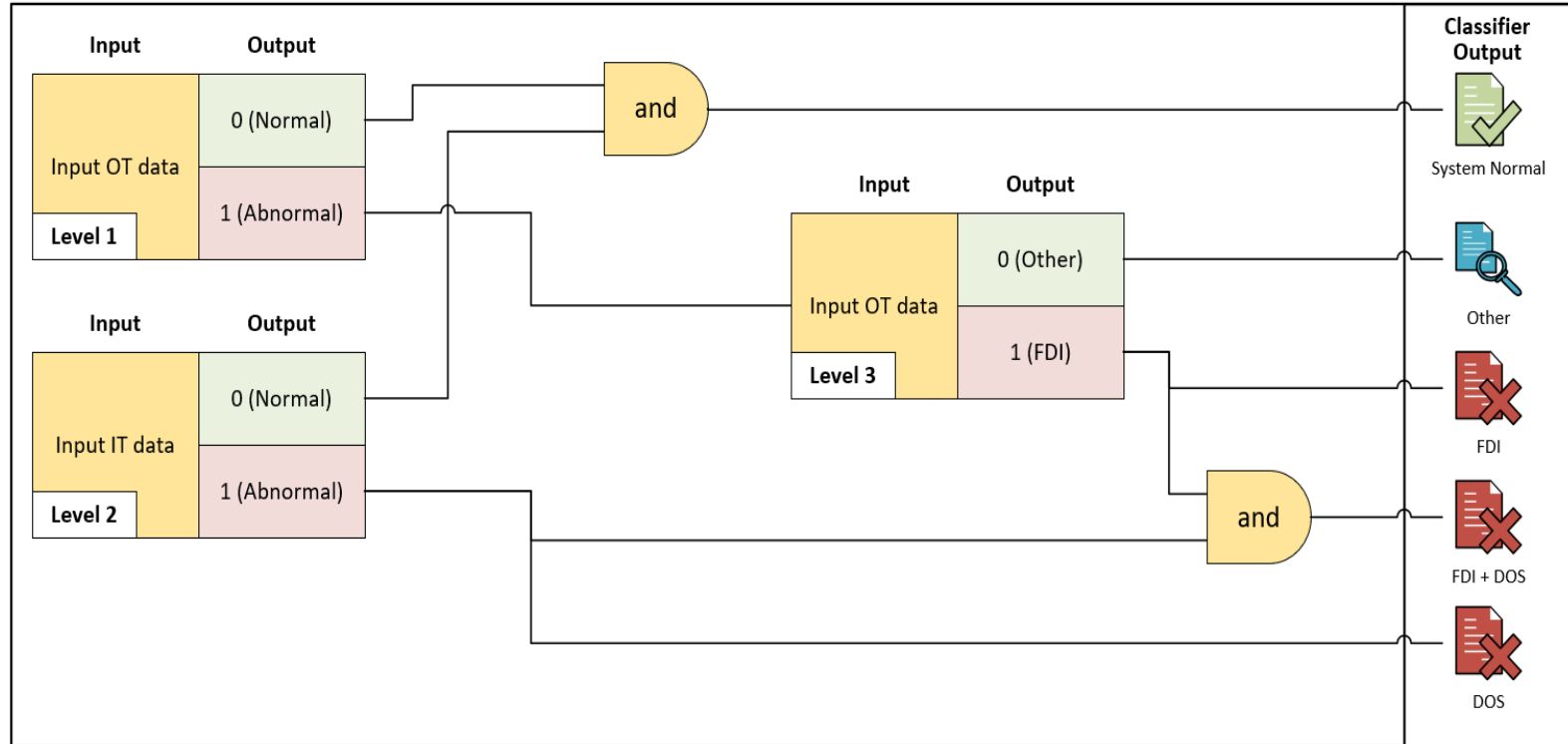
Lesson learned #1 : Data Artifacts



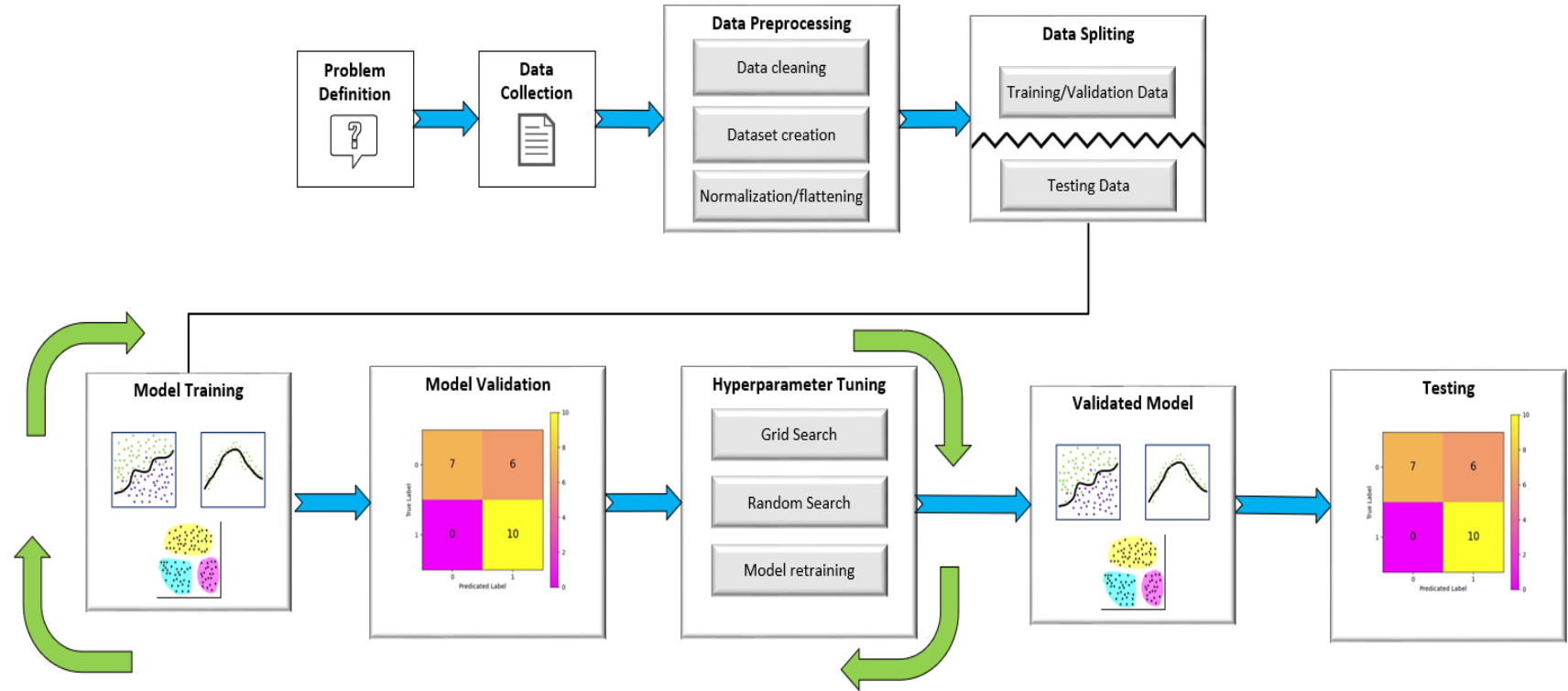
Lesson Learned #2: Use Case Progression



Lesson Learned #3: AI/ML Architecture



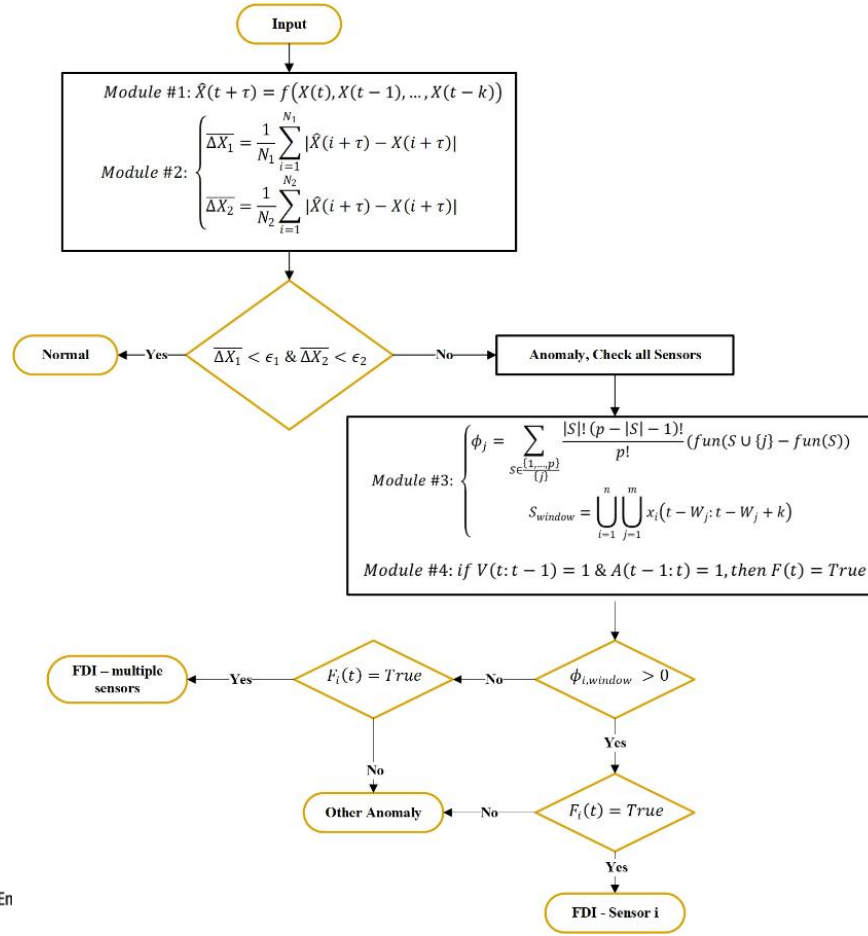
Lesson Learned #4: AI/ML Training and Testing



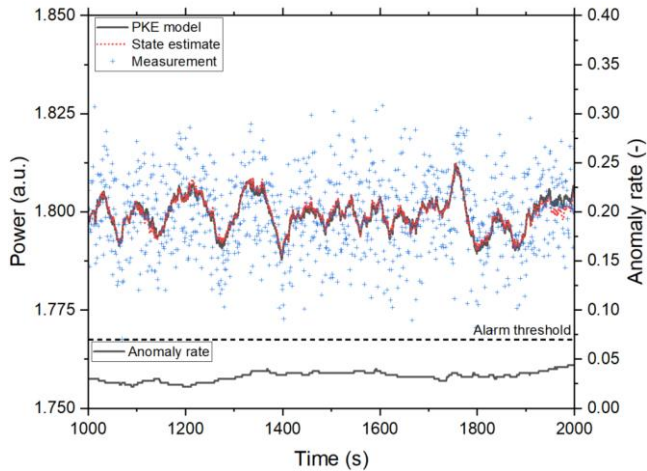
Example: Hyperparameter Tuning

Independent Variable	Description	Values
Window Length	Number of timesteps fed as input to the model	All levels: 1, 5, 10, 20, 30
Window Step	Number of points between consecutive time windows	All levels: 1, 3, 5, 7, 10
Training Balance Ratio	Ratio of normal to abnormal data during training	Level 1: 1, 3, 5, 10, 20, 30 Level 2: 1, 3, 5, 10, 20, 30 Level 3: 0.33
Validation Balance Ratio	Ratio of normal to abnormal data during validation	Level 1: 30 Level 2: 30 Level 3: 0.33
Testing Balance Ratio	Ratio of normal to abnormal data during testing	Level 1: 30 Level 2: 30 Level 3: 0.33
Scaling	Type of normalization applied to data	All levels: Min-max, standard
Algorithm	Type of AI/ML algorithm used	All levels: Decision Tree, Random Forest, Logistic Regression, Linear SVM, Naïve Bayes
Train/Validation/Test Split	Proportion of dataset saved for testing	All levels: 50/30/20, 60/20/20, 70/10/20

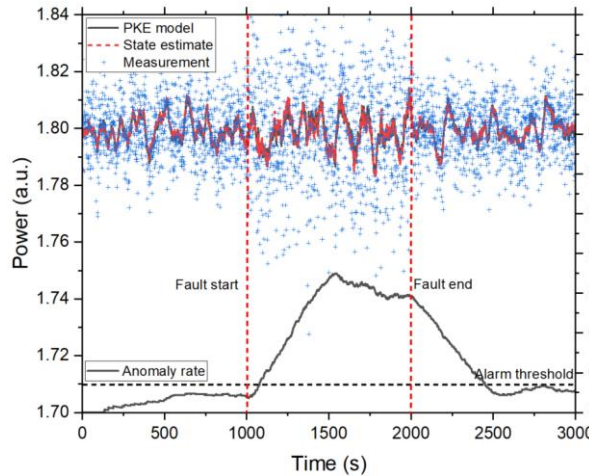
Lesson Learned #5: Interpretability



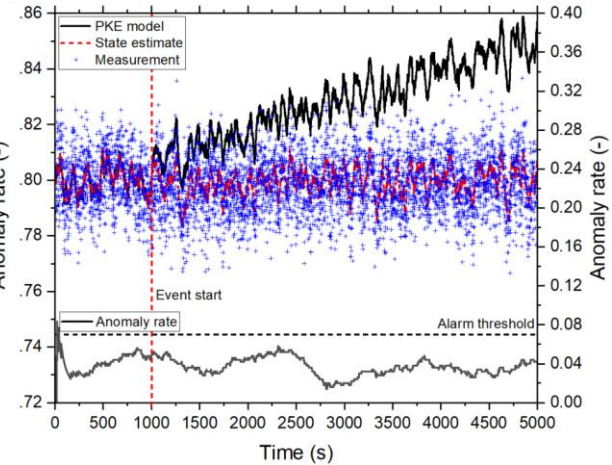
Performance Evaluation – Bias Injection Attack



Normal conditions



Increased sensor noise



Low level bias injection

Challenges

Can we design an unconditional security framework to fulfil cyber requirements?

Can we detect and characterize cyber or other events in real-time?

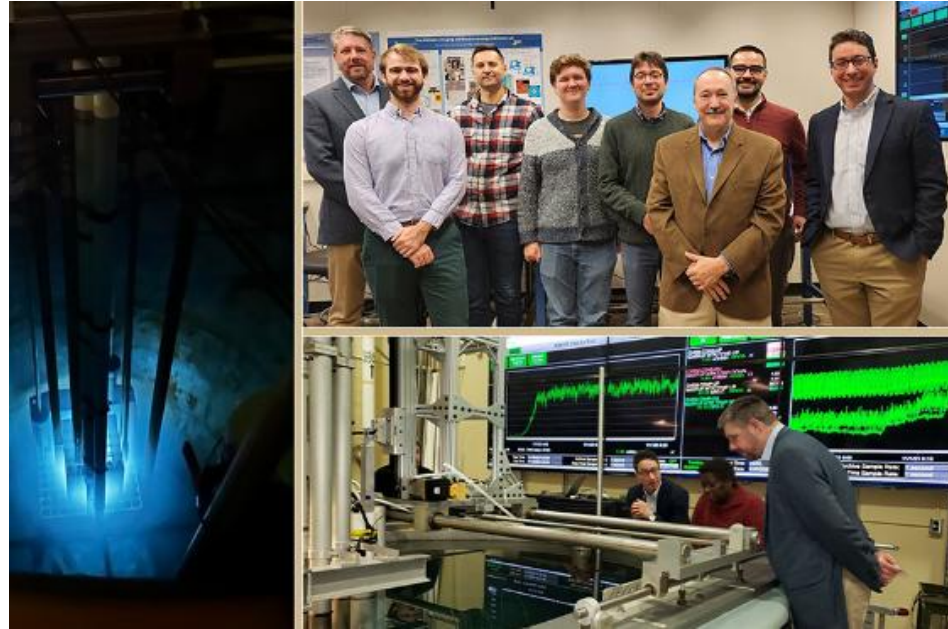
How can we license operators when 10CFR50 requires training on an existing facility?

Is (semi) autonomous control feasible?

Acknowledgements

This research is being performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Programs and the Nuclear Regulatory Commission.

We also thank Ben Cipiti and Katya Le Blanc at Sandia, John Jackson and Vivek Agarwal at INL, and Doug Eskins, Anya Kim, and Kaitlyn Cottrell from the NRC for fruitful discussions and expert input.





Questions?