CS637A: Embedded and Cyber Physical Systems

Design and Deployment of Resilient Control Execution Patterns

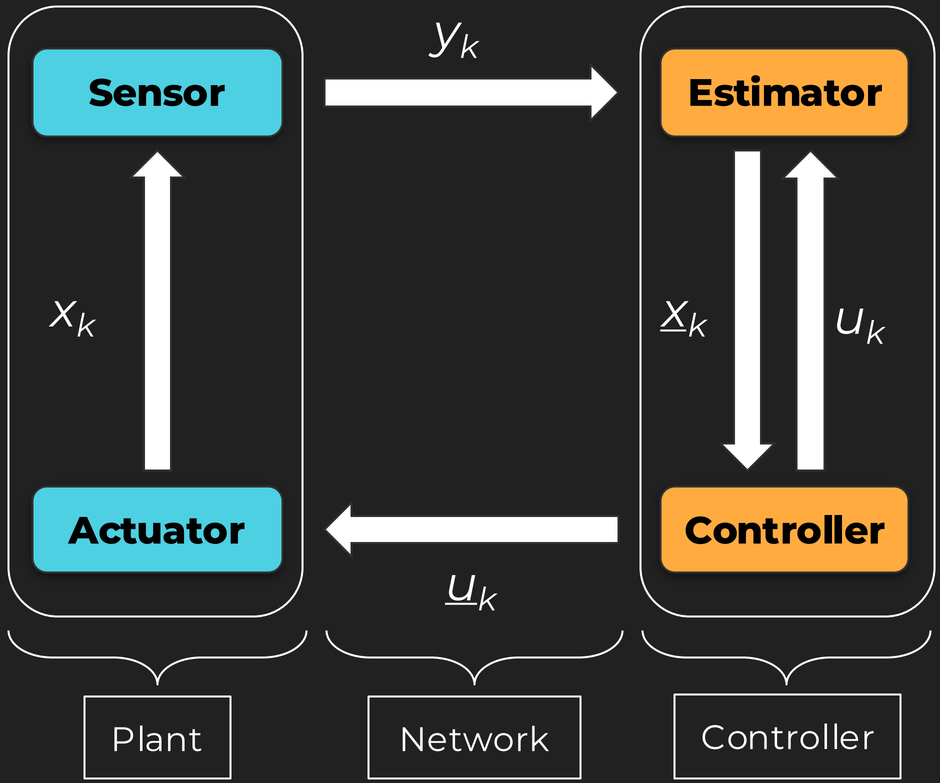
A Prediction, Mitigation Approach

Ipsita Koley , Sunandan Adhikary , Arkaprava Sain, Soumyajit Dey

(Presented in the Proceedings of ICCPS 2023)

Aravind Seshadri 210180	*Kapu Nirmal Joshua 210489*	*Sumay Avi 211071*	*Tanmay Purohit 211097*	*Raghav Shukla 210800*

Modern Cyber Physical System Architecture



Network connected CPS are designed as a closed loop systems

$$x_{k+1} = A\,x_k + B\,u_k + w_{k+1}$$

$$y_k = C\,x_k + v_k$$

$$y^{pred}_k = C(A\,x^{pred}_k + B\,u_k)$$

$$r_{k+1} = y_{k+1} - y^{pred}_{k+1}$$

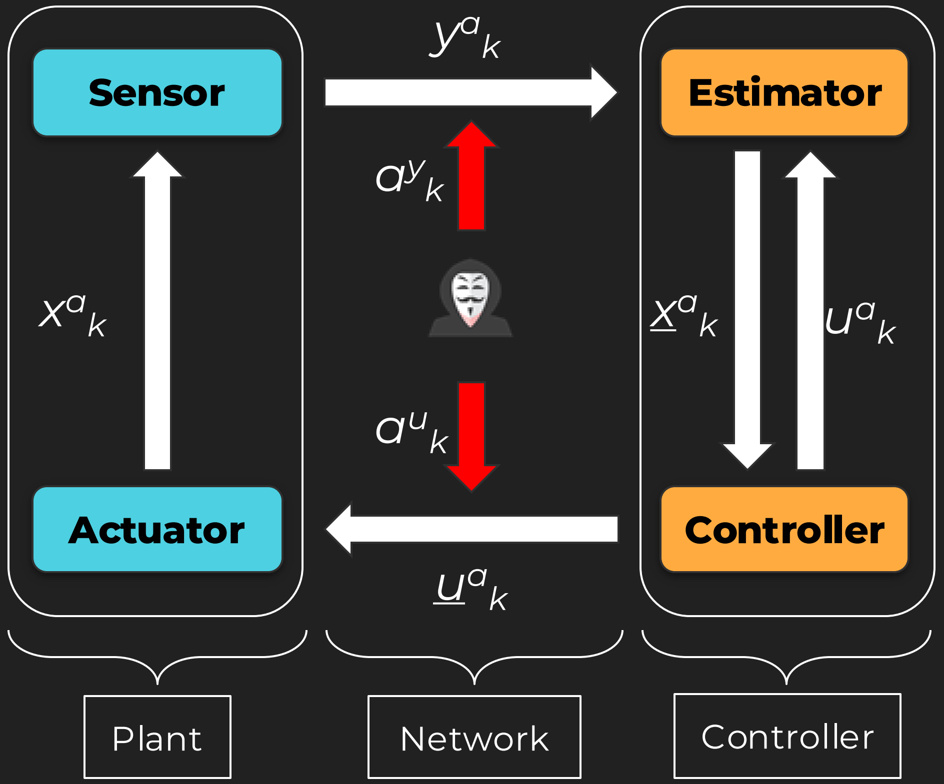
$$x^{pred}_{k+1} = A\,x^{pred}_k + B\,u_k + L\,r_k$$

$$u_k = -\,K\,x^{pred}_k$$

$$e_k = x_k - x^{pred}_k$$

Where can an attack/hack occur in such a system?

Modern Cyber Physical System Architecture



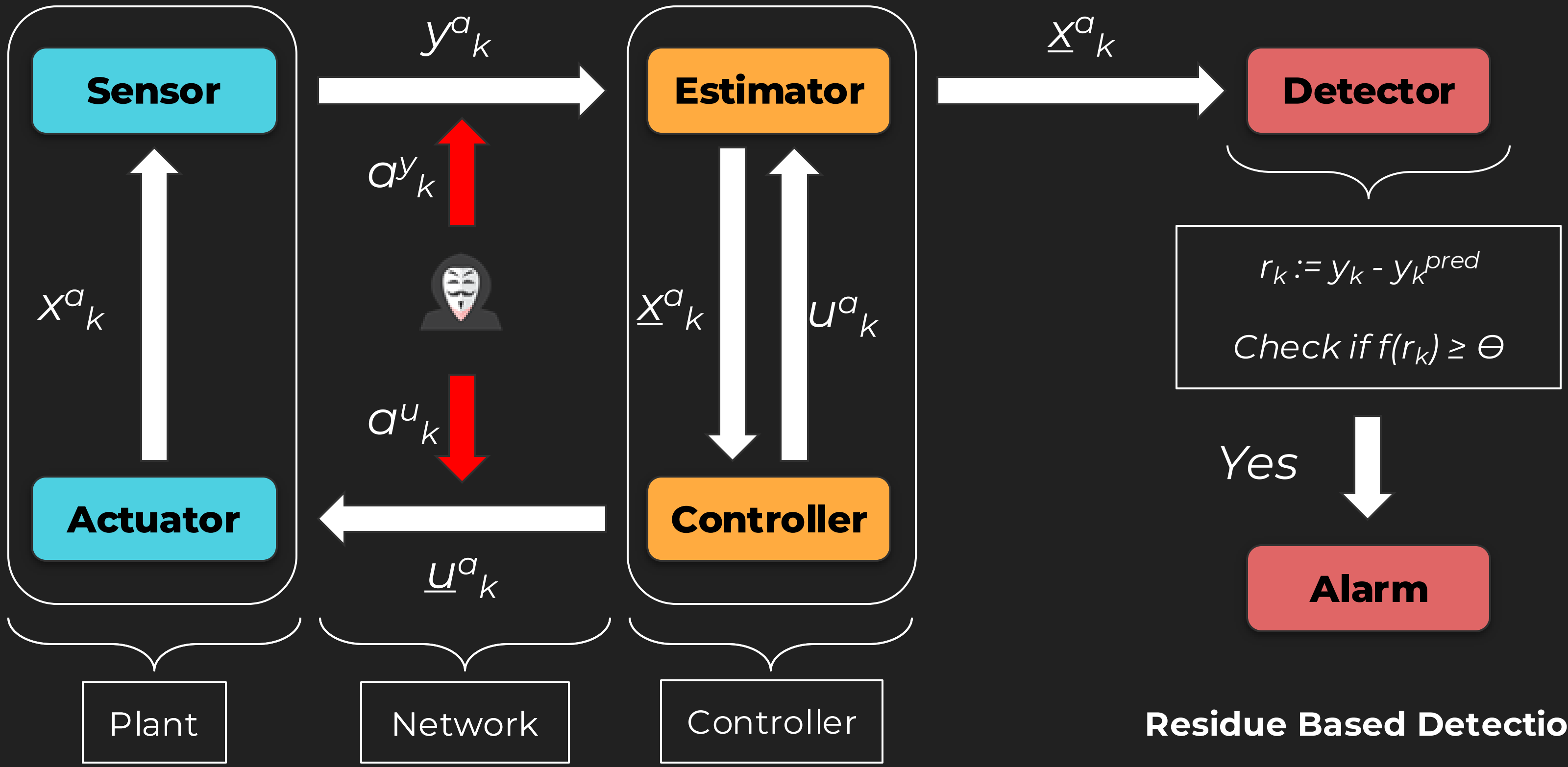
Network connected CPS are designed as a closed loop systems

Where can an attack/hack occur in such a system?

The Network Layer (*u* and *y*)

Attacks occur through False Data Injection (FDI) in both u and y

“Secure” Cyber Physical System Architecture



Residue Based Detection

Stealthy FDI Attacks - The Problem

Attack Vector | Attack Vector on u | Attack Vector on y

$$a = [a_1\ a_2\ a_3 \dots a_L]^T = [\ [a^u_1\ a^u_2 .. a^u_L]\ ,\ [a^y_1\ a^y_2 .. a^y_L]\ ]$$

Stealthy *if:*

$f(r_k) \leq \Theta \ \ \forall\ k = 1,2,3 \dots L$

where Θ is our safety threshold

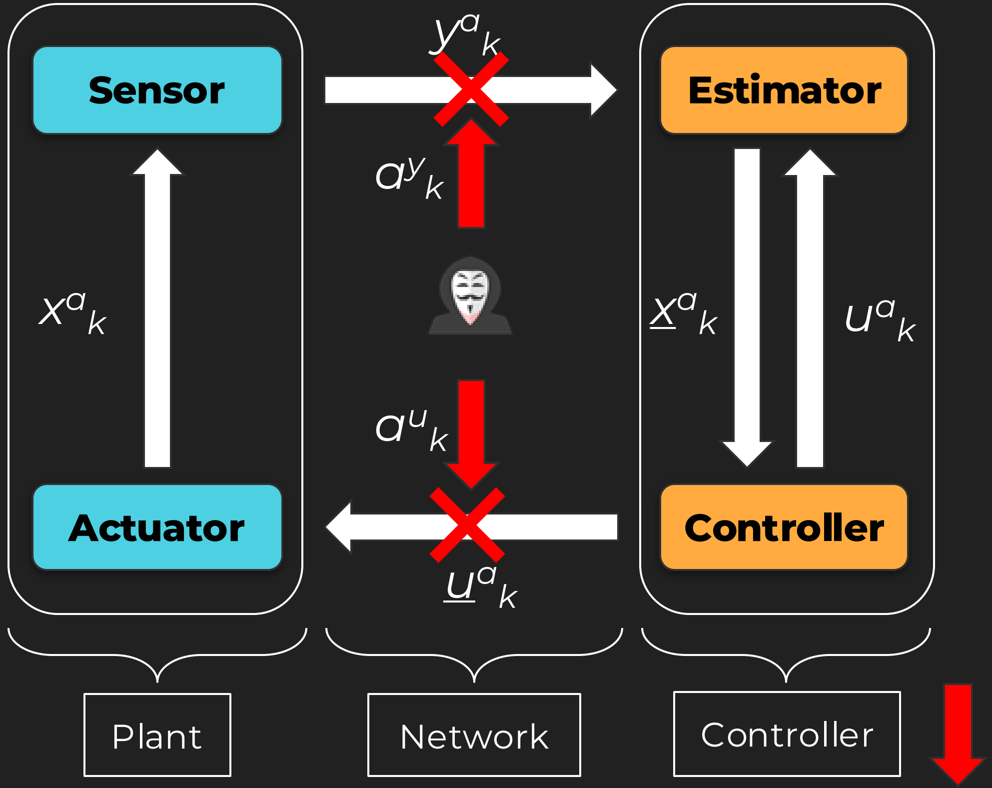
Successful *if:*

$\exists\ k \in \{1,2,3..L\}$ such that $x_k \notin X_s$

where X_s is our safety envelope

A **stealthy** and **successful** FDI attack could still damage our CPS!

Control Execution Skips - A Potential Solution



In a **control execution skip** at iteration *k*:

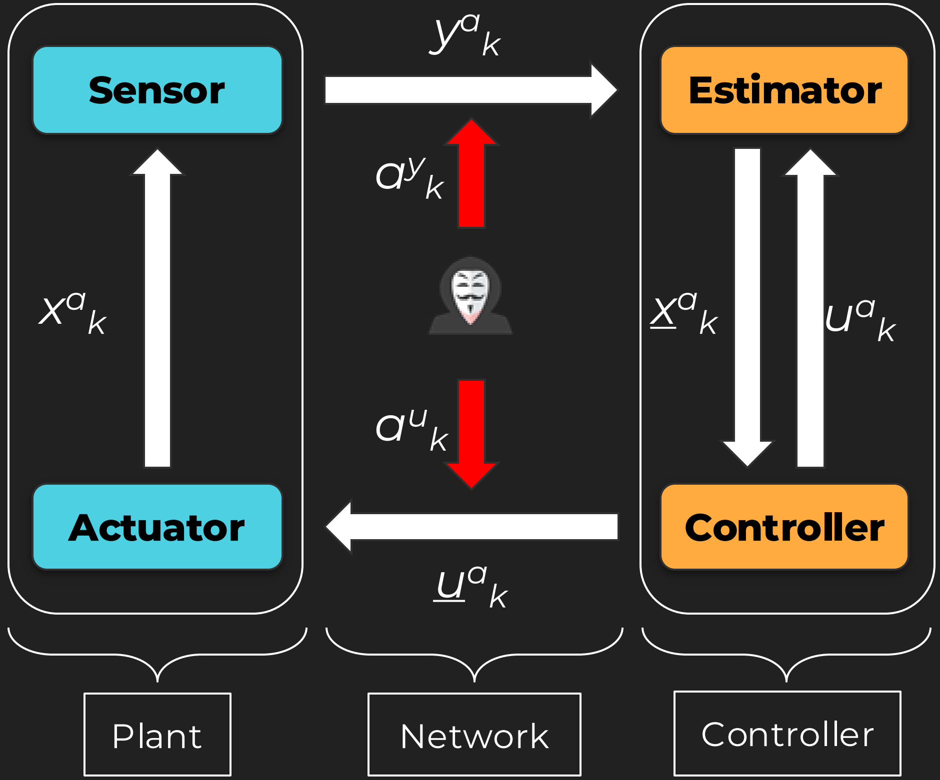
1. Block y_k from going to the estimator
2. Stop u_k from being recalculated

Control execution skips can lead to degraded controller performance

How do we use this paradigm to increase resilience to stealthy FDI attacks?

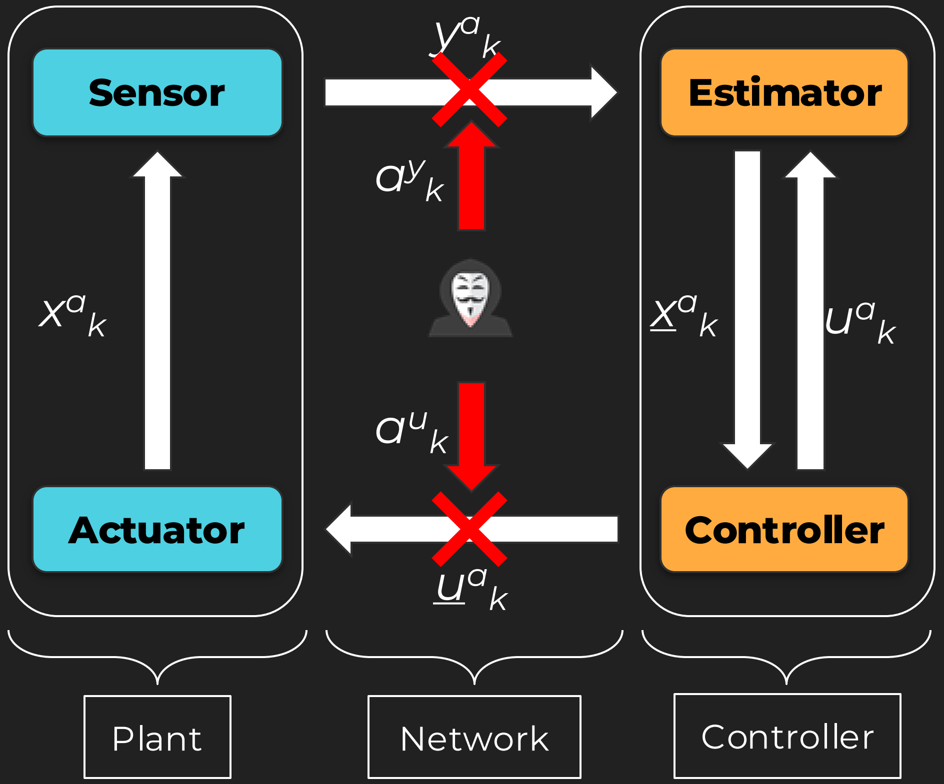
Control Execution Patterns

Let *1* := control line is open
And let *0* := control line is closed



$$\underbrace{1 \quad 1 \quad 1 \quad \cdots \quad 1}_{k \text{ executions}}$$

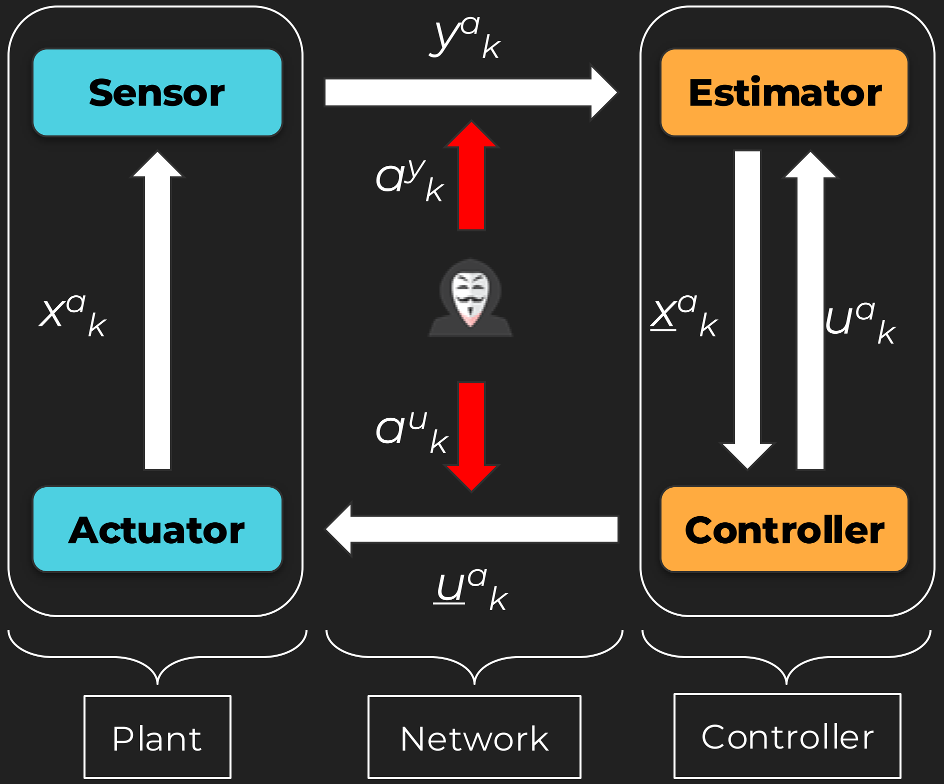
Control Execution Patterns



Let *1* := control line is open
And let *0* := control line is closed

$$\underbrace{1 \quad 1 \quad 1 \quad \cdots \quad 1}_{k \text{ executions}} \quad \underbrace{0 \quad 0 \quad \cdots \quad 0}_{l \text{ skips}}$$

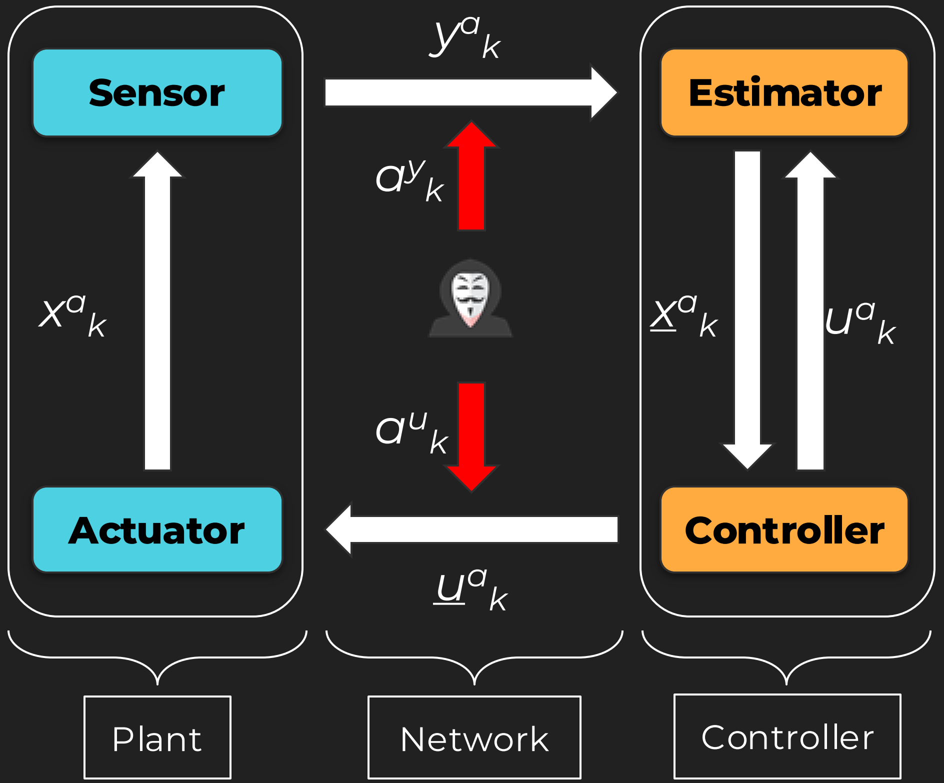
Control Execution Patterns



Let *1* := control line is open
And let *0* := control line is closed

$$\underbrace{1\quad 1\quad 1\quad \cdots\quad 1}_{k\ \text{executions}}\quad \underbrace{0\quad 0\quad \cdots\quad 0}_{l\ \text{skips}}\quad \underbrace{1\quad 1}_{(t\text{-}l\text{-}k)\ \text{executions}}$$

Control Execution Patterns



Let *1* := control line is open
And let *0* := control line is closed

$$\underbrace{1\ \ 1\ \ 1\ \cdots\ 1}_{k\ executions}\ \underbrace{0\ \ 0\ \cdots\ 0}_{l\ skips}\ \underbrace{1\ \ 1}_{(t-l-k)\ executions}$$

A t length periodic sequence such that:

1. $\pi[k] \in \{1,0\}^t$
1. $\pi[k] = \pi[k+t] = \mathcal{S}[k \bmod t]$

is called as a control sequence of length t

<u>*Notation Used:*</u> $1^k\ 0^l\ 1^{t-l-k}$

Proposed Approach: Overview Of Underlying Ideas

Control Execution Skip Patterns

Step 1: What is a control skip pattern?

Step 2: What is the criteria under which a control skip pattern will **help** us the most?
Derive criteria

Enhance Resilience

Compromise Performance

Step 3: Generate control execution sequences using above criteria through a Dynamic Programming approach

Attack Generation

Objective: Model an FDI Attacker

a) optimal FDI attack sequences
b) consumes minimum time

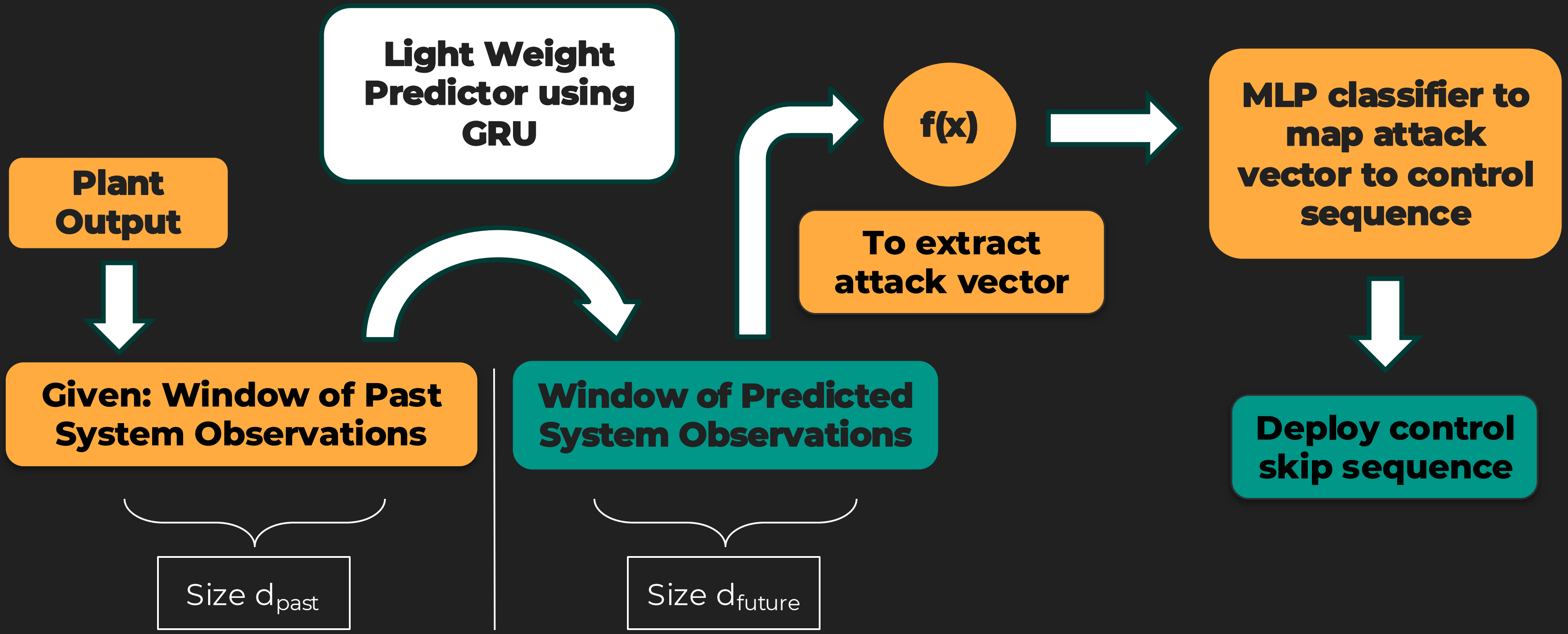
HOW

?

We generate a library of attack vectors

Formulate a constraint solving problem given specifications of CPS , initial conditions safe operating regions

Proposed Approach: Overview Of Underlying Ideas



Favorable Subpattern Synthesis

Feasibility of Aperiodic Control

Periodic control

$$\underbrace{1\quad 1\quad 1\quad \cdots\quad 1}_{t\ \textit{executions}}$$

Aperiodic Control

$$\underbrace{1\quad 1\quad 1\quad \cdots\quad 1}_{k\ \textit{executions}}\ \underbrace{0\quad 0\quad \cdots\quad 0}_{l\ \textit{skips}}\ \underbrace{1\quad 1}_{(t\text{-}l\text{-}k)\ \textit{executions}}$$

- Control Performance: In a window of t samples, to ensure the desired performance despite skipping, the control execution must maintain the minimum rate $r_{\min}$

  $\Rightarrow$ Controller must be executed $\lceil t \times r_{\min} \rceil$ times.

  $\therefore t - l \geq \lceil t \times r_{\min} \rceil$

Control Execution Pattern

Attack Induced Estimation Error:

$$\Delta e_k = e_k^a - e_k \qquad\qquad e_k = x_k - \hat{x}_k$$

$$\Delta r_k = r_k^a - r_k \qquad\qquad r = y_k - \hat{y}_k$$

Periodic Control (No skip)	Sole skip at (k+1)th step	Skipping from (k+1) to (k+l) steps
$\Delta e_k^p = \sum_{i=0}^{k-1} A^i (Ba_{k-1-i}^u - L\Delta r_{k-i})$	$\Delta e_{k+1}^{ap} = A\Delta e_k + Ba_{k-1}^u$	$\Delta e_{k+l}^{ap} = A^l \Delta e_k + \sum_{i=0}^{l-1} A^i B a_{k-1}^u$
$\Delta r_k = CA\Delta e_k + CBa_k^u + a_{k+1}^y$	$\Delta r_k = 0$	$\Delta r_{k+l} = 0$

$(\Delta e_0 = 0)$

Applicability of Aperiodic Control

- Theorem: Assuming control performance criteria is satisfied, control execution skips for consecutive l sampling instances after k periodic control executions will be effective when:

$$\left\|\Delta e_{k+l}^{p}\right\| > \left\|\Delta e_{k+l}^{ap}\right\|$$

- $\Rightarrow \left\|\sum_{i=0}^{l-1} L\Delta r_{k+1-i}\right\| > \left\|\sum_{i=0}^{l-1} A^{i} B\left(a_{k+l-1-i}^{u} - a_{k-1}^{u}\right)\right\|$

LHS

RHS

Algorithm to Calculate Advantage Matrix

Initialize

- D, subPatternList, t, k, ρ, l

Compute

- $\Delta r_k := CA\Delta e_k + CBa_k^u + a_{k+1}^y \;\; \forall \text{ k } \in [1, \text{t}]$

k: for loop
l: for loop

- Initialize: ρ[k][l] := 1^t ; lhs := LHS; rhs:= RHS
- If (|| lhs || > || rhs ||): proceed to the command below.
  - else: continue to next loop iteration.

(within nested loop, if condition)

- ρ[k][l] := $1^k\, 0^l\, 1^{t-l-k}$
- If (controller performance valid):
  - Compute D[k][k+l] := || lhs || - || rhs ||
- Else: ρ[k][l] := 1^t

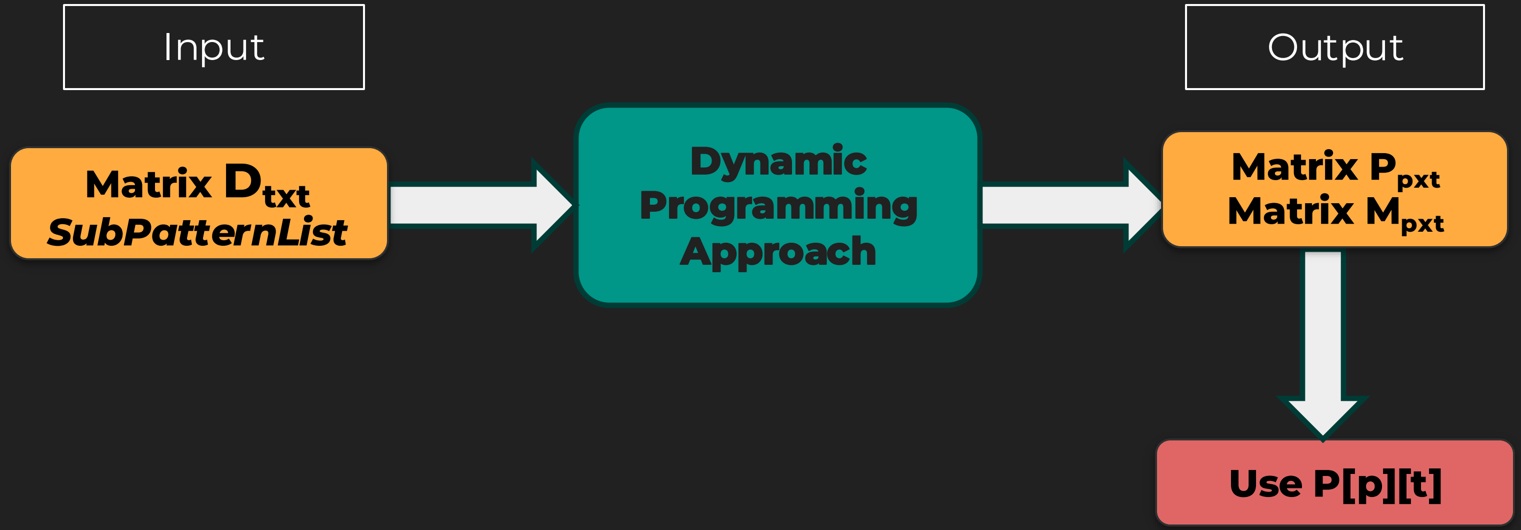
Store pattern

- subPatternList.append(ρ[k][l])

What is the Advantage Matrix?

- D quantifies the benefit of utilizing $\rho[k][l] := 1^{k}\, 0^{l}\, 1^{t-l-k}$ as the control sequence.
- It is only computed when there is lesser estimation error in aperiodic sequences.
- Time complexity of computation: $O(t^3)$
- $$D[k][l] = \left\|\Delta e_{k+l}^{p}\right\| - \left\|\Delta e_{k+l}^{ap}\right\|$$
$$= \left\|\sum_{i=0}^{l-1} L\Delta r_{k+1-i}\right\| - \left\|\sum_{i=0}^{l-1} A^{i}B(a_{k+l-1-i}^{u} - a_{k-1}^{u})\right\|$$

Optimal Resilient Attack Pattern Synthesis



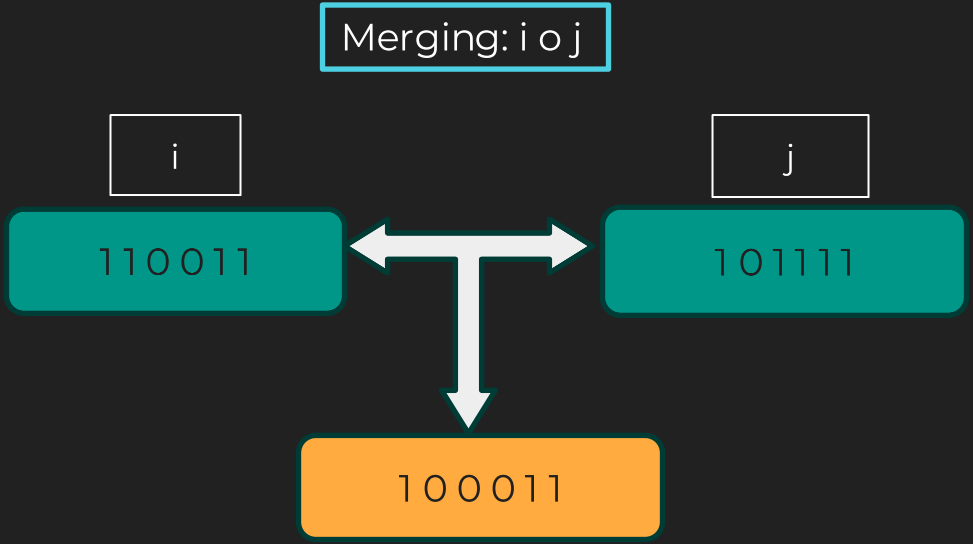
M[i][j]: **Maximum advantage** value by considering skips till jth position for first i subpatterns
P[i][j]: **Optimal t** length subpattern corresponding to M[i][j]

Some Notations

Let i^{th} subpattern list be ρ[k][l] where:
$\rho[k][l] = 1^{k}\ 0^{l-k}\ 1^{t-l}$

We define end0[i]=l and end1[i]=k

$$\underbrace{1\ \ 1\ \ 1\ \ \cdots\ \ 1}_{k\ executions}\ \ \underbrace{0\ \ 0\ \ \cdots\ \ 0}_{l-k\ skips}\ \ \underbrace{1\ \ 1}_{t-l\ executions}$$



Rate= $\frac{\sum_{i=1}^{t} pat[i]}{t}$
Rate is the number of ones divided by total length of sequence

Case 1: *i=1*

	j<end0(1)			Rate<r_{min}		*j* ≥end0(1) ∧ rate> r_{min}		
M[1]	**0**	**0**	**...**	**0**	**0**	**D[end1[1]][end0[1]]**	**...**	**D[end1[1]][end0[1]]**
P[1]	**1^t**	**1^t**	**...**	**1^t**	**1^t**	***SubPatternList*(i)**	**...**	***SubPatternList(i)***

$$M[1][j] = \begin{cases} \mathbf{0} & \text{if } j<\text{end0(1) or} \\ & \text{rate(pattern(1 to j))} < r_{min} \\ \mathbf{D[end1[1]]end0[1]]} & \text{otherwise} \end{cases}$$

$$P[1][j] = \begin{cases} \mathbf{1^t} & \text{if } j<\text{end0(1) or} \\ & \text{rate(pattern(1 to j))} < r_{min} \\ \boldsymbol{SubPatternList}\mathbf{[1]} & \text{otherwise} \end{cases}$$

Case 2: *i>1* and *j < end0(i)*

M[i-1]	0	0.02	...	0.08	0.08	...	0.08
M[i]	0	0.02	...	0.08	0.08	...	0.08

P[i-1]	1^t	1^t	...	1^t	110111	...	110011
P[i]	1^t	1^t	...	1^t	110111	...	110011

For j<end0(i): No new subpattern with non trivial advantage

M[i][j] = **M[i-1][j]** P[i][j] = **P[i-1][j]**

Case 3 Part(i): *$i>1$* and *$j \geq end0(i)$*

$\boxed{\text{Rate}(\textit{SubPatternList}(i)(1 \text{ to } j) < \mathbf{r}_{\mathbf{min}}}$

M[i-1]	0	0.02	...	0.08	0.08
M[i]	0	0.02	...	0.08	0.08

Consider $\rho(k,l)$= 11000011 as the ith subpattern
For j=6, we have rate=[110000]11
2/6=0.333 < $\mathbf{r}_{\mathbf{min}}$ =0.5

P[i-1]	1^t	1^t	...	1^t	1 1 0 1 1 1
P[i]	1^t	1^t	...	1^t	1 1 0 1 1 1

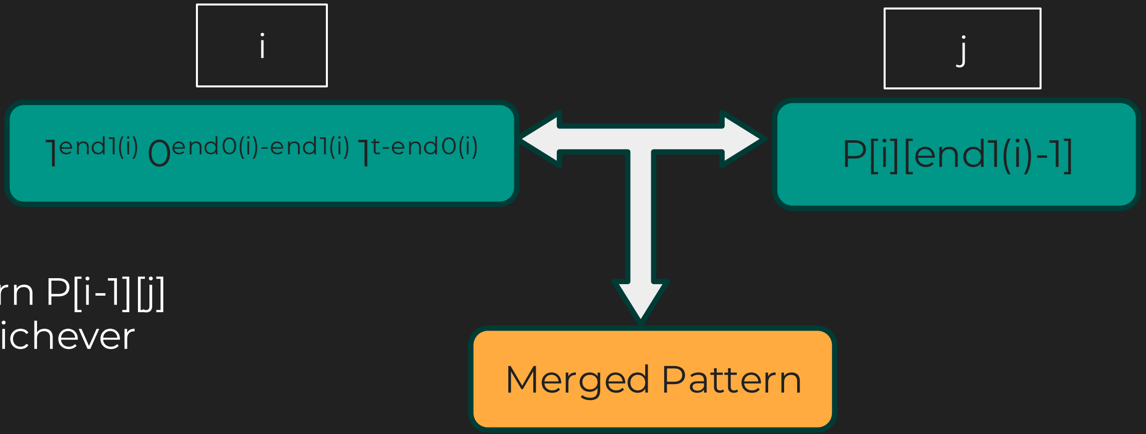
For $j<end0(i)$

M[i][j] = **M[i-1][j]**

P[i][j] = **P[i-1][j]**

Case 3 Part(ii) : i>1 and j≥end0(i)

$\boxed{\mathrm{Rate}(\mathit{SubPatternList}(\mathrm{i})) \geq \mathbf{r}_{\mathbf{min}}}$



If rate(merged_pattern)<$\mathbf{r}_{\mathbf{min}}$

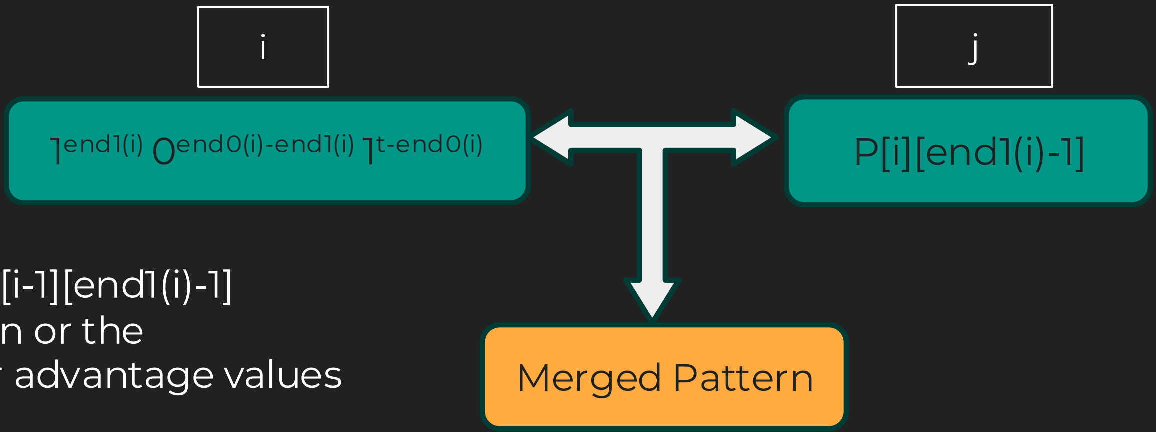
We choose the previous sub-pattern P[i-1][j] or the ith sub-pattern in the list, whichever gives a higher advantage

$$M[i][j] = \begin{cases} \mathbf{M[i-1][j]} & \text{if } M[i-1][j] > D[end1(i)][end0(i)] \\ \mathbf{D[end1[i]][end0[i]]} & \text{otherwise} \end{cases}$$

$$P[i][j] = \begin{cases} \mathbf{P[i-1][j]} & \text{if } M[i-1][j] > D[end1(i)][end0(i)] \\ \boldsymbol{\rho}\mathbf{[end1[i]][end0[i]]} & \text{otherwise} \end{cases}$$

Case 3 Part(iii) : i>1 and j≥end0(i)

$\boxed{\text{Rate}(\textit{SubPatternList}(\text{i})) \geq \mathbf{r_{min}}}$



If rate(mergedpattern)≥ r_{min}
Adv(merge)= D[end1(i)][end0(i)]+M[i-1][end1(i)-1]
We choose the previous subpattern or the merged subpattern based on their advantage values

$$M[i][j] = \begin{cases} \mathbf{M[i\text{-}1][j]} & \text{if } M[i\text{-}1][j] > Adv(merge) \\ \mathbf{Adv(merge)} & \text{otherwise} \end{cases}$$

$$P[i][j] = \begin{cases} \mathbf{P[i\text{-}1][j]} & \text{if } M[i\text{-}1][j] > Adv(merge) \\ \textbf{Merged Pattern} & \text{otherwise} \end{cases}$$

Attack Library Synthesis

Attack Model

FDI in actuator and sensor signal

FDI in Consecutive samples:
1) Max damage
2) Minimum time

Attack Vector Generation Strategy : Constraint Optimization Problem

$$CP:\ \exists\ a[1], a[2], a[3], \dots, a[t]$$
$$s.t.\ x_0^a = x;\ \hat{x}_0^a = x \ where\ x \in X_S$$

Initial conditions

State space equations

$$u_{i-1}^a = -K\hat{x}_{i-1}^a;\ \tilde{u}_{i-1}^a = u_{i-1}^a + a_{i-1}^u \forall\ i \in [1,t]$$

$$x_i^a = Ax_{i-1}^a + B\tilde{u}_{i-1}^a\ ;\ y_i^a = Cx_i^a + a_i^y\ \forall\ i \in [1,t]$$

$$r_{i-1}^a = y_i^a - C(A\hat{x}_{i-1}^a + Bu_{i-1}^a);\ \hat{x}_i^a = A\hat{x}_{i-1}^a + Bu_{i-1}^a + Lr_{i-1}^a \forall\ i \in [1,t]$$

Stealthiness condition, Safety envelope, Signal ranges

$$f(r_{i-1}^a) < Th;\ |y_i^a|, |a_i^y| < Y; |u_i^a|, |\tilde{u}_i^y|, |a_i^u| < U \quad \forall\ i \in [1,t]$$

$$x_i^a \in X_S\ \forall\ i \in [1, t-1];\ x_t^a \in X_S$$

FDI Attack Prediction

- ***Goal***: Given the values of Sensor Data in presence of attack vectors (y^a) and residual (r^a) for a number of previous timesteps (d-past), We need to predict the attack vectors for a future timeframe (d_{future}).

$$\{[y^a\ r^a]^{T}\}_{dpast} \xrightarrow{\text{GRU Based Predictor}} \{[a^u\ a^y]^{T}\}_{dfuture}$$

Implementation

- The GRU Predictor outputs the values of system outputs without the sensor attack ($\tilde{y}^a = y^a - a^y$) and residual (r^a) for future timesteps.
- Using these values, we will calculate the attack vector values for the sensor and actuator for these future timesteps.

$$\begin{bmatrix} \tilde{y}^a_{t_o+1} & \cdots & \tilde{y}^a_{t_o+dfuture+1} \\ r^a_{t_o+1} & \cdots & r^a_{t_o+dfuture+1} \end{bmatrix} = \text{Predictor}\left(\begin{bmatrix} y^a_{t_o-dpast} & \cdots & y^a_{t_o} \\ r^a_{t_o-dpast} & \cdots & r^a_{t_o} \end{bmatrix} \right)$$

Model Specifications and Training

- **The predictor is designed with 4 layers of neural networks consisting of 2 layers of Gated Recurrent Units (GRU) followed by 2 layers of fully connected layers.**

Simulation of system model in presence of attack vectors

Training Data

Training Labels

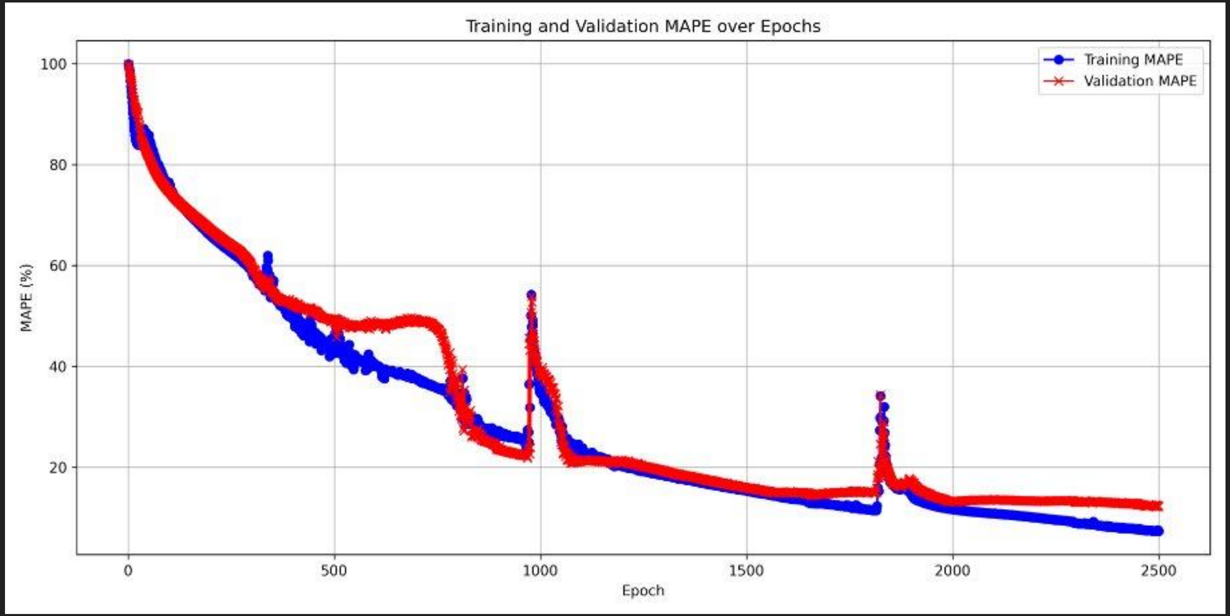
$$\{[y^a\ r^a]^\top\}_{dpast}$$

$$\{[\tilde{y}^a\ r^a]^\top\}_{dfuture}$$

Model Accuracy

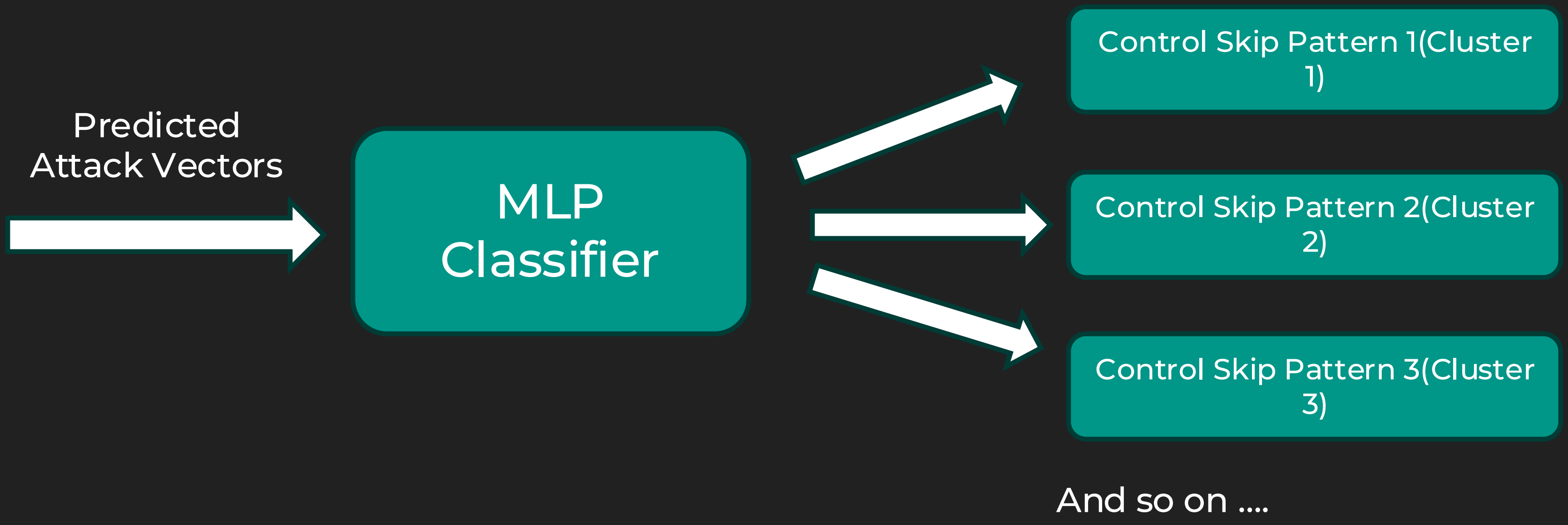
- Length of attack vectors – Lies in the range of 5 – 11
- d_{past} chosen as 3 and d_{future} chosen as 8
- For evaluation, the predicted outputs (predicted future attack vectors) are compared against the generated test data (future attack vectors in the attack library) to find the accuracy.

Training and Validation Loss Percentages.



MLP Classifier

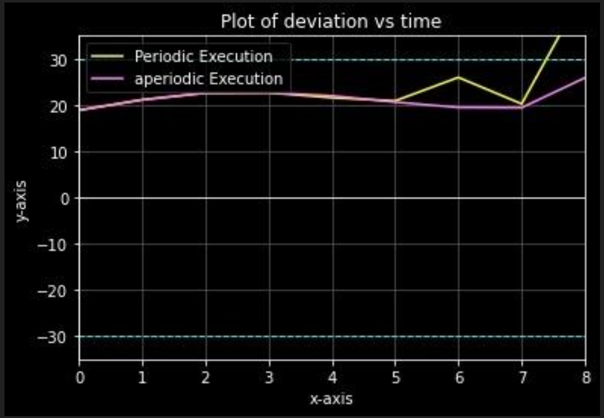
- Given a prediction of the most probable FDI attack vectors for future iterations, we need to deploy an optimal attack-resilient pattern to incapacitate it.



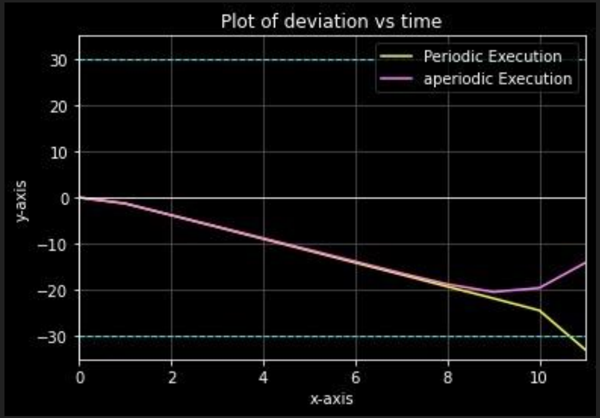
Labeling and Training

- We partition the library of attack vectors A_{lib} into a finite number of clusters, to use as training data.
- Formally, two attack vectors ai and aj will be placed in the same cluster C_k if the optimal attack-resilient control execution patterns returned by our proposed DP-based method for ai and aj are same.
- We then use this dataset to train a multi-layer perceptron (MLP)-based classifier that can map a predicted attack vector in runtime to a certain cluster and deploy the control execution skip pattern which labels that cluster.

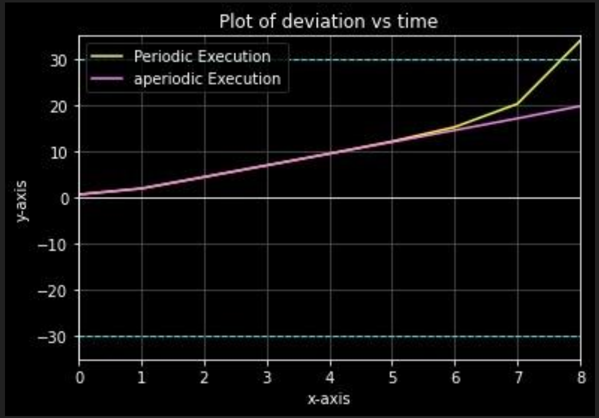
Results on the TTC Benchmark



Control Input 11001111



Control Input 10000011111



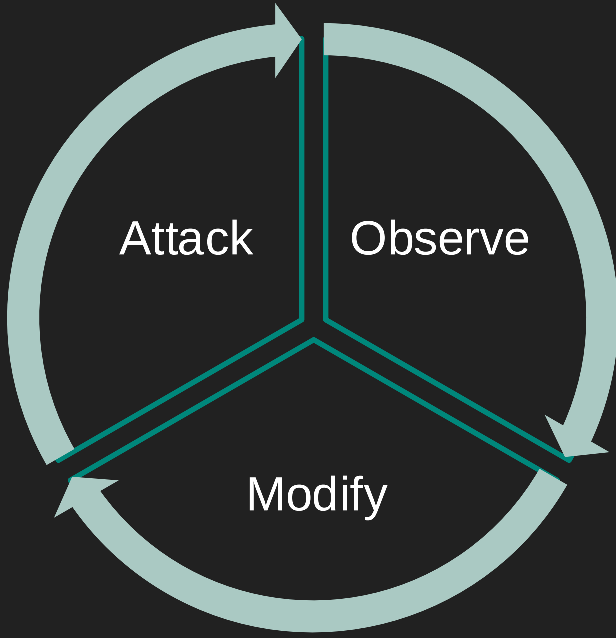
Control Input 11000011

Attack vector takes the system outside safety region
Control skip pattern improves resilience for countermeasures

Results on other benchmarks

Systems	Order	r_{min}	Initial State	Attack Length	Control Patterns	Advantages
TTC	2	0.51	[0.65,0.78]	10	$10^5 1^4$, $10^4 1^5$, 101^8	*15.43, 12.01, 5.43*
ESP	2	0.45	[-1.96,-3.93]	3	100	*2.43*
Fuel Injection	3	0.5	[-0.08,-0.53,-1.77]	5	$1^4 0$, $1^3 0^2$, $10^2 1^2$	*6.12, 4.45, 2.28*
Suspension Control	4	0.52	[-1.64,21.89,47.19,71.12]	7	$1010^2 1^2$, 101^5, $1^2 0^3 1^2$, $10^3 1^3$	*938.76, 879.87, 720.83, 661.78*
4-Car Platoon	8	0.5	[-0.55, 1.47, -5.29, -0.26, -1.84, -1.39, 4.47, 3.75]	26	$1^5 0^{13} 1^8$, $1^4 0^{13} 1^9$, $1^3 0^{13} 1^{10}$, $1^2 0^{13} 1^{11}$	*281.59, 255.03, 231.67, 209.67*

Limitations



Attacker Dynamically changes his attack patterns by learning the deployed execution sequence

Attack library limited to one type of attack:

Replay attack: Replays valid communication data from the past

Delayed Data injection

Contributions

Implemented the algorithms provided in the paper (only existing implementation)

Obtained similar results (graphs) provided in the paper – improved resilience!

Researched and found potential failures for the algorithms in the paper

CS637A: Embedded and Cyber Physical Systems

Thank You

Any Questions?

Aravind Seshadri	*Kapu Nirmal Joshua*	*Sumay Avi*	*Tanmay Purohit*	*Raghav Shukla*
210180	*210489*	*211071*	*211097*	*210800*