acrt AI, Computer Science and Robotics Technology

IntechOpen Journals

RESEARCH PAPER

A Convolutional Neural Network-LSTM Based Physical Sensor Anomaly Detector for Interdependent SCADA Controllers

Rishabh Das[1,*], Aaron Werth[2] and Thomas Morris[2]

1 Emerging Communication Technologies, Ohio University, USA
2 Computer Engineering, University of Alabama in Huntsville, USA
*Corresponding author. E-mail: rishabh.das@ohio.edu

Citation
Rishabh Das, Aaron Werth and Thomas Morris (2025), A Convolutional Neural Network-LSTM Based Physical Sensor Anomaly Detector for Interdependent SCADA Controllers. *AI, Computer Science and Robotics Technology* 4(1), 1–31.

DOI
https://doi.org/10.5772/acrt.20250037

Received: 28 March 2025
Accepted: 30 May 2025
Published: 18 June 2025

Abstract

This study outlines a novel intrusion detection system (IDS) to detect compromised sensor data anomalies in interdependent industrial processes. The IDS used a peer-to-peer communication framework which allowed multiple programmable logic controllers (PLCs) to communicate and share sensor data. Utilizing the shared sensor data, state estimators used a long short-term memory (LSTM) machine learning algorithm to identify anomalous sensor readings connected to neighboring PLCs controlling an interdependent physical process. This study evaluated the performance of the IDS on three industrial operations aligning to a midstream oil terminal. The framework successfully detected several multi-sensor compromises during mid-stream oil terminal operations. A set of performance evaluations also showed no impact on the real-time operations of the PLC and outlined the prediction latencies of the framework.

Keywords: embedded intrusion detection, interdependent process monitoring, mid-stream oil terminal, physical sensor spoofing, spatio-temporal detection framework

1. Introduction

Cyberattacks have impacted industrial control systems (ICS) and supervisory control and data acquisition (SCADA) for more than a decade. Some of these attacks have relied on deception by manipulating or reusing data when the data no longer represented the actual state of the critical infrastructure. Stuxnet manipulated data to appear normal to the user so that the user would not be aware of the actual physical state of the industrial process, eventually causing damage [1]. Operators consider sensor data as the ground truth and rely on the readings to achieve remote monitoring. Spoofing sensor values can create a disconnect between the operator's understanding and the actual state of the physical system. Such disconnects are dangerous and may lead the operator to make the wrong operational decision, causing damage to the critical infrastructure. Detecting such attacks is difficult due to modern critical infrastructure's sheer scale and complexity.

Critical infrastructure uses programmable logic controllers (PLCs) connected to sensors and actuators to control physical processes. The processes may influence system components in predictable ways, following established laws of physics, e.g., the flow rates and pressure follow fluid mechanics principles in a midstream oil terminal. Actuators such as pumps can alter these flow rates, thus affecting sensor readings. Statistical or machine learning models can capture such behaviors. The working hypothesis is that if statistical models learn the process-semantic patterns of inter-dependent industrial processes, the framework can identify spoofed sensor readings. This detection can also be scaled across large-scale distributed physical processes.

In a distributed infrastructure, an attacker can spoof sensor readings through a network intrusion or a supply chain compromise. In a network intrusion, the attacker can perform a man-in-the-middle (MiTM) attack and relay tampered traffic, while in a supply chain attack, the attacker inserts a software or hardware exploit inside the sensors. In both cases, the sensor communicates false readings to the PLCs. The falsified reading may send incorrect data to the user and elicit an improper operational response, potentially disrupting the physical process. Network-based intrusion detection system (IDS) can detect intrusions across industrial communications but cannot detect sensor compromise due to supply chain compromise [2–6]. Literature review also shows that existing IDSs focus on detecting attacks on one node and lack visibility of neighboring PLCs [7]. Moreover, the existing solutions lack embedded detection capability within the PLCs.

Following are the contributions of this study.

- This study presents an embedded IDS framework that can predict the state of peer node sensors if they control an interdependent physical process. If the sensor

states are highly correlated, the IDS can detect anomalies of sensors connected to neighboring PLCs.
- This study presents a hybrid spatio-temporal detection framework that uses a convolutional neural network (CNN) to capture spatial co-activations within a PLC scan, and a long short-term memory network (LSTM) to model temporal dependencies across scans.
- The IDS can be embedded inside the industrial controller to provide native detection.
- The IDS can detect falsified states of sensors due to supply chain attacks.

The contributions address the research gaps, as the existing IDS solutions focus primarily on network-based attacks or single-node anomalies. The following sections elucidate the design and implementation of three core modules: a data sensor module that reads the state of the sensors and actuators from the MODBUS memory of the PLC, a peer-to-peer network that circulates the current state of the PLCs with other edge nodes, and a module inside the analysis engine. These modules allow the framework to detect sensor anomalies across interconnected systems. Section 2 presents the literature review pertaining to intrusion detection systems and novelties of this work's IDS. Section 3 discusses the threat model including the software trojan exploit among others and explains the approach taken in this study to detect the threat. Section 4 evaluates the accuracy of the approach using three midstream oil terminal operations. Section 5 evaluates the impact of the IDS on PLC performance and Section 6 concludes the study.

2. *Literature review*

Past studies have focused on sensor data compromised by cyberattacks and have described these possible attacks, the proposed solutions and methods to detect and mitigate against these attacks. Cardenas in his studies has discussed attacks that affect certain supervisory, or control loops associated with industrial systems [8]. Attacks can affect the communication between the devices connected to sensors and a controller. Huang [9], in his study, which includes Cardenas as coauthor, characterized a threat model for a specific industrial control system—the Tennessee Eastman System. This ICS has various controllers and sensors associated with chemical reactions. The threat model involved the attacker providing deceptive sensor data to the controllers, which therefore acted incorrectly and caused the tanks to reach unsafe levels of pressure. In recent studies, Cardenas devised methods to detect compromised sensor data and employed an Advanced detection module (ADM) that used a linear mathematical model of the physical system, or plant [8]. With the model, the ADM determined if there was a discrepancy when compared to the sensor data that it received. The discrepancy was accumulated in a variable

called the CUSUM over time and compared to a threshold to determine if there was a compromised sensor. Combita and Alvero [2] developed a method to mitigate the threat model.

A survey described techniques in which a model of the physics in cyber-physical systems was used to detect anomalies for ICSs and critical infrastructure [7]. The survey highlighted the nature of sensor data as values at discrete points in time whose behavior can be modeled even when the physical system was subjected to various inputs and stimuli from actuators, such as pumps and valves among others depending on the given domain of engineering or physics. If there was a discrepancy from the expected values according to the model from the purported sensors and other information, it can be inferred that there was an anomaly, indicating a potential cyberattack. The survey also referenced disparate works in power systems, control systems, and cybersecurity and discusses them in a unifying taxonomy of various approaches. A later survey provided an update with similar techniques and more sophisticated attacks for the smart grid [10]. This later survey focused on situational awareness, and analyzed how various techniques of detection enabled that awareness. Another survey by Gaggero *et al.* discussed similar detection methods that specifically involved artificial intelligence (AI) for the smart grid [11]. Gaggero *et al.* discussed the ability of traditional machine learning and AI at large to learn patterns in data associated with the physical system to be able to detect anomalous events. Other studies have focused on power systems involving false data injection attacks (FDIA). Wang *et al.* presented a survey of methods to detect FDIAs in power systems, which involved state estimators to do so [3]. Ahmed *et al.* analyzed several studies including a taxonomy of the current countermeasures to defend against FDIA [4] for both structured and unstructured data.

Chromik, in several studies, discussed the use of power system models to detect FDIAs [5, 6]. Of particular relevance is a study in which a local monitoring detection system within substations was presented [12]. The current work is similar to this study in that the IDS discussed is also local to the relevant devices and physical system. However, the current work proposes an IDS that is not only local but also embedded in the devices, which allows for a reduced attack surface. Chromik proposed that in future work, the local IDSs of the various substations will communicate and share relevant data. This current work, on the other hand, involves the sharing of data among embedded IDSs in PLCs to allow for greater situational awareness. According to Hadžiosmanovic [13], certain variables within industrial control systems and critical infrastructure may be correlated indicating interdependencies. Therefore, sharing the data of these relevant variables is useful for the PLCs to have more accurate state estimation, which can aid in detecting anomalies.

In cyber-physical systems, interdependent processes are distributed across vast distances and generate time-synchronized data streams. These data streams exhibit spatio-temporal patterns, and the proposed IDS should be able to identify patterns for effective anomaly detection. CNN-LSTM models offer superior handling of spatial-temporal patterns in sensor-based time-series data, offering a stronger foundation than GRU, ARIMA, and even Transformer-based methods in several industrial and financial applications. Although studies have not applied CNN-LSTM for embedded intrusion detection, prior work has explored CNN-LSTM in domains like stock prediction, Internet of Things sensor data prediction, and other financial applications [14–18]. For example, Kumar *et al.* compared CNN, LSTM, Gated Recurrent Unit (GRU), and ARIMA in traffic forecasting and determined that CNN-LSTM was superior to ARIMA and CNN exhibited lower RMSE values across several datasets [14]. Ata *et al.* concluded that a CNN-based model effectively predicted traffic flow with erratic and extended patterns, outperforming established models in intelligent transportation systems [18]. For stock market analysis, Dwivedi *et al.* and Chen *et al.* compared the S-ARIMA model with CNN-LSTM [15, 16]. The CNN-LSTM model offered deeper insights into non-linear patterns. Liang *et al.* observed that CNN-LSTM performed better in volatile conditions than a standard ARIMA model [17]. Existing literature has established the effectiveness of CNN-LSTM combination in modelling spatio-temporal data streams. This study also used CNN-LSTM models within embedded architecture and a novel peer-to-peer data sharing framework for PLC-native intrusion detection.

3. Methodology

This section outlines a novel framework to detect attacks on sensors connected to neighboring nodes in an industrial control system. The framework uses three modules to get visibility of the physical process, states of neighboring PLCs, and establish a baseline of the system's normal behavior. The first module is a physical system data sensor; it allows the embedded IDS to monitor the states of the sensors or actuators. The second module is a peer-to-peer (P2P) communication framework. Sharing information through the P2P framework allows the embedded IDS to have visibility over the states of neighboring PLCs. The third module is a physical system anomaly detector. This module learns the sensors' behavior and infers the sensors' values using the actuators' states.

In every PLC cycle, the physical system data sensor takes a snapshot of the actuator and sensor readings from the hardware layer of the PLC. The instances of the embedded IDS share the states using the P2Pcommunication framework. Sharing allows the embedded IDS to have visibility of the entire physical process. Using the collective states of PLCs, the physical system anomaly detectors predict the potential values of the sensors and the actuators. The physical system anomaly

detector also maintains a confidence interval for each sensor reading. If the reported value is out of the confidence interval, the embedded IDS flags the reading as false and sends an alert to a designated computer using the incident response system. Figure 1 shows the architecture and workflows of the proposed framework.

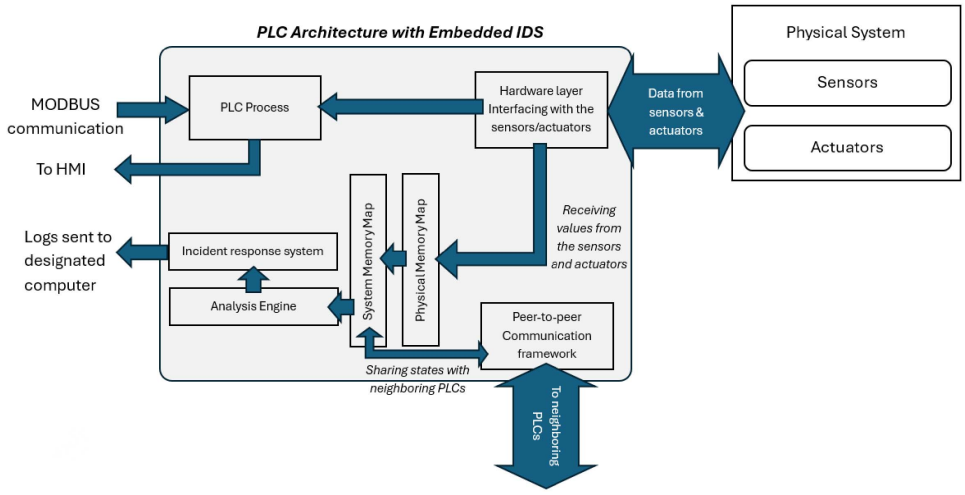

Figure 1. Modules inside the embedded IDS with the physical anomaly detector and the physical system data sensors.

The upcoming sections provide a detailed discussion of the three modules.

3.1. Physical system data sensors

The embedded IDS uses two physical system data sensors: physical memory map (PMM) and system memory map (SMM). PMM monitors the current state of the PLC and SMM maintains the current state of all interconnected nodes.

- Physical Memory Map (PMM): The PLC receives the states of the sensors and actuators from the hardware layer. The PMM takes a snapshot of the MODBUS memory and uses the address information from the PLC configuration file to interpret the values of the sensors and actuators. The readings in the PMM update every PLC scan cycle and mirror the states seen by the PLC CPU. If a sensor or actuator value is altered, the PMM reads the falsified states of the sensors and actuators.
- System Memory Map (SMM): The SMM takes the present state information from the PMM and shares it with other PLCs using the P2P communication framework. SMM receives the state information of the neighboring PLCs, and structures them based on the variable address of the ladder logic and peer node locations.

The proposed framework embeds the PMM and SMM modules inside the PLC. The case studies implemented these modules by adding additional code into an

open-source PLC, OpenPLC [19]. The PMM obtained the values from the hardware layer of the OpenPLC, and the SMM used a peer-to-peer network, as outlined in the next section, to share values with neighboring nodes.

3.2. Peer-to-peer communication framework

The peer-to-peer (P2P) communication architecture enables the embedded IDS to share the physical state of the actuators and sensors of each PLC with other edge nodes. A P2P network is a distributed application architecture in which interconnected nodes ("peers") communicate without using a centralized server. Each peer in this research is an instance of the embedded IDS inside the PLC. The nodes have equal administrative privileges and can function concurrently as clients and servers.

The P2P network uses an optimized depth-first routing algorithm [20] based on Gao's methodology [21]. Gao's approach combined star and mesh topology. Gao grouped the PLCs into k clusters based on the physical distance between the edge nodes. Each cluster behaves like a star topology. The cluster selects a random central node, and all edge nodes are directly connected to the central node. The intra-cluster connections use a partial mesh architecture. The partial-mesh architecture directly connects the central node of each cluster to two other central nodes using a point-to-point connection. The clustered topology considers the inter-PLC distances and allows the P2P network to scale for large systems.

For case studies involving virtual midstream oil terminals, outlined in Section 4, twelve PLCs control five subsystems (tank farms, pump houses, tanker truck gantry, pipeline transfer, and vessel operation) which were distributed across different distances were simulated. This study used the simulation distance of the subsystems to create the distance matrix for twelve PLCs and created five clusters: Cluster I—[PLC1, PLC2, PLC4, PLC5]—Controls Marine Tanker (MT) operation and the terminal-to jetty pipeline; Cluster II—[PLC10, PLC11, PLC12]—Controls Tank Farm (TF) operations; Cluster III—[PLC7, PLC8, PLC9]—Controls pumphouses; Cluster IV—[PLC6]—Controls Tanker Truck (TT) operations; and Cluster V—[PLC3]—Controls Pipeline Transfer operation (PLT).

The IDS uses a custom configuration file to define the P2P topology. Python script parses the "config.csv" and creates a logical overlay network encapsulated within a traditional TCP/IP routing framework. Figure 2 shows a sample configuration file of a P2P network topology in the virtual midstream oil terminal testbed.

The configuration file shown in Figure 2 illustrates the logical interconnectivity of each node with other neighboring edge nodes. An interpreter reading the configuration file considers a line starting with hash ("#") sign as a comment and skips the first line of the file having names of the configuration parameters. The rest

```
# Description of the network parameters------------------
# Node-> Name of the PLC
# Neighbors-> Neighbors to the PLC
# NIPs->IP address of the neighbor
# NodeIPs->IP address of the current node
# -------------------------------------------
Node|Neighbors|NIPs|NodeIPs
PLC1|PLC2,PLC3,PLC4,PLC5,PLC6|200.200.200.2,200.200.200.3,200.200.200.4,200.200.200.5,200.200.200.6|200.200.200.1
PLC2|PLC1|200.200.200.1|200.200.200.2
PLC3|PLC1,PLC9|200.200.200.1,200.200.200.9|200.200.200.3
PLC4|PLC1|200.200.200.1|200.200.200.4
PLC5|PLC1|200.200.200.1|200.200.200.5
PLC6|PLC1,PLC12|200.200.200.1,200.200.200.12|200.200.200.6
PLC7|PLC9|200.200.200.9|200.200.200.7
PLC8|PLC9|200.200.200.9|200.200.200.8
PLC9|PLC3,PLC7,PLC8,PLC12|200.200.200.3,200.200.200.7,200.200.200.8,200.200.200.12|200.200.200.9
PLC10|PLC12|200.200.200.12|200.200.200.10
PLC11|PLC12|200.200.200.12|200.200.200.11
PLC12|PLC6,PLC9,PLC10,PLC11|200.200.200.6,200.200.200.9,200.200.200.10,200.200.200.11|200.200.200.12
```

Figure 2. Configuration file for designing the topology of the P2P network.

of the file contains a line of network configuration parameters for each edge node. For each node, an operator configures the following four parameters:

- **Node:** Name of the PLC. In the case study, the virtual midstream oil terminal names the twelve PLCs as "PLC" followed by a sequential number between 1 and 12.
  Example: "PLC1".
- **Neighbors:** Names of the neighboring PLCs logically connected to the current PLC. The PLC names are represented as a list separated by commas.
  Example: "PLC2, PLC3, PLC4, PLC5, PLC6".
- **NIPs:** A list of IP addresses of logically connected neighboring nodes separated by commas.
  Example: "200.200.200.2, 200.200.200.3, 200.200.200.4, 200.200.200.5".
- **NodeIPs:** IP address of the current PLC.
  Example: "200.200.200.1".

A vertical bar ("|") separates each configuration parameter.

To identify the quantitative performance of network topologies, this study compared the network latency of the physical P2P topology developed using Gao's approach with four topologies from the existing literature [1, 22, 23]. The analysis considered line topology, ring topology, star topology, and full-mesh topology.

Line topology connects all edge nodes in a single line. Ring topology connects every edge node to two neighboring devices. A star topology connects all edge nodes to a centralized hub. In the case of the midstream oil terminal, one of the edge nodes acts as a central hub and connects to other edge nodes. A full-mesh topology establishes a direct point-to-point connection between each edge node and every other edge node.

The network latency test measures the time taken by each PLC to complete the sharing of physical states. For the midstream oil terminal, it was the sum of the time taken by each PLC to send its physical state information to the other eleven PLCs

and the time taken to receive the states of all eleven PLCs. For example, the network latency of PLC1 is

Network latency of PLC 1 = Time taken to send physical state information of PLC 1 to all other PLCs + Time taken to receive physical state information of every other PLC

Table 1. Network latency in milliseconds for different topologies.

Topology of the network	PLC1	PLC2	PLC3	PLC4	PLC5	PLC6	PLC7	PLC8	PLC9	PLC10	PLC11	PLC12	Average (in milliseconds)
Line [1]	10.82	14.16	12.41	15.72	16.52	11.32	18.57	15.63	15.81	12.94	15.36	14.73	14.49
Ring [22]	23.44	23.14	27.65	26.25	25.55	27.98	31.53	31.53	24.53	28.23	29.54	24.5	26.99
Full mesh [1]	2200.75	1948.35	1979.29	2175.12	2004.47	2218.48	2205.74	2788.18	2379.42	2248.85	2169.44	2572.38	2240.87
Star [23]	12.49	12.3	13.6	10.59	10.75	15.22	12.72	13.25	12.88	15.29	12.94	10.58	12.717
Gao's approach [21]	10.71	14.17	10.67	11.52	18.93	17.55	13.39	13.8	11.43	11.92	10.72	11.37	13.015

Star topology provided a lower average latency. However, Gao's approach provided more reliability, greater network bandwidth, and is cost-efficient. This is because star topology has a single point of failure; If the central node fails, the attached nodes get disabled. The central node has to handle heavy network traffic from all other nodes. A PLC with low computation capacity cannot handle such network traffic. Additionally, the star topology is a centralized architecture. Hence, each node, however far apart, needs a physical connection to the central node. This increases the construction cost of the network. Hence, this study used the topology designed using Gao's approach for the P2P network of the embedded IDS. Table 1 outlines the network latencies for different topologies.

3.3. Analysis engine—physical system anomaly detector

The physical system anomaly detector in the analysis engine runs every PLC cycle and uses state estimators to decide if the readings of the sensors are false. The state estimators model the values of a continuous or a discrete system parameter. It takes the older system conditions as input and predicts the state of the next time step. A data preprocessing unit aggregates the current readings from the PMM and SMM. After aggregation, the data preprocessing unit rearranges the readings and feeds the ordered list into a FIFO stack. This stack is coupled to the inputs of the module containing the state estimators. If the prediction is within the confidence interval, the physical system anomaly detectors mark the value as normal; otherwise, the incident response system logs the false value of the sensor. The analysis engine uses two types of state estimators: Type I and Type II. Each type performs a unique input-to-output mapping for state estimation.

A generic interconnected system with three PLCs is shown in Figure 3.

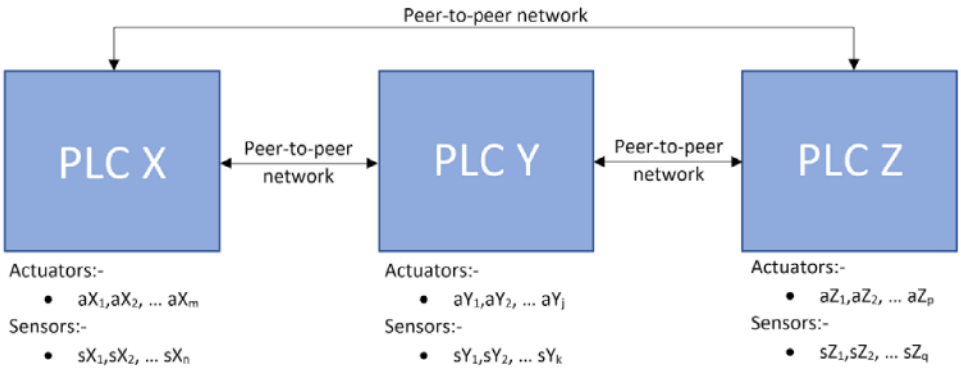

Figure 3. System with three interconnected PLCs.

In the example, PLC X has m actuators and n sensors. Sensor and actuator names start with "s" and "a" followed by the PLC name (X, Y, and Z) and the serial number. The naming for PLC Y and PLC Z follows the same convention.

The Type I state estimator takes the value of the actuators and changes in sensor value ($\Delta s_t = s_t - s_{(t-1)}$) at previous time stamps ($t-1, t-2, t-3$) as input and predicts the change in sensor value at the current timestamp (t). Each sensor attached to a PLC has a Type I state estimator. Hence, PLC X, Y, and Z have n, k, and q Type I state estimators, respectively. For a given states of the actuators, the Type I state estimator predicts the change in sensor value of the current timestamp. Hence, Type I state estimator for sensor sX_i predicting timestamp t based on actuator values and past sensor changes is expressed as

$$\Delta sX_{i,t} = f(\{a_{X,\,Y,\,Z(t-1:t-3)}, \Delta sX_{i(t-1:t-3)}\})$$

where:

- $a_{X,\,Y,\,Z\,(t-1:\,t-3)}$: actuator values from PLCs X, Y, Z at timestamps $t-1, t-2, t-3$.
- $\Delta sX_{i\,(t-1:\,t-3)}$: previous changes in sensor sX_i at timestamps $t-1, t-2, t-3$.

The Type II state estimator models correlated sensor values of interdependent processes in an industrial control system. An interdependent process is a large industrial system that uses multiple PLCs and adheres to a distributed control system architecture. In such a system, a change of an actuator on one of the PLC may cause changes in sensor states on other PLCs. An example of an interdependent industrial operation is the TT loading operation at the mid-stream oil terminal. The TT loading operation uses three interconnected PLCs: a PLC controlling the tank farm, a PLC controlling the pump house, and a PLC controlling the tanker trucks. During a TT loading operation, turning on pumps at the pump house increases the

flow rate at the outlet of the pump house and at the TT loading bay; Sensors P7SF2 and P6SF1 observe these changes. The PLC controlling the tanker truck observes a change in sensor reading even though the actuator states are the same. Here sensor P7SF2 and sensor P6SF1 show a statistical correlation. Using the change in values of one of the correlated sensors, the Type II state estimator can predict the change in the other.

The first step for modeling a Type II state estimator is identifying the correlated sensors. This research uses PCC [24] for measuring the linear relationship between two sensor readings. The PCC coefficient ranges from −1 to +1. An exact value of +1 or −1 implies that a linear equation can completely describe the relationship of the variables and a value of 0 implies no linear correlation between the variables. For this study, the interdependent processes in the midstream oil terminal require a PCC value of at least |0.9| for building accurate state models [25]. Hence, while identifying interdependent sensors, the case studies in the upcoming sections chose the sensor pairs only if the PCC value was greater than or equal to |0.9|.

Let's assume that sensor sX_1 and sensor sZ_1 are strongly linearly correlated, and $PCC\ (sX_1, sZ_1) \geq |0.9|$. At timestamp t, the Type II state estimator operating in PLC X for predicting the correlated sensor values of PLC Z receives the actuator values from PLCs X, Y, Z ($a_{X,\,Y,\,Z\ (t-1:\,t-3)}$) and changes in sensor values on PLC X ($\Delta sX_{1(t-1:\,t-3)}$) as inputs from the FIFO stack (timestamps $t-1, t-2, t-3$). The output of the type II estimator will predict the change in the correlated sensor sZ_1 on PLC Z at current timestamp t, $\Delta sZ_{1,\,t} = sZ_{1,\,t} - sZ_{1,\,(t-1)}$.

$$\Delta sZ_{1,\,t} = h(\{a_{X,\,Y,\,Z(t-1:t-3)}, \Delta sX_{1(t-1:t-3)}\})$$

h: represents the predictive state estimator model trained based on historical data exhibiting a strong correlation (PCC ≥ |0.9|) between sX_1 and sZ_1.

3.3.1. Choice of algorithm for Type I and Type II state estimators

The Type I and Type II state estimators are both one-step forecast models. These models take a stack of values containing parallel input time series of sensors and actuators values. The input time series are parallel because each series has an observation at the same time steps. The output depends on the input time series and predicts the value of the current timestamp.

The algorithm for state estimation in the midstream oil terminal needs to infer spatial and temporal structure from the data. The actuator and sensor data from the midstream oil terminal have a spatial structure. For example, a tanker truck loading operation in the midstream oil terminal involves the actuation of valves in three locations; the change in sensor readings are spatially linked to these locations. The state estimator has to infer the spatial relations from the data. Additionally, the

midstream oil terminal data exhibits a temporal structure. For example, turning a pump ON in the pumphouse increases the pressure and flow rate at the outlet of the pump house at a future timestamp.

Considering the requirement of spatial and temporal awareness, this study used a combination of CNN and LSTM model inside the state estimators. CNN-LSTM combination is the current state- of-the-art for modeling multiple parallel input time series with spatial and temporal structure [26, 27]. A deep CNN architecture extracts the spatial features from the 1D input vector and the LSTM model interprets the features across time steps. Together the hybrid CNN-LSTM model infers the spatial and temporal information of the data.

3.3.2. CNN-LSTM structure

CNN-LSTM combines two sub-models: CNN model for feature extraction and LSTM model for interpreting the features across time steps. The CNN model has three layers: a 1-D convolutional layer, a Max pooling layer, and a Flatten layer. Figure 4 shows the layout of the CNN layers and the LSTM backend components. The 1-D convolutional layer analyzes the rolling window and extracts the features. The kernel size parameter in the convolutional layer specifies the number of time steps included in each input sequence. In this research, the kernel size parameter was set to three to match the FIFO input sequence length of the Type I and II estimators. The matching kernel size ensured that the convolution layer captured the essential temporal dynamics from the last three timestamps of the sensor data. The number of filters was set to 64, which optimized computational complexity while capturing relevant spatial features. The model used ReLU activation to mitigate the vanishing gradient problem in deep neural networks. After the feature extraction, a Max pooling layer with a pool size of 2 with stride 2 reduced the dimensionality of the feature map, and a Flatten layer converted the multidimensional feature map to a single-dimensional array. A Time- distributed wrapper binds all three layers into a single CNN unit.

The CNN unit provides a single-dimensional feature vector to the LSTM network. Using these features, the LSTM learns the long-term dependencies between the timestamps [28]. The LSTM network has two layers: an LSTM layer and a Dense layer. The LSTM layer holds the neural network nodes for learning the behavior of the system. This study optimized the number of nodes using GridSearchCV [29]. An LSTM layer with 200 nodes provides the lowest Root-Mean-Square Error (RMSE) value for the operational data of the midstream oil terminal. The combination of dropout and recurrent dropout at 0.2 randomly drops connections during training and mitigates overfitting. The dropout allows for a better generalization across temporal sequences and reduces sensitivity to noisy sensor data. A single node dense layer with linear activation directly maps LSTM output to predicted continuous

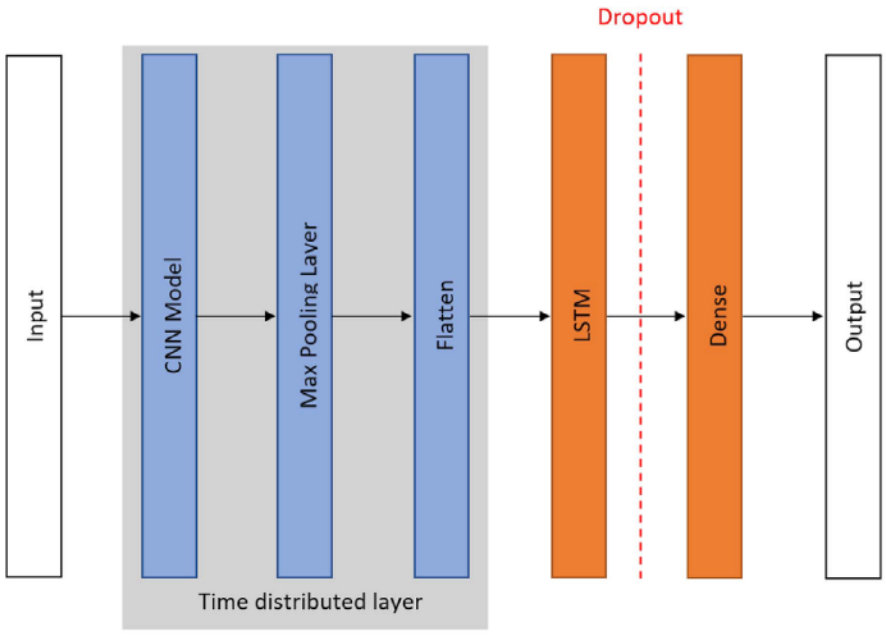

Figure 4. Type II state estimators of PLC X for predicting correlated sensor values of PLC Y.

sensor values. Linear activation ensures unbiased prediction suitable for regression problems in state estimators.

The training data used 37 h of routine operations from the oil terminal. During training, the model used five-fold time series cross-validation, where each fold advanced the validation block forward in time so that the model was never validated on the past data points. To prevent temporal leakage, the stream was split in arrival order into 70% training, 15% validation, and 15% hold-out test segments. A grid search explored the LSTM hidden units (64, 128, 200, 256), CNN kernels (3, 5, 7), dropout (0, 0.2, 0.3), and learning rate (1×10^{-3}, 5×10^{-4}, 1×10^{-4}). The configuration with 200 hidden LSTM units, a kernel length of five, a 0.2 dropout value, and a 5×10^{-4} learning rate provided the lowest validation RMSE.

3.3.3. Confidence interval and detection of anomaly using state estimators

The physical system anomaly detector received the predictions from the state estimators and maintained a dynamic confidence interval for each estimator. The dynamic confidence interval of a state estimator is the standard deviation of a rolling window containing 20 of its latest predictions. The dynamic confidence interval acts as a threshold while measuring the variation of the state prediction from the current sensor values [30]. If the sensor value is out of the confidence interval the physical system anomaly detector calls the incidence response system to log the false sensor value.

4. Case studies and evaluation

This section describes the effectiveness of the embedded intrusion detection system inside PLCs controlling a simulated midstream oil terminal [31, 32]. The midstream oil terminal included twelve operational stations adhering to the American Petroleum Institute (API) standards [33–39]. The stations used OpenPLC to control the mid-stream oil terminal processes. OpenPLC is an IEC 61,131–3 compliant open-source industrial controller for cybersecurity research [19]. Being open-source, researchers can access and modify the source code to test novel cybersecurity methodologies. OpenPLC also matches the comparison benchmark of commercial PLCs in metrics like real-time scan operations, ladder logic arithmetic, SCADA connectivity, and cyberattack response. Alves *et al.* provided a comprehensive comparison of OpenPLC with commercial PLCS like Schneider M221, Siemens S7-1214C, Omron CP1L-L20DR-D, and Allen-Bradley (A-B) MicroLogix 1400 [19]. Prior research on the midstream oil terminal testbed using OpenPLC as the controller device showed high-fidelity responses during cyber-attack scenarios and illustrated cyber-attacks' impact on connected processes [31].

The case studies in this section focused on three cargo operations in the mid-stream oil terminal: Tanker Truck loading operation, where an attacker compromised a single sensor across one PLC; Pipeline Transfer (PLT) operation, where an attacker compromised five sensors across one PLC; and Marine Tanker (MT) loading operation, where an attacker compromised three sensors across three PLCs.

These cases' studies were motivated by historical attacks and scenarios from existing literature. The case studies assumed that the attacker had access to the industrial sensor network and could falsify the sensor response. The attack privileges were similar to the Irongate scenario, which targeted industrial control systems (ICS) and spoofed sensor data to hide malicious activities [40]. Yang *et al.* [41] also outlined a similar sensor spoofing scenario where the attacker injected fake measurements into sensor networks to mislead decision-making. Studies ([40] and [41]) have shown scenarios where the attacker had complete control over the sensor responses. Attacks like Stuxnet have shown that similar sensor spoofing could be achieved even in an air-gapped network, where an attacker used infected USB drives to gain control over the network and eventually manipulated the programmable logic controller [42, 43, 44].

4.1. Tanker truck loading operation—(attack on single sensor)

The Tanker Truck (TT) loading operation involved three subsystems of the midstream oil terminal: Tank Farm (TF), Pumphouse, and TT gantry. During TT loading, the liquid cargo moved from the TF to the TT gantry through the

pumphouse. One of the adverse effects of cyber-attacks during the TT loading operation is an overflow scenario [31]. The midstream oil terminal simulated the sensor network in MATLAB. During the attack scenario, MATLAB activated a malicious Simulink block which delayed the level sensor (P6SL1) reading of the internal tanks of the TT by 7 s. This resulted in HMI reporting an older value different from the actual reading of the level sensor. The delayed response may have misled the operators in taking an incorrect control decision and create a cargo overflow scenario.

The cyberattack involved a **single PLC** (PLC6) and **spoofed the readings of one level sensor** (P6SL1) connected to it. The cyber attack is represented as Ex-14 in Figure 5.

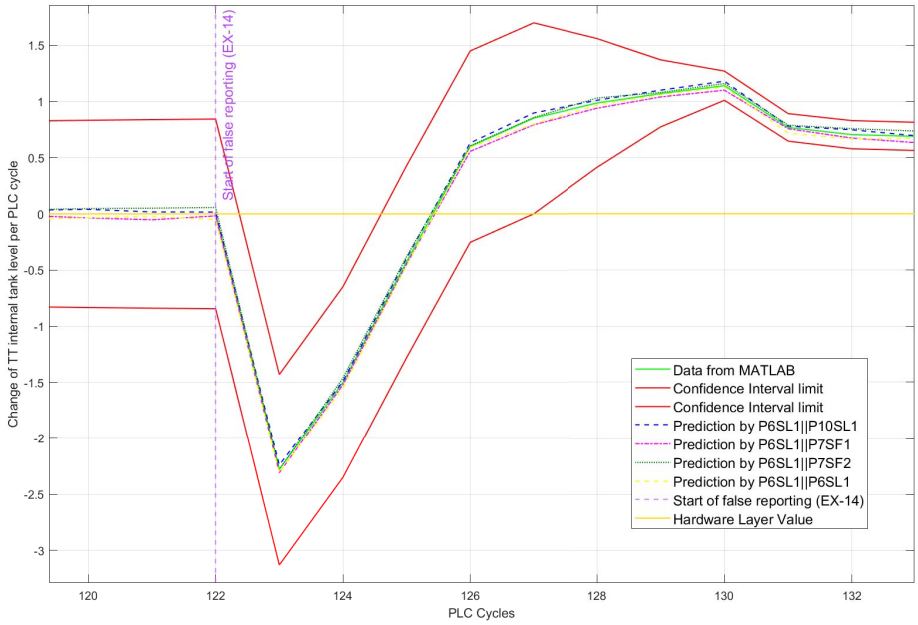

Figure 5. Prediction of the state estimators and the confidence interval of the hardware layer values.

Table 2 shows the list of sensors involved in the gasoline tanker truck loading operation. For each sensor, the embedded IDS has a Type I state estimator. We represent Type I state estimators using a specific nomenclature. The nomenclature uses the name of the sensor whose value is forecasted, followed by a subscript. The subscript represents the name of the sensor that inputs to the state estimator. For example, a Type I state estimator forecasting the value of sensor P10SL1 using the past observation of P10SL1 as represented as $P10SL1_{P10SL1}$.

The number of Type II state estimators depends on the statistical relationship between the sensor readings. Hence, this analysis used a dataset containing 37 h of

Table 2. List of sensors involved in a gasoline TT loading operation.

PLC name	Sensor name	Purpose
PLC10 (Gasoline tank farm)	P10SL1	TK11 Tank level
	P10SP1	Dispatch valve pressure
	P10SF1	Dispatch valve flow rate
PLC7 (Gasoline pump house)	P7SF1	Gasoline pump house inlet flow rate
	P7SF2	Gasoline pump house outlet flow rate
	P7SP1	Gasoline pump house inlet pressure
	P7SP2	Gasoline pump house outlet pressure
PLC6 (TT Gantry)	P6SF1	Bay 1 flow rate sensor
	P6SP1	Bay 1 pressure sensor
	P6SL1	Bay 1 level sensor 1
	P6SL2	Bay 1 level sensor 2

normal operational data of the midstream oil terminal to compute a matrix containing the absolute values of PCC of sensors responses. Table 3 illustrates the matrix with the absolute PCC values of the sensors. A Type II state estimator was constructed for sensors having PCC values higher than or equal to 0.9. The Type II estimator enabled the embedded IDS to predict the states of correlated sensors in neighboring nodes. Table 3 highlights the sensor pairs with PCC values great than equal to 0.9 showing strong linear relationship. For example, P7SF2 shows high linear relationship with P6SL2. A Type II state estimator can be constructed forecasting the value of sensor P7SF2 using past observations of P6SL2 (represented as $P7SF2_{P6SL2}$).

Figure 5 shows the predictions of the state estimators and the confidence intervals. The false reporting started at the 122nd cycle of the PLC and the hardware layer replaced the current observations of sensor P6SL1 with older values. The state estimators forecasting the response of P6SL1 predicted a value different from the hardware layer. Every estimator flagged it as an anomalous reading because the value of the hardware layer was out of the confidence interval.

Table 4 shows the predictions of the state estimator, actual values from MATLAB simulation, and the false readings from the hardware layer. Few state estimators flagged PLC cycle 125 and 126 as normal because the false readings overlapped with the actual values. The Type I states estimators in PLC 6 and the Type II state estimators in PLC 7 and 10 detected the false readings of sensor P6SL1.

4.2. Pipeline transfer (PLT) operation—(attack on multiple sensors)

The PLT operation transfers liquid cargo from a shore-side oil refinery to the mid-stream oil terminal using a 150 kilometer (km) pipeline. The pipeline transports

Table 3. Absolute values of Pearson Correlation Coefficient (PCC) of sensors.

	P10SL1	P10SP1	P10SF1	P7SF1	P7SP1	P7SF2	P7SP2	P6SF1	P6SP1	P6SL1	P6SL2
P10SL1				0.6251	0.9307	0.4751	0.3473	0.3156	0.1735	0.9482	0.9541
P10SP1				0.8215	0.9732	0.9213	0.4213	0.9012	0.7542	0.1247	0.1057
P10SF1				0.9644	0.6154	0.9124	0.5124	0.9249	0.6973	0.8503	0.8147
P7SF1	0.6251	0.8215	0.9644					0.9732	0.1743	0.9455	0.9403
P7SP1	0.9307	0.9532	0.6154					0.1745	0.0178	0.0943	0.0156
P7SF2	0.4751	0.9213	0.9124					0.9721	0.7149	0.9566	0.9512
P7SP2	0.3473	0.4213	0.5124					0.4541	0.7931	0.3150	0.4965
P6SF1	0.3156	0.9012	0.9249	0.9732	0.1745	0.9721	0.4541				
P6SP1	0.1735	0.7542	0.6973	0.1743	0.0178	0.7149	0.7931				
P6SL1	0.9482	0.1247	0.8503	0.9455	0.0943	0.9566	0.3150				
P6SL2	0.9541	0.1057	0.8147	0.9403	0.0156	0.9512	0.4965				

Table 4. Behavior of the state estimators during the cyber-attack.

				PLC Cycles									
				Normal	Normal	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor	Spoofed Sensor
				Cycle 121	Cycle 122	Cycle 123	Cycle 124	Cycle 125	Cycle 126	Cycle 127	Cycle 128	Cycle 129	Cycle 130
State Estimator Location	State Estimator Type	Sensor P6SL1	MATLAB Values (ground truth)	-5.27e-11	-4.25e-11	-2.28	-1.5	-0.424	0.6	0.852	0.985	1.07	1.14
			Hardware Layer	-5.27e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11	-4.25e-11
PLC 10	Type II	$P6SL1_{P10SL1}$	Predicted	-5.27e-11	-4.25e-11	-2.24	-1.49	-0.394	0.632	0.898	1.01	1.1	1.18
PLC 7	Type II	$P6SL1_{P7SF1}$	Predicted	-5.27e-11	-4.25e-11	-2.31	-1.52	-0.446	0.557	0.793	0.939	1.04	1.1
		$P6SL1_{P7SF2}$	Predicted	-5.27e-11	-4.25e-11	-2.28	-1.46	-0.379	0.606	0.858	1.03	1.08	1.16
PLC 6	Type I	$P6SL1_{P6SL1}$	Predicted	-5.27e-11	-4.25e-11	-2.31	-1.55	-0.46	0.592	0.799	0.966	1.06	1.13

Within Confidence interval Outside Confidence interval

the liquid cargo into one tank in the tank farm. Five pressure sensors (P3SP1, P3SP2, P3SP3, P3SP4 and P3SP5) monitor the state of the pipeline during the PLT operation. The attacker closed a valve at one end of the pipeline and spoofed the readings of the pressure sensors. The closing of the valve increased the pressure inside the pipeline. The operators do not see this change of state because of the false readings of the pressure sensors. This attack can eventually lead to a pipeline rupture similar to the Trans-Siberian explosion [45].

The cyber-attack involved a **single PLC** (PLC3) and **spoofed the readings of five pressure sensors** (P3SP1, P3SP2, P3SP3, P3SP4, and P3SP5).

In PLT operation, PLC 3 manages the 150 km pipeline. The pipeline transports the liquid cargo into a gasoline tank farm. Table 5 shows the list of sensors connected to PLC 3 and PLC 10. For each sensor, the embedded IDS has a Type I state estimator. Similar to Case study I, we constructed Type II state estimators for sensors pairs having PCC values higher than or equal to 0.9. Table 6 shows a matrix with the absolute PCC values of the sensors.

Table 5. List of sensors involved in a gasoline PLT operation.

PLC name	Sensor name	Purpose
PLC3 (Pipeline transfer)	P3SP1	Terminal side pressure
	P3SF1	Terminal side flow rate
	P3SP2	30 km from terminal pressure
	P3SF2	30 km from terminal flow rate
	P3SP3	60 km from terminal pressure
	P3SF3	60 km from terminal flow rate
	P3SP4	90 km from terminal pressure
	P3SF4	90 km from terminal flow rate
	P3SP5	120 km from terminal pressure
	P3SF5	120 km from terminal flow rate
PLC10 (Gasoline tank farm)	P10SP2	TK11 Receipt line pressure
	P10SF2	TK11 Receipt line flow rate
	P10SL1	TK11 Tank Level

Table 6. Absolute values of Pearson Correlation Coefficient (PCC) of sensors.

	P3SP1	P3SF1	P3SP2	P3SF2	P3SP3	P3SF3	P3SP4	P3SF4	P3SP5	P3SF5	P10SP2	P10SF2	P10SL1
P3SP1											0.9254	0.9569	0.194
P3SF1											0.1504	0.9721	0.9679
P3SP2											0.6301	0.8739	0.2005
P3SF2											0.2951	0.9831	0.9437
P3SP3											0.7239	0.796	0.154
P3SF3											0.1093	0.9732	0.9475
P3SP4											0.6501	0.6134	0.46
P3SF4											0.305	0.9243	0.918
P3SP5											0.6723	0.5987	0.1793
P3SF5											0.2109	0.9155	0.9172
P10SP2	0.9254	0.1504	0.6301	0.2951	0.7239	0.1093	0.6501	0.305	0.6723	0.2109			
P10SF2	0.9569	0.9721	0.8739	0.9831	0.796	0.9732	0.6134	0.9243	0.5987	0.9155			
P10SL1	0.194	0.9679	0.2005	0.9437	0.154	0.9475	0.46	0.918	0.1793	0.9172			

Figures 6, 7, 8, 9, and 10 show the predictions of the state estimators and the confidence intervals. The false reporting started after the 2nd cycle of the PLC and the hardware layer replaced the current observations of pressure sensors with older values. The state estimators forecasting the response predicted a value different from the hardware layer. Every estimator flagged it as an anomalous reading because the value of the hardware layer was out of the confidence interval. During such scenarios, instead of using the anomalous readings from the hardware layer, the

Type I state estimators $\mathbf{P3SP1}_{\mathbf{P3SP1}}$, $\mathbf{P3SP2}_{\mathbf{P3SP2}}$, $\mathbf{P3SP3}_{\mathbf{P3SP3}}$, $\mathbf{P3SP4}_{\mathbf{P3SP4}}$, and $\mathbf{P3SP5}_{\mathbf{P3SP5}}$ used its predictions as input.

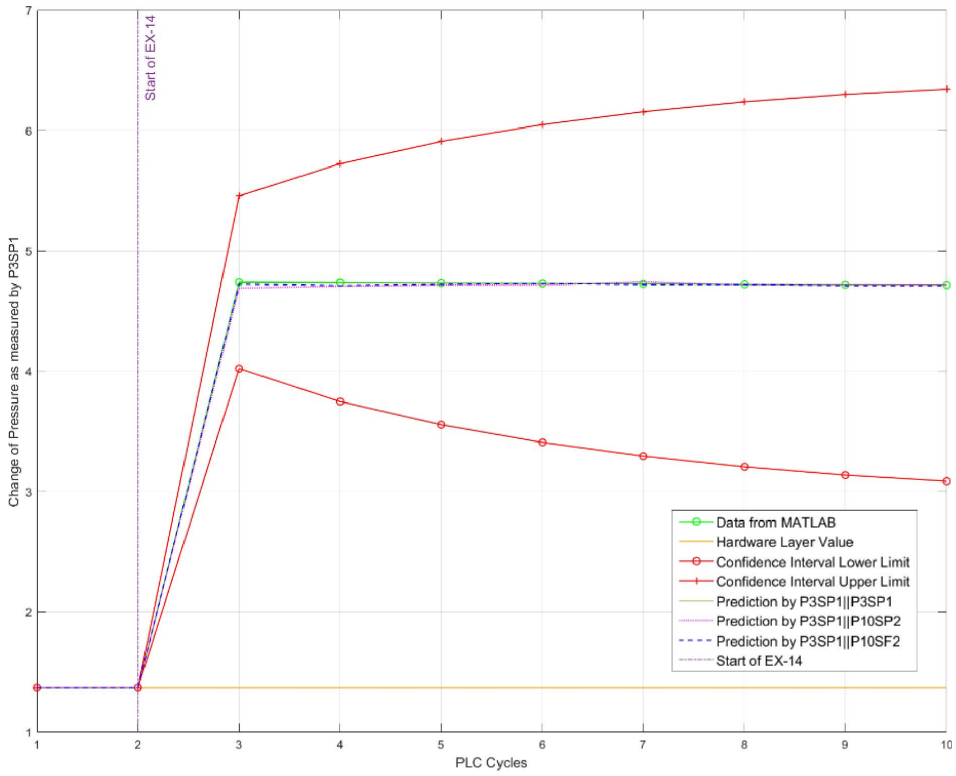

Figure 6. Prediction of the state estimators and the confidence interval of the hardware layer values.

Table 7 shows the responses of the state estimators during the cyber-attack. The false readings were from the pressure sensor of PLC 3. The Type II state estimators $\mathbf{P3SP1}_{\mathbf{P10SP2}}$ and $\mathbf{P3SP1}_{\mathbf{P10SF2}}$ placed inside PLC 10 successfully identified the incorrect values. The successful detection demonstrates that the proposed IDS effectively identifies cyberattacks on neighboring nodes managing interdependent processes. The Type I state estimators inside PLC3 were able to flag the false values from all sensors.

4.3. Marine tanker (MT) loading operation—(attack on multiple sensors across multiple PLCs)

The MT loading operation involves four subsystems: gasoline tank farm (PLC 10), gasoline pump house (PLC 7), loading marine tanker (PLC 5), and 12.5 kilometers (km) terminal-to-jetty pipeline (PLC1). The centrifugal pumps in the gasoline pump house transfer liquid cargo from the gasoline tank farm to the internal tanks of the Marine Tanker (MT) using the 12.5 km terminal-to-jetty pipeline. The attack

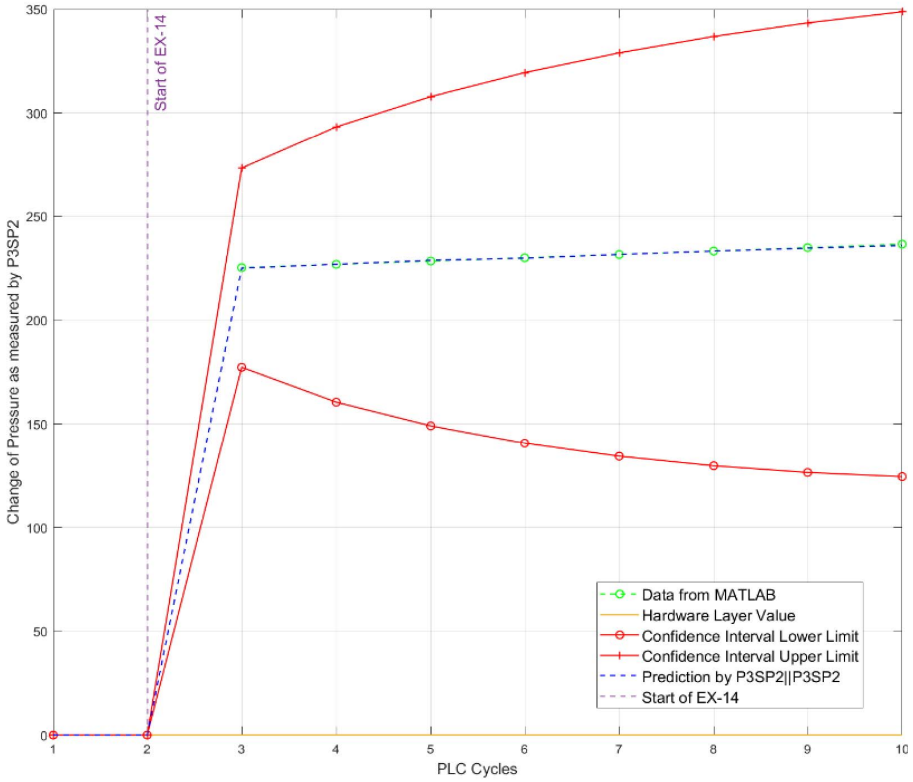

Figure 7. Prediction of the state estimators and the confidence interval of the hardware layer values.

scenario involved three PLCs during the MT loading operation: PLC 1, PLC 5, and PLC 10. The attacker closed a valve at one end of the jetty-to-terminal pipeline and spoofed the readings of the pressure sensors P1SP1, P7SP2, and P10SP7. Such scenarios can increase the pressure in the pipeline and cause pipeline rupture [31].

The cyberattack involved **three PLCs** (PLC10, PLC7, and PLC1) and **spoofed the readings of three pressure sensors** (P1SP1, P7SP2, and P10SP7).

Table 8 shows the list of sensors connected to PLC 1, PLC 5, PLC 7, and PLC 10. The embedded IDS inside each PLCs have a Type I state estimator. Type II state estimators were built for sensors having PCC values higher than or equal to 0.9. Table 9 shows a matrix with the absolute PCC values of the sensors.

Table 10 shows the response of the state estimators during the cyberattack. The false readings were from the pressure sensors connected to PLC 1, 7, and 10. During the attack, the Type II state estimators in PLC 5 identified the incorrect values from the pressure sensor connected to the neighboring node (PLC1). Additionally, the Type I state estimators inside PLC 1, 7, and 10 detected the incorrect values from pressure sensors connected. Figure 11 and 12 shows the predictions of the state estimators and the confidence intervals.

Table 7. Behavior of the state estimators during the cyberattack.

				PLC Cycles									
				Normal	Normal	Spoofed Reading	Spoofed Reading	Spoofed Reading	Spoofed Reading	Spoofed Reading	Spoofed Reading	Spoofed Reading	Spoofed Reading
	Sensor/Notation	MATLAB (Ground truth) /Hardware Layer		Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10
	Sensor P3SP1	MATLAB Values		1.371	1.371	4.7391	4.7354	4.7317	4.7282	4.7246	4.7212	4.7177	4.7144
		Hardware Layer		1.371	1.371	1.371	1.371	1.371	1.371	1.371	1.371	1.371	1.371
	Sensor P3SP2	MATLAB Values		0.0983	0.0983	225.3551	226.9389	228.5181	230.0972	231.6909	233.3256	234.9768	236.639
		Hardware Layer		0.0983	0.0983	0.0983	0.0983	0.0983	0.0983	0.0983	0.0983	0.0983	0.0983
	Sensor P3SP3	MATLAB Values		0.1613	0.1613	2.6323	3.4267	4.2086	4.8766	5.713	6.9122	7.4099	7.7066
		Hardware Layer		0.1613	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613
	Sensor P3SP4	MATLAB Values		0.4218	0.4218	1.9535	2.0838	2.2364	2.4053	2.5594	2.6412	2.7074	2.7714
		Hardware Layer		0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218	0.4218
	Sensor P3SP5	MATLAB Values		0.8893	0.8893	29.3407	29.3437	29.3571	29.3511	29.355	29.3602	29.3649	29.3697
		Hardware Layer		0.8893	0.8893	0.8893	0.8893	0.8893	0.8893	0.8893	0.8893	0.8893	0.8893
State Estimator Location	State Estimator Type	State Estimator Notation											
PLC3	Type I	$P3SP1_{P3SP1}$	Predicted	1.371	1.3711	4.732	4.7304	4.7301	4.73	4.729	4.723	4.7206	4.7181
		$P3SP2_{P3SP2}$	Predicted	0.0991	0.0997	225.0772	226.9749	228.9087	229.9065	231.7052	233.301	234.7206	236.0045
		$P3SP3_{P3SP3}$	Predicted	0.16	0.1601	2.4021	3.2209	4.1906	4.9731	5.679	6.792	7.5001	7.6061
		$P3SP4_{P3SP4}$	Predicted	0.4301	0.43	1.8991	1.973	2.2206	2.399	2.4919	2.7092	2.6907	2.7019
		$P3SP5_{P3SP5}$	Predicted	0.878	0.88	28.9346	29.07947	29.504	29.35	29.35	29.359	29.375	29.2909
PLC 10	Type II	$P3SP1_{P10SP2}$	Predicted	1.37	1.3705	4.6905	4.7069	4.7149	4.7146	4.7412	4.7154	4.715	4.7161
		$P3SP1_{P10SF2}$	Predicted	1.371	1.371	4.7237	4.7103	4.7214	4.7279	4.72	4.7194	4.7105	4.7101

Within Confidence interval Outside Confidence interval

Table 8. List of sensors involved in a gasoline PLT operation.

PLC name	Sensor name	Purpose
PLC1 (Loading marine tanker pipeline)	P1SP1	Terminal side pressure
	P1SF1	Terminal side flow rate
PLC5 (Loading marine tanker)	P5SP1	Marine tanker manifold pressure
	P5SF1	Marine tanker manifold flow rate
	P5SL1	Tank P1 Level
	P5SL2	Tank P2 Level
	P5SL3	Tank P3 Level
	P5SL4	Tank S1 Level
	P5SL5	Tank S2 Level
	P5SL6	Tank S3 Level
PLC7 (Gasoline pump house)	P7SF1	Gasoline pump house inlet flow rate
	P7SF2	Gasoline pump house outlet flow rate
	P7SP1	Gasoline pump house inlet pressure
	P7SP2	Gasoline pump house outlet pressure
PLC10 (Gasoline tank farm)	P10SP7	TK13 Dispatch line pressure
	P10SF7	TK13 Dispatch line flow rate
	P10SL3	TK13 Tank Level

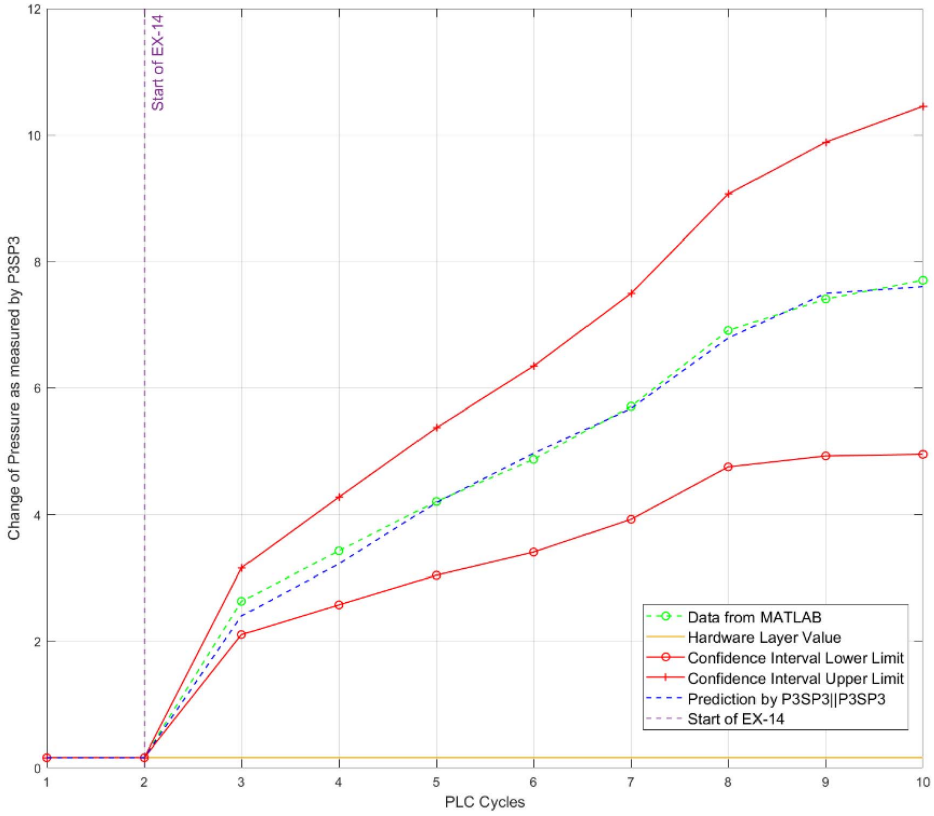

Figure 8. Prediction of the state estimators and the confidence interval of the hardware layer values.

Table 9. Absolute values of Pearson Correlation Coefficient (PCC) of sensors.

	P1SP1	P1SF1	P5SP1	P5SF1	P5SL1	P5SL2	P5SL3	P5SL4	P5SL5	P5SL6	P7SF1	P7SF2	P7SP1	P7SP2	P10SP7	P10SF7	P10SL3
P1SP1			0.9613	0.6743	0.3703	0.6014	0.1907	0.1864	0.1005	0.145	0.4831	0.9913	0.371	0.9973	0.291	0.1391	0.0027
P1SF1			0.7931	0.9869	0.914	0.9034	0.9347	0.9064	0.954	0.9017	0.8571	0.9993	0.6713	0.6412	0.1789	0.013	0.1006
P5SP1	0.9613	0.7931									0.0051	0.6597	0.1343	0.7430	0.1456	0.723	0.731
P5SF1	0.6743	0.9869									0.9301	0.9451	0.3219	0.4160	0.0129	0.0397	0.364
P5SL1	0.3703	0.914									0.673	0.9702	0.0947	0.112	0.71	0.1332	0.1131
P5SL2	0.6014	0.9034									0.7649	0.8901	0.3641	0.1360	0.354	0.6311	0.1145
P5SL3	0.1907	0.9347									0.8561	0.9313	0.0541	0.213	0.1546	0.546	0.1165
P5SL4	0.1864	0.9064									0.731	0.9064	0.0721	0.7045	0.1655	0.1457	0.1364
P5SL5	0.1005	0.954									0.8101	0.9431	0.0746	0.7601	0.3694	0.151	0.1516
P5SL6	0.145	0.9017									0.643	0.8846	0.03	0.1114	0.1664	0.1564	0.616
P7SF1	0.4831	0.8571	0.0051	0.9301	0.673	0.7649	0.8561	0.731	0.8101	0.643					0.1092	0.983	0.9136
P7SF2	0.9913	0.9993	0.6597	0.9451	0.9702	0.8901	0.9313	0.9064	0.9431	0.8846					0.306	0.9703	0.8012
P7SP1	0.371	0.6713	0.1343	0.3219	0.0947	0.3641	0.0541	0.0721	0.0746	0.03					0.956	0.034	0.0633
P7SP2	0.9973	0.6412	0.743	0.416	0.112	0.136	0.213	0.7045	0.7601	0.1114					0.013	0.1501	0.2643
P10SP7	0.291	0.1789	0.643	0.0129	0.71	0.354	0.1546	0.1655	0.3646	0.1654	0.1092	0.306	0.956	0.013			
P10SF7	0.1391	0.013	0.723	0.0397	0.1332	0.6311	0.546	0.1457	0.151	0.1564	0.983	0.9703	0.034	0.1501			
P10SL3	0.0027	0.1006	0.731	0.364	0.1131	0.1145	0.1165	0.1364	0.1516	0.616	0.9136	0.8012	0.0633	0.2643			

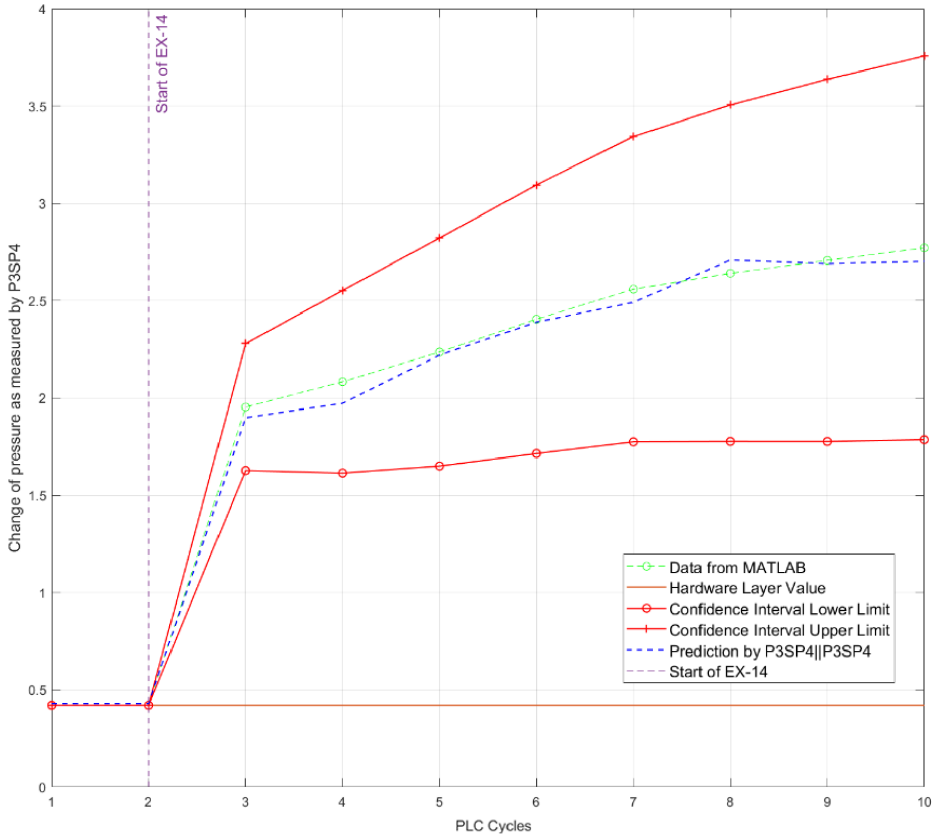

Figure 9. Prediction of the state estimators and the confidence interval of the hardware layer values.

Table 10. Behavior of the state estimators during the cyber-attack.

				PLC Cycles									
				Normal	Normal	Spoofed reading	Spoofed reading	Spoofed reading	Spoofed reading	Spoofed reading	Spoofed reading	Spoofed reading	Spoofed reading
				Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10
		Sensor	MATLAB (Ground Truth)/Hardware										
		Sensor P1SP1	MATLAB Values	-0.00004	-0.00004	2.98E-08	4.76E-08	8.15E-08	8.3E-08	7.99E-08	7.6E-08	7.28E-08	6.98E-08
		Sensor P1SP1	Hardware Layer	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-4E-05	-0.00004	-0.00004
		Sensor P7SP2	MATLAB Values	-0.00004	-0.00004	43022.08	72232.179	247666.6	272423	254843.6119	171334	123474.06	85512.771
		Sensor P7SP2	Hardware Layer	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-0.00004	-4E-05	-0.00004	-0.00004
		Sensor P10SP7	MATLAB Values	-1.8668	-1.8668	-643.394	-677.004	-689.0399	-496.194	-246.412	-0.5124	199.1277	330.3051
		Sensor P10SP7	Hardware Layer	-1.8668	-1.8667	-1.8667	-1.8667	-1.8667	-1.8667	-1.8667	-1.8667	-1.8667	-1.8667
State Estimator Location	State Estimator Type	State Estimator Notation											
PLC 1	Type I	$P1SP1_{P1SP1}$	Predicted	-0.0019	-0.002	0.0016	0.0015	0.0014	0.0007	0.0006	0.0002	0.0001	0.0001
	Type II	$P7SP2_{P1SP1}$	Predicted	-0.0045	-0.0032	42371.2	72612.001	235817.4	270310	225166.0373	168390	125178.9	91912.4
PLC 5	Type II	$P10SP7_{P5SP1}$	Predicted	-0.0028	0.0031	0.005	0.00514	0.00534	0.00594	0.00672	0.00691	0.00731	0.00781
PLC 7	Type I	$P7SP2_{P7SP2}$	Predicted	-0.0025	-0.0031	42051.09	52789.106	228009.3	269201	223997.4397	155533	101542.48	72451.112
	Type II	$P1SP1_{P7SP2}$	Predicted	-0.00011	-0.00054	0.004	0.0051	0.0059	0.00679	0.00796	0.00801	0.00812	0.00829
		$P1SP1_{P7SP2}$	Predicted	-0.00015	-0.00034	0.0029	0.0042	0.00515	0.00643	0.00685	0.00756	0.00768	0.00809
		$P10SP7_{P7SP1}$	Predicted	-1.8601	-1.871	-651.732	-691.0049	-676.7064	-522.109	-260.3104	-30.754	180.9431	315.4521
PLC 10	Type I	$P7SP7_{P10SP7}$	Predicted	-1.863	-1.864	-666.418	-693.4173	-674.1751	-470.414	-231.1465	-50.165	215.1647	324.1443

Within Confidence interval Outside Confidence interval

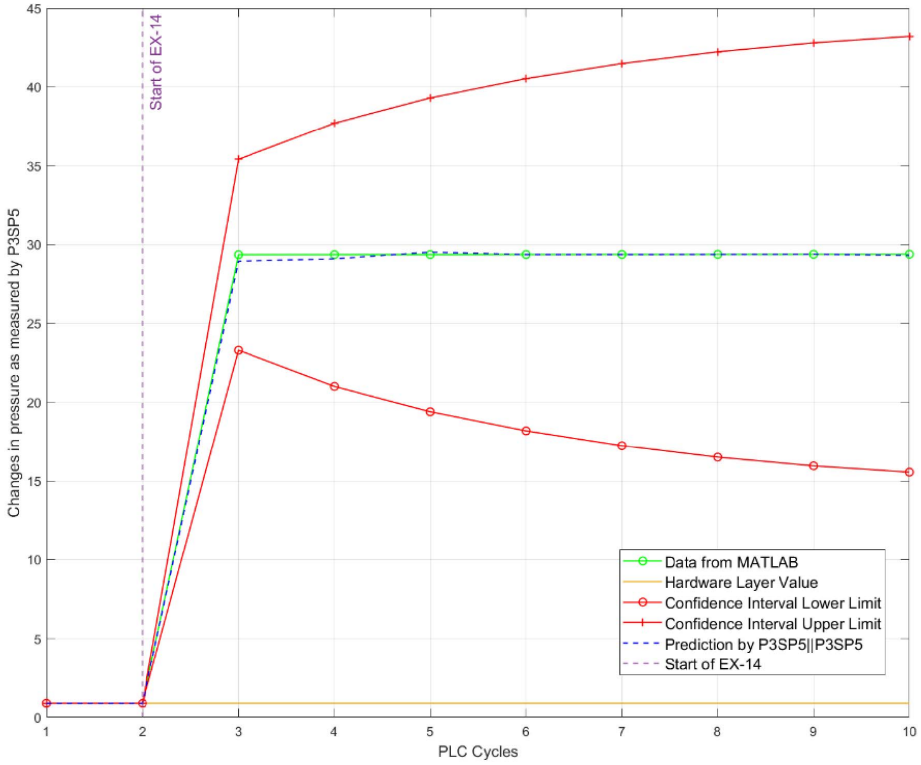

Figure 10. Prediction of the state estimators and the confidence interval of the hardware layer values.

5. *Impact on PLC performance*

This section describes two experiments to assess the impact of embedded IDS on the performance of the PLC: the first experiment measured the time taken by the embedded IDS to log a false sensor reading, the second experiment analyzed the effect of the embedded IDS on the real time performance of the PLC.

5.1. Response time of the embedded IDS

This analysis measured the time taken by the embedded intrusion detection system to respond to a false sensor reading. The embedded IDS operated inside the PLCs of the midstream oil terminal. The experiment repeated the attack scenario of case study I ten times during the TT loading operation. Each instance of the experiment measured the peer-to-peer network latency, time taken by the ensemble state estimators, and time taken by the incident response system. Figure 13 illustrates the average, median, and maximum response time of the embedded IDS.

The embedded IDS took an average time of 33.922 ms to predict and log a false sensor reading. The analysis also revealed that the maximum time taken to respond

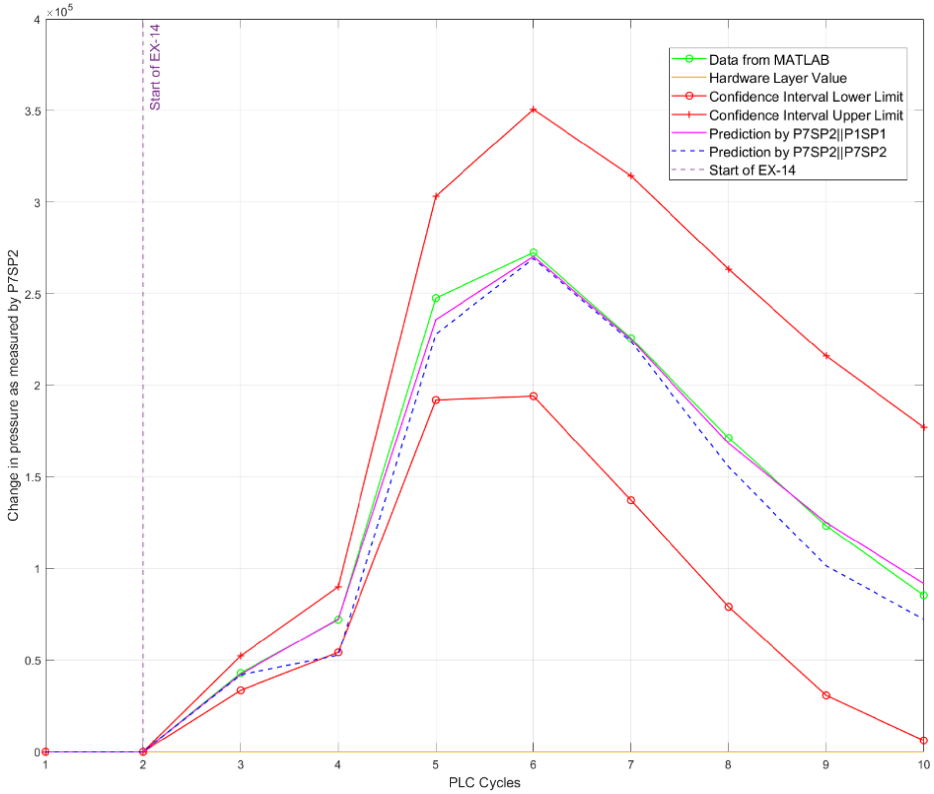

Figure 11. Prediction of the state estimators and the confidence interval of the hardware layer values.

was 40.3 ms. The disparity in response time was because of the network latency of the P2P network.

5.2. Effect On real-time performance of the PLC

This analysis repeated the experiment of case studies and documented the effect of embedded IDS with the physical system anomaly detector on the real-time performance of PLCs. The embedded IDS operated inside PLC 6 of the midstream oil terminal. The cycle time of the PLC was set to 50 ms, and an embedded logger monitored the cycle time for a period of 24 h. This experiment was conducted twice to examine the cycle time data, both without the IDS and with the embedded physical system anomaly detector.

Table 11 outlines the standard deviation and the average cycle of the PLC. The PLC maintained a mean cycle time of 50.04 ms and a standard deviation of 0.0142. The experiments showed that the embedded IDS had no impact on the real-time performance of the PLC.

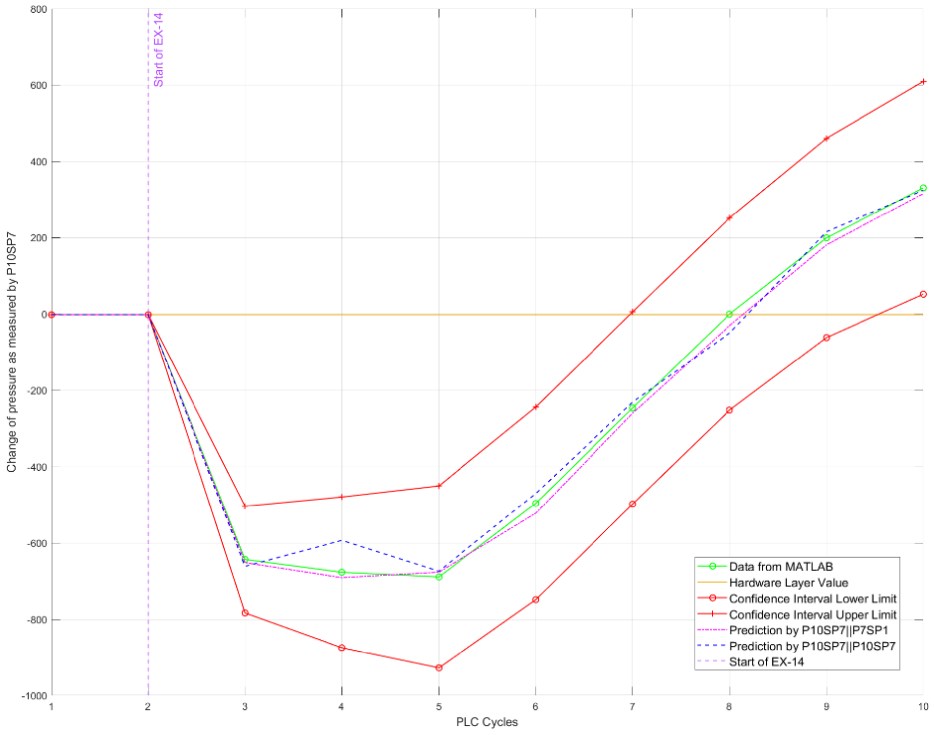

Figure 12. Prediction of the state estimators and the confidence interval of the hardware layer value.

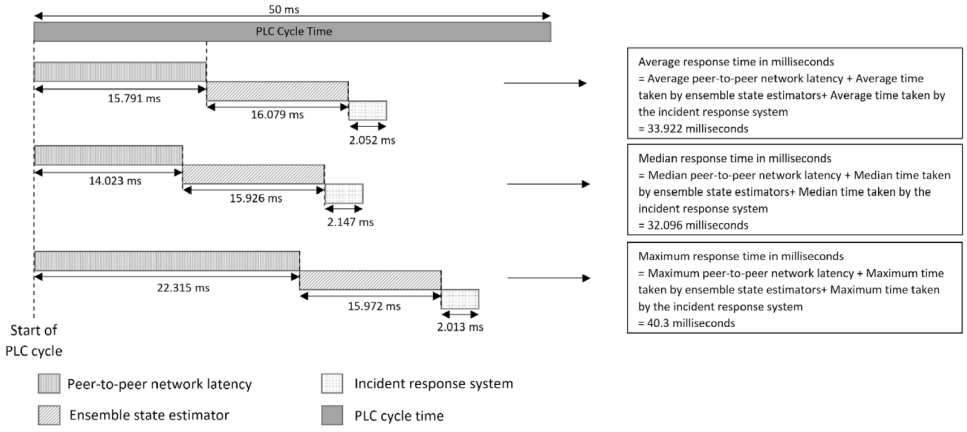

Figure 13. Response time of the embedded intrusion detection system.

Table 11. PLC cycle time in milliseconds.

PLC	Average	Standard deviation
PLC without IDS	50.04	0.0142
PLC with embedded network IDS and physical system anomaly detector	50.04	0.0142

6. Conclusion

This study elucidated the implementation of a distributed embedded intrusion detection system. The embedded IDS operated inside the PLCs and used a peer-to-peer network to share the physical states of the PLC with its neighboring nodes. Using the states of the PLCs, a CNN-LSTM-based state estimator predicted the potential values of the sensors. The embedded IDS categorized a sensor value as incorrect if the values received from the sensor network were out of the prediction interval of the state estimators and generated an alert. In interdependent processes with correlated sensor values, the embedded IDS can predict the state of the sensors connected to neighboring PLCs, facilitating the detection of attacks on connected systems.

This work used a to-scale midstream oil terminal to test the embedded IDS. Through three case studies involving tanker truck loading operations, pipeline transfer operations, and marine tanker loading, the IDS successfully detected falsified sensor data across single-sensor, multi-sensor, and multi-node spoofing. Two performance analyses determined that the proposed IDS achieved detection latencies within the operational constraints of programmable logic controllers (PLCs) without impacting real-time system operations. This study provides a robust framework for securing interconnected ICS controlling complex critical infrastructure.

The embedded IDS uses time-synchronized multivariate data streams (sensor, actuator, and control-flag values) that most modern SCADA installation archives for routine historical or alarm management purposes. The 1-D CNN layers learn the spatial co-activation patterns, and the LSTM network captures temporal dynamics independent of the absolute cycle time. Neither layer requires explicit knowledge of the physical system and learns the patterns in dimensionless sequences. As the CNN-LSTM system parameters are agnostic to physical units, the same model architecture can be effective on another plant whose tags relate to different physical processes. The hybrid CNN-LSTM combination makes the system process-agnostic and can scale across a broad range of critical infrastructure, including water, power, chemical, and discrete-manufacturing SCADA systems.

7. Limitations and future work

This research explored the detection of a cyberattack on a midstream oil terminal. The novel methodology identified falsified sensor readings on interconnected PLCs. Future work can investigate methodologies to enhance the outlined intrusion detection systems. Researchers can improve the latency of the peer-to-peer network to achieve more efficient data sharing between peer nodes. A lower latency in the

peer-to-peer network can enable the IDS to work within PLCs with lower cycle times. Another research path can compare the performance of different algorithms for state estimation. A comparison helps to identify algorithms that achieve faster performance with higher accuracy. Researchers can also use ICSs from different domains to investigate the performance of the IDS.

Besides exploring IDS enhancements, future work can investigate residual attack vectors arising from platform vulnerabilities, time synchronization attacks, and resource hogging attacks. An unpatched PLC with platform vulnerabilities like buffer overflow bugs, scan-cycle denial-of-service flaws, and remote code execution can compromise the PLC. The embedded IDS relies on a time-aligned tag stream. A time synchronization attack that skews NTP clocks can degrade detection performance without altering process variables. The proposed system may be vulnerable to resource exhaustion attacks like CPU hogging or memory by fragmentation. Resource exhaustion can force the IDS to shed load or miss scan deadlines.

Author contributions

Das, Rishabh: Conceptualization, Data curation, Investigation, Methodology, Visualization, Writing – original draft; **Werth, Aaron:** Writing – review & editing, Writing – original draft; **Morris, Thomas:** Funding acquisition, Investigation, Methodology, Supervision, Writing – review & editing.

Funding

This work was supported at least in part by the National Science Foundation through Grants 1623657 and 1431484.

Ethical statement

Not applicable.

Data availability

Source data is not available for this article.

Conflicts of interest

The authors declare no conflict of interest.

Acknowledgements

Parts of this manuscript were previously published in the doctoral thesis by the same author: Rishabh Das. An embedded defense-in-depth module for detecting cyberattacks on interdependent SCADA controllers. 2020. University of Alabama in Huntsville. Available at: https://louis.uah.edu/uah-dissertations/204.

References

1 Altuhafi AW. A review on peer-to-peer live video streaming topology. *Int J Comput Appl*. 2013;**68**(5):6–14. doi:10.5120/11573-6881.

2 Combita LF, Cardenas A, Quijano N. Mitigating sensor attacks against industrial control systems. *IEEE Access*. 2019;**7**: 92444–92455. doi:10.1109/ACCESS.2019.2927484.

3 Wang Q, Tai W, Tang Y, Ni M. Review of the false data injection attack against the cyber-physical power system. *IET Cyber-Phys Syst: Theory Appl*. 2019;**4**(2):101–107. doi:10.1049/iet-cps.2018.5022.

4 Ahmed M, Pathan ASK. False data injection attack (FDIA): an overview and new metrics for fair evaluation of its countermeasure. *Complex Adapt Syst Model*. 2020;**8**(1):4. doi:10.1186/s40294-020-00070-w.

5 Chromik JJ. Process-aware SCADA traffic monitoring: a local approach [DSI Ph.D. Thesis Series; 19-009]. Enschede: University of Twente; 2019. 231 p. doi:10.3990/1.9789036548014.

6 Chromik JJ, Remke A, Haverkort BR. Improving SCADA security of a local process with a power grid model. In: *Proceedings of the 4th International Symposium for ICS & SCADA Cyber Security Research*. BCS Learning and Development Ltd.; 2016. p. 1–10. doi:10.14236/ewic/ICS2016.13.

7 Giraldo J, Urbina D, Cardenas A, Valente J, Faisal M, Ruths J, et al. A survey of physics-based attack detection in cyber-physical systems. *ACM Comput Surv*. 2018;**51**(4):1–36. doi:10.1145/3203245.

8 Cárdenas AA, Amin S, Lin ZS, Huang YL, Huang CY, Sastry S. Attacks against process control systems. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security - ASIACCS '11*. Association for Computing Machinery; 2011. p. 355–366. doi:10.1145/1966913.1966959.

9 Huang YL, Cárdenas AA, Amin S, Lin ZS, Tsai HY, Sastry S. Understanding the physical and economic consequences of attacks on control systems. *Int J Crit Infrastruct Prot*. 2009;**2**(3):73–83. doi:10.1016/j.ijcip.2009.06.001.

10 Nafees MN, Saxena N, Cardenas A, Grijalva S, Burnap P. Smart grid cyber-physical situational awareness of complex operational technology attacks: a review. *ACM Comput Surv*. 2023;**55**(10):1–36. doi:10.1145/3565570.

11 Gaggero G, Girdinio P, Marchese M. Artificial intelligence and physics-based anomaly detection in the smart grid: a survey. *IEEE Access*. 2025;**13**: 23597–23606. doi:10.1109/ACCESS.2025.3537410.

12 Chromik JJ, Remke A, Haverkort BR. An integrated testbed for locally monitoring SCADA systems in smart grids. *Energy Inform*. 2018;**1**: 1–29. 56. doi:10.1186/s42162-018-0058-7.

13 Hadžiosmanović D, Sommer R, Zambon E, Hartel P. Through the eye of the PLC: semantic security monitoring for industrial processes. In: *Proceedings of the 30th Annual Computer Security Applications Conference*. Association for Computing Machinery; 2014. p. 126–135. doi:10.1145/2664243.2664277.

14 Kumar BP, Hariharan K, Shanmugam R, Shriram S, Sridhar J. Enabling internet of things in road traffic forecasting with deep learning models. *J Intell Fuzzy Syst*. 2022;**43**(5):6265–6276. doi:10.3233/JIFS-220230.

15 Dwivedi S, Attry A, Parekh D, Singla K. Analysis and forecasting of time-series data using S-ARIMA, CNN and LSTM. In: *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE; 2021. p. 131–136. doi:10.1109/ICCCIS51004.2021.9397134.

16 Chen M. Comparative analysis of forecasting Chevron's crude oil stock performance with machine learning techniques. *Adv Econom Management Political Sci*. 2024;**86**(1):21–27. doi:10.54254/2754-1169/86/20240935.

17 Liang L. ARIMA with attention-based CNN-LSTM and XGBoost hybrid model for stock prediction in the US stock market. *SHS Web Conf*. 2024;**196**: 02001. doi:10.1051/shsconf/202419602001.

18 Mohammad Ata KI, Hassan MK, Ismaeel AG, Al-Haddad SAR, Alquthami T, Alani S. A multi-Layer CNN-GRUSKIP model based on transformer for spatial-TEMPORAL traffic flow prediction. *Ain Shams Eng J*. 2024;**15**(12):103045. doi:10.1016/j.asej.2024.103045.

19 Alves T, Morris T. OpenPLC: an IEC 61,131–3 compliant open source industrial controller for cyber security research. *Comput Secur*. 2018;**78**: 364–379. doi:10.1016/j.cose.2018.07.007.

20 Sur S, Srimani PK. A depth-first search routing algorithm for star graphs and its performance evaluation. *Math Comput Model*. 1994;**19**(9):35–52. doi:10.1016/0895-7177(94)90039-6.

21 Ma Y, Tan Z, Chang G, Gao XA. P2P network topology optimized algorithm based on minimum maximum K-means principle. In: *2009 Ninth International Conference on Hybrid Intelligent Systems*. IEEE; 2009. p. 396–399. doi:10.1109/HIS.2009.193.

22 Condie T, Kamvar S, Garcia-Molina H. Adaptive peer-to-peer topologies. In: *Proceedings of the Fourth International Conference on Peer-to-Peer Computing*. IEEE Computer Society; 2004. p. 53–62. doi:10.1109/PTP.2004.1334931.

23 Xu Y, Chi D, Min G. *The topology of P2P network*, vol. 3 (8), Mianyang, Sichuan, China: School of Information Engineering, Southwest University of Science and Technology; 2012.

24 Pearson K. VII. Note on regression and inheritance in the case of two parents. *Proc R Soc. Lond*. 1895;**58**: 240–242. doi:10.1098/rspl.1895.0041.

25 Li L, Lu Z, Zhou C. Importance analysis for models with correlated input variables by the state dependent parameters method. *Comput Math Appl*. 2011;**62**(12):4547–4556. doi:10.1016/j.camwa.2011.10.034.

26 Donahue J, Hendricks L, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Trans Pattern Anal Mach. Intell*. 2017;**39**(4):677–691. doi:10.1109/TPAMI.2016.2599174.

27 Vinyals O, Toshev A, Bengio S, Erhan D. Show and tell: a neural image caption generator. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2015. p. 3156–3164. doi:10.1109/CVPR.2015.7298935.

28 Olah C. Understanding LSTM networks. Colah's Blog [Internet]. 2015 [cited 2025 Jun 14]. Available from: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

29 3.2. Tuning the hyper-parameters of an estimator—scikit-learn 0.21.3 documentation. Scikit-learn.org [Internet]. 2019 [cited 2025 Jun 14]. Available from: https://scikit-learn.org/0.21/modules/grid_search.html.

30 Zhu L, Laptev N. Deep and confident prediction for time series at Uber. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE; 2017. p. 103–110. doi:10.1109/ICDMW.2017.19.

31 Das R, Morris T. Modeling a midstream oil terminal for cyber security risk evaluation. In: Staggs J, Shenoi S, editors. *Critical Infrastructure Protection XII. ICCIP 2018. IFIP Advances in Information and Communication Technology*. vol. 542, Springer; 2018. doi:10.1007/978-3-030-04537-1_9.

32 Das R. An embedded defense-in-depth module for detecting cyberattacks on interdependent SCADA controllers [dissertations]. 2020; 204 p. https://louis.uah.edu/uah-dissertations/204.

33 American Petroleum Institute. *Specification for Electric Motor Prime Mover for Beam Pumping Unit Service*. 1st ed. API SPEC 11L6, Washington, DC: API; 1993.

34 American Petroleum Institute. *Specification for End Closures, Connectors and Swivels*. 2nd ed. API SPEC 6H, Washington, DC: API; 1998.

35 American Petroleum Institute. *Specification for Line Pipe*. 43rd ed. API SPEC 5L, Washington, DC: API; 2004.

36 American Petroleum Institute. *Specification for Bolted Tanks for Storage of Production Liquids*. 15th ed. API SPEC 12B, Washington, DC: API; 2008.

37 American Petroleum Institute. *Specification for Pipeline Valves*. 23rd ed. API SPEC 6D, Washington, DC: API; 2008.

38 American Petroleum Institute. *Loading and Unloading of MC 306/DOT 406 Cargo Tank Motor Vehicles*. API RP 1007, Washington, DC: API; 2011.

39 American Petroleum Institute. *Line Markers and Signage for Hazardous Liquid Pipelines and Facilities*. 5th ed. API RP 1109, Washington, DC: API; 2017.

40 Tian J, Tan R, Guan X, Xu Z, Liu T. Moving target defense approach to detecting stuxnet-like attacks. *IEEE Trans Smart Grid*. 2020;**11**(1):291–300. doi:10.1109/TSG.2019.2921245.

41 Yang N, Zhong Y, Li Y, Shi L. Model-unknown spoofing attack via false data injections. In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE; 2023. p. 1814–1819. doi:10.1109/CDC49753.2023.10383617.

42 Masood R, Um-e-Ghazia , Anwar Z. SWAM: Stuxnet worm analysis in metasploit. In: *2011 Frontiers of Information Technology*. IEEE; 2011. p. 142–147. doi:10.1109/FIT.2011.34.

43 Lindsay JR. Stuxnet and the limits of cyber warfare. *Secur Stud*. 2013;**22**(3):365–404. doi:10.1080/09636412.2013.816122.

44 Langner R. Stuxnet: dissecting a cyberwarfare weapon. *IEEE Secur Priv*. 2011;**9**(3):49–51. doi:10.1109/MSP.2011.67.

45 Banks W. Developing Norms for Cyber Conflict (February 22, 2016) [Internet]. doi:10.2139/ssrn.2736456.