IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 July 2023, accepted 3 August 2023, date of publication 7 August 2023, date of current version 18 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3303015

RESEARCH ARTICLE

A Machine Learning Based Framework for Real-Time Detection and Mitigation of Sensor False Data Injection Cyber-Physical Attacks in Industrial Control Systems

MARIAM ELNOUR[1], MOHAMMAD NOORIZADEH[1], MOHAMMAD SHAKERPOUR[2], NADER MESKIN[1], (Senior Member, IEEE), KHALED KHAN[2], (Senior Member, IEEE), AND RAJ JAIN[3], (Life Fellow, IEEE)

[1]Department of Electrical Engineering, Qatar University, Doha, Qatar
[2]Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[3]Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA

Corresponding author: Nader Meskin (nader.meskin@qu.edu.qa)

This work was supported by National Priorities Research Programme (NPRP) through the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP 10-0206-170360. The open-access publication of this article was funded by Qatar National Library.

ABSTRACT In light of the advancement of the technologies used in industrial control systems, securing their operation has become crucial, primarily since their activity is consistently associated with integral elements related to the environment, the safety and health of people, the economy, and many others. This work presents a distributed, machine learning based attack detection and mitigation framework for sensor false data injection cyber-physical attacks in industrial control systems. It is developed using the system's standard operational data and validated using a hybrid testbed of a reverse osmosis plant. A MATLAB/Simulink-based simulation model of the process validated with actual data from a local plant is used. The control system is implemented using Siemens S7-1200 programmable logic controllers with 200SP Distributed Input/Output modules. The proposed solution can be adopted in the existing industrial control systems and demonstrated effective performance in real-time detection and mitigation of actual cyber-physical attacks launched by compromising the communication links between the process and the programmable logic controllers.

INDEX TERMS Attack detection, attack mitigation, industrial control system (ICS), false data injection (FDI), support vector machine (SVM).

I. INTRODUCTION

Industrial control systems (ICSs) are automation systems used in social and critical infrastructures, manufacturing and industrial facilities, etc. They play essential roles in realizing their control functions and ensuring their safety. They consist of electrical devices, mechanical devices, and computers. They include manual operations supervised by humans. Additionally, several types of control systems and instrumentation are used to control and regulate the industrial process, such as programmable logic controllers (PLCs), distributed control systems (DCS), supervisory control and data acquisition (SCADA) systems, process control systems (PCS), and others.

In light of the advancement and sophistication in the Internet of Things (IoT) technologies deployed in ICSs, securing their operation has become increasingly critical and disconcerting. That being indicated, several cyber-attack incidents on critical infrastructures were recorded in the past years [1], [2], [3], [4], [5], [6]. The structure and sophistication of attacks are constantly evolving as advanced means are deployed to launch stealthy and deleterious attacks. Energy industries and critical infrastructures are among the most vulnerable domains of cyber-attacks [7]. Last year, an electricity

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Yu.

grid in India was compromised, resulting in a massive power outage [8], while the Ukrainian power grid was subjected to an unsuccessful cyber-attack by the Russians in 2022, aiming to cause blackouts affecting about two million people [9]. Additionally, a cyber-attack disrupted the operation in the European oil facilities [10], and another took place in a middle east-based petrochemical plant in 2017 [11]. The operation of ICSs is consistently associated with crucial elements related to the environment, the safety and health of people, the economy, and many others [12].

The research in the cybersecurity of ICSs has been evolving and ongoing towards achieving solutions to identify the occurrence of attacks and potentially mitigate their effects on the system. Even though extensive research has been conducted regarding attack detection and mitigation in ICSs and industrial IoT systems, the majority are concerned with DoS attacks in software-defined networks (SDNs) such as in [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22] using data-based approaches. In [23], a supervised machine learning based intrusion detection system was proposed. A rule-based mitigation strategy was adopted for low-rate DoS attacks in an SDN environment emulated using Mininet, a tool for creating virtual network topologies. Moreover, the authors in [24] and [25] surveyed several works for DoS attack detection and mitigation in SDNs using other approaches such as pattern recognition, machine learning, rule-based, fuzzy logic, etc.

The false data injection (FDI) attack was initially introduced for ICSs of power systems, specifically in the smart grid domain [26]. Generally, an attacker compromises sensor readings to disrupt the industrial process or drive the process's state to instability. Due to the rapid growth of industrial IoT systems and today's increasingly critical cyber-world of networked systems, FDI attacks are considered one of the top-priority issues [27]. FDI attacks are critical in ICSs as they control delicate processes that usually have alarming environmental, social, and/or economic impacts.

A. PREVIOUS WORKS

Several works were developed to detect FDI attacks as surveyed in [28], presenting various state-of-the-art machine learning based FDI attack detection strategies for power systems. Model-based approaches were extensively used to incorporate the mitigation problem as in [29], in which an observer-based framework was proposed and applied for smart grid systems. In [30], joint static state and dynamic state estimation models were used to detect FDI attacks using weighted least squares (WLS) and extended Kalman filter (EKF) with exponential weighting function (WEKF) to improve the robustness.

In [31], the performance and resilience of a linear cyber-physical control system (CPCS) with attack detection and reactive attack mitigation were investigated for power grids. It addressed the problem of deriving an optimal sequence of FDI attacks that maximizes the state estimation error of the power system. In [32], an attack detection and mitigation strategy for ICSs using Kalman filters (KFs) was presented for sensor FDI attacks and validated using a simulation model of a three-tank system. In addition, model-based FDI attack detection and mitigation approaches using observers and KFs were proposed for power systems in [33], [34], and [35], and simulation tools were used for validation.

In [36], an observer-based resilient control strategy was developed for variable-speed wind turbines against FDI attacks. In [37], a multi-agent model-based DoS and FDI attack detection and mitigation framework was proposed using the physical and the cyber characteristics of the plant for distribution of the automation system in power systems, and in [38], state estimation was used to detect and mitigate the same type of hybrid attacks in multi-area power systems. The authors in [39] investigated the dynamic event-triggered fuzzy control of DC microgrids with FDI attacks and imperfect premise matching. Moreover, recently, transfer learning was applied for anomaly detection in ICSs to solve the problem of limited and/or imbalanced datasets [40].

Additionally, data-driven approaches and machine learning algorithms were employed to develop FDI attack detection systems as in [41] for industrial IoT systems, in [42], and [43] for power systems, and in [44] for a water treatment plant. A neural network (NN)-based attack mitigation strategy was proposed in [50] for power systems. A FDI attack detection and mitigation approach was implemented for power systems in [52] based on Kullback-Liebler (KL) divergence. The detection was made based on the discrepancy between the Gaussian distributions of the actual and expected data. Then a self-belief value was generated to modify the distributed control protocol accordingly using Raspberry Pi modules. A supervised data-driven analytical method employing a margin setting algorithm (MSA) was demonstrated in [45] using simulation tools to detect FDI cyber-physical attacks in microgrids. In addition, in [46], a supervised detection strategy was proposed utilizing a NN-based autoencoder (AE) and an Extra Trees (ETs) classifier for FDI attacks detection in smart grids aiming to address the computational complexity issue of data-driven approaches.

In [47], a NN-based distributed intrusion detection for FDI attacks in smart grids was proposed to address the over-fitting issue of machine learning based approaches for large-scale systems. A supervised approach was presented in [48] for FDI attacks in energy management systems of smart power grids, while in [49], a transformer and long short-term memory (LSTM) networks-based detection framework was deployed for the same system. Additionally, in [57], a gated recurrent unit (GRU) NN-based control strategy was proposed to eliminate FDI attacks in DC microgrids. From the communication network perspective, a hidden Markov model (HMM) was employed in [51] to develop a detection and prediction module to monitor the IoT devices supported by a distributed trust management module to establish trust between devices. Finally, a bandwidth optimization problem was formulated for trusted devices' bandwidth allocation.

TABLE 1. Previous studies in FDI attack detection and mitigation for ICSs.

Reference	Problem	Approach	System	Type of attacks	Validation
[29]	Detection	Model-based	Power system	FDI	Simulation
[30]	Detection	Model-based	Power system	FDI	Simulation
[31]	Detection & Mitigation	Model-based	Power system	FDI	Simulation
[32]	Detection & Mitigation	Model-based	ICS	FDI	Simulation
[33]–[35]	Detection & Mitigation	Model-based	Power system	FDI	Simulation
[36]	Detection & Mitigation	Model-based	Wind turbines	FDI	Simulation
[37]	Detection & Mitigation	Model-based	Power system	FDI, DoS	Simulation
[38]	Detection & Mitigation	Model-based	Power system	FDI, DoS	Simulation
[39]	Detection & Mitigation	Model-based	Power system	FDI	Hybrid
[41]	Detection	Data-driven (NN)	Industrial IoT	FDI	Simulation
[42]	Detection	Data-driven (NN)	Power system	FDI	Simulation
[43]	Detection	Data-driven (PCA, kNN)	Power system	FDI	Simulation
[44]	Detection	Data-driven (NN)	Water treatment plant	FDI	Practical
[45]	Detection	Data-driven (MSA)	Power system	FDI	Simulation
[46]	Detection	Data-driven (ETs and NN)	Power system	FDI	Simulation
[47]	Detection	Data-driven (NN)	Power system	FDI	Simulation
[48]	Detection	Data-driven (ML)	Power system	FDI	Simulation
[49]	Diagnosis	Data-driven (NN)	Power system	FDI	Simulation
[50]	Mitigation	Data-driven (NN)	Power system	FDI	Simulation
[51]	Detection & Mitigation	Data-driven (HMM)	IoT network	FDI	Simulation
[52]	Detection & Mitigation	Data-driven (KL divergence)	Power system	FDI	Hybrid
[53]	Detection & Mitigation	Model-based and data-driven (NN)	Power system	FDI	Simulation
[54], [55]	Detection & Mitigation	Model-based and data-driven (NN)	Power system	FDI	Simulation
[56]	Detection & Mitigation	Model-based and data-driven (NN)	IoT network	FDI	Hybrid

k-Nearest Neighbor (kNN), Neural Network (NN), Hidden Markov Model (HMM), Principal Component Analysis (PCA), Kullback-Liebler (KL), Margin Setting Algorithm (MSA), Machine Learning (ML).

Moreover, a combination of model-based and data-based approaches was used to detect and mitigate FDI attacks in power systems. For example, in [53], a hyper basis function neural network (HBF-NN)-based observer was proposed to detect, isolate, and mitigate FDI attacks in microgrid systems with electric vehicles (EVs) and it was validated using simulation. The authors in [54] used a model-based parameter estimation model for attack detection and a NN-based forecasting model for mitigation, and [55] employed a KF fused with a three-layer NN-based observer. Additionally, an extended observer-based hybrid tracking control strategy relying on discrete-time sliding function and NNs for a networked system with FDI attacks was proposed in [56].

After examining the existing works for attack detection and mitigation in ICSs, the following observations were noted:

- Several existing works examined the system's status from the network perspective, specifically against DoS attacks, as they are common and known to be challenging to mitigate. However, addressing FDI attacks are as crucial since their impacts on the ICS and the IoT system are consequential and can be very costly,
- Most of the works were validated using simulation tools (e.g., MATLAB/Simulink, virtual networks, etc.) due to their low cost, ease of accessibility, and high flexibility compared to other options. However, more concrete and practical validation means are favorable for more corroborated and substantiated findings,
- The model of the system or expert knowledge were required for several of the presented approaches in the literature. However, data-based approaches provide a superior solution since most recent ICSs and IoT systems have a historian server for continuous data logging,
- Data-driven frameworks either required the availability of labeled training data (i.e., supervised learning), had poor to limited scalability, and/or suffered from high computational overhead.

A summary is presented in TABLE 1 of the existing research works for FDI attack detection and mitigation in ICSs.

B. AIM AND CONTRIBUTION

We aim to tackle the limitations above by applying a conventional machine learning based approach for real-time detection and mitigation of sensor FDI attacks and validating the proposed framework using a hardware setup of the ICS. First, we employ machine learning algorithms to develop a black-box model of the system under study to identify the occurrence of a sensor FDI attack and its magnitude. Machine learning regression algorithms were commonly used to model nonlinear systems for forecasting and estimating the system behavior as in [58] and [59] using support vector machine (SVM), in [60] using k-nearest neighbor, in [61] using decision trees (DTrees), and in [62] and [63] using NNs. The black-box model is intended to provide the expected healthy version of the system dynamics. Then, the detection is made based on the discrepancy between the actual value and the black-box model predicted value. Finally, mitigation is carried out by passing the corrected sensor value to the ICS.

The contribution of this work is the successful application of a machine learning based real-time attack detection and mitigation framework for sensor FDI cyber-physical attacks in ICSs validated on industrial-grade automation hardware from Siemens. It is tested in real-time using a hybrid testbed consisting of a calibrated MATLAB/Simulink model of the process and Siemens S7-1200 PLCs with 200SP Distributed I/O modules on which the proposed framework is implemented, and the cyber-physical FDI attacks are injected. The proposed framework has the following prominent characteristics:

1) Development: It is a scalable and data-driven approach that can be developed using the system's normal operational data only and without the need for knowledge of the system's mathematical model.
2) Flexibility: It is a machine-learning and residual-based framework that can be adjusted by performing thresholds-reassignments and/or models' refinement in case of changes in the environment (e.g., increased noise level) or as necessary.
3) Application: It can be easily adopted into the existing ICSs and integrated into the plant's control system. It is programmed in PLCs from Siemens of a hybrid testbed of a reverse osmosis (RO) plant, a popular ICS. A MATLAB/Simulink-based model of the RO plant that was calibrated and validated with operational data from a local plant is used [64], while the control system is realized using Siemens S7-1200 PLCs with 200SP Distributed I/O modules. The proposed framework is implemented in real-time on the PLCs and validated with actual online cyber-physical attacks that are launched after compromising the communication links between the plant and the PLCs [65].

The paper is organized as follows. First, in Section II, we feature the details of the hybrid testbed used to demonstrate and validate the work. Next, we present the description of the proposed FDI sensor attack detection and mitigation framework, the underlying theory of the machine learning algorithms used, and the development details in Section III. Then, the details of the validation phase of the proposed solution in terms of the evaluation metrics used and the results obtained are presented and demonstrated in Section IV. Finally, conclusions and future work are summarized in Section V.

II. DESCRIPTION OF THE HYBRID REVERSE OSMOSIS TESTBED

The cyber-physical system under study is a two-pass RO plant presented in [64]. As demonstrated in FIGURE 1, it is divided into three processes as described below with the water flow being regulated using motorized valves and pumps:

1) In the pre-treatment stage, the raw feed water is conditioned before entering the RO process. Water filtration and chemical dosing for anti-scaling and adjusting pH levels take place to maintain the lifetime of the RO membrane and the quality of the product water. Firstly, the raw seawater is stored in the water intake storage tank before passing through a DAF system to remove total suspended solids (TSS) from the water stream and then stored in the DAF storage tank. In the next stage, the DAF tank's water is divided into two streams in preparation for the next stage. For each line, a pump is used to force the water through the disk filter to further filter the water before going to the next stage.
2) The RO process is used to remove the salt to produce fresh water. The RO unit has three streams, one is the feed water inflow stream and the other two are the outflow streams, which are the concentrated (brine or reject) solution stream and the product (permeate or freshwater) stream. Pumps provide water flow from the RO storage tanks through the pressure pumps then through the RO 1 unit and the ERD that make the two water streams. The second stream through the ERD aims to increase the efficiency of the RO plant by capturing and utilizing the hydraulic energy from the high pressure reject stream of the RO 1 unit. The product water of the RO 1 unit is pressurized through the RO 2 unit in which the semi-final product water is produced.
3) In the post-treatment stage, the product water of the RO 2 unit is stored and ready for distribution for human use. Ideally, the process involves chemical dosing of minerals, disinfection, and adjusting the pH level. However, those processes are not realizable by simulation means due to their complexity [64].

TABLE 2. List of equipment used in the RO plant simulator.

Stage	Equipment details	
Pre-treatment	Water intake storage tank	
	Water intake pump	
	DAF storage tank	
	Disk filters pressure pumps	
	RO storage tanks	
	RO supply pumps	
RO process	RO units	HP pump
		Pressure booster pump
	ERD	Pressure booster pump
Post-treatment	Product water tank	
	Distribution pump	

The plant process is realized using a simulation model developed in MATLAB/Simulink 2018b, while the control system is implemented using industrial-grade automation hardware from Siemens via Siemens TIA Portal V15. The main details of the RO plant simulator are summarized in TABLE 2 and TABLE 3. The control system of the plant contains twelve PID control loops controlling the level of the tanks (x_1, x_2, x_4, x_5), the flow booster pumps' outlet flow rate (x_{10}, x_{11}, x_{17}), and the pressure booster pumps' outlet pressure (x_8, x_9, x_{12}, x_{13}, x_{16}). A total of five S7-1200 PLCs are used for implementing the control system. ET 200SP Distributed I/O hardware from Siemens interfaces the sensors and actuators with the plant process in each stage. The control system hardware is networked in a PROFINET subnet. The list of controlled and measured variables and control signals

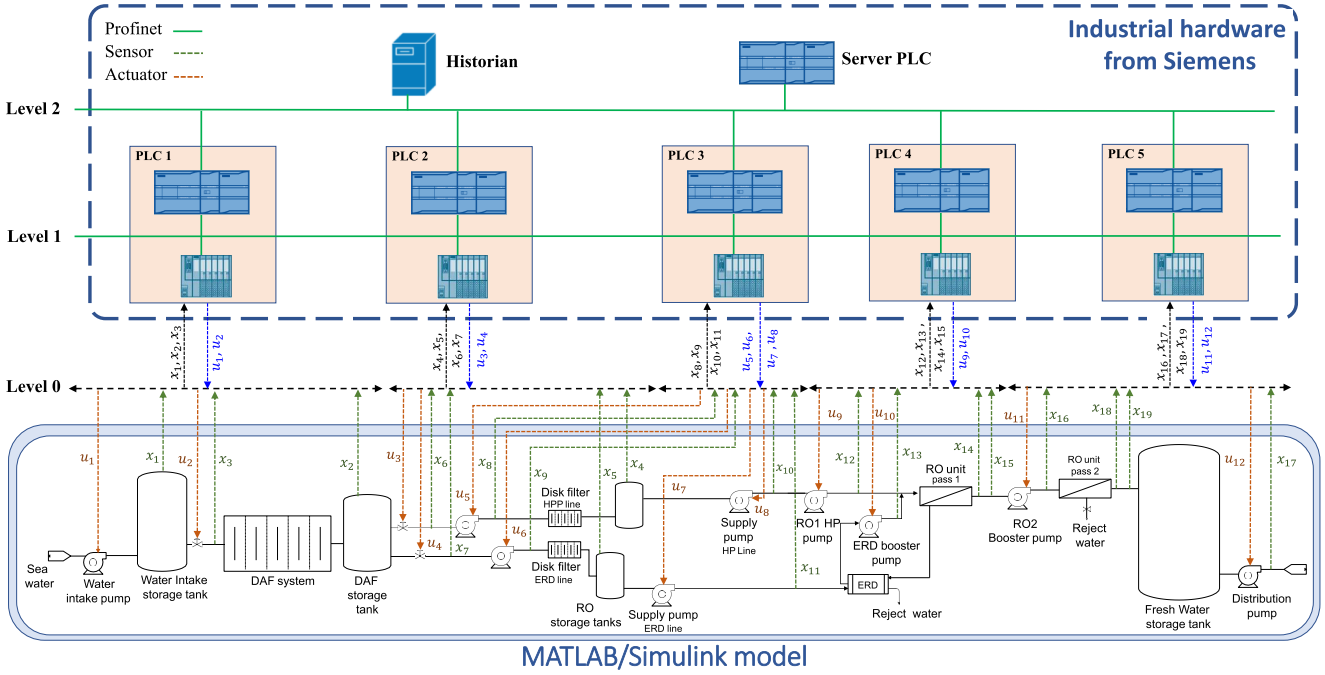


FIGURE 1. The hybrid RO testbed.

TABLE 3. Description of the variables of the RO plant.

PLC	Variable description	Symbol
1	Water level of the intake tank (m)	x_1
	Water level of the DAF tank (m)	x_2
	Outlet flow rate of the intake tank (kg/s)	x_3
2	Water level of the RO tank - HPP line (m)	x_4
	Water level of the RO tank - ERD line (m)	x_5
	Outlet flow rate of the DAF valve - HPP line (kg/s)	x_6
	Outlet flow rate of the DAF valve - ERD line (kg/s)	x_7
3	Outlet pressure of the DF pressure pump - HPP line (bar)	x_8
	Outlet pressure of the DF pressure pump - ERD line (bar)	x_9
	Outlet flow rate of the RO 1 HPP line supply pump (kg/s)	x_{10}
	Outlet flow rate of the RO 1 ERD line supply pump (kg/s)	x_{11}
4	Outlet pressure of the RO 1 pressure pump - HPP line (bar)	x_{12}
	Outlet pressure of the RO 1 pressure pump - ERD line (bar)	x_{13}
	RO 1 product water concentration (ppm)	x_{14}
	RO 1 product water flow rate (kg/s)	x_{15}
5	Outlet pressure of the RO 2 pressure pump (bar)	x_{16}
	Distribution pump flow rate (kg/s)	x_{17}
	RO 2 product water concentration (ppm)	x_{18}
	RO 2 product water flow rate (kg/s)	x_{19}

TABLE 4. The list of variables and control signals of the PLCs of the testbed.

PLC	Control signals	Controlled variables	Measured variables
1	u_1, u_2	x_1, x_2	x_3
2	u_3, u_4	x_4, x_5	x_6, x_7
3	u_5, u_6, u_7, u_8	x_8, x_9, x_{10}, x_{11}	-
4	u_9, u_{10}	x_{12}, x_{13}	x_{14}, x_{15}
5	u_{11}, u_{12}	x_{16}, x_{17}	x_{18}, x_{19}

of each PLC is presented in TABLE 4. The RO plant consists of two types of actuation:

- *Direct actuation*, in which the control system directly modulates the actuator to achieve the control objective such as controlling the outlet pressure/flow rate of a pump as the case with the **flow pumps** x_{10}, x_{11}, and x_{17}, and the **pressure pumps** x_8, x_9, x_{12}, x_{13}, and x_{16}. In these cases, a single input-single output model can be established between the control input and the controlled variable.
- *Indirect actuation* in which the control system modulates an actuator that is indirectly coupled with the controlled variable, e.g., controlling the levels of the tanks. That is, the controller attempts to maintain the tank level by modulating the speed of the flow pump as necessary. In the case of indirect actuation, a single input-single output model between the control input and the controlled variable cannot be established.

III. THE PROPOSED SENSOR FDI ATTACK DETECTION AND MITIGATION FRAMEWORK

A. METHODOLOGY

The basic steps for the application of the proposed sensor FDI attack detection and mitigation framework are as follows:

- Development phase:
  1) Identification of the control loops pertinent to the proposed framework
  2) The design and formulation of the prediction models of the identified control loops
     a) Identification of the inputs and outputs of each of the prediction models
     b) Collection and preprocessing of the historical data of the identified control loops

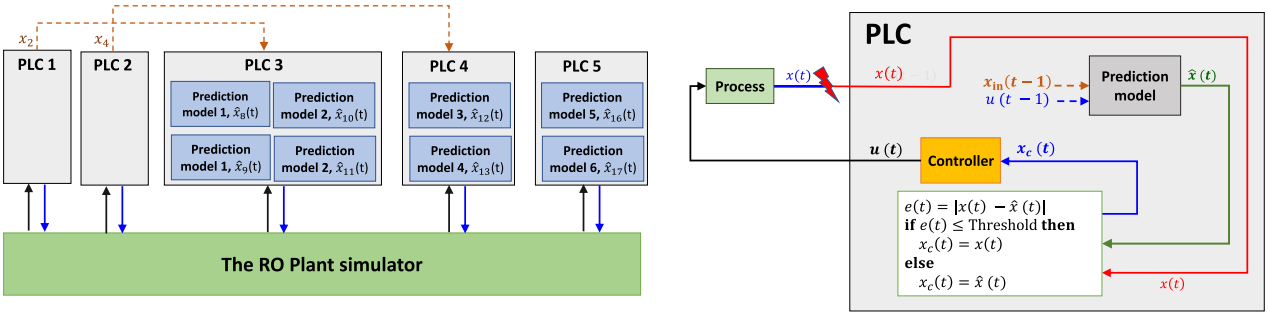


FIGURE 2. The FDI sensor attack detection and mitigation framework consisting of 6 prediction models.

c) Training of the machine learning based prediction models

- Application phase:
  1) Interpretation and expression of the machine learning models in programmable formulations
  2) Incorporation of the programmable formulations of the prediction models in the PLCs
  3) Implementation of the proposed framework's logic in the PLCs

In this work, the proposed sensor FDI attack detection and mitigation framework is applied to mitigate the attacks targeting the eight direct actuation control loops implemented in PLC 3 - PLC 5. As presented in FIGURE 2, it is composed of six dynamical prediction models for the four pressure pumps and the two flow pumps as described in TABLE 5. For example, Prediction model 1 represents the dynamics of the low-pressure pump that is used to boost the water pressure at the inlet of the disk filters in both the HPP and ERD lines in two control loops described by the pairs (x_8,u_5) and (x_9,u_6). Similarly, Prediction model 2 represents the dynamics of the flow pump used to supply water to the HPP and ERD lines in another control loop described by the pairs (x_{10},u_7) and (x_{12},u_8).

The prediction model is used to produce the prediction $\hat{x}(t)$ of the sensor reading $x(t)$ using the past control signal $u(t-1)$ given the past sensor reading $x(t-1)$ and the relevant sensor readings if exist as described in TABLE 6. The latter is determined based on the known physical interdependently between the system's variables. For example, the RO 1 pressure pump's outlet pressure (x_{12}) depends on the pump's inlet pressure, which depends on the level of the RO 1 tank (x_4). Finally, an attack is detected if the absolute difference between the actual and the predicted sensor reading $e(t) = |x(t) - \hat{x}(t)|$ exceeds a predefined threshold. Once a sensor FDI attack is detected, meaning the actual sensor reading has been compromised, the predicted sensor reading is sent to the controller to mitigate the attack effect.

B. TESTBED DATASETS

The training of the prediction models was conducted using datasets collected from the testbed for the system's normal operation at a rate of 1 sample/second. In TABLE 7, the details of the datasets used to develop the prediction models are presented. The size of the datasets varied for the different models for two reasons: 1) The control loops have different time responses, and 2) The dynamical behavior of some pumps has multiple operational modes. For example, the settling time for the RO 2 pressure pump, whose outlet pressure is predicted using Model 5, is about 20 minutes, while the dynamics of the distribution pump, whose outlet flow rate is predicted using Model 6, change at an hourly rate. Additionally, the behavior of some pumps, which are the DF pressure pumps and the RO 1 pressure pump, is dependent on the settings of the water levels of the preceding tanks, which have multiple operational setpoints.

TABLE 5. The description of dynamical prediction models of the plant's pumps.

Model	Pump type	Operating range	Used to predict:
1	DF pressure pump	4 - 15 bar	x_8, x_9
2	DF supply flow pump	1500 - 5000 kg/s	x_{10}, x_{11}
3	RO 1 high pressure pump	20 - 90 bar	x_{12}
4	RO 1 ERD pressure pump	20 - 90 bar	x_{13}
5	RO 2 high pressure pump	10 - 40 bar	x_{16}
6	Distribution flow pump	5000 - 20000 kg/s	x_{17}

TABLE 6. The input-output details of the prediction models of the proposed sensor FDI attack detection and mitigation framework.

PLC	Model	Output ($\hat{x}$)	Input	
			Sensor reading (x_{in})	Control signal (u)
3	1	$\hat{x}_8$	x_2 ,x_8	u_5
	1	$\hat{x}_9$	x_2 ,x_9	u_6
	2	$\hat{x}_{10}$	x_{10}	u_7
	2	$\hat{x}_{11}$	x_{11}	u_8
4	3	$\hat{x}_{12}$	x_4, x_{12}	u_9
	4	$\hat{x}_{13}$	x_{12}, x_{13}	u_{10}
5	5	$\hat{x}_{16}$	x_{16}	u_{11}
	6	$\hat{x}_{17}$	x_{17}	u_{12}

C. DEVELOPMENT OF THE MACHINE LEARNING BASED PREDICTION MODELS

1) DESCRIPTION OF THE MACHINE LEARNING ALGORITHMS

In this section, the underlying theory of the machine learning algorithms used is presented.

TABLE 7. The details of the training datasets for the prediction models of the proposed sensor FDI attack detection and mitigation framework.

Model	Dataset size	Details
1	3000	Combined record of the DF pressure pumps x_8 and x_9 each at 3 different setpoints of the DAF tank level
2	1000	Combined record of the DF supply flow pumps x_{10} and x_{11}
3	400	At 3 different setpoints of the RO 1 HPP line tank level
4	400	At 3 different setpoints of the RO 1 HPP line tank level
5	800	N/A
6	1200	N/A

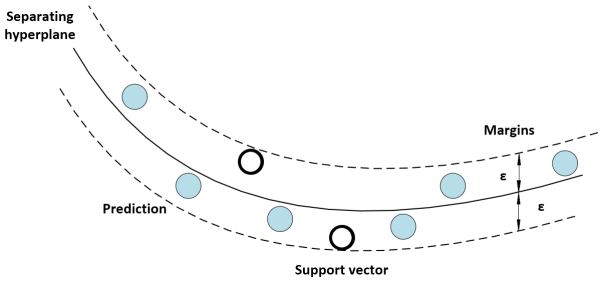


FIGURE 3. Example of SVM regression. The empty circle represents two support vectors [67].

A: Support Vector Machine (SVM)

SVM analysis is a commonly used machine learning algorithm for classification and regression. It is a statistical learning algorithm that uses the concept of decision planes that utilize decision boundaries to optimally separate data into the different categories [66]. Given the regression problem $Y = g(X)$ where $Y = [y_1, y_2, \ldots, y_m] \in \mathbb{R}^{m\times 1}$ and $X = [x_1, x_2, \ldots, x_m] \in \mathbb{R}^{m\times n}$ in which m is the number of training samples and n is the number of data features (attributes), ε-SVM regression model attempts to find a function $f(x)$ that deviates from y_i by a value no greater than ε for each training point x_i, $i = 1, 2, \ldots, m$, and at the same time is as flat as possible. The SVM regression model is:

$$f(x) = \sum_{i=1}^{P} \alpha_i G(sv_i, x) + b, \tag{1}$$

where $P \in \mathbb{R}$ is the number of support vectors, $sv_i \in \mathbb{R}^{1\times n}$ for $i = 1, \ldots, P$ is the i-th support vector, $b \in \mathbb{R}$ is the bias term, $G(*)$ is the kernel function, and α_i, $i = 1, \ldots, P$ are the Lagrangian multipliers. The SVM problem is solved by optimizing the following cost function:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j G(x_i, x_j) + \varepsilon\sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m} y_i\alpha_i,$$

subject to

$$\sum_{k=1}^{m}\alpha_k = 0, \quad \forall i : 0 \le \alpha_k \le C, \tag{2}$$

where $m \in \mathbb{R}$ is the number of training samples, $C \in \mathbb{R}$, and $\varepsilon \in \mathbb{R}$. The hyper-parameters of a SVM regression model are:

- **Kernel function**, $G(x_j, x_k)$: Kernel functions transform the original data to a higher dimensional space where they can be linearly separated.
  1) Polynomial Kernel is $G(x_j, x_k) = (1 + x_i x_k^{\mathrm{T}})^p$, where p is a hyper-parameter representing the order of the polynomial function.
  2) Gaussian Kernel is $G(x_j, x_k) = \exp(||x_j - x_k||^2)$.
- **Epsilon**, ε: It is half of the width of the epsilon-insensitive band demonstrated in FIGURE 3.
- **Box Constraint**, C: It controls the penalty imposed on observations that lie outside the epsilon margin (ε) and works as a regularization parameter to help in preventing over-fitting. It controls the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than ε are accepted.

B: K-Nearest Neighbour (KNN)

kNN regression is a non-parametric algorithm used to approximate the association among independent variables. The output is predicted by local interpolation of the targets associated with the nearest neighbors in the training set. It is a memory-based algorithm and cannot be realized by a closed-form model such that the training samples are required at run-time, and predictions are made directly from the sample relationships [68]. Hence, it is impractical and computationally expensive for large and complex regression problems. The hyper-parameters of the kNN algorithm are:

- **Number of neighbors k:** It is the number of samples closest in distance to the predictors of the target point.
- **Distance**, d: It is the metric used to evaluate the distance between the target's predictors and the k nearest neighbors. Several distance metrics can be used: Cityblock, Chebychev, Correlation, Cosine, Euclidean, Hamming, Jaccard, Mahalanobis Seuclidean, Spearman, or Minkowski with the exponent, E as an additional hyper-parameter associated with this metric. The definitions of the metrics can be found in [69].
- **Distance weight**, w_d: It is the distance weighting function which can be:
  - Equal weight: Each neighbor gets equal weight.
  - Inverse weight: Each neighbor gets a weight of the reciprocal of the distance ($1/d$) between this neighbor and the point being processed.
  - Squared-inverse weight: Each neighbor gets a weight of the reciprocal of squared the distance ($1/d^2$) between this neighbor and the point being processed.

C: Decision Trees (DTrees)

A binary DTree is developed based on a sequential decision process using the classification and regression tree (CART) algorithm. Starting from the root, recursive binary splitting is performed in which at every node of the tree, a feature

is evaluated, and two branches are yielded until a final leaf is reached [67]. The final leaf represents the final output. Decision trees are simple in their dynamics and efficiency. They can be employed to solve regression problems. The objective at each node is to find the best feature and its split value for partitioning the remaining data in the node into one of two regions such that the overall error between the actual output and the predicted output is minimized. The cost function of a regression DTree at the ith node is expressed as:

$$\text{MSE}_i = \frac{1}{m_i}\sum_{j=1}^{m_i}\left(y_j - \hat{y}_i\right)^2,$$
$$\hat{y}_i = \sum_{j=1}^{m_i} y_j, \tag{3}$$

where m_i is the number of training samples in the ith node, y is the actual response, and $\hat{y}$ is the predicted response computed as the average prediction of the node.

- **Maximum number of splits**: It represents the maximum number of decision splits (or branches) in the tree reflecting its maximum depth.
- **Minimum leaf size**: It represents the minimum number of training samples of leaf nodes.
- **Number of variables to sample**: It represents the number of predictors or features to select at random for each split at the node.

D: Neural Networks (NNs)

A feed-forward NN is a directed computational structure that connects an input layer to an output one, as demonstrated in FIGURE 4. The hidden layers are the building blocks of the NN composed of several nodes (or neurons), and together they determine the complexity of the network. Neurons have complete pairwise connections with the adjacent ones, and the output of one layer is the input of the subsequent layer. Those connections are represented by a set of parameters known as the weights **W** and biases **b** that are to be adjusted through an iterative training procedure using the back-propagation algorithm to minimize the cost function [70].

The computation performed at each layer of the network can be expressed as follows:

$$a^{[l]} = f\left(W^{[l]}a^{[l-1]} + b^{[l]}\right), \tag{4}$$

where $a^{[l-1]}$ and $a^{[l]}$ are the input and output of the lth layer, respectively, $f(*)$ is the activation function, $W^{[l]} \in \mathbb{R}^{N_l \times N_{l-1}}$ is the weights matrix of layer l, $b^{[l]} \in \mathbb{R}^{N_l}$ is the bias, and N_i is the number of neurons in the ith layer. For a regression problem, the cost function is as follows:

$$MSE = \frac{1}{m}\sum_{j=1}^{m}\left(y_i - \hat{y}_i\right)^2, \tag{5}$$

where m is the number of training samples, y is the true output, and $\hat{y}$ is the predicted output produced by the network. The hyper-parameters of a NN are as follows:

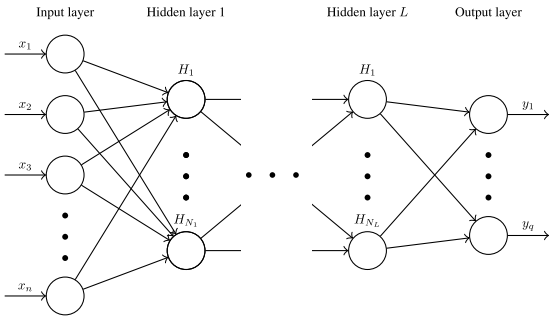


FIGURE 4. A standard fully connected feed-forward NN with n inputs, q outputs, and L layers.

- **Number of layers**, L
- **Number of neurons**, N
- **Learning rate**, α: It controls how much to change the model in response to the estimated error each time the model weights are updated.
- **Activation functions** $f(*)$: It enables the network to learn and approximate complex functional mappings between the inputs and outputs. For every node in the layer, a transformation function is applied to the weighted sum vector to produce the layer's output. There are several types of activation functions, and the most used are:
  - Sigmoid (σ): It is expressed as:

$$y = f(x) = \sigma(x) = \frac{1}{1+\exp(-x)}. \tag{6}$$

  - Hyperbolic tangent (Tanh): It produces an output with values between -1 and 1 and is expressed as:

$$y = f(x) = \tanh(x) = \frac{2}{1+\exp(-2x)} - 1. \tag{7}$$

  - Rectified linear unit (ReLU): It returns 0 if the input is negative and its mathematical form is,

$$y = f(x) = \max(0, x). \tag{8}$$

2) TRAINING OF THE MACHINE LEARNING MODELS

The prediction models were developed using machine learning regression algorithms, which are SVM, kNN, DTrees, and NNs in which a mapping function F was learned to make a prediction of the sensor reading $x(t)$ given $x_{in}(t-1)$ and $u(t-1)$ as mentioned previously. The prediction model is:

$$F = \text{Train\_MLregressor}\left(x(t), [x_{in}(t-1), u(t-1)]\right), \tag{9}$$

and the prediction is computed by:

$$\hat{x}(t) = F\left(x_{in}(t-1), u(t-1)\right). \tag{10}$$

The training of the models using the different machine learning algorithms was conducted using MATLAB 2018b on a PC with 64 GB RAM and 8-core AMD Ryzen 9 3800X CPU with 3.9 GHz speed, and a 64-bit Windows 10 Pro OS.

TABLE 8. The ranges of hyper-parameters of the machine learning algorithms for models' tuning using Bayesian optimization.

Algorithm	Parameter	Range
SVM	Kernel Function (G)	Polynomial, Gaussian
	Polynomial order (p)	1 - 2
	Box constraint (C)	5 - 10000
	Epsilon (ε)	0 - 50
kNN	Number of neighbours (k)	1 - $m/2$
	Distance (d)	Cityblock, Chebychev, Correlation, Cosine, Euclidean, Hamming Jaccard, Mahalanobis, Minkowski, Seuclidean, Spearman
	Distance weight (w_d)	equal, inverse, squared-inverse
	Exponent (E)	0.5 - 3
DTree	Minimum leaf size	1 - $m/2$
	Maximum number of splits	1 - m
	Number of variables to sample	1 - n
NN	Number of layers	1 - 2
	Number of neurons	1 - 20
	Activation function(f)	Sigmoid, ReLU, Tanh
	Learning rate (α)	1.0E-5 - 1.0E-1

where m is the number of training samples, and n is the number of features (or variables).

Data preprocessing was performed on the raw data to normalize the sensor data values in the range of 0 to 1. The optimization of the hyper-parameters was performed using a 10-fold cross-validation and Bayesian optimization algorithm for the hyper-parameters ranges presented in TABLE 8. In Bayesian optimization, a posterior distribution of functions that best describes the objective function is constructed. The optimization algorithm keeps track of past iterations to find better choices for the next set of hyper-parameters to evaluate. As the number of evaluations increases, the posterior distribution improves, and the algorithm becomes more confident in choosing the hyper-parameters set worth exploring [71].

The Bayesian optimization process for training and tuning the prediction models is demonstrated in FIGURE 5, while the final obtained models are presented in TABLE 9. For FIGURES 5 (a) - (e), the x-axis represents the evaluation iteration, and the y-axis represents the optimization objective function, which is the 10-fold cross-validation Mean Squared Error (MSE) between the true and the predicted values. TABLE 10 lists the Root Mean Squared Error (RMSE) of the optimized prediction models on the whole training dataset. It can be noticed that the tuning process was smooth for the six SVM-based, kNN-based, and DTree-based prediction models in which the training converged in less than ten iterations with a maximum of RMSE of 1.3 on average per model, as demonstrated in FIGURE 5 (g). However, for the NN-based models, the convergence was slow, and the optimized cross-validation error was relatively high, as demonstrated in FIGURE 5 (h). It can be attributed to two factors: (i) the limited data used to train the network, and (ii) the diversity of the observations in the data that is to be realized by a 2-layer NN with a maximum of 20 neurons per layer. It is worth noting that we limited the maximum number of layers and the maximum number of neurons per layer to 2 and 20, respectively, to limit the computational complexity of the NN-based model.

IV. EVALUATION AND DISCUSSION

A. INJECTION AND DATA COLLECTION OF FDI ATTACKS ON THE TESTBED

Sensor FDI cyber-physical attacks were launched through the communication channels between the I/O modules and the corresponding PLC under three main assumptions, which are: 1) the adversary had field-level access to the plant, 2) the adversary had sufficient knowledge about the process of the RO plant that qualified to launch the attacks, and 3) the absence of actuator attacks. The full details of the testbed and the sensor FDI cyber-physical attacks' injection methodology can be found in [65].

In this work, scaling sensor attacks were considered in which the channel data $X(t)$ under attack is scaled by a constant factor $\lambda > 0$ of the actual data. The characteristic feature of scaling attacks is introducing a time-varying bias to the measurement signal. The attack magnitude builds up and linearly scales with time based on the sensor's measurements under attack [72]. If not detected and eliminated at an early stage, the impact of scaling attacks can be critical when integral sensors of the industrial process are targeted [73], [74], [75]. The real-time channel data $\widetilde{X}(t)$ subjected to a scaling attack can be expressed as:

$$\widetilde{X}(t) = \begin{cases} X(t) & t < t_a^s \text{ or } t \geq t_a^e \\ \lambda X(t) & t_a^s \leq t < t_a^e \end{cases}, \tag{11}$$

where t_a^s and t_a^e are times of the start and end of the attack, respectively. An attacks dataset was collected to evaluate the ability of the SVM-based, kNN-based, DTree-based, and NN-based prediction models to predict the sensors' readings under routine and attack scenarios and compare their performance. The attack dataset contains 10498 samples, with about 43% of attack data samples. It is worth noting that attack mitigation was not executed during this phase. The attacks were launched by compromising the links between the simulated plant and the PLC I/O modules one at a time. At each time, scaling sensor attacks were injected to all the sensors' measurements involved in the control loops in the PLC under attack using a scaling factor λ in the range of 70 - 150%, as presented in TABLE 11.

B. EVALUATION METRICS

Accuracy is the conventional metric for assessing the capability of prediction models. It is measured by the degree of closeness between the predicted and the actual value. However, especially for data-driven models, more broad and representative evaluation criteria must be considered to account for the models' diverse characteristics in terms of precision, efficiency, generalization ability, etc., stemming from the various characteristics of the algorithms used to develop them. Therefore, the following evaluation metrics were considered:

- **Mean Absolute Error (MAE)**: It measures the average magnitude of the errors between the actual values x_i and the predicted values $\hat{x}_i$ for $i = 1, \ldots, m$, where m is the

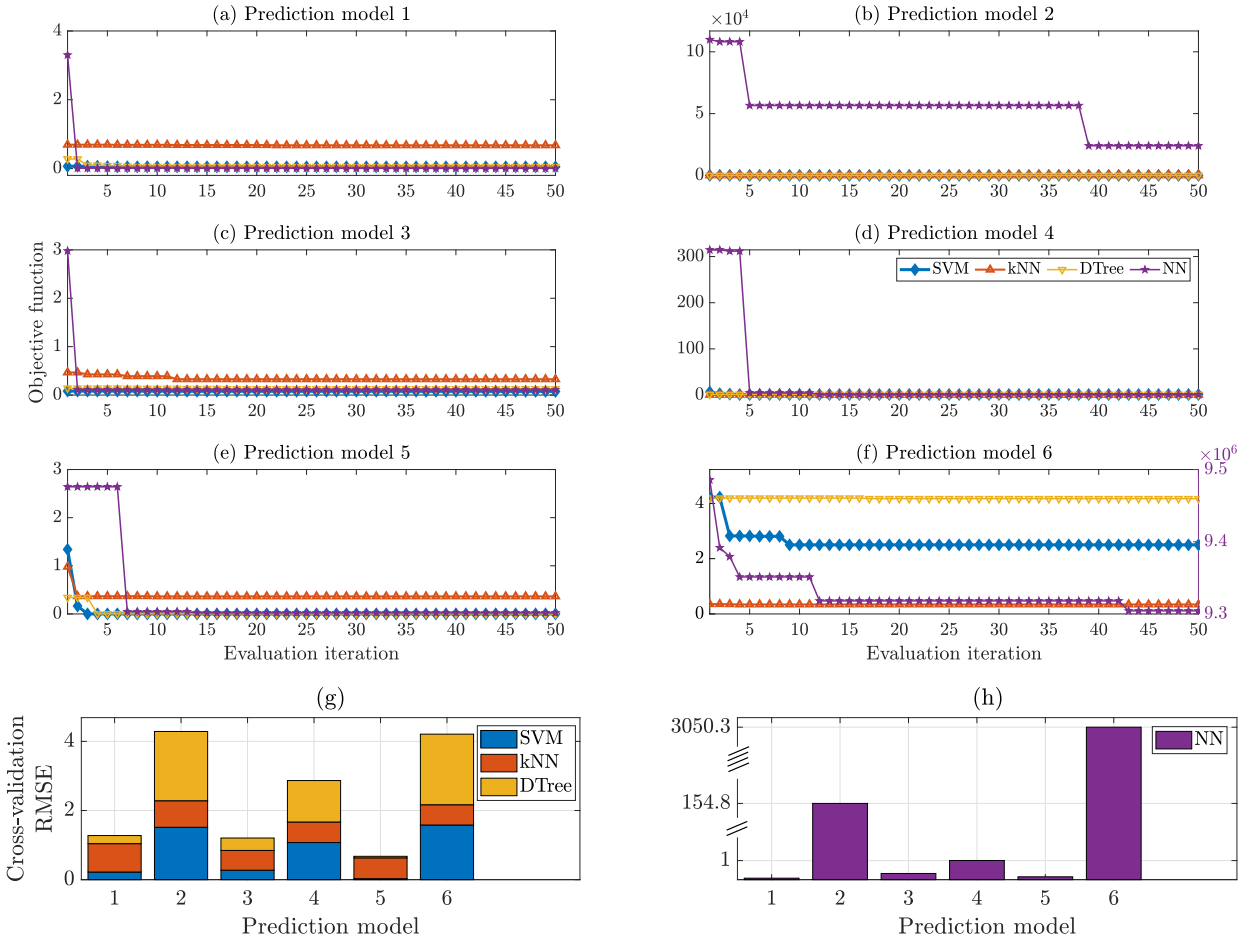


FIGURE 5. The traces of the hyper-parameters tuning process of the prediction models training using Bayesian optimization for 50 iterations for the four machine learning regression algorithms. (a) - (f) The objective function plots of prediction model 1 - prediction model 6 for the Bayesian optimization-based hyper-parameters tuning process of the four machine learning regression models. (g) and (h) The comparison results of the performance of the various machine learning regression algorithms for the six prediction models in terms of the cross-validation RMSE.

TABLE 9. The details of the obtained prediction models using Bayesian optimization.

Prediction Model			1	2	3	4	5	6
Parameters	SVM	Kernel Function (G)	Polynomial, $p = 2$	Polynomial, $p = 2$	Gaussian	Gaussian	Polynomial, $p = 3$	Polynomial, $p = 3$
		Box constraint (C)	21.7454	4989.4271	165.7554	349.5467	2.5434	9926.1602
		Epsilon (ε)	0.1195	0.9580	0.1154	0.6816	0.0295	1.1002
	kNN	Number of Neighbours (k)	8	4	3	2	1	3
		Distance (d)	Hamming	Minkowski	Jaccard	Seuclidean	Seuclidean	Minkowski
		Distance weight (w_d)	inverse	equal	inverse	equal	squared inverse	equal
		Exponent (E)	N/A	0.5153	N/A	N/A	N/A	2.4528
	DTree	Minimum leaf size	16	3	7	3	2	6
		Maximum number of splits	1514	54	182	40	539	54
		Number of variables to sample	3	2	3	3	2	2
	NN	Number of layers	1	1	1	1	1	2
		Number of neurons	2	9	8	10	12	[20,20]
		Activation function (f)	Sigmoid	Sigmoid	ReLU	ReLU	Tanh	[Tanh, ReLU]
		Learning rate (α)	7.30E-04	1.39E-05	6.05E-04	1.00E-05	6.43E-05	2.19E-05

total number of observations [76]. It is expressed as:

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|x_i - \hat{x}_i\right|. \tag{12}$$

- **Correlation coefficient** ρ: It is a statistical measure of the strength of the relationship between the change of two variables [77]. It ranges between -1 to 1, such that a correlation coefficient of 1 implies a perfect positive correlation, while a correlation coefficient of -1 represents a perfect negative correlation. A correlation coefficient of 0 means there is no linear relationship between the trend of the two variables. A decent prediction model has a positive and close to 1 correlation coefficient between the actual values $\mathbf{x}$ and predicted values $\hat{x}$.

TABLE 10. The training error of the finalized prediction models for the different machine learning regression models.

Prediction Model	RMSE			
	SVM	kNN	Dtree	NN
1	0.2226	0.0214	0.2116	0.2591
2	2.9477	4.2445	4.6267	191.2617
3	0.2747	0.0362	0.2858	0.3035
4	1.4648	1.5475	1.4043	4.3267
5	0.0284	0.0277	0.0310	0.2375
6	3.3283	3.3795	7.6883	3286.6072

TABLE 11. The list of injected attacks - the attacks dataset.

PLC	Scaling factor, λ	Compromised sensor measurement, $x(t)$	Duration (min)
3	80%	x_8, x_9, x_{10}, x_{11}	2
	90%		3
	120%		3
	150%		4
4	120%	x_{12}, x_{13}	10
	150%		9
	95%		26
	80%		4
5	80%	x_{16}, x_{17}	4
	90%		2
	120%		3
	130%		5

The correlation coefficient can be expressed as:

$$\rho = \frac{1}{m-1}\sum_{i=1}^{m}\left(\frac{x_i-\mu_x}{\sigma_x}\right)\left(\frac{\hat{x}_i-\mu_{\hat{x}}}{\sigma_{\hat{x}}}\right), \tag{13}$$

where μ_x, and σ_x are the mean and standard deviation of $x = \{x_i\}$ for $i = 1, \ldots, m$, respectively, and $\mu_{\hat{x}}$, and $\sigma_{\hat{x}}$ are the mean and standard deviation of $\hat{x} = \{\hat{x}_i\}$ for $i = 1, \ldots, m$, respectively.

- **Computational complexity**: The computational complexity is determined by the number of resources required for execution in terms of time and memory requirements [78]. For example, the computational complexity of data-driven models can be concluded from the time required to develop the model described by the **training time** (T_d), the time required to make predictions on new data described by the **evaluation time** (T_r), and the memory utilization of the model (M_d).

C. COMPARISON RESULTS OF THE VARIOUS REGRESSION ALGORITHMS

The performance of the different machine learning regression algorithms is compared in TABLE 12 and summarized in FIGURE 6. By conducting the evaluation using the attacks dataset, we observed that the DTree-based, kNN-based, and NN-based had low training time with an average of around 54 seconds. However, the models' capability in producing accurate predictions was poor at an average MAE of 318. The NN-based models scored the highest MAE of 480.9 due to the limited training data and our constraints on the networks' size.

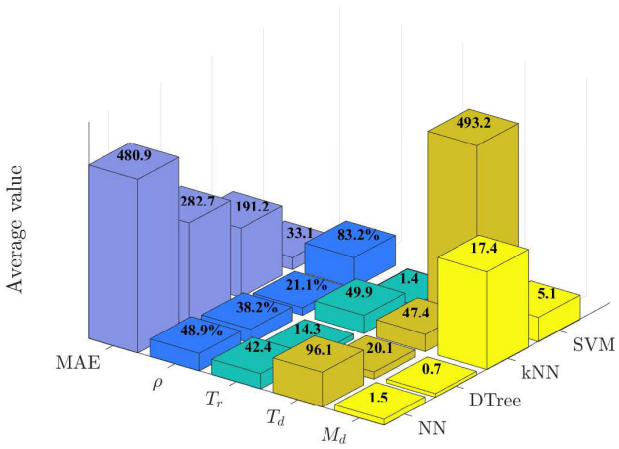


FIGURE 6. A synoptic comparison among SVM-based, kNN-based, DTree-based, and NN-based models in terms of prediction capability and computational complexity.

Similarly, even though the training error of the DTree-based models was low, they had the second-highest MAE, which can be attributed to the fact that they are known to have poor generalization ability. The kNN-based prediction models had the highest memory requirement and evaluation time, and the lowest correlation coefficient between the predicted and actual sensors' readings. They are computationally expensive, inefficient, and sensitive to outliers in the data.

Even though the time required to develop the SVM-based models was the highest, they achieved the best prediction performance with an average of 83.2% correlation between the actual and predicted sensor readings. Furthermore, the evaluation time was less than 2 seconds, with the minimum value of the MAE compared to the others.

D. COMPARATIVE ANALYSIS OF PREVIOUS WORKS

Regarding previous works summarized in TABLE 1, a predominant trend was the reliance on mathematical models in tackling the FDI attack detection problem. Specifically, references [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39] employed model-based approaches, while [53], [54], [55], [56] utilized hybrid approaches. Additionally, the majority of these studies underwent testing and validation using simulation tools.

The target system under this study has a pronounced non-linear nature [64], rendering the application of model-based approaches extremely arduous. It is crucial to acknowledge the significant challenges and limitations associated with deploying FDI attack detection and mitigation approaches that rely on the mathematical model of the system. That is, implementing these methods in actual hardware poses complexity and necessitates specific considerations that are beyond the scope of this work. In fact, the application of such approaches in this scenario may not only be exceedingly difficult but also unreasonable and impractical.

TABLE 12. The evaluation results of the prediction models using the different machine learning regression algorithms using the attacks dataset.

Sensor	Prediction model	MAE	Correlation, ρ	Evaluation time, T_r (sec)	Training time, T_d (sec)	Model size, M_d (kB)
x_8	SVM	**0.0790**	**0.9052**	**1.3281**	840.1622	5.28
	kNN	6.5666	-0.0075	166.4688	84.7597	71.95
	DTree	0.1555	0.6083	19.7188	**18.8295**	2.28
	NN	0.5440	0.8546	39.1719	187.5219	**0.57**
x_9	SVM	0.1140	0.8816	**1.2969**	840.1622	5.28
	kNN	3.0894	-0.0049	173.7188	84.7597	71.95
	DTree	**0.0931**	0.6860	21.3281	**18.8295**	2.28
	NN	0.4171	**0.8877**	38.9688	187.5219	**0.57**
x_{10}	SVM	**7.5104**	**0.8705**	**1.3125**	468.2094	9.72
	kNN	63.7649	0.0578	20.6250	36.0854	15.26
	DTree	63.5513	-0.0339	13.2969	**22.0225**	0.87
	NN	161.6922	0.6171	39.0781	44.6998	**0.78**
x_{11}	SVM	**12.8346**	**0.9139**	**1.4219**	468.2094	9.72
	kNN	126.1994	-0.3774	19.3281	36.0854	15.26
	DTree	127.4323	-0.0297	14.9219	**22.0225**	0.87
	NN	317.8730	0.5268	37.5938	44.6998	**0.78**
x_{12}	SVM	**0.8212**	**0.8062**	**1.3438**	33.4024	2.02
	kNN	5.4250	0.0340	5.3750	33.1485	7.20
	DTree	2.6431	0.0344	12.0625	**19.0396**	**0.55**
	NN	1.3217	0.5782	37.2656	104.3387	0.81
x_{13}	SVM	**1.9430**	**0.8828**	**1.4063**	44.2893	2.46
	kNN	4.9570	0.6630	19.7188	34.5192	10.54
	DTree	5.7401	0.6887	14.0156	**19.6590**	**0.65**
	NN	3.8332	0.3463	37.5313	60.3719	0.89
x_{16}	SVM	**0.1906**	**0.7440**	**1.2656**	675.2317	2.95
	kNN	0.2658	0.5952	17.2344	38.3532	12.80
	DTree	0.3892	0.1258	15.1719	**17.9494**	1.30
	NN	0.4109	0.5269	41.1406	141.3534	**0.87**
x_{17}	SVM	**137.2483**	**0.7407**	**1.3906**	534.4787	11.64
	kNN	755.2271	0.4705	13.3906	39.5550	19.20
	DTree	1171.9298	0.6775	8.9375	**21.1927**	**0.28**
	NN	1921.0351	0.0328	55.5469	47.3780	4.75

With a targeted focus, we compared the proposed framework with others developed in previous studies selectively and purposefully. The emphasis of this comparative analysis lies on frameworks that exclusively employ data-driven methods and closely match the contextual requirements of the system under study. A qualitative comparison of data-driven methods of previous works is presented in TABLE 13. One of the key remarks is that several of the existing works handle attack detection as a supervised classification problem, which requires the acquisition of representative labeled data of both attack and normal observations, i.e., [42], [43], [45], [46], [47], and [48]. Some approaches are more cohesive choices for addressing the problem from the communication and networks perspective (i.e., [51]), which falls beyond the purview of this study. That is, conversely, this work presents a detection and mitigation framework that does not rely on labeled data and examines the operation of the system using the process data. Additionally, the method proposed in [49] was deemed unsuitable for this specific scenario as transformer-based and LSTM-based models, which are known to be computationally demanding, are useful for large and complex systems [79]. For simpler problems - such as the one presented in this work-, more straightforward and simple algorithms, e.g., SVM, are sufficient and more efficient. Furthermore, attack mitigation received scant attention, lacking the level of focus and emphasis it deserved as it was only addressed in [50], [51], and [52] mostly from the communication perspective of the IoT system.

In terms of application, the control system was implemented in Simulink to allow testing and comparing our detection performance to the ones of the proposed approaches in [41], and [44] using an AE-based detector and in [52] using KL divergence-based detection criterion. A single-layer AE made of an encoder going from the input layer to the bottleneck and the decoder from the bottleneck to the output layers (see FIGURE 7) was trained to learn to reconstruct the sensor measurements of interest to this study and the final network architecture was concluded with 4 units in the bottleneck layer. For the KL divergence-based approach, the detection was implemented using Equation (14) assuming that the distribution of variables of interest can be approximated by a Gaussian distribution. The statistical properties of the normal operation mode, i.e., the reference sequence, were determined and the observation window, i.e., the sequence length size, was empirically set to 30 seconds. The thresholds were set based on the normal operation/reference data that were used to develop the AE-based and the KL divergence-based detection models.

$$D_{\mathrm{KL}}(X\|Z) = \frac{1}{2}\left(\log\left(\frac{|\Sigma_Z|}{|\Sigma_X|}\right) + \mathrm{tr}\left(\Sigma_Z^{-1}\Sigma_X\right) + (\mu_X-\mu_Z)^\top\Sigma_Z^{-1}(\mu_X-\mu_Z) - N\right), \quad (14)$$

where μ and Σ are the mean and the covariance of the sequence, $\mathrm{tr}\{\}$ is the trace operator, and N is the sequence length, given that X and Z are Gaussian distributed sequences.

FIGURE 8 - FIGURE 11 demonstrate the performance comparison results of our proposed detection framework with the AE-based and KL divergence-based strategies. Those figures show the following: (i) the variable's actual value on

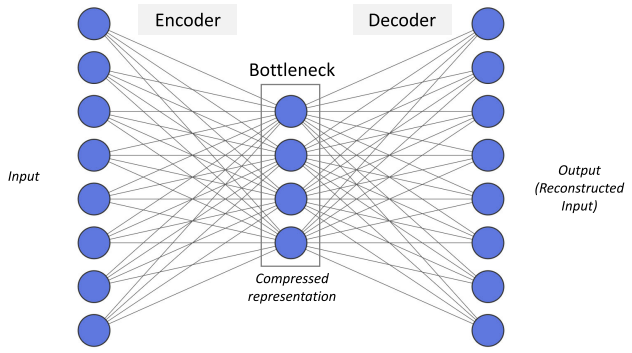


FIGURE 7. The architecture of the AE.

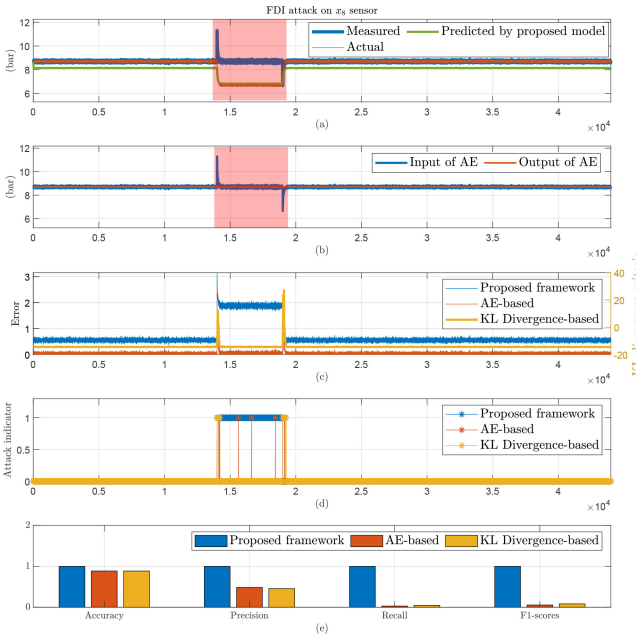


FIGURE 8. Comparison of the attack detection performance of the proposed framework, AE-based [41], [44], and KL divergence-based [52] approaches under attack on the measurements of $x_8(t)$ sensor that resulted in reducing its value from 8 to 6.4 bar (highlighted in red). (a) The measured, actual, and predicted (by the proposed framework) values of $x_8(t)$ sensor. (b) The input and output of the AE-based model. (c) The attack detection evaluation for the three approaches. (d) Their attack indicators. (e) Their binary classification performance.

the plant side depicted by the "**Actual**" plot, (ii) the measured value seen by the PLCs shown by the "**Measured**" plot, (iii) the predicted value by the proposed framework, (iv) the input and output of the AE-based detection model, (v) the value of the detection evaluation for the three approaches, i.e., error, and KL divergence criteria, (vi) their attack indicators, and (vii) their binary classification performance.

The attack indicators of the AE-based and KL divergence-based detection approaches were flagged during the transient phases (See FIGURE 8d and FIGURE 9d), which are the start and end of the attack, mostly, as they solidly represent anomalous behavior that was successfully detected. Referring to FIGURE 8a and FIGURE 9a, when the attack was injected, by maliciously increasing the sensor reading that was sent to

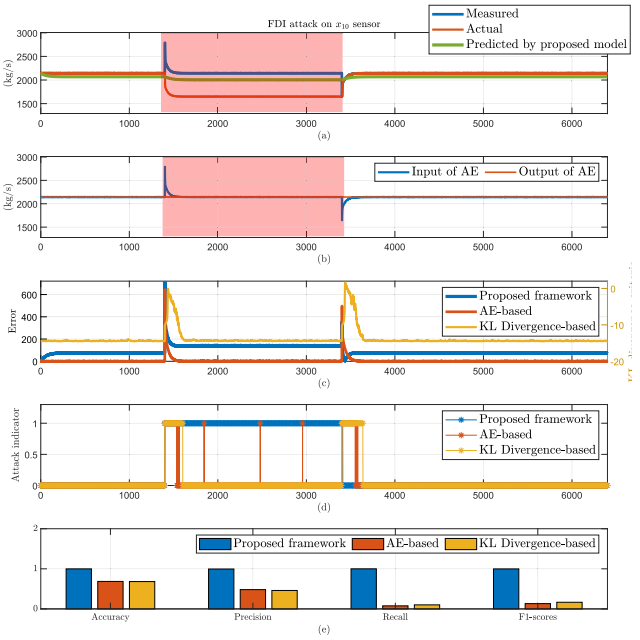


FIGURE 9. Comparison of the attack detection performance of the proposed framework, AE-based [41], [44], and KL divergence-based [52] approaches under attack on the measurements of $x_{10}(t)$ sensor that resulted in reducing its value from 2100 to 1600 kg/s (highlighted in red). (a) The measured, actual, and predicted (by the proposed framework) values of $x_{10}(t)$ sensor. (b) The input and output of the AE-based model. (c) The attack detection evaluation for the three approaches. (d) Their attack indicators. (e) Their binary classification performance.

the controller, the latter was fouled into ultimately regulating that part of the process at a lower value than desired. This happened because the closed-loop control system continuously monitors the system's output and adjusts its inputs to maintain the desired value/reference. It uses feedback to make corrective adjustments and control the system's behavior. Similarly, when the attack was terminated and hence the attack component was removed, the sensor reading sent to the controller was abruptly reduced and entered into another transient phase where the controller worked to regulate the process at the reference value.

However, the AE-based and KL divergence-based detection methods demonstrated unsatisfactory overall detection performance with precision of around 40% and recall of 10% (See FIGURE 8e and FIGURE 9e). That is, in the event of FDI sensor attacks, the controller will adjust the inputs based on the compromised readings. Over time, the closed-loop control effectively masks the observable effects of the FDI attack on the process measurements, creating the illusion of terminated or mitigated attacks (See FIGURE 8a and FIGURE 9a). While the attack's impact on the system is still present, the controller inevitably obscures the FDI attacks, preventing the measurement indicators from reflecting the true consequences of the attack. As a result, the attack remains hidden from detection, giving the impression of normalcy despite the underlying alteration inflicted by the intrusion.

TABLE 13. Comparative analysis of previous studies in data-driven methods for FDI attack detection and mitigation.

Reference	Unsupervised method?	Description	Comparison results	Validation
[41]	✓	A NN-based AE was used to detect FDI attacks in industrial IoT systems. The AE was trained with the healthy sensors data where the same data were set to its inputs and outputs. Given a predefined threshold, the discrepancy between the inputs and outputs of the AE indicated an attack occurrence.	It does not depict the dynamic behavior of the process and only focuses on the detection problem	Simulation
[42]		Bus-based FDI attacks were detected, and affected buses were identified using an ELM-based One-Class-One-Network (OCON) framework	It implements a supervised classification FDI attacks detection framework and hence requires labeled datasets and adequate attack logs	Simulation
[43]		A methodology was proposed to handle concept drift due to branch outage contingencies to enable the detection of stealthy FDI attacks after concept drift. The proposed method identified the critical concepts based on the severity of the change in the underlying data distribution.	It targets FDI attack detection under concept drifts for supervised classification methods that require labeled datasets and adequate attack logs	Simulation
[44]	✓	A NN-based AE was used to detect FDI attacks in a water treatment testbed. The AE was trained using the healthy sensors and the error signal indicated an attack occurrence based on a predefined threshold.	It does not incorporate the dynamics of the process and only focuses on the detection problem	Practical
[45]		Data-centric paradigm employing the MSA was used to detect attacks in a six-bus power network	It implements a supervised classification FDI attack detection framework and hence requires labeled datasets and adequate attack logs	Simulation
[46]		An AE was used to produce a new representation in lower dimensions that is then used to train an extra trees classifier for FDI attack detection.	It implements a supervised classification FDI attacks detection framework and hence requires labeled datasets and adequate attack logs	Simulation
[47]		A distributed supervised NN-based approach was proposed for the detection and localization of sparse stealthy FDI attacks in smart grid systems	It implements a supervised classification FDI attack detection framework and hence requires labeled datasets and adequate attack logs	Simulation
[48]		A data-driven machine learning based framework was proposed for stealthy FDI attacks detection on state estimation in power systems using ensemble learning, where multiple classifiers were used and decisions by individual classifiers were further classified.	It implements a supervised classification FDI attacks detection framework and hence requires labeled datasets and adequate attack logs	Simulation
[49]	✓	A forecasting model using a transformer and LSTM networks was proposed for the detection and isolation of FDI attacks in smart grids.	It is computationally expensive and not practical for the system under study, and is better suited for large and extended water distribution systems as an example	Simulation
[50]	✓	A reference tracking application was deployed to remove the attack component and use the correct value of the DC voltage level as the input to the controllers, in which its value was estimated by a NN, and then used as a reference in the reference tracking application	As presented in Table 12, the performance of the SVM-based model was superior to the NN-based one for the same amount and quality of system data in terms of prediction accuracy, correlation, and evaluation time.	Simulation
[51]	✓	A HMM was used to observe the behavior of IoT devices and predict their future actions for FDI attack detection and then they were mitigated through the communication channels by formulating a bandwidth optimization problem to meticulously allocate bandwidth to trusted devices	The FDI attack detection and mitigation problem was addressed from communication channels' perspective which is outside the scope of this work	Simulation
[52]	✓	The KL divergence-based criterion was used to detect and mitigate FDI attacks in distributed control of microgrids with respect to the actual and expected local signal frequency. The calculated KL divergence was used to find the probability of the attack's presence on neighbors of an agent and the trustworthiness of the agent's own outgoing information to modify distributed control protocols accordingly	The FDI attack mitigation problem was tackled by modifying the control protocol in terms of data exchange and communication channels among agents of the distributed framework	Hybrid

Our proposed framework successfully overcame this problem in which the attack indicator remained ON for the whole duration of the attack. The deployed SVM model evaluates the expected system response under the consequential control command due to the attack and predicts the actual process behavior under the attack. The discrepancy between predicted and measured readings shall reveal the hidden attacks, as demonstrated by the error signal of the proposed framework in FIGURE 8c and FIGURE 9c.

The AE-based and KL divergence-based detection frameworks are more reliable for offline attack detection or for detecting unconcealed or overt attacks, as demonstrated in

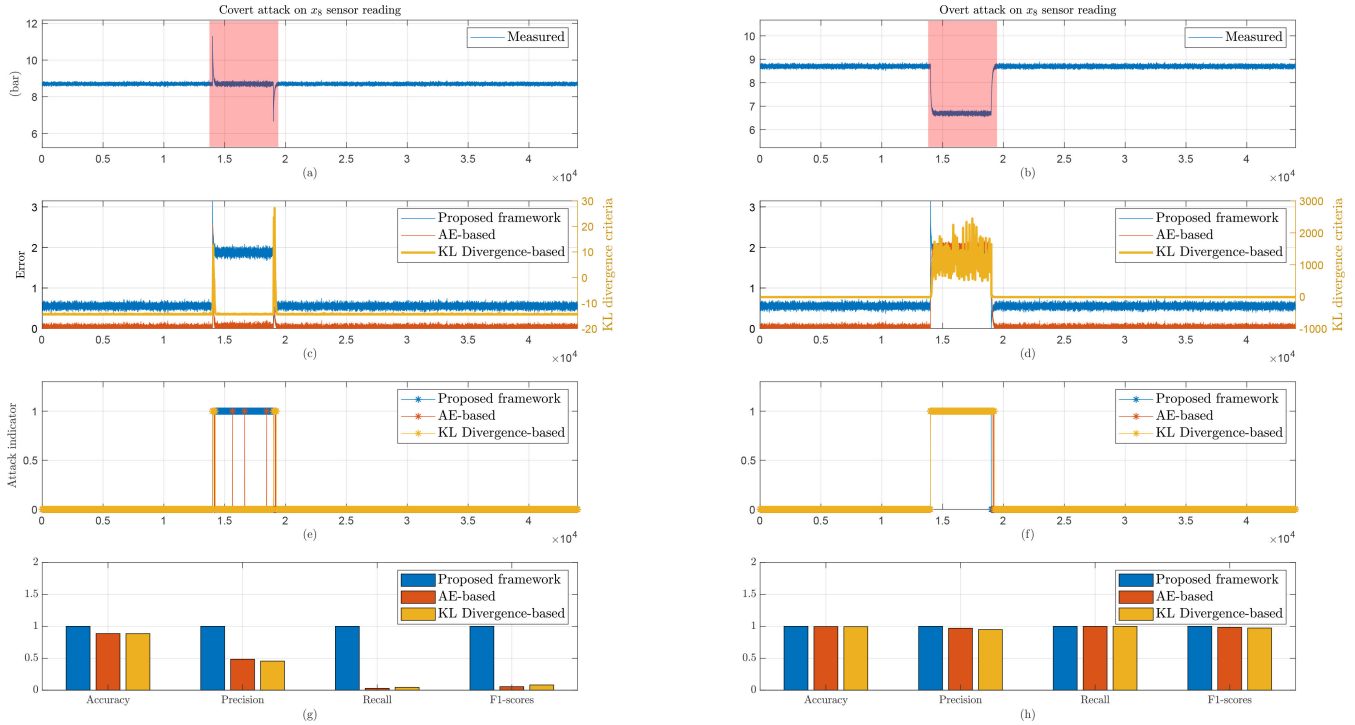


FIGURE 10. Comparison of the attack detection performance of the proposed framework, AE-based [41], [44], and KL divergence-based [52] approaches under covert and overt attacks on the measurements of $x_8(t)$ sensor that resulted in reducing its value from 8 bar to 6.9 bar (highlighted in red). Under the covert attack, the true impact remained concealed and the process appeared to be operating normally. In the event of the overt attack, the attack's consequences became rapidly apparent, and hence likely detectable. For each evaluated scenario, the plots are: (a) - (b) The measured, actual, and predicted (by the proposed framework) values of $x_8(t)$ sensor. (c) - (d) The input and output of the AE-based model. (e) - (f) The attack detection evaluation for the three approaches. (g) - (h) Their attack indicators. (i) - (j) Their binary classification performance.

FIGURE 11. Comparison of the attack detection performance of the proposed framework, AE-based [41], [44], and KL divergence-based [52] approaches in the case of operating mode changes, i.e., setpoint change. The $x_9(t)$ sensor was subjected to an attack that increased its value (highlighted in red). The detection performance was evaluated under two operation modes, i.e., setpoints of $x_9(t)$ sensor. For each evaluated scenario, the plots are: (a) - (b) The measured, actual, and predicted (by the proposed framework) values of $x_9(t)$ sensor. (c) - (d) The input and output of the AE-based model. (e) - (f) The attack detection evaluation for the three approaches. (g) - (h) Their attack indicators. (i) - (j) Their binary classification performance.

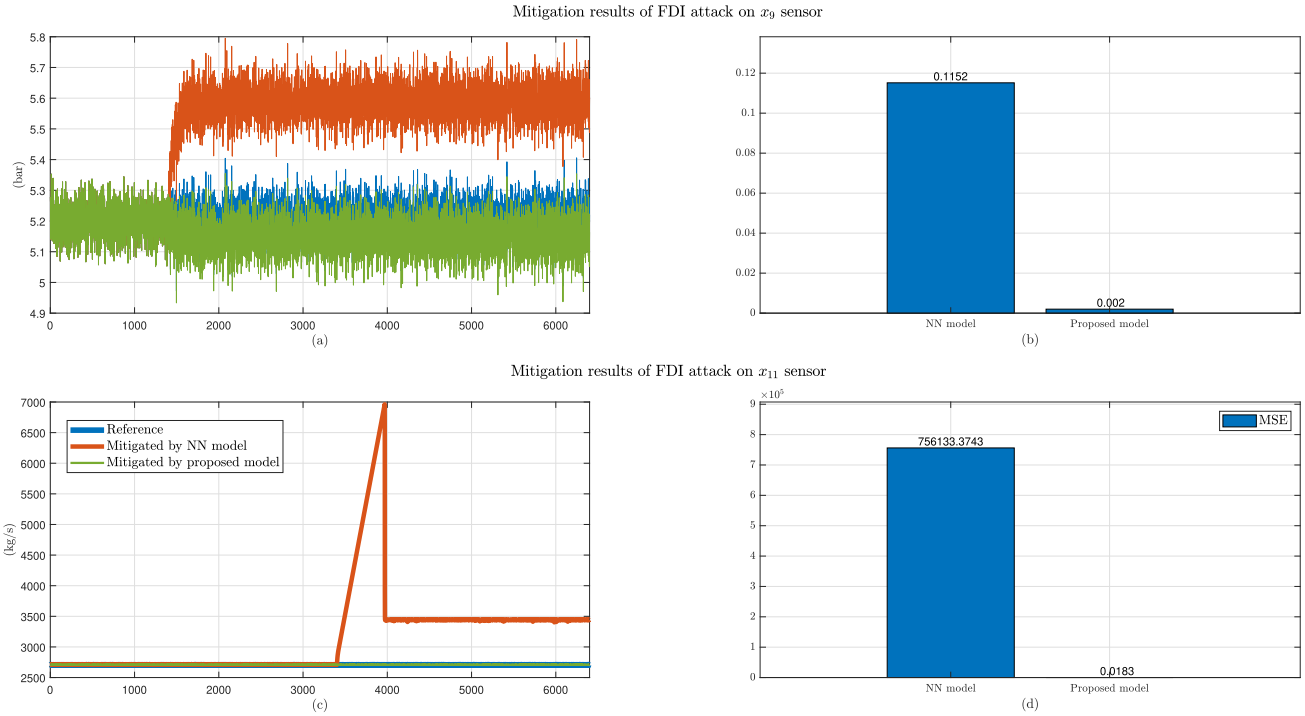


FIGURE 12. Comparison of the proposed mitigation framework and the NN-based approach in [50]. The reference healthy operation is shown in the blue plots. The sensor's reading mitigation results of the NN-based approach are shown in the brown plots, and the ones of the proposed framework are shown in the green plots. The MSE was computed between measurements obtained during the reference healthy operation and those obtained through mitigation in the event of an attack. (a) - (b) The mitigation of FDI attack in $x_8(t)$ sensor. (a) - (b) The mitigation of FDI attack in $x_{11}(t)$ sensor.

FIGURE 10, which shows a comparison of their performance in the two attack scenarios against the detection performance of our proposed framework. Under a covert attack, the true impact remained concealed and the process appeared to be operating normally during which the AE-based and the KL divergence-based detection models failed. In the event of an overt attack, the attack's consequences became rapidly apparent and hence detectable. Unlike the proposed framework, they are unreliable in the event of concealed attacks, leading to alarmingly poor detection performance.

Moreover, they do not depict the dynamic behavior of the observed variables, so they are more suitable for processes with static properties, such as for monitoring the frequency of power lines. For the system considered in this work, they should be revised and updated in case of operating mode shifts such as setpoint changes, as neglecting to address this issue could lead to an increased false alarm rate. This was demonstrated in FIGURE 11 where $x_9(t)$ sensor was subjected to an attack that increased its value. In the standard mode of the system operation, $x_9(t)$ sensor is regulated at a pressure of 5.2 bar, upon which the development of the AE-based and KL divergence detection strategies was carried out. Upon encountering the FDI attack, we observed a consistent outcome akin to the previous evaluations, which was the detection of the attack's onset and cessation by the AE-based and KL divergence-based models. However, they experienced a catastrophic failure for the same attack evaluated when the setpoint of $x_9(t)$ sensor was increased to 8 bar. This exemplifies a plausible real-world scenario wherein the operation team chooses to adjust the reference accordingly in order to attain the desired system performance. This situation requires updating the AE-based and KL divergence-based approaches, unlike our proposed framework, which consistently maintained exceptional detection performance in both evaluated scenarios. The AE-based approach suffers from poor scalability as prominent changes in the system may mandate concrete updates on the AE model. Unlike the AE-based method, the KL divergence offers benefits such as quantifying dissimilarity and flexibility, but it has limitations related to assumptions about data distribution and the lack of contextual information.

Additionally, the NN-based mitigation model proposed in [50] was compared with the prediction model deployed in our proposed framework. In the previous section, it was found that the NN-based models did not demonstrate the highest level of accuracy in their predictions, scoring the largest MAE of 480.9, under constrained training data and design parameters. This shall result in suboptimal performance. To showcase this, we compared the mitigation performance of the NN-based model with the deployed SVM-based as demonstrated in FIGURE 12, in which we observed a notably high MSE between measurements obtained during the reference healthy operation and those obtained through the NN-based mitigation method under attack. However, when we applied the proposed mitigation framework, the MSE error was noticeably lower. Overall, the proposed framework

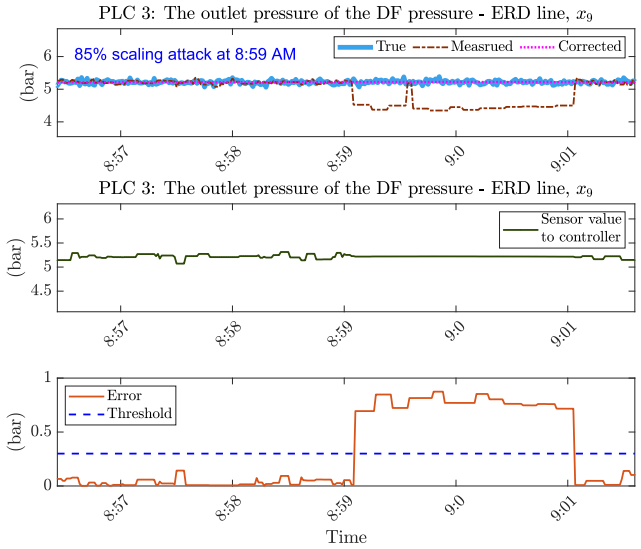


FIGURE 13. The detection and mitigation of the 85% scaling attack on sensor $x_9(t)$ in PLC3.

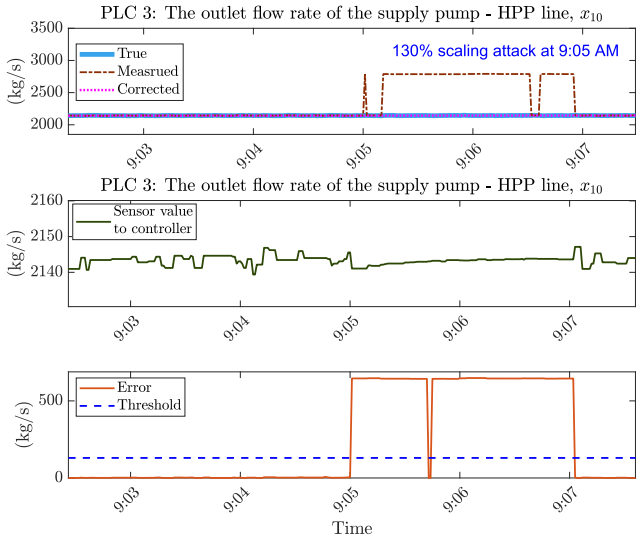


FIGURE 14. The detection and mitigation of the 130% scaling attack on sensor $x_{10}(t)$ in PLC3.

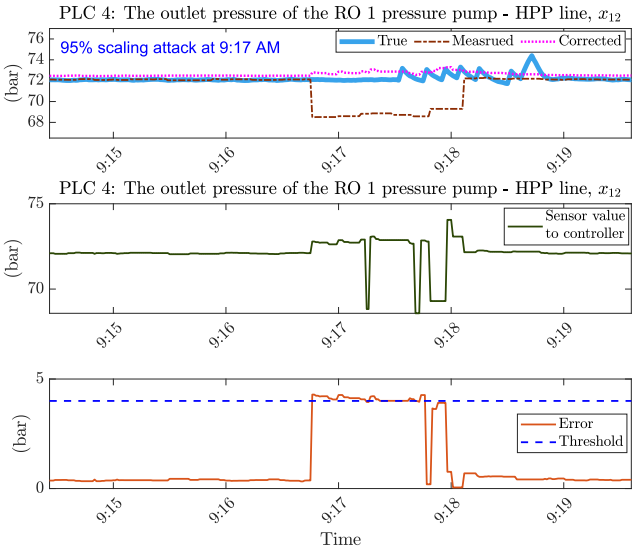


FIGURE 15. The detection and mitigation of the 95% scaling attack on sensor $x_{12}(t)$ in PLC4.

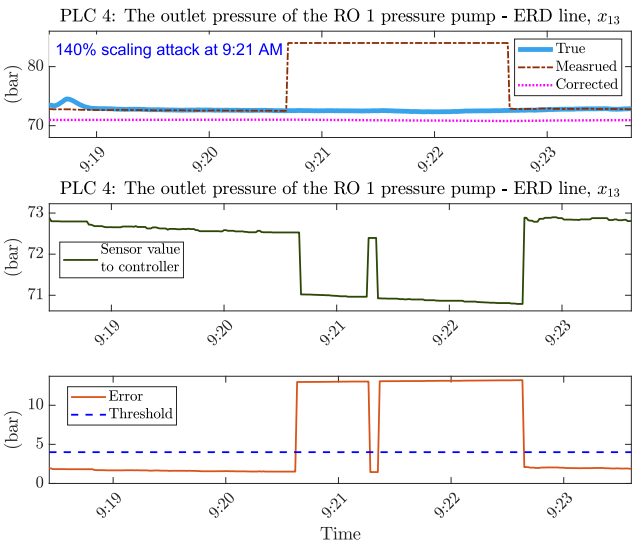


FIGURE 16. The detection and mitigation of the 140% scaling attack on sensor $x_{13}(t)$ in PLC4.

was superior to the ones proposed in previous works on aspects of accuracy and reliability of detection and mitigation, scalability, and flexibility.

E. PRACTICAL DEPLOYMENT AND APPLICATION: CASE STUDIES

For the practical application phase, the SVM-based prediction models were used to implement the real-time sensor FDI attack detection and mitigation framework in the S7-1200 PLCs. Firstly, the SVM models' parameters, **sv**, b, and α, as presented in Equation (2), were extracted and stored in the appropriate PLC, as shown in FIGURE 2a. Then in each PLC, Equation (2) was implemented, and the PLC's control logic was updated to incorporate the strategy demonstrated in FIGURE 2b.

We assumed the absence of attacks targeting the control loops responsible for regulating the water levels in the tanks (i.e., all the tanks are full and capable of supplying the required flow). The detection thresholds were set to be about 5% of the nominal value of the sensor readings to preserve a safe margin. Hence, the attacks are only detectable if the sensor readings fell below 0.95% or increased above 1.05% of their nominal values bearing in mind that the ±5% change in the sensor readings is tolerable for the RO plant without causing severe damage. Nevertheless, threshold reassignment can always be performed if needed.

In addition, it was observed that the prediction error increased rapidly during transient at startup or when the set-points were changed. Therefore, the mitigation strategy was

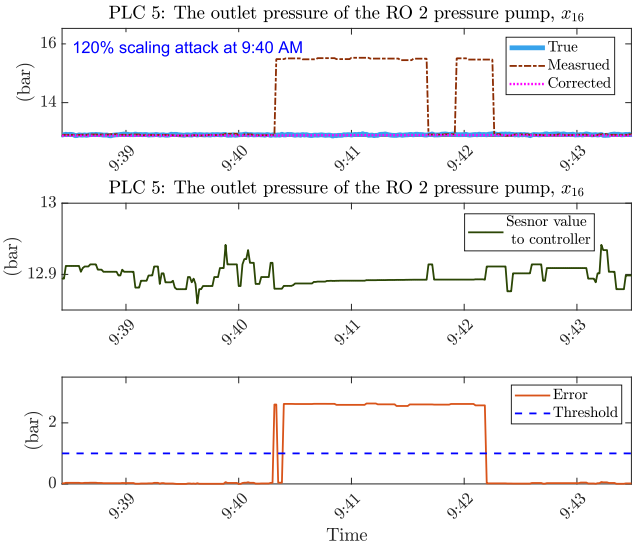


FIGURE 17. The detection and mitigation of the 120% scaling attack on sensor $x_{16}(t)$ in PLC5.

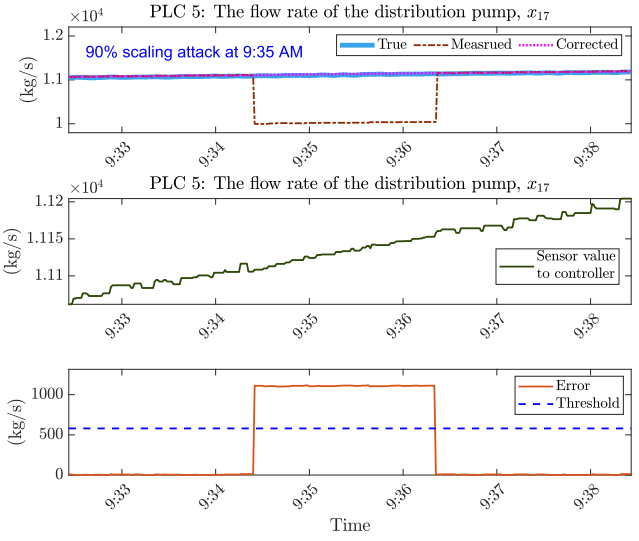


FIGURE 18. The detection and mitigation of the 90% scaling attack on sensor $x_{17}(t)$ in PLC5.

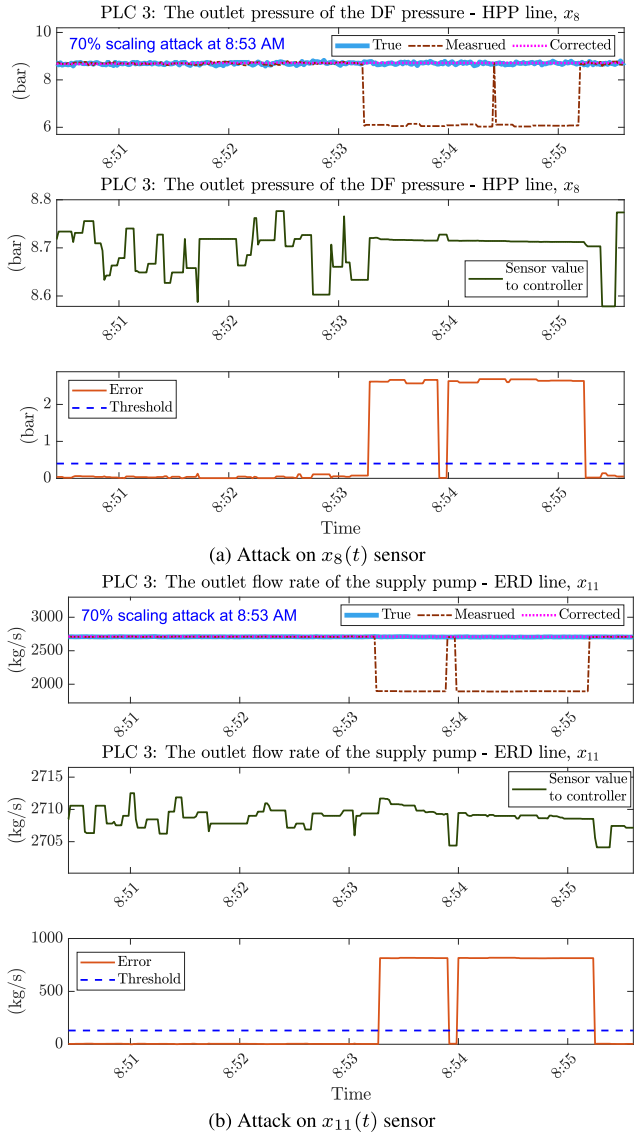
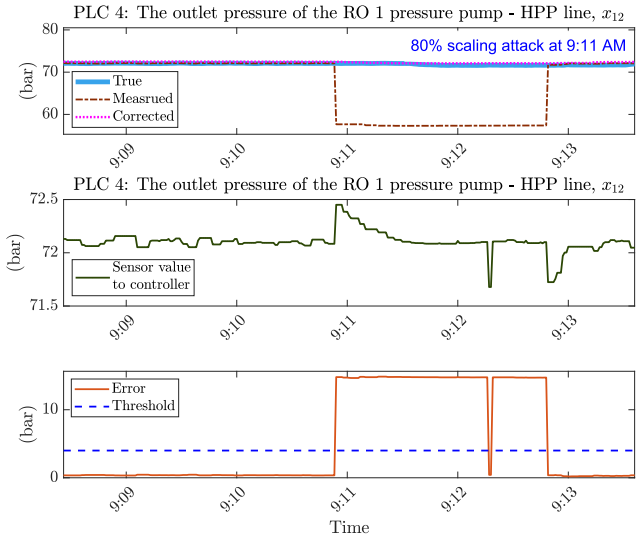


FIGURE 19. The detection and mitigation of the multiple attacks of 70% scaling attack on $x_8(t)$ and $x_{11}(t)$ sensors in PLC3.

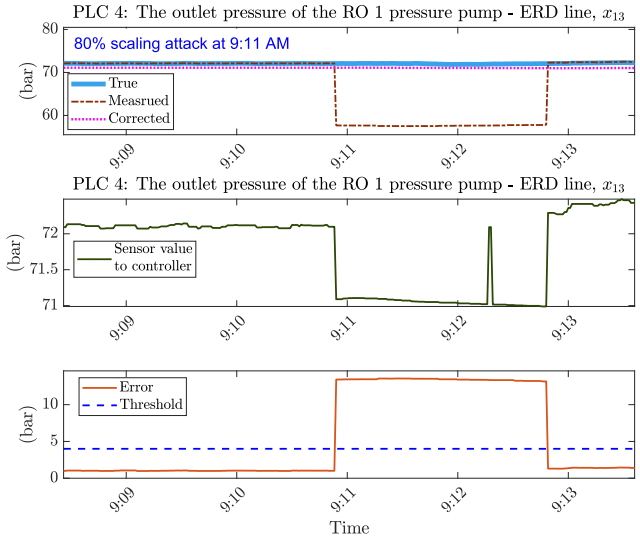
enabled only after passing the transient phase, and logic was implemented in the PLCs for thresholds reassignment to 20% of the nominal sensors values when a change in the setpoints is detected. After a period corresponding to the settling time of the control loop, the thresholds would be automatically reset to the steady-state settings. The performance of the proposed framework under sensor FDI attacks of a duration of 2 minutes is demonstrated in FIGURE 13 - FIGURE 20. The true plot in blue represents the variable's actual value on the plant side. The brown dashed plot represents the measured value seen by the PLCs. The corrected magenta dotted plot represents the predicted value by the mitigation strategy, and the green plot represents the sensor value sent to the PID controller.

1) ATTACKS ON PLC 3

Sensor FDI attacks on two sensor measurements in PLC 3 are demonstrated in FIGURE 13 and FIGURE 14. At 8:59, an 85% scaling sensor attack was injected to the sensor measurement $x_9(t)$, which represents the outlet pressures of the DF pressure pumps in the ERD line. Consequently, the $x_9(t)$ sensor measurement was dropped to 4.5 bar. At 9:05, the reading of the flow sensor $x_{10}(t)$ of the HPP line supply pump was increased to 2800 kg/s due to a 130% scaling attack. In all attack cases, the actual values of the controlled variables were not altered because the attacks were instantly detected online and mitigated by sending the corrected sensor values to the PID controllers inside PLCs. The attacks were mitigated in the sense that the control system's response was based on an estimate of what was happening in the plant rather than the falsified data due to the attacks. Once the attack



(a) Attack on $x_{12}(t)$ sensor



(b) Attack on $x_{13}(t)$ sensor

FIGURE 20. The detection and mitigation of the multiple 80% scaling attacks on $x_{12}(t)$ and $x_{13}(t)$ sensors in PLC4.

was terminated, the actual sensor measurements would be passed to the controllers. Attacks on $x_8(t)$ and $x_9(t)$ sensors can potentially damage the disk filters or disrupt the plant operation in the subsequent stages, and attacks on $x_{10}(t)$ and $x_{11}(t)$ sensors can result in underflowing or overflowing the RO water storage tanks.

2) ATTACKS ON PLC 4 AND PLC 5

FIGURE 15 - FIGURE 18 represent sensor FDI attacks on each sensor measurement in PLC 4 and PLC 5. At 9:17, the reading of the pressure sensor $x_{12}(t)$ of the RO 1 HP pump was subjected to a 95% scaling attack, and at 9:21, the reading of the pressure sensor $x_{13}(t)$ of the RO 1 ERD pump was increased to 85 bar. For PLC 5, the sensor reading $x_{16}(t)$ of the outlet pressure of the RO 2 pressure pump was falsified at 9:40 to appear 20% more than its actual value, and a 90% scaling attack was injected into the distribution pump's flow sensor $x_{17}(t)$. Similarly, all attacks were instantly detected and mitigated successfully. The plants' processes controlled by PLC 4 and PLC 5 are crucial and critical. For instance, attacks on the sensors of the pressure pumps $x_{12}(t)$, $x_{13}(t)$, and $x_{16}(t)$ that result in increasing the actual outlet pressure can damage the RO units, which are very sensitive and expensive as well as the pressure pumps. If the attacks decrease the actual outlet pressure of the pumps, the freshwater production process is interrupted. Similarly, attacks targeting the flow pump sensor $x_{17}(t)$ can interrupt or disrupt the water distribution stage.

3) SIMULTANEOUS ATTACKS

The prediction models of the attack detection and mitigation framework are independent. Therefore, the proposed solution was capable of detecting and mitigating simultaneous attacks successfully, as demonstrated in FIGURE 19 for multiple 70% scaling attacks on $x_8(t)$ and $x_{11}(t)$ sensors in PLC 3 and FIGURE 20 for multiple 80% scaling attacks on $x_{12}(t)$ and $x_{13}(t)$ sensors in PLC 4.

V. CONCLUSION

This work presented a distributed, SVM-based attack detection and mitigation framework for sensor FDI cyber-physical attacks in ICSs. It was developed using the system's normal operational data and can be easily adopted in the existing ICSs. It was validated using a hybrid testbed of a RO plant in which a validated MATLAB/Simulink-based simulation model of the process was used, while the control system was implemented using Siemens S7-1200 PLCs with 200SP Distributed I/O modules. The proposed attack detection and mitigation strategy was programmed in the PLCs and tested with actual cyber-physical attacks injected by compromising the communication links between the simulated environment and the PLCs.

The proposed detection and mitigation framework is a residual-based strategy such that when the error between the measured and the predicted sensor value exceeds a predefined threshold; a sensor attack is identified, and the falsified measured value is replaced with the predicted one. It showcased exceptional superiority over existing approaches reported in the literature. It demonstrated effective performance in real-time detecting and mitigating single and simultaneous scaling sensor attacks of magnitudes ranging between 70% and 140%. It represents a compelling solution to maintain the operation of the ICS by assisting in providing resilient control of its sub-systems.

It is worth mentioning that, when inspecting the process data, sensor FDI attacks may share the exact characteristics of sensor faults (i.e., bias, drifting, etc.) [80]. However, faults are usually assumed to be random, independent events with a fixed failure-rate probability. On the contrary, FDI attacks can be carefully designed by clever attackers with the intent to cause the greatest possible damage, which may thus result in more severe consequences [81]. Hence, the proposed mit-

igation approach shall detect and mitigate sensor faults as well. Additionally, ideally, we expect this framework to be accompanied by a fault detection and diagnosis system such that it shall only be enabled if the system is fault-free. Hence, the catastrophic impact on the operation of these systems in the presence of system faults is avoided. For further research, one can study the combined fault and attack detection and mitigation to maintain the continuity of attack detection and mitigation even when the system shows signs of failure.

Moreover, the future work includes developing solutions for (i) the distinction between FDI attacks and faults by inspecting the network traffic, (ii) the detection and mitigation of the sensor FDI attacks in the indirect actuation control loops, and (iii) the detection and mitigation of simultaneous sensor and actuator FDI attacks. Additionally, we plan to investigate other types of attacks.

ACKNOWLEDGMENT

The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] *Ukraine Power Cut was Cyber-Attack*. Accessed: May 10, 2019. [Online]. Available: https://www.bbc.com/news/technology-38573074

[2] J. Summers and M. Walstrom. (Aug. 2018). *Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks*. The Henry M. Jackson School of International Studies. Accessed: May 10, 2019. [Online]. Available: https://jsis.washington.edu/news/cyberattack-critical-infrastructure-russia-ukrainian-power-grid-attacks

[3] P. Hafezi. (Nov. 2010). *Iran Admits Cyber Attack on Nuclear Plants*. Reuters. Accessed: May 10, 2019. [Online]. Available: https://www.reuters.com/article/us-iran/iran-admits-cyber-attack-on-nuclear-plants-idUSTRE6AS4MU20101129

[4] I. G. Macola. (Apr. 2020). *The Five Worst Cyberattacks Against the Power Industry Since 2014*. Power Technology. Accessed: Nov. 21, 2020. [Online]. Available: https://www.power-technology.com/features/the-five-worst-cyberattacks-against-the-power-industry-since2014/

[5] N. Perlroth and C. Krauss. (May 2018). *A Cyberattack in Saudi Arabia Had a Deadly Goal. Experts Fear Another Try*. The New York Times. Accessed: Nov. 21, 2019. [Online]. Available: https://www.nytimes.com/2018/03/15/technology/saudi-arabia-hacks-cyberattacks.html

[6] M. Hill. (Mar. 2016). *Water Treatment Plant Hit by Cyber-Attack*. Info Security. Accessed: Nov. 21, 2019. [Online]. Available: https://www.infosecurity-magazine.com/news/water-treatment-plant-hit-by/

[7] C. Harris. (May 2022). *The New Era of Cyber-Attacks—Who is Most at Risk this Year?* Infosecurity Group. Accessed: May 17, 2022. [Online]. Available: https://www.bbc.com/news/technology-61416320

[8] *Chinese Hackers Targeted India's Power Grid System Through Malware: Report*. Deccan Chronicle. Accessed: May 17, 2022. [Online]. Available: https://www.deccanchronicle.com/nation/current-affairs/010321/chinese-hackers-targeted-indias-power-grid-system-through-malware-re.html

[9] J. Tidy. (Apr. 2022). *Ukrainian Power Grid Lucky to Withstand Russian Cyber-Attack*. BBC News. Accessed: May 17, 2022. [Online]. Available: https://www.bbc.com/news/technology-61085480

[10] J. Tidy. (Feb. 2022). *European Oil Facilities Hit by Cyber-Attacks*. BBC News. Accessed: May 17, 2022. [Online]. Available: https://www.bbc.com/news/technology-60250956

[11] (May 2022). *Cyber Attacks on the Power Grid*. Security Boulevard. Accessed: May 17, 2022. [Online]. Available: https://securityboulevard.com/2022/05/cyber-attacks-on-the-power-grid

[12] M. Elnour, N. Meskin, K. Khan, and R. Jain, "A dual-isolation-forests-based attack detection framework for industrial control systems," *IEEE Access*, vol. 8, pp. 36639–36651, 2020.

[13] M. Imran, M. H. Durad, F. A. Khan, and H. Abbas, "DAISY: A detection and mitigation system against denial-of-service attacks in software-defined networks," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1933–1944, Jun. 2020.

[14] N. N. Tuan, P. H. Hung, N. D. Nghia, N. V. Tho, T. V. Phan, and N. H. Thanh, "A DDoS attack mitigation scheme in ISP networks using machine learning based on SDN," *Electronics*, vol. 9, no. 3, p. 413, Feb. 2020.

[15] D. Yin, L. Zhang, and K. Yang, "A DDoS attack detection and mitigation with software-defined Internet of Things framework," *IEEE Access*, vol. 6, pp. 24694–24705, 2018.

[16] I. Ko, D. Chambers, and E. Barrett, "Adaptable feature-selecting and threshold-moving complete autoencoder for DDoS flood attack mitigation," *J. Inf. Secur. Appl.*, vol. 55, Dec. 2020, Art. no. 102647.

[17] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3559–3570, Apr. 2020.

[18] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, and Z. Anwar, "Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks," *IEEE Access*, vol. 7, pp. 34885–34899, 2019.

[19] M. V. O. de Assis, L. F. Carvalho, J. J. P. C. Rodrigues, J. Lloret, and M. L. Proença, Jr., "Near real-time security system applied to SDN environments in IoT networks using convolutional neural network," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106738.

[20] S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song, and K. Ren, "Detection and mitigation of DoS attacks in software defined networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1419–1433, Jun. 2020.

[21] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced support vector machine- (ASVM) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–12, Mar. 2019.

[22] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, Jan. 2016.

[23] J. A. Pérez-Díaz, I. A. Valdovinos, K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[24] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," *Comput. Sci. Rev.*, vol. 37, Aug. 2020, Art. no. 100279.

[25] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," *Arabian J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, Feb. 2017.

[26] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 3153–3158.

[27] M. Ahmed and A.-S.-K. Pathan, "False data injection attack (FDIA): An overview and new metrics for fair evaluation of its countermeasure," *Complex Adapt. Syst. Model.*, vol. 8, no. 1, pp. 1–14, Dec. 2020.

[28] L. Cui, Y. Qu, L. Gao, G. Xie, and S. Yu, "Detecting false data attacks using machine learning techniques in smart grid: A survey," *J. Netw. Comput. Appl.*, vol. 170, Nov. 2020, Art. no. 102808.

[29] J.-J. Yan, G.-H. Yang, and Y. Wang, "Dynamic reduced-order observer-based detection of false data injection attacks with application to smart grid systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 6712–6722, Oct. 2022.

[30] P. Hu, W. Gao, Y. Li, F. Hua, L. Qiao, and G. Zhang, "Detection of false data injection attacks in smart grid based on joint dynamic and static state estimation," *IEEE Access*, vol. 11, pp. 45028–45038, 2023.

[31] S. Lakshminarayana, J. S. Karachiwala, T. Z. Teng, R. Tan, and D. K. Y. Yau, "Performance and resilience of cyber-physical control systems with reactive attack mitigation," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6640–6654, Nov. 2019.

[32] L. F. Cómbita, Á. A. Cárdenas, and N. Quijano, "Mitigating sensor attacks against industrial control systems," *IEEE Access*, vol. 7, pp. 92444–92455, 2019.

[33] A. Ashok, M. Govindarasu, and V. Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1636–1646, May 2018.

[34] M. Khalaf, A. Youssef, and E. El-Saadany, "Joint detection and mitigation of false data injection attacks in AGC systems," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 4985–4995, Sep. 2019.

[35] M. Al Janaideh, E. Hammad, A. Farraj, and D. Kundur, "Mitigating attacks with nonlinear dynamics on actuators in cyber-physical mechatronic systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4845–4856, Sep. 2019.

[36] S. Zhao, Q. Yang, P. Cheng, R. Deng, and J. Xia, "Adaptive resilient control for variable-speed wind turbines against false data injection attacks," *IEEE Trans. Sustain. Energy*, vol. 13, no. 2, pp. 971–985, Apr. 2022.

[37] I.-S. Choi, J. Hong, and T.-W. Kim, "Multi-agent based cyber attack detection and mitigation for distribution automation system," *IEEE Access*, vol. 8, pp. 183495–183504, 2020.

[38] X. Chen, S. Hu, Y. Li, D. Yue, C. Dou, and L. Ding, "Co-estimation of state and FDI attacks and attack compensation control for multi-area load frequency control systems under FDI and DoS attacks," *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 2357–2368, May 2022.

[39] F. Li, K. Li, C. Peng, and L. Gao, "Dynamic event-triggered fuzzy control of DC microgrids under FDI attacks and imperfect premise matching," *Int. J. Electr. Power Energy Syst.*, vol. 147, May 2023, Art. no. 108890.

[40] W. Wang, C. Wang, Z. Wang, M. Yuan, X. Luo, J. Kurths, and Y. Gao, "Abnormal detection technology of industrial control system based on transfer learning," *Appl. Math. Comput.*, vol. 412, Jan. 2022, Art. no. 126539.

[41] M. M. N. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8462–8471, Sep. 2020.

[42] D. Xue, X. Jing, and H. Liu, "Detection of false data injection attacks in smart grid utilizing ELM-based OCON framework," *IEEE Access*, vol. 7, pp. 31762–31773, 2019.

[43] M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdinian, and B. Mohammadi-Ivatloo, "Ensuring cybersecurity of smart grid against data integrity attacks under concept drift," *Int. J. Electr. Power Energy Syst.*, vol. 119, Jul. 2020, Art. no. 105947.

[44] M. R. G. Raman, W. Dong, and A. Mathur, "Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant," *Comput. Secur.*, vol. 99, Dec. 2020, Art. no. 102055.

[45] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26022–26033, 2017.

[46] S. H. Majidi, S. Hadayeghparast, and H. Karimipour, "FDI attack detection using extra trees algorithm and deep learning algorithm-autoencoder in smart grid," *Int. J. Crit. Infrastruct. Protection*, vol. 37, Jul. 2022, Art. no. 100508.

[47] J. Shi, S. Liu, B. Chen, and L. Yu, "Distributed data-driven intrusion detection for sparse stealthy FDI attacks in smart grids," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 3, pp. 993–997, Mar. 2021.

[48] M. Ashrafuzzaman, S. Das, Y. Chakhchoukh, S. Shiva, and F. T. Sheldon, "Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning," *Comput. Secur.*, vol. 97, Oct. 2020, Art. no. 101994.

[49] A. Baul, G. C. Sarker, P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, "XTM: A novel transformer and LSTM-based model for detection and localization of formally verified FDI attack in smart grid," *Electronics*, vol. 12, no. 4, p. 797, Feb. 2023.

[50] M. R. Habibi, H. R. Baghaee, T. Dragicevic, and F. Blaabjerg, "False data injection cyber-attacks mitigation in parallel DC/DC converters based on artificial neural networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 2, pp. 717–721, Feb. 2021.

[51] H. Moudoud, Z. Mlika, L. Khoukhi, and S. Cherkaoui, "Detection and prediction of FDI attacks in IoT systems via hidden Markov model," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2978–2990, Sep. 2022.

[52] A. Mustafa, B. Poudel, A. Bidram, and H. Modares, "Detection and mitigation of data manipulation attacks in AC microgrids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2588–2603, May 2020.

[53] E. Tian, Z. Wu, and X. Xie, "Codesign of FDI attacks detection, isolation, and mitigation for complex microgrid systems: An HBF-NN-based approach," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 26, 2022, doi: 10.1109/TNNLS.2022.3230056.

[54] M. Ghafouri, M. Au, M. Kassouf, M. Debbabi, C. Assi, and J. Yan, "Detection and mitigation of cyber attacks on voltage stability monitoring of smart grids," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5227–5238, Nov. 2020.

[55] A. Abbaspour, A. Sargolzaei, P. Forouzannezhad, K. K. Yen, and A. I. Sarwat, "Resilient control design for load frequency control system under false data injection attacks," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 7951–7962, Sep. 2020.

[56] L. Xu, Y. Chen, M. Li, L. Zhang, and G. Mahmoud, "Extended observer-based hybrid tracking control strategy for networked system with FDI attacks," *Asian J. Control*, vol. 25, no. 4, pp. 3092–3104, Jul. 2023.

[57] Q. He, P. Shah, and X. Zhao, "Resilient operation of DC microgrid against FDI attack: A GRU based framework," *Int. J. Electr. Power Energy Syst.*, vol. 145, Feb. 2023, Art. no. 108586.

[58] W. Li, L. Xie, and Z. Wang, "Two-loop covert attacks against constant value control of industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 663–676, Feb. 2019.

[59] X. Feng, C. Weng, X. He, X. Han, L. Lu, D. Ren, and M. Ouyang, "Online state-of-health estimation for Li-ion battery using partial charging segment based on support vector machine," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8583–8592, Sep. 2019.

[60] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying *k*-nearest neighbor to time series forecasting," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 2019–2037, Oct. 2019.

[61] Z. Yu, F. Haghighat, B. C. M. Fung, and H. Yoshino, "A decision tree method for building energy demand modeling," *Energy Buildings*, vol. 42, no. 10, pp. 1637–1646, Oct. 2010.

[62] D.-C. Wu, B. B. Asl, A. Razban, and J. Chen, "Air compressor load forecasting using artificial neural network," *Exp. Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114209.

[63] C. Deb, L. S. Eang, J. Yang, and M. Santamouris, "Forecasting diurnal cooling energy load for institutional buildings using artificial neural networks," *Energy Buildings*, vol. 121, pp. 284–297, Jun. 2016.

[64] M. Elnour, N. Meskin, K. M. Khan, R. Jain, and S. Z. H. Siddiqui, "Full-scale seawater reverse osmosis desalination plant simulator," in *Proc. 21st IFAC World Congr.*, 2020, pp. 1–8.

[65] M. Noorizadeh, M. Shakerpour, N. Meskin, D. Unal, and K. Khorasani, "A cyber-security methodology for a cyber-physical industrial control system testbed," *IEEE Access*, vol. 9, pp. 16239–16253, 2021.

[66] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer, 1999.

[67] G. Bonaccorso, *Machine Learning Algorithms: Popular Algorithms for Data Science and Machine Learning*. Birmingham, U.K.: Packt, 2018.

[68] B. Boehmke and B. M. Greenwell, *Hands-on Machine Learning with R*. Boca Raton, FL, USA: CRC Press, 2019.

[69] *Classification Using Nearest Neighbors*. MathWorks. Accessed: Dec. 10, 2020. [Online]. Available: https://www.mathworks.com/help/stats/classification-using-nearest-neighbors.html

[70] L. De Marchi and L. Mitchell, *Hands-On Neural Networks: Learn how to Build and Train your First Neural Network Model Using Python*. Birmingham, U.K.: Packt, 2019.

[71] R. Andonie, "Hyperparameter optimization in learning systems," *J. membrane Comput.*, vol. 1, no. 4, pp. 279–291, Dec. 2019.

[72] D. Muniraj and M. Farhood, "A framework for detection of sensor attacks on small unmanned aircraft systems," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 1189–1198.

[73] H. M. S. Ahmad and N. Meskin, "Cyber attack detection for a nonlinear binary crude oil distillation column," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 212–218.

[74] H. Zhong, D. Du, C. Li, and X. Li, "A novel sparse false data injection attack method in smart grids with incomplete power network information," *Complexity*, vol. 2018, pp. 1–16, Nov. 2018.

[75] E. Drayer and T. Routtenberg, "Detection of false data injection attacks in smart grids based on graph signal processing," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1886–1896, Jun. 2020.

[76] M. Harrison, *Machine Learning Pocket Reference: Working with Structured Data in Python*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[77] E. W. Weisstein. (2006). *Correlation Coefficient*. MathWorld—A Wolfram Web Resource. [Online]. Available: https://mathworld.wolfram.com/CorrelationCoefficient.html

[78] D. Z. Du and K. I. Ko, Theory of Computational Complexity, vol. 58. Hoboken, NJ, USA: Wiley, 2011.

[79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[80] A. W. Werth, "Towards distinguishing between cyber-attacks and faults in cyber-physical systems," Ph.D. dissertation, Dept. Elect. Eng., Vanderbilt Univ., Nashville, TN, USA, 2014.
[81] M. S. Mahmoud and Y. Xia, *Cloud Control Systems: Analysis, Design and Estimation*. New York, NY, USA: Academic, 2020.

MARIAM ELNOUR received the B.Sc. and M.Sc. degrees from Qatar University, Doha, Qatar. She has been a Research Assistant with Qatar University, since 2019. Her research interests include artificial intelligence and machine learning and their applications in energy optimization, anomaly diagnosis, and cybersecurity in industrial control systems.

MOHAMMAD NOORIZADEH received the B.Sc. degree from Qatar University, Doha, Qatar, in 2015. He has been a Research Assistant with Qatar University, since 2015. His research interests include machine learning, automation, control, and robotics.

MOHAMMAD SHAKERPOUR was born in Isfahan, Iran, in 1999. He is currently pursuing the bachelor's degree in computer engineering with Qatar University. He has been an Undergraduate Research Assistant with the KINDI Centre, Qatar University, since 2018. His research interests include robotics, cyber-security of industrial control systems, wireless security, and artificial intelligence.

NADER MESKIN (Senior Member, IEEE) received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 1998, the M.Sc. degree from the University of Tehran, Tehran, in 2001, and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 2008. He was a Postdoctoral Fellow with Texas A&M University at Qatar, Doha, Qatar, from January 2010 to December 2010. He is currently an Associate Professor with Qatar University, Doha, and an Adjunct Associate Professor with Concordia University. He has published more than 190 refereed journals and conference papers. He is the coauthor of the book *Fault Detection and Isolation (FDI): Multi-Vehicle Unmanned Systems* (Springer, 2011) (with K. Khorasani). His research interests include FDI, multiagent systems, active control for clinical pharmacology, and linear parameter varying systems.

KHALED KHAN (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science and informatics from the Norwegian University of Science and Technology, the Ph.D. degree in computing from Monash University, Australia, and the bachelor's degree from the University of Dhaka. He also completed some intensive courses, such as cybersecurity risk management offered by Harvard University, a course on the economics of blockchain provided by the Massachusetts Institute of Technology (MIT), and another course on blockchain technology provided offered by the University of California at Berkeley, Berkeley. He is currently an Associate Professor with the Department of Computer Science and Engineering, Qatar University. Prior to these, he was with Western Sydney University, Australia, as a Senior Lecturer and the head of postgraduate programs for several years. He has published more than 100 technical papers and has edited four books. His research interests include human factors in cybersecurity, secure software engineering, cloud computing, measuring security, and trust in computer software. He has secured over US$6 million worth of external research funding. He was the founding Editor-in-Chief of the *International Journal of Secure Software Engineering* (IJSSE), from 2009 to 2017. He is also the Emeritus Editor-in-Chief of *IJSSE*.

RAJ JAIN (Life Fellow, IEEE) received the B.S. degree in electrical engineering from APS University, Rewa, India, in 1972, the M.S. degree in computer science controls from IISc, Bengaluru, India, in 1974, and the Ph.D. degree in applied mathematics/computer science from Harvard University, in 1978. He was one of the co-founders of Nayna Networks Inc., San Jose, CA, USA's next-generation telecommunications systems company. He was a Senior Consulting Engineer with Digital Equipment Corporation, Littleton, MA, USA, and then a Professor in computer and information sciences with The Ohio State University, Columbus, OH, USA. He is currently the Barbara J. and Jerome H. Cox Junior Professor in computer science and engineering with Washington University in St. Louis. He holds 14 patents and has written or edited 12 books, 16 book chapters, more than 65 journals and magazine papers, and more than 105 conference papers. He is a fellow of ACM and AAAS. He received the ACM SIGCOMM Award 2017, ACM SIGCOMM Test of Time Award 2006, CDAC-ACCS Foundation Award 2009, and ranks among the top cited authors in computer science.