IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 6, 2021, accepted January 15, 2021, date of publication January 20, 2021, date of current version January 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053135

A Cyber-Security Methodology for a Cyber-Physical Industrial Control System Testbed

MOHAMMAD NOORIZADEH[1], **MOHAMMAD SHAKERPOUR**[2], **NADER MESKIN**[1], **(Senior Member, IEEE)**, **DEVRIM UNAL**[2], **(Senior Member, IEEE)**, **AND KHASHAYAR KHORASANI**[3], **(Member, IEEE)**
[1]Department of Electrical Engineering, Qatar University, Doha, Qatar
[2]KINDI Center for Computing Research, Qatar University, Doha, Qatar
[3]Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

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

This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) through the National Priorities Research Program (NPRP) under Grant 10-0105-17017. The work of Nader Meskin and Khashayar Khorasani were supported by the North Atlantic Treaty Organization (NATO) through the Emerging Security Challenges Division Program.

ABSTRACT Due to recent increase in deployment of Cyber-Physical Industrial Control Systems in different critical infrastructures, addressing cyber-security challenges of these systems is vital for assuring their reliability and secure operation in presence of malicious cyber attacks. Towards this end, developing a testbed to generate real-time data-sets for critical infrastructure that would be utilized for validation of real-time attack detection algorithms are indeed highly needed. This paper investigates and proposes the design and implementation of a cyber-physical industrial control system testbed where the Tennessee Eastman process is simulated in real-time on a PC and the closed-loop controllers are implemented on the Siemens PLCs. False data injection cyber attacks are injected to the developed testbed through the man-in-the-middle structure where the malicious hackers can in real-time modify the sensor measurements that are sent to the PLCs. Furthermore, various cyber attack detection algorithms are developed and implemented in real-time on the testbed and their performance and capabilities are compared and evaluated.

INDEX TERMS Industrial control systems, cyber attack, attack detection algorithm, man-in-the-middle attack, hybrid testbed.

I. INTRODUCTION

Recent technological advances in control, computing, and communications have generated intense interest in development of new generation of highly interconnected and sensor rich systems that is known as critical Cyber-Physical Systems (CPS) infrastructure with application to variety of engineering domains such as process and automation systems, smart grid and smart cities, and healthcare systems. These complex systems are becoming more distributed and computer networked that have necessitated the development of novel monitoring, diagnostics, and distributed control technologies. Supervisory Control And Data Acquisition (SCADA) systems, Wireless Sensor Networks (WSN), and PLCs, are now established paradigms that are utilized in many critical CPS infrastructure.

On the other hand, the envisaged complex CPS infrastructure do more than ever require development of novel and proactive security technologies, as these systems are continuously being targeted by cyber attacks and intrusions by intelligent malicious adversaries. The adversaries are capable of attacking core control systems that are employed in all key cyber-physical systems infrastructure. These scenarios do not exist and are not possible or similar to security challenges that are present in traditional IT systems. Therefore, there exists an urgent need to study the vulnerabilities, analyze the risks, and develop defensive and mitigation mechanisms for critical CPS infrastructure.

Due to sensitivity and high importance of the safety critical systems in real life, any research activity that is directly applied to the physical infrastructure can lead to disruption,

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

unexpected damages or losses, and hence development of testbeds that mimic behavior of CPS in a small-scale fashion is highly essential for development of various cyber-security technologies. In this paper, a hybrid cyber-physical testbed for industrial control systems is developed and various types of real cyber attack scenarios are injected and implemented. Moreover, online real-time cyber attack detection algorithms are proposed to provide a comprehensive solution to the cyber-security of cyber-physical industrial control systems (ICS).

ICS testbeds generally consist of two main components, namely the physical process and the field devices such as PLC, HMI, RTU, etc. Depending on implementation methods, ICS testbeds are classified into three main categories as follows [1]: I) simulation testbed in which both components of the ICS are solely based on computer simulation [2], II) physical testbed where real physical parts are used in both components [3], and III) hybrid testbed in which a combination of simulation and physical testbed is considered where some components of the testbed such as the physical process is simulated and the rest are based on actual physical parts [4], [5].

In this paper, the hybrid testbed architecture is selected for development of the ICS testbed, where the Tennessee Eastman (TE) plant is simulated inside a PC and the remaining parts are implemented using actual industrial hardware. The TE plant is selected as the industrial process for our developed cyber-security testbed due to the following reasons. First, the TE model is a well-known chemical process that is used in control systems research and its dynamics are well-understood. Second, it should be properly controlled otherwise small disturbances will drive the system towards an unsafe and unstable operation. The inherent unstable open-loop property of the TE process presents a real-world scenario in which a cyber attack could correspond to a real risk to human safety, environmental safety, and economic viability. Third, the TE process is complex, coupled and highly nonlinear, and has many degrees of freedom by which to control and perturb the dynamics of the process. Finally, various simulations of the TE process have been developed with readily available reusable code design by Ricker [6].

Finally, from the anomaly detection perspectives, the cyber attack detection algorithms can be divided into five main categories, namely: linear, proximity-based, probabilistic, outlier ensembles, and neural networks approaches [7]. Therefore, in order to have a comprehensive comparison for cyber attack detection approaches that fit the TE process, the following algorithms have been chosen from various categories such as: Principal Component Analysis (PCA), One-Class Support Vector Machines (OCSVM), Local Outlier Factor (LOF) k-Nearest-Neighbors (kNN), and Isolation Forest (IF). Comparative studies are conducted based on the cyber attack detection time and the confusion matrix performance metrics where subsequently, the OCSVM and kNN are demonstrated to yield promising performance for accomplishing the cyber attack detection objective.

A. BACKGROUND

Cyber attacks on TE processes are also investigated in the literature. In [8], an integrity attack is injected on the manipulated variable signals and the corresponding sensor measurements are observed by correlation-based clustering algorithm. Different studies have been conducted on finding the optimal time to launch the Denial of Service (DoS) attack on either the sensor or actuator signals in the TE process [9]–[11]. Several cyber attack detection methods such as model-based approaches [12], [13], clustering-based approaches [14], Gaussian mixture models [15], and RNN-based approaches [16] are developed for detection of different cyber attacks on the TE process. However, all of the above work are based on the simulated TE process and cyber attacks are mainly emulated inside the simulation file.

Furthermore, several recent ICS testbeds for investigating cyber security are developed in the literature and Table 1 presents comparisons among these testbeds for diverse range of applications that are based on **TYPE** (simulation (S), physical (P), real ICS (R), and hybrid (H)), **Process**, **Data Type** (network data (NET) and process data (PR)), **Detection Method**, **Attacks**, **Attack Type** (emulation (E) and physical (P)). As shown in this table, in [17]–[25] cyber-physical testbeds are developed for the physical water system and different case studies in terms of data type, communication and attack injection/detection are presented. In [17], a model-based detection approach is developed to detect three different attacks by using network data. Also, a physics-based detection approach is presented in [18] in order to detect stealthy vulnerability by using the process data. In [19], an Intrusion Detection System (IDS) approach is developed to detect four various attacks by using network data. In [20], different data-driven intrusion detection algorithms are developed using the network data from the Modbus communication protocol. In [21]–[25], water system testbeds are developed based on the Ethernet/IP as the communication protocol.

A power system testbed is designed and implemented in [26]–[28]. A simulation testbed is used in [26] and in [27] a physical testbed is developed and different attack detection algorithms are developed by using both the network and the process data. In [29], [30], a simplified version of the Tennessee Eastman process is utilized as the physical plant in the testbed and model-based attack detection algorithms are proposed for the simulation-based testbeds without considering any physical hardware in the simulator.

B. CONTRIBUTIONS

In this paper, a full version of the nonlinear chemical process of the Tennessee Eastman process is used as the physical process in the developed hybrid testbed. Moreover, based on the structure and features of PROFINET as the industrial field bus that is used in the Siemens distributed I/O, the actual real-time false data injection cyber attack is implemented

TABLE 1. Overview of the existing testbeds for cyber-security study.

TESTBED	TYPE	Process	Data Type (NET,PR)	Communication Type	Detection Method	Attacks	Attack Type Emulation Physical
[17]	P	Realistic water system emulator	NET	Modbus	Model-Based	DoS[1], FDI[2], Replay	P
[18]	S, P, R	Water treatment	PR	Modbus	Physics-Based	Stealthy	p
[19]	P	Water level control and air pollution control	NET	Modbus	IDS[3]	Reconnaissance, Response Injection, Command Injection, DoS	p
[20]	P	Water treatment	NET	Modbus	RF, DT, LR[4], NB[5]KNN.	Reconnaissance, Command injection, (DoS)	P
[21]–[23]	P	Water treatment (SWAT)	PR,NET	Ethernet	Model-Based	Reconnaissance, FDI, Physical	P
[24], [25]	P	Water Distribution (WADI)	PR	Ethernet	Model-Based	FDI	P
[26]	S	Distribution substation of a power grid	NET	Modbus	NA[6]	Reconnaissance, DoS	E
[27]	P	Two-loop nuclear power system	NET, PR	NA	Defense-in-depth, Data-driven, (AAKR)[7], KNN[8] DT[9], Bagging, RF[10]	MITM[11], FDI, (DoS) Data exfiltration, Data tampering	P
[28]	P	3 phase power distribution system	NET	Modbus	Mitigation techniques	FRE[12], MITM, PWB[13], Web based	P
[29]	S	Simplified Tennessee Eastman	PR	NA	IDS	FDI, DoS	E
[30]	S	Simplified Tennessee Eastman	NET	NA	IDS	NA	E
This paper	**H**	**Tennessee Eastman**	**PR**	**Profinet**	**PCA[14],OCSVM[15], LOF[16], KNN, IF[17]**	**FDI**	**P**

[1]Denial of Service, [2]False Data Injection, [3]Intrusion Detection System, [4]Logistic Regression, [5] Naïve Bayes , [6]Not Available, [7]Auto-Associative Kernel Regression, [8]k-Nearest-Neighbors, [9]Decision Tree, [10]Random Forest, [11]Man In The Middle, [12]Firmware Reverse Engineering, [13]Protocol Weakness Based, [14]Principal Component Analysis , [15]One-Class Support Vector Machines , [16]Local Outlier Factor , [17]Isolation Forest,

through the man-in-the-middle (MITM) architecture on the developed testbed. This is achieved by utilizing Address Resolution Protocol such that the cyber hacker acts as the MITM in the closed-loop system and modifies the sensor measurements sent to the PLC or the actuator commands that are sent to the distributed I/O. Furthermore, various real-time online cyber attack detection algorithms are developed and implemented on the testbed and their performance capabilities are compared and evaluated. Consequently, this is the first work in the literature that completely simulates a full-version of the Tennessee Eastman Process using a hybrid testbed. In other words, this work provides a comprehensive solution for the cyber-security of ICS enabled with the following main contributions:

1) A hybrid testbed is developed by using the simulated full-version of the Tennessee Eastman Process as a non-linear unstable process and the Siemens field devices such as PLC and distributed I/O, whereas the previous work in [29], [30] only considered the simplified version of TE without having *any* actual hardware in the testbed.
2) Real-time false data injection cyber attacks are implemented by compromising the PROFINET field-bus protocol for the first time in the literature, where as shown in Table 1, all of the previous works are based on either the Modbus or the Ethernet communication protocols.
3) Several online cyber attack detection methodologies such as PCA, OCSVM, LOF, KNN, and IF are developed and implemented for real-time detection of cyber attacks in the supervisory level of the testbed. In contrast, in most of the previous work in the literature the detection algorithms are implemented off-line after collecting the data from the testbed.

The remainder of this paper is organized as follows. In Section II, the developed hybrid ICS testbed is presented. Section III provides details on PROFINET field bus protocol that is used in the testbed and in Section IV, the implementation of false data injection cyber attack is described and introduced. Section V presents the proposed cyber attack detection methodologies and in Section VI their performances are quantitatively demonstrated, validated, and verified subject to various cyber attack scenarios. Finally, in Section VII, conclusions and future work are provided.

II. HYBRID ICS TESTBED

The cyber-physical ICS includes three main components namely, a physical plant to be controlled, an embedded system for implementing the controller, and a communication network for exchanging the information between the controller and the plant. In the developed testbed, these components are all considered where the plant is simulated inside a PC, the controller is implemented on an actual hardware (PLCs) and finally the communication is established by using the industrial protocol namely, PROFINET.

As shown in Fig. 1, the developed testbed is partitioned into four layers: (1) the Tennessee Eastman plant that is simulated by a PC, (2) the field devices that are emulated by using DAQ and the Siemens distributed I/O, (3) the control layer implementation using the Siemens PLCs, and (4) the supervisory layer using additional Siemens PLC and web-server. Moreover, the mathematical model of the TE process is implemented and simulated in Matlab/Simulink environment and the controllers are implemented by using the PLCs. The interface between the plant simulation and the PLCs is accomplished by using the DAQ boards and the distributed I/O modules. The DAQ boards generate voltages that are proportional to various plant variables and also acquire the input voltages as the actuator command signals from the controller. Hence, by using the DAQs, different sensors and actuators inside the plant are emulated in the testbed. The distributed I/O modules provide the interface between the plant sensors/actuators and the PLCs. Consequently, the DAQ boards and the distributed I/O modules emulate the layer 1 within the industrial automation hierarchy, namely the field layer.

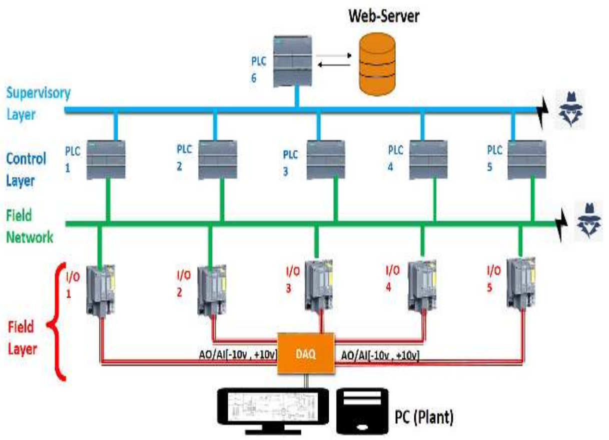


FIGURE 1. The developed hybrid ICS testbed.

A. TENNESSEE EASTMAN (TE) PROCESS SIMULATION

The TE process is first described by Down and Vogel in 1993 [6], [31] and is modeled through fifty (50) nonlinear and coupled differential equations [32]. It consists of five major operational units, namely: (1) chemical reactor, (2) product condenser, (3) recycle compressor, (4) vapor-liquid separator, and (5) product stripper. Two liquid products (G, H) are produced by using A, C, D, and E gaseous reactants with B and F as inert and byproduct, respectively. The chemical reactions are irreversible and can be presented as follows:

$$A(g) + C(g) + D(g) \rightarrow G(l), \quad \text{Product 1,}$$
$$A(g) + C(g) + E(g) \rightarrow H(l), \quad \text{Product 2,}$$
$$A(g) + E(g) \rightarrow F(l), \quad \text{Byproduct,}$$
$$3D(g) \rightarrow 2F(l), \quad \text{Byproduct.}$$

The TE process is a nonlinear open-loop unstable process which reaches its shutdown constraints in less than 2 hours. Accordingly, a controller is required to maintain the system in the steady state and the process variables at desired values, and to enforce hard constraints on the process variables such as the reactor pressure, the reactor level, the reactor temperature, among others [31], [33].

The TE process has 12 manipulated variables (XMVs), 41 measured variables (XMEAS), and 20 different process disturbances (IDVs) which can be chosen by the user [6]. The output measurements (XMEAS) of the plant are divided into 22 continuous-time and 19 discrete-time measurements. In the developed testbed in this work, only 9 inputs and 16 continuous-time outputs are used as specified in Tables 2 and 3, respectively. It should be noted that the time unit of the original TE process model was in hours which is not suitable for a real-time simulation. Thus, in order to make the process real-time, the model is modified accordingly by changing the state dynamics of the system and correspondingly the controller gains.

TABLE 2. Manipulated variables used in the testbed.

Variable name	Variable number	Units
D-feed flow	XMV 1	kg/h
E-feed flow	XMV 2	kg/h
A-feed flow	XMV 3	kscmh
A and C feed flow	XMV 4	kscmh
Purge valve	XMV 6	%
Separator pot liq. flow	XMV 7	m3/h
Stripper liq. prod. flow	XMV 8	m3/h
Reactor cw. flow	XMV 10	m3/h
Condenser cw. flow	XMV 11	m3/h

TABLE 3. Process measurements used in the testbed.

Variable name	Variable number	Units
A feed	XMEAS 1 (y_1)	kscmh
D feed	XMEAS 2 (y_2)	kg/h
E feed	XMEAS 3 (y_3)	kg/h
A and C feed	XMEAS 4 (y_4)	kscmh
Reactor pressure	XMEAS 7 (y_7)	kPa gauge
Reactor level	XMEAS 8 (y_8)	%
Reactor temperature	XMEAS 9 (y_9)	C
Purge rate	XMEAS 10 (y_{10})	kscmh
Prod. separator temperature	XMEAS 11 (y_{11})	C
Prod. separator level	XMEAS 12 (y_{12})	%
Prod. separator underflow	XMEAS 14 (y_{14})	m3/h
Stripper level	XMEAS 15 (y_{15})	%
Stripper underflow	XMEAS 17 (y_{17})	m3/h
A Concentration	XMEAS 23 (y_{23})	mol %
C Concentration	XMEAS 25 (y_{25})	mol %
G Concentration	XMEAS 40 (y_{40})	mol %

B. FIELD DEVICE AND CONTROL LAYERS

In the developed ICS testbed, the Siemens S7-1200 PLC CPU and the SIMATIC EP 200SP distributed I/O modules are used. For establishing the interface between the simulated process on the PC, and PLCs and distributed modules, MF644 and MF634 DAQ boards are used mainly due to a high number of analog inputs/outputs and their compatibility with MATLAB/Simulink. Each I/O module contains 4 analog inputs and 2 analog outputs and in order to connect all PLCs with all I/Os, the Siemens CSM 1277 switch modules are used.

As shown in Fig. 1, DAQ boards convert sensor measurements from the TE process (implemented by a PC) to analog signals and feed them to the distributed I/O modules. At the same time, DAQ boards receive the actuator signals from the distributed I/Os and feed them back to the TE process that is simulated by Simulink. All communications between the distributed I/O modules and PLCs are based on the PROFINET protocol which is an Open Real-time Industrial Ethernet Standard Protocol which can be used for virtually any function that is required in automation, namely: discrete, process, motion, peer-to-peer integration, vertical integration, and safety, among others.

As shown in Table 4, the closed-loop controller scheme for the testbed contains 9 main Proportional-Integral (PI) controllers on five PLCs that are regulating the flow rate of each valve and the 8 internal PI loops for generating the internal set-points and variables that are needed in the main PI controllers. Accordingly, all the PI controllers' gains have been selected from the original paper in [31]. Subsequently, in order to convert the process to a real-time process in terms of process run time, all the T_i's gains are multiplied by 3600. The corresponding measurements and control inputs for each I/O module and the corresponding PLC are specified in Fig. 2. Moreover, as illustrated in this figure, XMEAS 17 and the production rate (FP as the internal variable in the PLC1) are also required by the other PLCs, which are implemented by using the Siemens S7 communication protocol.

TABLE 4. Distribution of the TE control blocks in PLCs.

PLC	Input Signal	Output Signal	Controller Gains		
			K_C	T_i	T_S
Link #1 (PLC1)	XMEAS2	XMV 1	1.60E-06	0.06	1.8
	XMEAS3	XMV 2	1.80E-06	0.06	1.8
	XMEAS40	Internal Variable	-0.4	6000	1.8
	XMEAS17	Internal Variable	3.2	7200	1.8
Link #2 (PLC2)	XMEAS 8	Internal Variable	0.8	3600	1.8
	XMEAS 9	XMV 10	-8	450	1.8
	XMEAS 11	XMV 11	-4	900	1.8
Link #3 (PLC3)	XMEAS 12	Internal Variable	-0.001	12000	1.8
	XMEAS 14	XMV 7	0.0004	0.06	1.8
	XMEAS 15	Internal Variable	-0.0002	12000	1.8
	XMEAS 17	XMV 8	0.0004	0.06	1.8
Link #4 (PLC4)	XMEAS 1	XMV 3	0.01	0.06	1.8
	XMEAS 4	XMV 4	0.003	0.06	1.8
	XMEAS 23	Internal Variable	0.0002	3600	36
	XMEAS 25	Internal Variable	0.0003	7200	36
Link #5 (PLC5)	XMEAS 7	Internal Variable	-0.0001	1200	1.8
	XMEAS 10	XMV 6	0.01	0.06	1.8

C. SUPERVISORY LAYER

As depicted in Fig 1, the supervisory layer that consists of the PLC 6 is the last layer of the TE testbed. Each Siemens S7-1200 contains internal memory that can be accessed

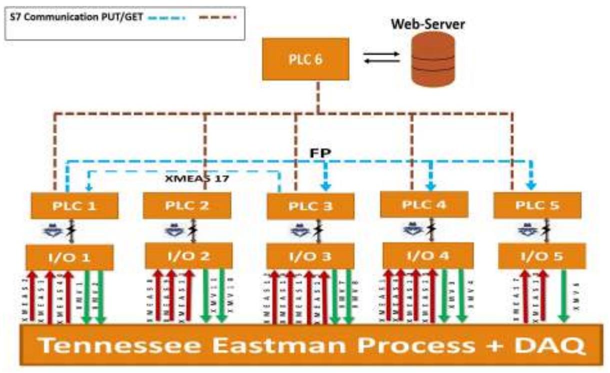


FIGURE 2. The TE process block diagram.

through a web-server. In other words, the web-server provides a local cloud that allows the user access and control over the PLC internal memory, stop/run PLC and many other features remotely (through the PLC static IP address). In the developed testbed as shown in Fig 2, by using the Siemens internal communication protocol known as the S7 communication, all measurements and actuator data of each PLC are transferred to and are stored in the PLC 6 internal memory. Subsequently, these data can be downloaded from the web-server for training or for online cyber attack detection purposes as will be presented and described in Section V.

D. VULNERABILITIES AND CYBER ATTACK GATEWAYS AND POINTS

Figure 2 illustrates the cyber attack gateways and points on the testbed where the malicious hackers can gain access to the communication link between the PLC and I/O modules. By accessing each communication link, the malicious hackers can inject different cyber attacks on the sensor measurements as well as actuator commands corresponding to that communication link. For example, as shown in Fig. 2 and Table 4, if the hacker accesses the communication link between the PLC1 and the I/O module 1 (labeled as communication link #1), then the sensor measurements XMEASs 2, 3, 17 and 40 and the actuator commands XMVs 1 and 2 can be compromised.

III. COMMUNICATION PROTOCOLS

A. PROFINET

Siemens S7-1200 utilizes the PROFINET protocol suite as an industrial Ethernet standard and S7-communication protocol in order to communicate with other network nodes. PROFINET protocol is the standard protocol which is being facilitated heavily by Siemens as one of the main industrial Ethernet communication protocols. It has inherited its architecture from the native OSI model of TCP/IP for cyclic and acyclic data and UDP/IP for context management. A PROFINET architecture/system requires at least three nodes to operate, namely: the IO Controller (PLC), the IO Module (Sensor and Actuator), and the IO Supervisor

(Engineering Station or HMI Device). Moreover, PROFINET inherits variety of Information Technology protocols within its substructure to establish and maintain connectivity, which is susceptible to similar structure cyber attack surfaces that are present in the standard Ethernet environments.

One of the main characteristics of the PROFINET protocol suite that distinguishes itself from the other ICS protocols is that it prioritizes the type of communication based on real-time requirements. Consequently, as shown in Fig. 3, two channels are being introduced as Real-Time (RT) and Non-Real-Time (NRT) and both channels coexist in the Application Relation (AR) between the IO Device and the IO Controller. An Application Relation is a state which both the IO Device and the IO Controller need to converge to, in order to initialize the transmission of the Cyclic Data. However, a handshake is a perquisite to this state which is being conducted by the profinet Context-Manger (PN-CM).

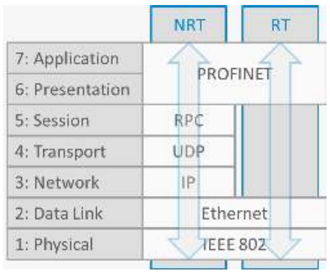


FIGURE 3. PROFINET IO RT and the NRT Stack[18].

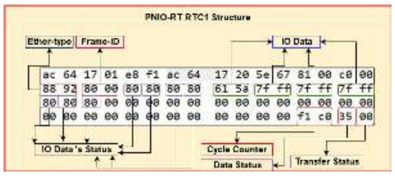
In terms of the C.I.A security aspects (confidentiality, integrity and availability) of the PROFINET protocol, it is shown in this work that to compromise confidentiality of the cyclic data, through an Address Resolution Protocol (ARP)-compromising attack a hacker can read the data in the plain text; to compromise integrity, the hacker can inject false data through the network switch; and to compromise availability, a port stealing attack would make the service temporarily unavailable.

B. PROFINET IO REAL-TIME PROTOCOL STRUCTURE

In order to guarantee real-time synchronicity in data transmission, certain layers of the OSI model have been omitted in PROFINET IO (PNIO) as illustrated in Fig. 3, which results in lower overhead communication flows. Hence, as shown in Fig. 4, in the real-time structure the dissection of a frame only consists of the Ethernet Header and the PROFINET Application Layer and which is specified as follows:

a) Frame-ID: Indicates the type of the frame which is set as 8,000 for cyclic real-time data.
b) IO Data: Sensor measurements and actuator signals are referred to as IO Data.
c) IO Data's Status: Represents the status of a given variable in the frame.

[18]Courtesy of profinetuniversity.com



```
ac 64 17 01 e8 f1 ac 64  17 20 5e 67 81 00 c0 00
88 92 80 00 80 80 80 80  61 5a 7f ff 7f ff 7f ff
80 80 80 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 f1 c0 35 00
```

FIGURE 4. PROFITNET IO real-time packet structure.

d) Cycle Counter: An incremental value, which is being incremented from the source, with an error checking purpose.
e) Data Status: Indicates the validity of the entire packet.

In the IO module, the data cycle update time, which is denoted by dt, can be set based on the system requirements from 2 to 512 msec, which represents the rate of data exchange between the IO module and the PLC. In the developed testbed, given the slow behavior of the TE process, this value is set to $dt = 512$ msec, which implies that 4 data samples are communicated in each full cycle (2 seconds). Fig. 5 shows the cycle counter corresponding to $dt = 512$ msec.

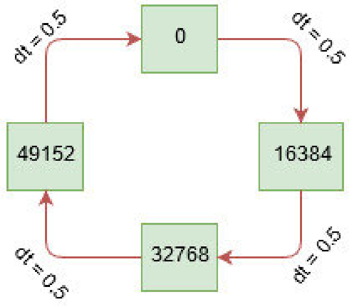


FIGURE 5. Cycle counters corresponding to $dt = 512$ msec.

IV. CYBER ATTACK INJECTION

In this section, our methodology for injecting cyber attacks on the developed testbed is presented. Generally, different protocols enable various attack surfaces such as the Data Integrity (DI) attack (e.g. manipulating sensor measurements), and Denial-of-Service (DoS) which causes disruption of communication flow among entities. In an ICS architecture, cyber attacks can be categorized into two general types, namely as configuration and operational attacks. In the configuration attack, the malicious hacker targets the configuration protocols of the ICS, and consequently gets access to full control of the system. On the other hand, in the operational attacks, the malicious hacker mainly targets the operational communication protocol such as the PROFINET IO Real-time data, in which critical field data are transferred. For this cyber attack to take place, it is assumed that:

(i) The hacker has a field level access to the IO Module and PLCs.

(ii) Hacker has knowledge of the physical system, implying that, he/she is aware of what is being transmitted from the sensors and what are being transferred to actuators.

In [34], the authors exploit a vulnerability of the PROFINET Discovery and Basic Configuration Protocol (DCP) to inject DoS attacks through port stealing, against the application relation between the IO Controller and the IO Device. This type of cyber attack is not designed to be stealthy and has a higher probability of detection. An early attempt for false data injection through port stealing is presented in [35] although the developed attacks are *not* implemented on a real testbed.

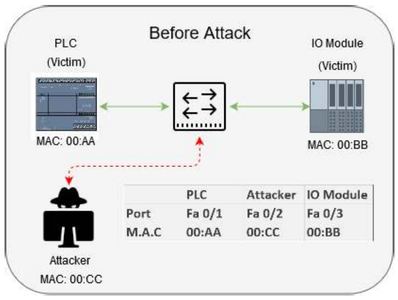
In this paper, based on the structure and features of the PROFINET, a false data attack is injected into the PROFINET IO. Real-time data through the man-in-the-middle (MITM) structure is also validated on the developed testbed. This is mainly achieved by utilizing the ARP in which the port of the victim on the shared medium (such as a switch) is stolen and the hacker acts as a Man-in-the-Middle (MITM) in the closed-loop system that can modify the sensor measurements that are sent to the PLC.

The PROFINET IO devices do not have any endpoint security functionality [36] which allows cyber attacks feasible once a malicious hacker has a physical access to a device or its network connections. One of the most effective and damaging cyber attacks on the PROFINET IO devices is the MITM cyber attack.

The MITM cyber attack will be implemented in our developed testbed, by utilizing the Port Stealing methodology. In the Port Stealing attack, the switch MAC table is compromised such that the hacker's MAC address is registered in place of the victim. Therefore, the intended port from the I/O module is stolen by the hacker, and consequently he/she can transmit false data to the PLCs.

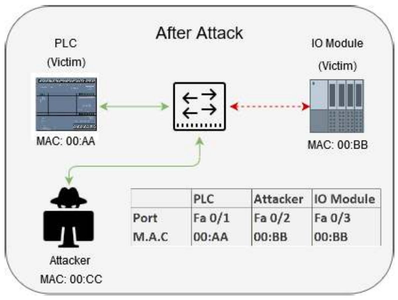
Port Stealing is an active cyber attack which allows a hacker to sniff packets in a switched network as well as modify packets by injecting new packets. This cyber attack targets the Application Relationship between the IO Controllers and the IO devices. Successful Port Stealing requires the hacker to synchronize with the real-time data communication and establish a race condition. The complete Port Stealing strategy is developed as follows:

- **ARP Flooding**: First, an ARP packet is constructed by setting the packet destination and source MAC address to the hacker MAC and the victim MAC, respectively. Subsequently, by injecting high flow rates of ARP packets into the switch, the intended port victim is stolen. As shown in Figs. 6 and victim 7, the MAC table of the switch is modified after the ARP flooding and the MAC address of the hacker is set as the MAC address of the IO module in the MAC address table.
- **Receiving Data**: In this step, the hacker receives data from the victim and modifies the sensor readings according to his/her knowledge of the process. The data received by the hacker is the raw IO data from the PROFITNET IO real-time packet as depicted in Fig. 4.



	PLC	Attacker	IO Module
Port	Fa 0/1	Fa 0/2	Fa 0/3
M.A.C	00:AA	00:CC	00:BB

FIGURE 6. Data exchange configuration before the ARP flooding.



	PLC	Attacker	IO Module
Port	Fa 0/1	Fa 0/2	Fa 0/3
M.A.C	00:AA	00:BB	00:BB

FIGURE 7. Data exchange configuration after the ARP flooding.

Next, the hacker needs to map the raw IO data into an actual sensor reading in order to be able to modify it precisely so that it will result in a desired effect to the system. Here the assumption is that the hacker has knowledge of the physical process and the control system, and therefore can map the raw IO data to the actual sensor readings. Therefore, the hacker will be able to choose values that are not easily detectable by the operator, thus a stealth cyber attack will be realized and accomplished.

- **Forwarding the Manipulated Data**: In this step, the main MITM cyber attack is implemented whereby the hacker re-crafts the received frames and forwards the modified frames back to the victim. However, the received frames cannot only be forwarded back to the network due to existence of the cycle counter in the frame. There exists a threshold for the number of missing packets per cycle and its value can be set inside the TIA portal tool. Therefore, in order to overcome this issue, the re-crafted packets are sent in a full cycle. Moreover, as the hacker and the IO modules are simultaneously sending the data to the PLC, a race condition is established between them in which the behavior of the system depends on the sequence or timing of events. With respect to Fig. 5, the race condition occurs if the hacker can send false data between the state transitions, therefore the false data crafted by the hacker would arrive at the victim before the actual data. The significance of the hacker to win the race condition is that the hacker is capable of injecting false measurement data to the system. However, this injection has to be sustained, at ideally every, or practically at most state transitions, for the hacker to be successful in winning the race condition. After winning the race condition, the hacker can receive the RTC1 frames which contain IO Data variables (process data). In order to increase the success probability of the cyber attack, the PLC should continually receive the mal-crafted data rather than the original data, therefore the hacker should send each mal-crafted data for more than one cycle.

Fig. 8 depicts the entire process of implementing the false data injection cyber attack on the PROFINET. It should be noted that due to precise timing and synchronicity which are required in order to inject data into the PLC, we have used the C language and *libpcap* put reference library in order to make this methodology possible. The *libpcap* library works by capturing all the frames that are coming out of the physical medium into the data link layer. The alternative to using *libpcap* library is to use a packet capture software such as the Wireshark, however for our purposes this is not suitable since Wireshark captures and saves packets offline.

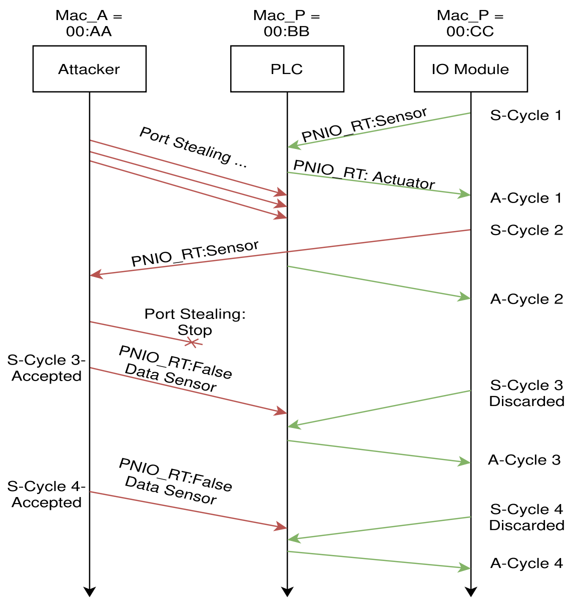


FIGURE 8. False Data Injection (FDI) through the port stealing.

One important point regarding the implemented cyber attack is that if the hacker continues the port stealing for a long duration, this will disrupt the communication between the PLC and the IO. In this case, the attack becomes a Denial-of-Service (DoS) attack which can be easier to detect by operators. By stopping the port stealing step after a given time duration, such as 1 sec, the attacker is able to start the frame manipulation without disrupting the communication.

V. CYBER ATTACK DETECTION (CAD) SCHEME

In order to detect cyber attacks in our developed testbed, several machine learning-based detection strategies are proposed and implemented. As shown in Fig. 9, the cyber detection scheme is divided into three main steps, namely, (a) pre-processing, (b) main scheme, and (c) post-processing.

A. PRE-PROCESSING

In order to have a dataset with zero mean and unit variance (standardization), data normalization is performed. The key feature of data normalization is that it will boost the learning speed and optimizes the algorithm accordingly. Moreover, there are several available techniques for data normalization, based on the nature/requirement of the algorithms itself.

B. MAIN SCHEMES

Broadly speaking, anomaly detection schemes can be divided into five main categories, namely (1) linear, (2) proximity-based, (3) probabilistic, (4) outlier ensembles, and (5) neural networks [7]. Consequently, in order to provide a comprehensive comparative study and evaluation the following schemes are chosen belonging to different categories:

- Linear: Principal Component Analysis (PCA) and One-Class Support Vector Machines (OCSVM).
- Proximity-Based: Local Outlier Factor (LOF) and k-Nearest-Neighbors (kNN).
- Outlier Ensembles: Isolation Forest (IF).

1) PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA) [37] is a method widely used to determine dominant subspaces in datasets based on eigenvectors of the covariance matrix that is designated as the principle components. An anomaly detection technique can be developed based on variations from the nominal dominant subspaces in the dataset. Generally, the use of major components indicates global deviations from the majority of results, whereas the use of minor components may suggest smaller local deviations. Indeed, as illustrated in Algorithm 1, by performing the Singular Value Decomposition (SVD) over the normalized data, the eigenvalues and eigenvectors can be determined. Moreover, by computing the PCA-reconstructed representation from $\hat{\mathcal{X}} = \mathcal{X}TT^{\mathrm{T}}$, the approximated value ($\hat{\mathcal{X}}$) can be obtained. Therefore, by computing the maximum Euclidean distance between the normalized training data and the approximated one in the training set, threshold values can be determined. Consequently, for the testing data point (D), if the distance between the existing instance and the corresponding approximated value of that instance is above a given threshold value, then the instance can be considered and classified as a cyber attack.

2) ONE-CLASS SUPPORT VECTOR MACHINES (OCSVM)

In the one-class support vector machine as a semi-supervised anomaly detection approach, the aim is to determine a hypersphere in the feature space with the minimum radius that contains all or most of the data points corresponding to the healthy operation of the system [38]. The hypersphere has two main parameters, namely the radius R_T and its center a which are obtained by solving an optimization problem as explained in Algorithm 2. Once, these parameters are obtained through the training stage, for each test data point D, one can obtain the distance between the data point and the hypersphere center a and if this distance is greater than R_T,

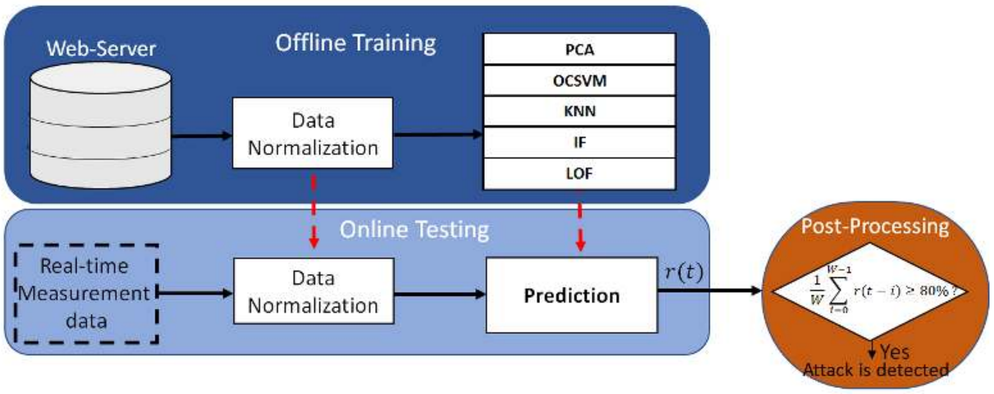


FIGURE 9. The proposed data-driven cyber attack detection methodology where $r(t)$ denotes the decision flag corresponding to the real-time data point $x(t)$.

```
Algorithm 1 Principle Component Analysis (PCA)
Training:
Input: $\mathcal{X}$ - Training data, $p$ - number of components to keep for PCA transformation.
Output: Threshold $T_r$
 1: Calculate SVD of the training data ($\mathcal{X}$)
 2: Construct the transformation matrix $T$ by selecting the $p$ dominant eigenvectors
 3: Calculate the PCA-reconstructed representation, $\hat{\mathcal{X}} = \mathcal{X}TT^T$
 4: Find the Euclidean distance between $\hat{\mathcal{X}}$ and $\mathcal{X}$, $E = distance(\hat{\mathcal{X}}, \mathcal{X})$
 5: Set the threshold as $T_r = \max(E)$
Testing:
Input: $D$ - test data, $T_r$.
Output: Test data flag $r$
 1: Calculate the PCA-reconstructed representation of the testing data, $\hat{D} = DTT^T$
 2: Calculate the Euclidean distance between $\hat{D}$ and $D$, $e = distance(\hat{D}, D)$
 3: if $e < T_r$ then
 4:     $D$ is normal, $r = 1$.
 5: else
 6:     $D$ is abnormal, $r = 0$.
 7: end
```

```
Algorithm 2 One-Class Support Vector Machine (OCSVM) Algorithm
Training:
Input: $x_i$ - training data ($i \in \{1, 2, 3, \ldots, N\}$), $C$.
Output: The hypersphere centre $a$ and its radius $R_T$.
  Optimize $\alpha_i$, $i = 1, \ldots, N$ in
    $$\min L(\alpha) = \sum_{i,j=1}^{N} \alpha_i\alpha_j K(x_i, x_j) - \sum_{i,j=1}^{N} \alpha_i K(x_i, x_i)$$
  subject to $0 < \alpha_i < C$ and $\sum_{i=1}^{N} \alpha_i = 1$ where $K(x_i, x_j) = exp(\frac{-||(x_i - x_j)||^2}{\sigma^2})$.
  Compute the centre ($a$) and the radius ($R_T$) of the hypersphere from: $a = \sum_{i=1}^{N} \alpha_i x_i$ and
    $$R_T^2 = \max_k K(x_k, x_k) - 2\sum_{i=1}^{N} \alpha_i K(x_k, x_i) + \sum_{i,j=1}^{N} \alpha_i\alpha_j K(x_i, x_j)$$
Testing:
Input: $D$ - test data, the hypersphere centre $a$ and its radius $R_T$.
Output: Test data flag $r$
  Compute
    $$R(D) = K(D, D) - 2\sum_{i=1}^{N} \alpha_i K(D, x_i) + \sum_{i,j=1}^{N} \alpha_i\alpha_j K(x_i, x_j)$$
  if $R(D) > R_T$ then
    $D$ is abnormal, $r = 1$.
  else
    $D$ is normal, $r = 0$.
  end
```

then the point is classified as an anomaly, otherwise it is assigned as a healthy data. The only two hyper-parameters for the OCSVM are C and σ, where C is controlling the influence of slack variables in the optimization process, and can be obtained from $C = \frac{1}{\nu N}$, where ν represents the trade-off between the overfitting and the generalization accuracy, and σ is the kernel coefficient.

3) k-NEAREST-NEIGHBORS (kNN)

The k-nearest-neighbor global unsupervised anomaly detection scheme is a simple way to determine irregularities for not to be mistaken with the kNN classification scheme [39]. As the name suggests, it specializes on global anomalies and is unable to identify local anomalies. In this approach, the hyper-parameter k denote the number of nearest neighbors.

During the training phase, the decision score a_{score} corresponding to all the training points are computed as the *Largest* distance to their k nearest neighbors [40] using the Ball-tree algorithm. The maximum value of a_{score} corresponding to the training data is set as the threshold T_r. Then, as illustrated in Algorithm 3, for each test data point D, the decision score $a_{\text{score}}(D)$ is compared with the computed threshold to detect a cyber attack.

Algorithm 3 Nearest-Neighbor Algorithm

```
Training:
Input: $x_i$ - training data ($i \in \{1, 2, 3, \dots, N\}$), $k$,
Output: Threshold $T_r$
1: for $i = 1, \dots, N$ do
2:     Compute the $k$ nearest neighbors of $x_i$ using the Ball-tree algorithm.
3:     Compute the decision score ($a_{score}(x_i)$) as the largest distance between $x_i$ and its nearest neighbors.
4: end
5: Set threshold as $T_r = \max_i(a_{score}(x_i))$
Testing:
Input: $D$ - test data, $x_i$ - training data, $T_r$,
Output: Test data flag $r$
1: Compute the $k$ nearest neighbors of $D$ using the Ball-tree algorithm.
2: Compute the decision score ($a_{score}(D)$) as the largest distance between $D$ and its nearest neighbors.
3: if $a_{score}(D) > T_r$ then
4:     $D$ is abnormal, $r = 1$.
5: else
6:     $D$ is normal, $r = 0$.
7: end
```

4) LOCAL OUTLIER FACTOR (LOF)

The local outlier factor (LOF) approach [41] is the most well-known local anomaly detection algorithm. In this algorithm, the concept of local anomalies is utilized where the LOF score is determined by matching the Local Reachability Density (LRD) of the record with respect to the LRDs of its k-nearest neighbors as illustrated in Algorithm 4. In this approach, first for the test data point D and the training set $\mathcal{X}$, the k-distance $\mathcal{D}_k(D)$ is defined as $\mathcal{D}_k(D) = d(D, x), x \in \mathcal{X}$, where (a) there exist at least k data points $x' \in \mathcal{X}$ such that $d(D, x) \leq d(D, x')$, and (b) there exist at most $k - 1$ data points $x' \in \mathcal{X}$ such that $d(D, x) < d(D, x')$, with $d(D, x)$ denoting the distance between the point D and x that can be found by using different norms. Next, the k-distance neighborhood $\mathcal{N}_k(D)$ is defined as follows

$$\mathcal{N}_k(D) = \{x \in \mathcal{X} | d(D, x) \leq \mathcal{D}_k(D)\}.$$

It should be noted that the cardinality of $\mathcal{N}_k(D)$ that is denoted by $|\mathcal{N}_k(D)|$ can be generally greater than k. Then, the reachability distance of D with respect to $x \in \mathcal{X}$ is defined as

$$\mathcal{R}_k(D, x) = \max\{d(D, x), \mathcal{D}_k(x)\}.$$

Algorithm 4 Local Outlier Factor (LOF) Algorithm

```
Input: $x_i$ - training data, $k$, $D$ - testing data
Output: Test data flag $r$
1: Find the $k$-distance neighborhood $\mathcal{N}_k(D)$
2: Compute the Local Reachability Density (LRD);
       $LRD(D) = \frac{|\mathcal{N}_k(D)|}{\sum_{x \in \mathcal{N}_k(D)} \mathcal{R}_k(D, x)}$
3: Compute $LOF(D) = \frac{\sum_{x \in \mathcal{N}_k(D)} LRD(x)}{LRD(D)|\mathcal{N}_k(D)|}$
4: if $LOF(D) > threshold$ then
5:     $D$ is abnormal, $r = 1$.
6: else
7:     $D$ is normal, $r = 0$.
8: end
```

Next, the Local Reachability Density $LRD(D)$ and the Local Outiler Factor $LOF(D)$ are obtained as explained in Algorithm 4. Finally, the test data point is classified as abnormal if $LOF(D) > 1$.

5) ISOLATION FOREST (IF)

The Isolation Forest (IF) scheme which is an unsupervised machine learning technique [42], [43] is now used as the strategy for performing the cyber attack detection objective. The key advantage of IF with respect to other anomaly detection schemes are as follows: (I) IF scheme does not utilize any distance or density measure to detect an anomaly, which eliminates a major computational cost of distance computations, and (II) it has a linear time-complexity with a constant training time and a minimal memory requirement [44]. These are two key features that are essential for online implementation of the IF for a real-time cyber attack detection process in industrial control systems.

Cyber attack detection using the IF is performed in two stages, namely: (1) training, and (2) real-time testing. In the training phase, isolation trees are constructed by using the sub-samples of the normal healthy system operational dataset. In the online testing phase, the real-time data are fed to the trained IF for performing the cyber attack detection objective.

In the training phase, given the training set $\mathcal{X} = \{x_1, \dots, x_N\}, x_i \in \mathbb{R}^d$ corresponding to the normal operation of the system, m different isolation trees $\mathcal{T}_i, i = 1, \dots, m$ are constructed by recursively splitting a sub-sample $\mathcal{X}_i \subset \mathcal{X}$ until all the data points in $\mathcal{X}_i$ are isolated. For each isolation tree $\mathcal{T}_i$, the sub-sample $\mathcal{X}_i$ is randomly selected without replacement from $\mathcal{X}$ using two hyper-parameters, n as the number of data-point used to train each tree, and $f \leq d$ as the number of features that are selected for training that isolation forest, i.e. $\mathcal{X}_i = \{x_{i,1}, \dots, x_{i,n}\}, x_{i,j} \in \mathbb{R}^f$. Each tree is specified by a set of nodes that are indexed by the pair (j, k), where j denotes the depth of the node and k is the index of that node in the given depth, where $0 \leq k \leq 2^j - 1$. Each internal node (j, k) has two children $(j+1, 2k)$ and $(j+1, 2k+1)$ and the root node is denoted by $(0, 0)$.

The isolation forest as shown in Algorithm 5 is operating based on the concept of binary recursively splitting the feature space $\mathbb{R}^d$ by each isolation tree $\mathcal{T}_i$ as well as with randomly selecting a split feature $q \in \{1, \ldots, f\}$ and its split value p within the selected feature range. The scheme will be initiated with the root node $(0, 0)$ and the training set $\mathcal{X}_i$ and the training set for each node, denoted by $\mathcal{X}_i^{j,k}$ is obtained recursively as follows. At the node (j, k), the data-points are splitted into two subsets $\mathcal{X}_i^{j+1,2k} = \{x_{i,j} \in \mathcal{X}_i^{j,k}, j = 1, \ldots, n | x_{i,j}^q < p\}$ and $\mathcal{X}_i^{j+1,2k+1} = \{x_{i,j} \in \mathcal{X}_i^{j,k}, j = 1, \ldots, n | x_{i,j}^q \geq p\}$, until all samples are isolated, where $x_{i,j}^q$ corresponds to the qth element of $x_{i,j}$. In each splitting step at node (j, k), two children nodes $(j+1, 2k)$ and $(j+1, 2k+1)$ with the corresponding training datasets $\mathcal{X}_i^{j+1,2k}$ and $\mathcal{X}_i^{j+1,2k+1}$ are generated. These can be an internal node if it is still possible to split the corresponding subset or an external node corresponding to the last node in the branch when the size of the data subset of that region is 1, or the maximum tree depth is reached. In case of an internal node, the data subsets $\mathcal{X}_i^{j+1,2k}$ and $\mathcal{X}_i^{j+1,2k+1}$ are further splitted until an external node is reached.

Algorithm 5 Train $\mathcal{T}_i$ ($\mathcal{X}_i$)

```
Input: $\mathcal{X}_i$ - input data
Output: an $\mathcal{T}_i$
1:  Initialization: The root node with index $(0, 0)$ and the training set $\mathcal{X}_i^{0,0} = \mathcal{X}_i$. Set $j = k = 0$.
2:  if $\mathcal{X}_i^{j,k}$ cannot be divided then
3:      The node $(j, k)$ is designated as the external node and no division will be performed for this node.
4:  else
5:      The node $(j, k)$ is designated as the internal node
6:      Randomly select a feature $q \in \{1, \ldots, f\}$
7:      Randomly select a splitting value $p$ between the minimum and the maximum values of the feature $q$ in $\mathcal{X}^{j,k}$
8:      Set $\mathcal{X}_i^{j+1,2k} = \{x_{i,j} \in \mathcal{X}_i^{j,k}, j = 1, \ldots, n | x_{i,j}^q < p\}$
9:      Set $\mathcal{X}_i^{j+1,2k+1} = \{x_{i,j} \in \mathcal{X}_i^{j,k}, j = 1, \ldots, n | x_{i,j}^q \geq p\}$
10:     Recursion: Go to step 2 and continue splitting the nodes $(j+1, 2k)$ and $(j+1, 2k+1)$
11: end
```

The general concept for the cyber attack detection strategy by utilizing the IF is justified and rationalized by the fact that in the process of splitting data, the cyber attacks are different from the normal points and they can be isolated closer to the root of the tree. Consequently, they have a shorter path from the root. In the real-time cyber attack detection process, for each new sample measurement D, the path length in the ith tree $\mathcal{T}_i$, as denoted by $h_i(D)$, is obtained by counting the number of edges from the root node to an external node as the sample D is splitted through the isolation tree $\mathcal{T}_i$. Consequently, the average path length of all trees is obtained as follows:

$$h_{\text{avg}}(D) = \frac{1}{m} \sum_{i=1}^{m} h_i(D), \tag{1}$$

Next, a score value is assigned to the new sample measurement D as

$$s(D) = 2^{-\frac{h_{\text{avg}}(D)}{H}}, \tag{2}$$

where H denotes the average expected path length of the trees in the forest and is given by

$$H = 2\ln(n-1) + 1.2 - 2(n-1)/N, \tag{3}$$

with N denoting as the total number of data-points in $\mathcal{X}$. Finally, the cyber attack is detected by using the detection threshold ϵ as follows:

$$r = \begin{cases} 0 & \text{if } s(D) > \epsilon \\ 1 & \text{if } s(D) \leq \epsilon. \end{cases} \tag{4}$$

C. POST-PROCESSING

As illustrated in Fig. 9, in the post-processing stage, an observation window of the last W data-points is used to perform the cyber attack detection decision making process. Specially, if 80% (80% is chosen based on the mesh search) of W flags $r(\tau)$, $\tau \in [t - W, t]$ corresponding to the the last W data points $x(t)$ are isolated as anomaly by the anomaly detection scheme, then the current data-point $x(t)$ is identified as a cyber attack. The main goal of the window-based post-processing scheme is to reduce the number of false alarms and to produce a smoother decision making process.

VI. PERFORMANCE EVALUATION AND ASSESSMENT

In this section, evaluation and validation of our proposed cyber attack detection schemes are provided and demonstrated for the developed TE testbed infrastructure.

A. DATASET

As previously indicated, the proposed methodologies of this work are demonstrated by using the real datasets that are generated from the implemented ICS testbed. The generated dataset consists of 25 variables such that 16 variables are corresponding to the sensor measurements and 9 variables are corresponding to the actuator signals. Two types of datasets are generated, where initially the testbed was run for almost 72 hours under the normal condition (that is, cyber attack free) for generating the training set of the size (25×96827) (after removing the initial transient behavior), i.e. $N = 96827$. Subsequently, the testbed was run several times subject to different cyber attack scenarios and different cyber attack gateways and points.

Towards this end, false data injection (FDI) cyber attacks are injected in the communication channels between the I/O modules and the corresponding PLC through online scaling the sensor measurement data with the scaling factor λ. Four different cyber attack scaling scenarios are considered as

$\lambda \in \{0.98, 0.96, 0.94, 0.92\}$, with the cyber attack duration of two hours. For instance, PLC 3 receives four measurements, namely, y_{12}, y_{14}, y_{15}, and y_{17} and the cyber attack on PLC 3 can be modeled as:

$$y_{ia} = y_i \times \lambda, \quad i = 12, 14, 15, 17 \tag{5}$$

where y_{ia} corresponds to the i-th measurement under cyber attack. Figure 10 illustrates the FDI on y_{12} and y_{15} in PLC 3 for all the four scaling cyber attack scenarios. These four cyber attack scenarios are repeated for all the five PLCs, and hence 20 different cyber attack scenarios are injected. Consequently, the test dataset with the size of (25×128159) has been generated such that 68113 out of 128159 correspond to cyber attacks and the rest are healthy data. The sampling time for data logging was 2 seconds for both datasets.

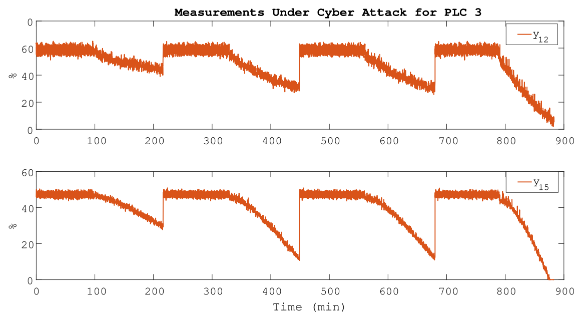


FIGURE 10. Measurements under cyber attacks for PLC3.

B. TRAINING OF PROPOSED METHODOLOGIES

The training of the proposed schemes and structures are performed by using an open-source Machine Learning library for the Python programming language, which is known as the Scikit-learn library and PyOD toolbox [7], [45].

Furthermore, the training is performed by using an 8-fold cross-validation such that each structure is trained 8 times. Moreover, the hyper-parameters of each scheme are set based on the mesh-search around the recommended values in PyOD [7].

C. PERFORMANCE EVALUATION METRICS

The confusion matrix is a form of contingency table with two dimensions identified as True and Predicted, and a set of classes corresponding to both dimensions, as presented in Table 5. The following detection and classification performance metrics are derived from the confusion matrix [46] as follows:

TABLE 5. The confusion matrix.

		Predicted	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

1) ACCURACY

Accuracy specifies the closeness of measurements to a specific category/class and it is computed as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

2) RECALL

Recall is the True Positive Rate (TPR) and is computed as:

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

3) PRECISION

Precision is the Positive Predictive Value (PPV) and is computed as:

$$PPV = \frac{TP}{TP + FP} \tag{8}$$

4) F1 SCORE

F1 Score is the harmonic average of the precision and recall, where it is at its best at a value of 1, implying perfect precision and recall and is computed by:

$$F1 = 2\frac{PPV * TPR}{PPV + TPR} \tag{9}$$

It should be noted that the main aim of this section is to perform a quantitative comparison study of various cyber attack detection schemes is presented using the real-time data generated by the developed testbed.

D. COMPARATIVE TESTING AND VALIDATION RESULTS

In this subsection, a quantitative comparison study of various cyber attack detection schemes is presented. As previously indicated, the field data are collected in real-time from the PLC's local cloud. Therefore, by implementing the cyber attack detection schemes on the process data in real time, the status of the data can be determined online.

Table 6 provides the efficiency of the proposed schemes. As illustrated is Table 6, the IF has the worst performance over the provided datasets due to high oscillation in the detection signal (high number of false negative alarms), while it has the fastest training time (speed) in comparison with the other techniques. Moreover, the OCSVM scheme has achieved quite promising results as compared to other methods. In general, the training speed is directly proportional to the characteristics of the scheme. For instance, the IF infrastructure is based on combination of multiple decision trees (binary) which leads to having a considerably fast training speed. On the other hand, OCSVM scheme calculates the decision boundaries about the data points, and hence its training speed is slow. Table 7 shows the cyber attack detection time (DT) corresponding to various cyber attack scenarios. Overall, as expected from Table 6, the OCSVM and kNN have the fastest detection times and by increasing the cyber attack severity (λ), the cyber attack detection times are generally improved. However, for the IF algorithm, due

TABLE 6. Performance of the proposed schemes.

ML Algorithm	Accuracy	Recall	Precision	F1
PCA	0.9754	0.9568	0.9968	0.9764
OCSVM	**0.9910**	**0.9876**	**0.9954**	**0.9915**
LOF	0.9863	0.9774	0.9969	0.9870
kNN	0.9869	0.9896	0.9859	0.9877
IF	0.9628	0.9338	0.9960	0.9639

TABLE 7. The cyber attack detection time (DT).

DT	λ	PCA	OCSVM	LOF	kNN	IF
PLC 1	98%	9:30	1:22	4:30	1:20	1:22
	96%	3:56	1:22	1:22	1:22	1:22
	94%	1:34	1:22	1:22	1:22	1:22
	92%	1:20	1:20	1:20	1:20	1:20
PLC 2	98%	1:00	1:00	1:00	0:54	0:58
	96%	1:18	1:14	1:18	0:52	0:48
	94%	0:46	0:46	0:46	0:42	0:44
	92%	0:58	0:58	0:58	0:56	1:00
PLC 3	98%	22:24	4:48	10:32	1:32	2:30
	96%	10:0	1:26	4:58	1:26	1:38
	94%	9:24	1:22	4:42	1:20	2:18
	92%	5:48	1:24	2:28	1:22	2:16
PLC 4	98%	13:40	2:00	6:48	1:16	1:36
	96%	6:22	1:20	1:32	1:14	1:24
	94%	3:38	1:22	1:26	1:22	0:18
	92%	1:28	0:02	1:24	0:02	1:00
PLC 5	98%	1:24	1:24	1:24	1:18	11:20
	96%	1:30	1:30	1:30	1:30	4:18
	94%	1:22	1:22	1:22	1:22	12:28
	92%	1:22	1:22	1:20	1:18	11:56
Average DT		**4:56**	**1:26**	**2:36**	**1:11**	**3:06**

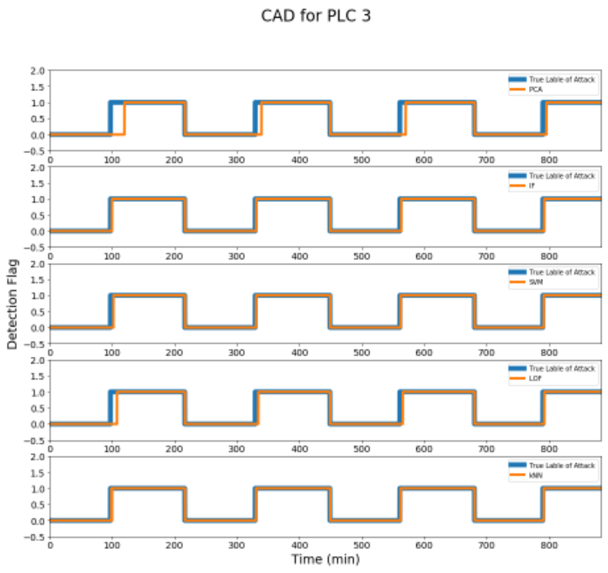


FIGURE 11. Cyber attack detection of the PLC3.

to high oscillations in the original signal and effects of post processing algorithm, by increasing λ the detection times are not improved.

Figures 11 and 12 depict the performance of various cyber attack detection schemes for the scenarios of PLC3 and PLC5, respectively, where the flag "0" represents the healthy data and the flag "1" represents the cyber attack data. The scaling factors in these figures are $\lambda = 0.98, 0.96, 0.94$ and $\lambda = 0.92$, respectively. It should be noted that PLC 5 is the least sensitive one in terms of cyber attack detection due to low number of direct measurements as provided in Table 4. As shown in Figure 12, this fact leads to false negative alarms generation using IF while the other algorithms can still detect the attacks on PLC 5 without any false negative alarm.

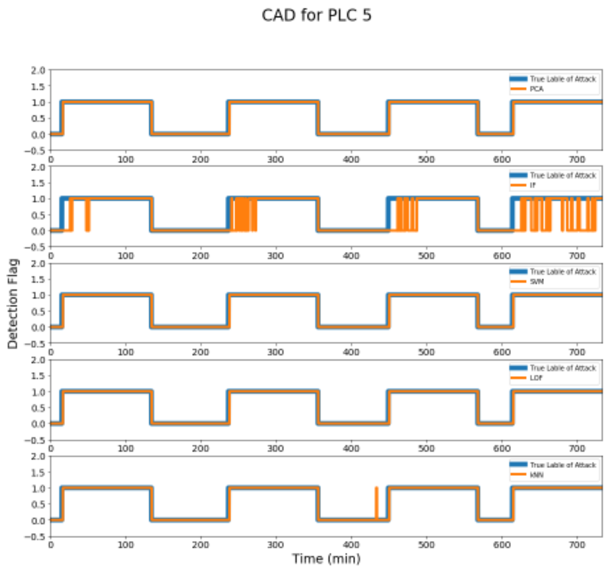


FIGURE 12. Cyber attack detection of the PLC5.

E. THE COMPUTATIONAL COMPLEXITY

The computational complexity analysis of machine learning algorithms can be generally performed by computing the O-notion of each algorithm, which, represents the rate of the growth or the decline of the algorithm computational complexity. In case of the nearest-neighbor based algorithms, the computational complexity of identifying the nearest-neighbors is $O(N^2)$ (where N is the number of samples) and the remaining computations, such as density or the LOF computations, can be ignored in the operations (less than 1% of the runtime). The complexity of the single-class SVM-based scheme is difficult to compute since it depends on the number of support vectors and therefore on the data properties and characteristics of the results. Furthermore, the σ tuning of the SVMs that are used has a significant effect on the runtime as the computations have quadratic complexity. Nevertheless, the complexity of the OCSVM scheme can be scaled between $O(dN^2)$ to $O(dN^3)$, where d denotes the number of features. The computational complexity of the PCA scheme is $O(d^2\ N + d^3)$, and thus it relies strongly on the number of measurements. If the number of dimensions is low, the scheme in practice represent as among the fastest algorithms in our studies. Finally, the complexity of the IF scheme can be obtained to be $O(tNlogN)$, where t denotes the number of trees [42].

VII. DISCUSSION AND CONCLUSION

In this paper, a hybrid testbed is developed and implemented for an industrial control system (ICS) through real-time simulating the Tennessee Eastman (TE) process as the physical component of the testbed and implementing the other layers of the ICS using Siemens modules, such as PLC and distributed I/O. Due to various security aspects of ICS, there are many constraints and challenges in obtaining actual field data. Therefore, by generating and logging the data from the physical part of the proposed testbed, a dataset as close as possible to the real field data is generated. Accordingly, by using this dataset, the impact of various real-time cyber attacks on the system and the corresponding proposed online detection approaches are studied. The Man-In-The-Middle (MITM) cyber attacks are directly implemented on the PROFINET communication protocols such that the malicious hacker can modify the sensor measurements that are sent to the PLC. Subsequently, several cyber attack detection approaches have been developed and implemented in real-time. Table 6 shows the overall performance of each cyber attack detection methodology under various malicious attack scenarios. Furthermore, Table 7 provides the cyber attack detection time for each scheme. Although, all the evaluated schemes have been able to detect the cyber attacks before shut-downing of the plant, however, the OCSVM scheme shows the best performance for this particular application. This study that is based on the proposed testbed can aid in determining the optimum approach for a particular ICS process that is based on specified constraints (e.g. the plant shutdown condition) and requirements (e.g. the plant production rate).

It should be emphasized that none of the previous works in the literature have considered the full Tennessee Eastman process in their developed testbed. Also, to the best of the authors' knowledge, none of the previous work have worked on the PROFINET protocol for injecting real-time cyber attacks. Moreover, in most of the previous work, the cyber attack detection algorithms are implemented off-line after collecting the data from the testbed where as in this work, the cyber attack detection schemes are implemented all in real-time in the supervisory level of the testbed. Hence, in this work the online performance for our proposed cyber attack detection schemes are demonstrated and provided.

Future work will involve the implementation of more complex multi-point cyber attacks on the testbed and evaluation of the performance of cyber attack detection and mitigation schemes in real-time on the testbed.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A survey of industrial control system testbeds," in *Proc. Nordic Conf. Secure IT Syst.* Stockholm, Sweden: Springer, 2015, pp. 11–26.

[2] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of SCADA control systems (TASSCS)," in *Proc. ISGT*, Jan. 2011, pp. 1–7.

[3] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi, "A control system testbed to validate critical infrastructure protection concepts," *Int. J. Crit. Infrastruct. Protection*, vol. 4, no. 2, pp. 88–103, Aug. 2011.

[4] H. Gao, Y. Peng, K. Jia, Z. Dai, and T. Wang, "The design of ICS testbed based on emulation, physical, and simulation (EPS-ICS Testbed)," in *Proc. 9th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Oct. 2013, pp. 420–423.

[5] I. N. Fovino, M. Masera, L. Guidi, and G. Carpi, "An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants," in *Proc. 3rd Int. Conf. Human Syst. Interact.*, May 2010, pp. 679–686.

[6] A. Bathelt, N. L. Ricker, and M. Jelali, "Revision of the tennessee eastman process model," *IFAC-Papers Line*, vol. 48, no. 8, pp. 309–314, 2015.

[7] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, Jan. 2019.

[8] A. Winnicki, M. Krotofil, and D. Gollmann, "Cyber-physical system discovery: Reverse engineering physical processes," in *Proc. 3rd ACM Workshop Cyber-Phys. Syst. Secur.*, Apr. 2017, pp. 3–14.

[9] M. Krotofil, A. Cardenas, and K. Angrishi, "Timing of cyber-physical attacks on process control systems," in *Proc. Int. Conf. Crit. Infrastruct. Protection*, pp. 29–45, Springer, 2014.

[10] M. Krotofil, A. A. Cárdenas, B. Manning, and J. Larsen, "CPS: Driving cyber-physical systems to unsafe operating conditions by timing DoS attacks on sensor signals," in *Proc. 30th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, 2014, pp. 146–155.

[11] M. Krotofil, A. Cardenas, J. Larsen, and D. Gollmann, "Vulnerabilities of cyber-physical systems to stale data—Determining the optimal time to launch attacks," *Int. J. Crit. Infrastruct. Protection*, vol. 7, no. 4, pp. 213–232, 2014.

[12] Z.-S. Lin, A. A. Cárdenas, S. Amin, H.-Y. Tsai, Y.-L. Huang, and S. Sastry, "Security analysis for process control systems," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, 2009, pp. 1–13.

[13] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, 2011, pp. 355–366.

[14] I. Kiss, P. Haller, and A. Bereá, "Denial of service attack detection in case of tennessee eastman challenge process," *Procedia Technol.*, vol. 19, pp. 835–841, Dec. 2015.

[15] I. Kiss, B. Genge, and P. Haller, "A clustering-based approach to detect cyber attacks in process control systems," in *Proc. IEEE 13th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2015, pp. 142–148.

[16] P. Filonov, F. Kitashov, and A. Lavrentyev, "RNN-based early cyber-attack detection for the tennessee eastman process," 2017, *arXiv:1709.02232*. [Online]. Available: http://arxiv.org/abs/1709.02232

[17] G. Bernieri, E. Etchevés Miciolino, F. Pascucci, and R. Setola, "Monitoring system reaction in cyber-physical testbed under cyber-attacks," *Comput. Electr. Eng.*, vol. 59, pp. 86–98, Apr. 2017.

[18] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1092–1105.

[19] C.-T. Lin, S.-L. Wu, and M.-L. Lee, "Cyber attack and defense on industry control systems," in *Proc. IEEE Conf. Dependable Secure Comput.*, Aug. 2017, pp. 524–526.

[20] M. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, p. 76, Aug. 2018.

[21] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-Phys. Syst. Smart Water Netw. (CySWater)*, Apr. 2016, pp. 31–36.

[22] S. Adepu and A. Mathur, "Distributed attack detection in a water treatment plant: Method and case study," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 86–99, Jan. 2021.

[23] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security*, G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, Eds. Cham, Switzerland: Springer, 2017, pp. 88–99.

[24] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: A water distribution testbed for research in the design of secure cyber physical systems," in *Proc. 3rd Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, 2017, pp. 25–28.

[25] V. K. Mishra, V. R. Palleti, and A. Mathur, "A modeling framework for critical infrastructure and its application in detecting cyber-attacks on a water distribution system," *Int. J. Crit. Infrastruct. Protection*, vol. 26, Sep. 2019, Art. no. 100298.

[26] V. S. Koganti, M. Ashrafuzzaman, A. A. Jillepalli, and F. T. Sheldon, "A virtual testbed for security management of industrial control systems," in *Proc. 12th Int. Conf. Malicious Unwanted Softw. (MALWARE)*, Oct. 2017, pp. 85–90.

[27] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4362–4369, Jul. 2019.

[28] R. Negi, P. Kumar, S. Ghosh, S. K. Shukla, and A. Gahlot, "Vulnerability assessment and mitigation for industrial critical infrastructures with cyber physical test bed," in *Proc. IEEE Int. Conf. Ind. Cyber Phys. Syst. (ICPS)*, May 2019, pp. 145–152.

[29] X. Li, C. Zhou, Y.-C. Tian, N. Xiong, and Y. Qin, "Asset-based dynamic impact assessment of cyberattacks for risk analysis in industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 608–618, Feb. 2018.

[30] X. Li, C. Zhou, Y.-C. Tian, and Y. Qin, "A dynamic decision-making approach for intrusion response in industrial control systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 2544–2554, May 2019.

[31] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.

[32] N. L. Ricker and J. H. Lee, "Nonlinear modeling and state estimation for the tennessee eastman challenge process," *Comput. Chem. Eng.*, vol. 19, no. 9, pp. 983–1005, Sep. 1995.

[33] G. Ravi Sriniwas and Y. Arkun, "Control of the tennessee eastman process using input-output models," *J. Process Control*, vol. 7, no. 5, pp. 387–400, Oct. 1997.

[34] S. Mehner and H. König, "No need to marry to change your name! Attacking profinet io automation networks using DCP," in *Proc. Int. Conf. Detection Intrusions*, 2019, pp. 396–414.

[35] J. Akerberg and M. Bjorkman, "Exploring security in PROFINET IO," in *Proc. 33rd Annu. IEEE Int. Comput. Softw. Appl. Conf.*, 2009, pp. 406–412.

[36] PROFIBUS & PROFINET International (PI), Karlsruhe, Germany. *PROFINET Security Guideline*. Accessed: Feb. 14, 2020. [Online]. Available: https://www.profibus.com/download/profinet-security-guideline

[37] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," Naval Res. Lab., Center High Assurance Comput. Syst., Washington, DC, USA, Tech. Rep. OMB No. 0704-0188, 2003.

[38] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[39] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, pp. 1–31, 2016.

[40] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Helsinki, Finland: Springer, 2002, pp. 15–27.

[41] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000,

[42] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[43] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, pp. 1–39, Mar. 2012.

[44] 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.

[45] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[46] K. M. Ting, "Confusion matrix," in *Encyclopedia Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA: Springer, 2010, p. 209.

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 working as an Undergraduate Research Assistant with the KINDI Center for Computing Research, Qatar University, since 2018.

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 at Texas A&M University at Qatar, Doha, Qatar, from January 2010 to December 2010. He is currently an Associate Professor with Qatar University, and an Adjunct Associate Professor with Concordia University. He has published more than 220 refereed journal and conference papers. His research interests include FDI, multiagent systems, active control for clinical pharmacology, cyber-security of industrial control systems, and linear parameter varying systems.

DEVRIM UNAL (Senior Member, IEEE) received the M.Sc. degree in telematics from Sheffield University, U.K., and the Ph.D. degree in computer engineering from Bogazici University, Turkey, in 1998 and 2011, respectively. He is currently a Research Assistant Professor of Cyber Security with the KINDI Center for Computing Research, College of Engineering, Qatar University. His research interests include cyber-physical systems and IoT security, wireless security, artificial intelligence, and next generation networks.

KHASHAYAR KHORASANI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1981, 1982, and 1985, respectively. From 1985 to 1988, he was an Assistant Professor with the University of Michigan at Dearborn, and he has been with Concordia University, Montreal, Canada, since 1988, where he is currently a Professor and a Concordia University Tier I Research Chair with the Department of Electrical and Computer Engineering and the Concordia Institute for Aerospace Design and Innovation (CIADI). His main areas of research interests include nonlinear and adaptive control, cyber-physical systems and cybersecurity, intelligent and autonomous control of networked unmanned systems, fault diagnosis, isolation and recovery (FDIR), diagnosis, prognosis, and health management (DPHM), satellites, unmanned vehicles, and neural networks/machine learning. He has authored/coauthored over 450 publications in these areas. He has served as an Associate Editor for the IEEE Transactions on Aerospace and Electronic Systems.