IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2020, accepted February 15, 2020, date of publication February 19, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975066

A Dual-Isolation-Forests-Based Attack Detection Framework for Industrial Control Systems

MARIAM ELNOUR[1], 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 the Qatar National Research Fund (a member of Qatar Foundation) through the National Priorities Research Program (NPRP) under Grant NPRP 10-0206-170360. The findings achieved herein are solely the responsibility of the authors.

ABSTRACT The cybersecurity of industrial control systems (ICSs) is becoming increasingly critical under the current advancement in the cyber activity and the Internet of Things (IoT) technologies, and their direct impact on several life aspects such as safety, economy, and security. This paper presents a novel semi-supervised dual isolation forests-based (DIF) attack detection system that has been developed using the normal process operation data only and is demonstrated on a scale-down ICS known as the Secure Water Treatment (SWaT) testbed and the Water Distribution (WADI) testbed. The proposed cyber-attack detection framework is composed of two isolation forest models that are trained independently using the normalized raw data and a pre-processed version of the data using Principal Component Analysis (PCA), respectively, to detect attacks by separating-away anomalies. The performance of the proposed method is compared with the previous works, and it demonstrates improvements in terms of the attack detection capability, computational requirements, and applicability to high dimensional systems.

INDEX TERMS Attack detection, principal component analysis (PCA), isolation forest (IF), industrial control systems, cybersecurity.

I. INTRODUCTION

Industrial control systems (ICSs) are composed of electrical and mechanical devices, computers, and manual operations supervised by humans. They are mainly used for partial or full automation control in industrial plants and critical infrastructures such as manufacturing industries, chemical plants, power generation and distribution systems, water treatment plants, and others [1]. Their operation has a direct impact on the environment, the safety and health of people, the economy, and national security. Concerns about the security of industrial control systems are increasing, given the growing sophistication of cyber activities. The advancement in the industrial Internet of Things (IoT) technologies is creating more potential threat points and vulnerabilities in the system. There have been a number of cyber-attacks on critical infrastructures in the past few years [2]–[4], and research in cybersecurity of industrial control systems has been evolving to overcome the challenges and vulnerabilities in the current industrial attack detection systems.

Attack detection systems are designed to monitor the events taking place in an information system in order to identify signs of security issues. Anomaly detection is the most commonly used approach for attack detection, which is the process of identifying anomalous events that do not conform to the expected behavior of the system. The main underlying advantage of the anomaly detection approach is its ability to detect unseen and new attacks. Anomaly detection-based attack detection approaches can be implemented using a variety of Machine Learning (ML) algorithms such as Support Vector Machine (SVM) [5], [6], Principal Component Analysis (PCA) [7], Neural Networks [8], clustering analysis [9], Negative Selection Algorithm (NSA) [10], and others. They can be divided into unsupervised, supervised, and semi-supervised learning approaches. In the unsupervised method, the model is developed using unlabeled data that contain normal and anomalous samples, while the labeled normal and attack data are used in the supervised learning scheme.

The associate editor coordinating the review of this manuscript and approving it for publication was Ana Lucila Sandoval Orozco.

However, in the semi-supervised approach, the model is developed using the normal operation data only.

The work presented in this paper is demonstrated using the datasets obtained from the iTrust Lab testbeds, which are the Secure Water Treatment (SWaT) testbed and the Water Distribution (WADI) testbed. There have been several works in attack detection using the SWaT dataset as in [10]–[22] and limited work using the WADI dataset as in [17]. Most of the previous works on attack detection utilized the normal process data using several ML algorithms such as Negative Selection Algorithm (NSA) [10], Singular Value Decomposition (SVD) [11], Standard Neural Networks (NNs) [12], [13], Convolutional Neural Networks (CNNs) [14], Recurrent Neural Networks (RNNs) [15], [16], and Generative Adversarial Network (GAN) implemented using the Long-Short-Term Memory (LSTM) network [17]. They are based on constructing a model that is able to profile normal system behavior, and then non-conforming observations are identified as anomalies. In [18], an attack detection approach is proposed based on a graphical model developed using a probabilistic deterministic real-time automaton model and a Bayesian network, named as the Time Automata and Bayesian netwORk (TABOR) approach. In [19], supervised learning is used to develop a detection model using SVM. A network-based attack detection system is proposed in [20] to detect attacks in particular communication links in the SWaT testbed. In addition, model-based attack detection methods are proposed in [21], [22] for the SWaT system using approximated discrete models in which invariants are derived from process dynamics and state entanglement among the physical components, to detect attacks.

From the computational overhead aspect, model-based approaches are considered relatively more efficient than data-driven ones for large-sized systems [23]. In addition, the computational complexity differs among the different Machine Learning algorithms as it is well known that CNNs and RNNs involve extensive computations in both training and evaluation phases, while NNs have less computational requirement ranging from average to high [24]. Comparatively, standard ML algorithms such as SVD, PCA, SVM, NSA, etc. are characterized by their low to average computational complexity depending on the problem size [25], [26].

However, model-based approaches in [21], [22] have some limitations such as modeling approximations given the complexity associated with some processes in the system (i.e., the chemical processes, etc.), which affect the detection accuracy. Nevertheless, the difficulty, effort, and time requirements for the system modeling rise with the increase in the complexity of the system, and the reliability of the detection approach is likely to degrade. Even though in [21] the authors proposed an approach for analyzing the security matter of the SWaT testbed such as the vulnerabilities of the system and the possible attack scenarios that can be discovered, the possibility of using this approach in launching attacks that cannot be detected by other approaches, specifically the data-driven methods, depends on the quality of the used system models. In addition, developing high-fidelity system models becomes more challenging as the complexity, the size, and the non-linearity of the system increase.

Methods proposed in [10]–[13], [15], [16] might have the drawbacks of high missed alarm rate and poor performance for high dimensional data. In addition, some approaches have high computational cost such as in [14]–[17], and others, e.g., [10], [11] do not make full use of the process information by disregarding the actuator signals that may contain valuable input about the process status. In addition, the approach proposed in [18] requires that the variables selection must be made manually and empirically by the designer based on the dynamic behavior. The disadvantage of the supervised learning-based attack detection system proposed in [19] is its dependency on the attack data- which are scarce - and the low accuracy of the detection model under new and unseen attack scenarios. TABLE 1 presents a summary of the previous works done using the SWaT and WADI datasets for intrusion and attack detection.

In this paper, we present a dual isolation forests-based (DIF) attack detection framework for industrial control systems in water treatment plants. The two isolation forest models are trained independently, one using the normalized raw data and the other using a pre-processed version of the data using PCA. The idea behind using two models is to inspect the data in two representations; one in the original data space and the other in the principal component space, thus, elevating the capability of the detection approach. Its main objective is to address the limitations of the previous works given that isolation forests have low computational complexity and high applicability to complex and high dimensional data. They can be used on mixed datasets-containing continuous and discrete variables- that facilitates harnessing the available data when developing the model. They can be used in both semi-supervised and unsupervised learning schemes. Unlike most of the previous works, they are based on pointing out anomalies using the concept of isolation, which improves the attack detection capability. There have been a couple of implementations of isolation forest-based approaches for attack detection, such as in [27] for smart grid networks and in [28] for information security.

The contributions of this work can be summarized as follows:

1) A dual-isolation forests-based attack detection framework is proposed for industrial control systems in water treatment plants utilizing the normal process data of actuator signals and sensor measurements.
2) The proposed approach is based on the principle of separating-away observations that are anomalous, which improves its ability to detect attacks.
3) Due to the nature of the isolation forest, it can harness the available information about the process by analyzing the relations between the different system variables, which are the sensor measurements and the actuator signals.

TABLE 1. Summary of the previous works on the iTrust Lab datasets for attack and intrusion detection.

Ref	Dataset	Method	Process/Network Data	Computational complexity
[10]	SWaT	Data-driven - Semi-supervised (NSA)	Process (sensors)	Average
[11]	SWaT	Data-driven - Semi-supervised (SVD)	Process (sensors)	Low
[12]	SWaT	Data-driven - Semi-supervised (NN)	Process (sensors and actuators)	Average
[13]	SWaT	Data-driven - Semi-supervised (NN)	Process (sensors)	Average
[14]	SWaT	Data-driven - Semi-supervised (1D-CNN)	Process (sensors and actuators)	High
[15]	SWaT	Data-driven - Semi-supervised (RNN)	Process (sensors and actuators)	High
[16]	SWaT	Data-driven - Semi-supervised (RNN)	Process (sensors and actuators)	High
[17]	SWAT & WADI	Data-driven - Semi-supervised (GAN)	Process (sensors and actuators)	High
[18]	SWaT	Data-driven - Semi-supervised (TABOR)	Process (sensors and actuators)	Average
[19]	SWaT	Data-driven - Supervised (SVM)	Process (sensors and actuators)	Average
[20]	SWaT	Data-driven - Unsupervised	Network	Average
[21], [22]	SWaT	Model-based	Process (sensors and actuators)	Low

4) It can exploit the available data of the system by learning from the process data in the original, as well as the PCA-transformed representations.
5) It provides an efficient solution in terms of computational complexity when compared to Deep Learning-based approaches.

The paper is organized as follows. The description of the systems under study is presented in Section II. In Section III, the details of the proposed approach are presented. The models training procedure and the used performance evaluation metrics are explained in Section IV, along with the evaluation and comparison results. Finally, conclusions and future work are summarized in Section V.

II. SYSTEM DESCRIPTION

The work presented in this paper utilizes the experimental process data from the Secure Water Treatment (SWaT) testbed [29], [30] and the Water Distribution (WADI) testbed [31] developed by iTrust Lab at Singapore University of Technology and Design in order to promote research work in the area of cybersecurity of ICSs. The details of the two testbeds are presented in the following subsections.

A. SECURE WATER TREATMENT (SWAT) TESTBED

The SWaT testbed is a scaled-down water treatment plant that is composed of 6 processes, as demonstrated in FIGURE 1, and is capable of producing 5 gallons per minute of fresh water. The data were collected for a total of 11 days in which 36 different attacks were injected during the last four days by hijacking the packets in the communication links between the SCADA system and the Programmable Logic Controllers (PLCs) comprising around 6% of the total data samples. The network packets were altered to reflect the spoofed values from the sensors [29]. The dataset consists of measurements from a total of 25 sensors for water level, flow rate, pressure, and chemical decomposition, and signals from 26 actuators, such as pumps and valves. The description of the SWaT attack scenarios is provided in TABLE 2.

B. WATER DISTRIBUTION (WADI) TESTBED

The WADI testbed is an operational testbed supplying 10 gallons per minute of filtered water. It represents a scaled-down version of a large water distribution network in a city. It contains three distinct control processes labeled as P1 to P3, as presented in FIGURE 2, each controlled by its own set of PLCs. It consists of a number of large water tanks that supply water to consumer tanks. The dataset captures the testbed operation for 16 days; it consists of a total of 59 sensor measurements and 45 actuator signals. It also includes the control signals of 7 actuators with their setpoints. The dataset contains 15 attacks that were injected during the last 2 days of the testbed operation targeting the components of the cyber-physical system with the intention of interrupting the water supply to the consumer tanks. They were conducted by opening valves and spoofing sensor readings. A description of the WADI attack scenarios is provided in TABLE 3.

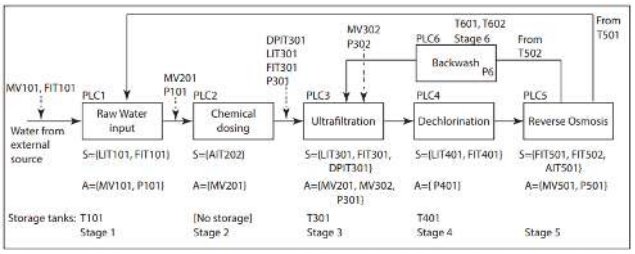

FIGURE 1. The physical water treatment process in the SWaT testbed. P1 through P6 indicate the six stages in the SWaT process - with each having its dedicated PLC - starting with the raw water intake, then the pre-treatment and filtration stage, and finally the reverse osmosis process. Solid arrows indicate the flow of water or chemicals in the dosing station. Dashed arrows indicate potential cyber-attack points. LIT: Level Indicator and Transmitter; Pxxx: Pump; AITxxx: Property indicator and Transmitter; DPIT: Differential Pressure Indicator and Transmitter [32].

III. PROPOSED METHOD

The proposed framework is developed utilizing the normal process data of the actuator signals and sensor measurements, and it is composed of two Isolation Forest (IF) models. The first IF model is developed using the normalized raw data while the second IF model is trained after performing PCA on the normalized continuous-time system variables, as illustrated in FIGURE 3. The aim of the dual isolation-forests framework is to exploit the system data by examining it using two representations to extract useful information that improves the process of separating-away/isolating anomalies.

TABLE 2. Description of the attack scenarios on the SWaT testbed.

Attack	Description	Impact	Attacker intent met?
1	MV-101 is open while it should be closed	Tank overflow	Yes
2	P-102 is turned ON while it should be OFF	Pipe bursts	Yes
3	LIT-101 reading is increased 1 mm per second	Tank underflow and damage P-101	Yes
4	MV-504 is open while it should be closed	No impact	No
5	AIT-202 reading is reduced below the nominal value	AIT-504 increased, and drainage did not start	No
6	LIT-301 reading is increased above the maximum limit	Tank underflow and damage P-101	Yes
7	DPIT-301 reading is increased above the nominal value	Backwash process re-started, decrease Tank-401 water level, increase Tank 301 water level	Yes
8-9	FIT-401 reading is reduced below the nominal value	UV process shutdown and P-501 turns OFF;	Yes
10	MV-304 is closed while it should be open	MV-304 was closed	No
11	MV-303 is stuck at the closed position	No impact	No
12	LIT-301 reading is decreased by 1 mm per second	Tank overflow	Yes
13	MV-303 is stuck at the closed position	Halt stage 3 operation	Yes
14-15	AIT-504 reading is increased above the nominal value	No impact	No
16	MV-101 is stuck at the open position, and LIT-101 reading is set as 0.7 m	Tank overflow	Yes
17	UV-401 is OFF while it should be ON, AIT-502 reading is increased above the nominal value, and P-501 is stuck at ON mode	Reduced output at FIT-502	No
18	DPIT-301 reading is increased above the nominal value, MV-302 is stuck at the open position, and P-602 is stuck at OFF mode	System freeze	Yes
19	P-203 and P-205 are turned OFF while they should be ON	No impact	No
20	LIT-401 reading is increased above the nominal value, and P-205 is stuck at ON mode	Tank overflow	Yes
21	P-101 is stuck at ON mode while it should be OFF, and LIT-301 reading is set at 0.8 m	Tank-101 underflow and Tank-301 overflow	Yes
22	P-302 is stuck at ON mode, and LIT-401 reading is set at 0.6 m	Tank overflow	Yes
23	P-302 is turned OFF while it should be ON	Inflow to Tank-401 stops	Yes
24	P-201, P-203, and P-205 are turned ON while they should be OFF	No impact	No
25	P-101 and MV-101 are stuck at ON mode while they should be OFF, and LIT-101 reading is set at 0.7 m	Tank-101 underflow and Tank-301 overflow	Yes
26	LIT-401 reading is decreased below the minimum level	Tank overflow	Yes
27	LIT-301 reading is increased above the maximum level	Tank underflow and damage P-302	Yes
28	LIT-101 reading is increased above the maximum level	Tank underflow and damage P-101	Yes
29	P-101 is turned OFF while it should be ON	Backup pump P-102 turned ON	No
30	P-101 is turned OFF while it should be ON, and P-102 is stuck at OFF mode	Outflow stops	Yes
31	LIT-101 reading is decreased below the minimum level	Tank overflow	Yes
32	P-501 is turned OFF while it should be ON, and FIT-502 reading is set above the nominal value	P-501 did not turn off, FIT-502 decreased, Speed of P-501 increased	No
33	AIT-402 and AIT-502 readings are set to 260	No impact	No
34	FIT-401 and AIT-501 readings are set to 0.5 and 140 mV	No impact	No
35	FIT-401 reading is set to zero	P-402 did not close	No
36	LIT-301 reading is decreased by 0.55 per second	Tank water level decreases	No

TABLE 3. Description of the attack scenarios on the WADI testbed.

Attack	Description	Impact	Attacker intent met?
1	1-MV-001 is open while it should be closed	Tank overflow	No
2	1-FIT-001 reading is tampered with	Change Water quality	Yes
3-4	1-AIT-001 reading is tampered with	Tank underflow	Yes
5	2-MCV-101 to 2-MCV-601 are closed while they should be open	Interfere with the water distribution process	Yes
6	2-MCV-101 and 2-MCV-201 are open while they should be closed	Interfere with the water distribution process	Yes
7	1-AIT-002 reading is tampered with and open 2-MV-003 while it should be closed	Change Water quality	Yes
8, 11, 12	2-MCV-007 is open while it should be closed	Interfere with the water distribution process	No
9	1-P-006 is turned ON while it should be off	Pipe bursts	Yes
10	Cause damage to 1-MV-001 and raw water pump	Tank underflow	Yes
13	Reduce pressure pump setpoint	Interfere with the water distribution process	Yes
14	Stop chemical dosing pumps	Change Water quality	Yes
15	AIT-001 reading is tampered with	Tank overflow	Yes

In the following subsections, we provide the details and the theoretical background of the algorithms used in the proposed method.

A. DATA PRE-PROCESSING ALGORITHMS

Machine Learning (ML) is about data analysis using algorithms and statistical models in order to build models capable of predicting outcomes given the input data. Machine Learning-based models are highly dependent on the data used to develop them. The performance and accuracy of the ML models are tied to the quality and representation of the data used. The model's ability to learn and extract useful information for the purpose of the application can be limited if the raw data are complex, redundant, contaminated with noise, etc. Hence, data pre-processing is an essential step in Machine Learning applications to improve learning. It involves data normalization, feature selection/extraction, dimensionality reduction, noise filtering, etc. There are various data pre-processing approaches that are commonly used. The following subsections present the ones used in this work.

1) DATA NORMALIZATION

Data normalization is performed by shifting the data to have a zero mean, and it may include standardization, which is done by scaling the data to have a unit variance. Data normalization

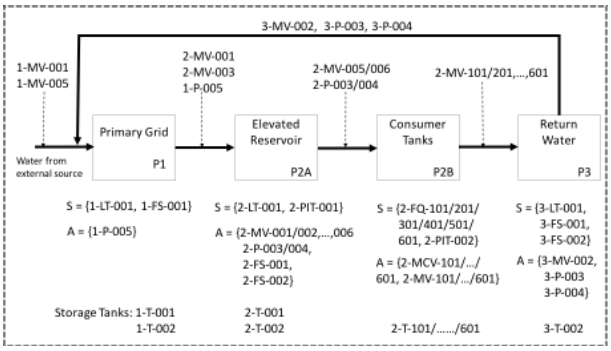

FIGURE 2. There are three processes in the WADI testbed labeled as P1 to P3. P1 is the primary grid process in which the water intake from the SWaT testbed product water or from the return water from P3 in WADI is stored in two storage tanks T-001 and T-002. The storage water tanks in P1 supply water to two elevated reservoir tanks in P2, which is the water distribution process to the six consumer tanks based on the demand. In P3, the recycled water is sent back to P1 once consumer tanks meet their demands. Solid arrows indicate the flow of water and sequence of processes. S and A represent sets of sensors and actuators, respectively. 1-LT-001: level sensor in stage 1 and tank 1; 1-FS-001: flow meter 1 in stage 1; 1-T-001: Tank 1 in stage 1; 2-MV-001: motorized valve 1 in stage 2; 2-MCV-101: motorized consumer valve 1 in stage 2; and 3-P-004: water pump 4 at stage 3 [31].

is useful to speed up the learning/training of the model and to optimize the algorithm results since most of the ML algorithms are about solving optimization problems (maximization/minimization), and hence depending on the nature of the data, the learning of the ML-models can be slow and even fall short due to any local optima. Data standardization is usually performed before applying machine learning algorithms that assume that the input data follow a Gaussian Distribution. The analysis is simpler if the input data follow the standard normal distribution of a zero mean and unit variance.

2) PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA is a multivariate statistical analysis method defined as a linear transformation of a set of correlated variables into a new set of uncorrelated variables. It is widely used in data dimensionality reduction. Given a measurement data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ where n is the number of variables, and m is the number of observations, a PCA model is developed using the normalized data matrix $\bar{\mathbf{X}} \in \mathbb{R}^{m\times n}$ by optimizing the correlation matrix $\mathbf{C} \in \mathbb{R}^{n\times n}$ to find a new set of bases that are uncorrelated to represent the data, namely the principal components (PCs). The correlation matrix is calculated as:

$$\mathbf{C} = \frac{\bar{\mathbf{X}}^{\mathrm{T}}\bar{\mathbf{X}}}{m-1}. \tag{1}$$

Then, the eigenvalue decomposition of the correlation matrix is found by:

$$\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}, \tag{2}$$

where $\mathbf{V} \in \mathbb{R}^{n\times n}$ is the matrix of the eigenvectors associated with each of the eigenvalues of $\mathbf{C}$, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix of the eigenvalues of $\mathbf{C}$ with λ_1 and λ_n are the largest and the lowest eigenvalues, respectively. The projection matrix $\mathbf{P} \in \mathbb{R}^{n\times l}$ is used to transform the data onto the new feature subspace. It is composed of the first l eigenvectors of the correlation matrix that are associated with the largest eigenvalues. That is, $\mathbf{V} = [\mathbf{P}, \tilde{\mathbf{P}}]$ where $\mathbf{P} \in \mathbb{R}^{n\times l}$, $\tilde{\mathbf{P}} \in \mathbb{R}^{n\times(n-l)}$, and l is the number of PCs. It is determined based on the desired explained cumulative variance contribution. PCA transforms the data into two subspaces; the principal components subspace (PCS) and the residual subspace (RS). The data transformation of a normalized measurement vector $x \in \mathbb{R}^{1\times n}$ to the new data vector $\hat{x} \in \mathbb{R}^{1\times l}$ in the PCS is expressed as:

$$\hat{x} = x\mathbf{P}. \tag{3}$$

B. ISOLATION FOREST-BASED ANOMALY DETECTION APPROACH

Isolation Forest (IF) is an unsupervised Machine Learning algorithm that is used for anomaly detection [33], [34]. It is an ensemble regressor encompassing a number of isolation trees in which each tree is trained on a random subset of the training data, as described in Algorithm 1. The parameters associated with an isolation forest are:

1) The number of trees ($n_{\text{estimators}}$),
2) The maximum number of observations ($m_{\max}$) representing the size of the data subset used to train each tree,
3) The maximum number of features ($n_{\max}$) representing the subset of the data features used to train each tree.

Algorithm 1 Train *Forest* $(X, n_{\text{estimators}}, m_{\max}, n_{\max})$

Input: X - input data, $n_{\text{estimators}}$ - number of trees, $m_{\max}$ - size of data subset, $n_{\max}$ - features of data subset
Output: a set of $n_{\text{estimators}}$ *iTrees*
  Initialize *Forest*
  for $i = 1$ to $n_{\text{estimators}}$ **do**
    $X' \leftarrow \text{sample}(X, m_{\max}, n_{\max})$
    $Forest \leftarrow Forest \cup iTree\left(X'\right)$
  end
  return *Forest*

As shown in Algorithm 2, the isolation forest uses the concept of isolation to separate-away anomalies in which recursive binary splitting is performed by each isolation tree (*i*Tree) for the random data subset X' by randomly selecting a split feature q and its split value p -that is within its range- yielding a left X_{l} and right X_{r} data subsets each time until all samples are isolated. Each split produces a node, which can be an internal node if there are further possible splitting in the corresponding split regions or an external node meaning it is 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 the case of an internal node, the data subsets of the two branches of the node X_{l} and X_{r} are further split until an external node is reached. The information associated with the external node is the size of the data subset in that region.

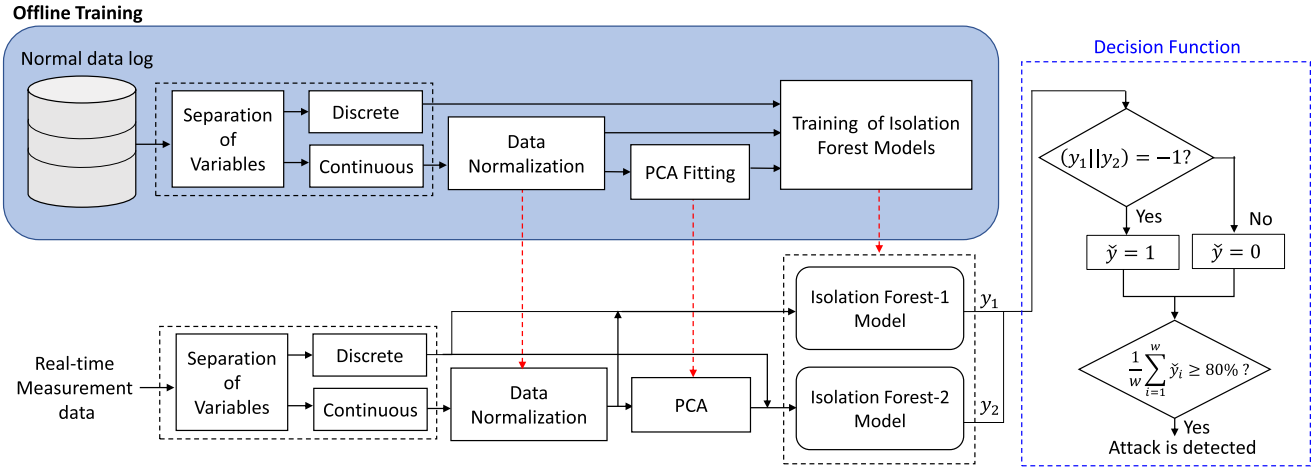

FIGURE 3. The framework of the dual isolation forests-based attack detection approach. It starts with the offline training in which the first step is separating the continuous and discrete variables. For the SWaT dataset, the system has a total of 51 variables with 26 discrete variables and 25 continuous variables, while for the WADI dataset, there are a total of 117 variables in which 53 are discrete, and 64 are continuous variables. PCA is performed on the normalized continuous variables to retain a cumulative explained variance of 95%, and the numbers of PCs found for the SWaT and the WADI datasets are 12 and 33, respectively. The original and the PCA-processed versions of the datasets are used to develop the isolation forests models. The same steps are followed in the online detection using the PCA and isolation forests models obtained from the offline training along with the information about the data normalization procedure -indicated by the red dotted arrows-. The final decision is made by observing the predictions of the two isolation forest models for a time frame of w time instants.

Algorithm 2 Train *iTree* (X')

Input: X' - input data
Output: an *iTree*
  if X' cannot be divided **then**
    return *external*Node{Size$\leftarrow |X'|$}
  else
    let Q be the list of features in X'
    randomly select a feature $q \in Q$
    randomly select a split point p between max and min values of feature q in X'
    $X_\mathrm{l} \leftarrow$ filter $(X', q < p)$
    $X_\mathrm{r} \leftarrow$ filter $(X', q \geq p)$
    return *internal*Node{Left $\leftarrow$ *i*Tree(X_l),
    Right $\leftarrow$ *i*Tree(X_r), FeatureSplit $\leftarrow q$, SplitValue $\leftarrow p$}
  end

Anomalies are different from normal observations, and they can be easily isolated. Hence, it is expected that they will be closer to the root and hence have a shorter path. The anomaly detection for a given data sample x is made upon the score $s(x)$ relative to the detection threshold ϵ as follows:

$$s(x) = 2^{-\frac{\bar{h}(x)}{\mathrm{H}}}, \tag{4}$$

$$\mathrm{H} = 2\ln(m_{\max} - 1) + 1.2 - 2(m_{\max} - 1)/m, \tag{5}$$

where H is the average expected path length of trees in the forest provided that anomalies are labeled as -1 while normal observations are labeled with 1, and $\bar{h}(x)$ denotes the average path length on all trees defined as:

$$\bar{h}(x) = 1/n_{\text{estimators}} \sum_{i=1}^{n_{\text{estimators}}} h_i(x). \tag{6}$$

Here, $h_i(x)$ is the path length of the ith tree determined by the number of edges in the tree. Then, the anomaly is detected using the following function:

$$y = \begin{cases} 1 & \text{if } s(x) > \epsilon \\ -1 & \text{if } s(x) \leq \epsilon. \end{cases} \tag{7}$$

For the proposed DIF-based attack detection framework presented in FIGURE 3, the two isolation forest models yield the outputs y_1 and y_2, which in turn -through a logical operation- produce the attack indicator $\check{y}$. That is, if either of y_1 or y_2 is -1, the attack indicator $\check{y}$ is one; otherwise, $\check{y}$ is zero. The decision function of the dual isolation forests-based attack detection approach is made by checking an observation window of length w such that an attack is detected if the attack indicator $\check{y}$ is 1 for at least 80% of the observation period.

IV. EVALUATION

This section presents the evaluation of the proposed attack detection framework in terms of the used performance metrics, datasets description, models training details, and the evaluation results.

A. PERFORMANCE METRICS

The confusion matrix is a form of contingency table with two dimensions identified as True and Predicted, and a set of classes in both dimensions, as presented in TABLE 4. The following performance metrics are derived from the confusion matrix [35]:

1) PRECISION

It is also called the Positive Predictive Value (PPV), which is a measure of the closeness of the set of predicted results, and

TABLE 4. The confusion matrix.

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

it is expressed as:

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

2) RECALL

It is also known as the True Positive Rate (TPR) and is calculated by:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

3) $\text{F}_1 - \text{score}$

It is the harmonic average of the precision and recall, where it is at its best at a value of 1, meaning perfect precision and recall. It is given by:

$$\text{F}_1 - \text{score} = 2\frac{\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} \tag{10}$$

B. DATASETS

As mentioned previously, the work presented in this paper is demonstrated using the iTrust Lab datasets, which are the SWaT and the WADI. The SWaT dataset consists of a total of 51 variables, 25 of which are sensor measurements (all are continuous variables), and 26 variables are actuator signals (all are discrete variables). The WADI dataset comprises of 117 variables with 59 sensor measurements (44 are continuous variables, and 15 are discrete variables), 45 actuator signals (7 are continuous variables, and 38 are discrete variables), 7 controller output signals (all are continuous variables), and 6 time-varying setpoints (all are continuous variables). The SWaT and the WADI datasets contain two data logs, the first one contains normal process data only collected for 7 and 14 days, respectively, while the second one consists of data for the system operation under both normal and attack scenarios for 4 and 2 days, respectively, at a sampling time of 1 second.

The first step is to clean the second data log by removing the data collected during 1 hour after each attack was terminated because the system behavior in that period is vague and might result in biasing the performance evaluation of the developed models. That is, it represents a recovery period from the attack impact during which the system stabilizes back to its steady-state normal behavior. Considering the actual labeling of this time period, the observations are labeled as normal and attack-free time instants. While behavior-wise, they are anomalous, which in turn would induce false positives and bias the performance evaluation of the proposed approach.

The normal and attack observations in the second log are separated since it was noticed that the normal operational data in the second logs seem to represent a different operational mode -different distribution-. When developing the machine learning model, the distribution of the training and the validation datasets should be the same. The dataset used to train and develop the Machine Learning model should be representative of the system operation. Finally, the steady-state combined normal process data from the two logs are used for developing the proposed detection approach. The attack data subset is used to test the detection model performance.

C. MODEL TRAINING

The training of the isolation forest models is conducted using Scikit-learn library, which is an open-source Machine Learning library for the Python programming language [36]. It is conducted using 5-fold cross-validation such that each IF model is trained 5 times using 80% of the training dataset for training and 20% for validation, selected randomly. Grid search is utilized for model tuning given the limited number of hyper-parameters associated with the isolation forest model for the ranges presented in TABLE 5 and with the objective of achieving a maximum false alarm rate of 5% on the training dataset. The PC used for the training has 64 GB RAM and 12-cores AMD Ryzen 9 3900X CPU with 3.8 GHz speed using 64 bit Windows 10 Pro OS.

TABLE 5. Ranges of the hyper-parameter values for the grid search.

$n_{\text{estimators}}$	m_{max}	n_{max}
100 - 500	2^8 - 2^{16}	5 - 50

The two IF models are trained independently using the normalized raw data and the PCA-processed data, respectively. PCA is applied to the continuous-time variables to retain an explained cumulative variance of 95%. For instance, FIGURE 4 and FIGURE 5 present examples of the SWaT dataset visualizations before and after PCA processing. It can be seen that the normal and the attack observations are somewhat fused when viewing the data in the original representation while they are decoupled in the PCA-transformed representation.

The details of the best models of the two isolation forests are presented in TABLE 6. As inferred from [33], the performance of the isolation forest converges in terms of the number of trees $n_{\text{estimators}}$ used, and it was found converging at 100 and 250 for the SWaT models, and at 100 and 150 for the WADI models, respectively with minimal further improvements in the detection performance at a higher cost of the training time. FIGURE 6 and FIGURE 7 represent a demonstration example on the SWaT dataset for the effect of varying the number of trees by the Receiver Operating Characteristic (ROC) curves.

In addition, the effect of the number of features n_{max} used to train trees in the isolation forest model was minor while the size of the data subset m_{max} used for training each tree showed noticeable effects on the model's performance since the data subset size determines the average path length as

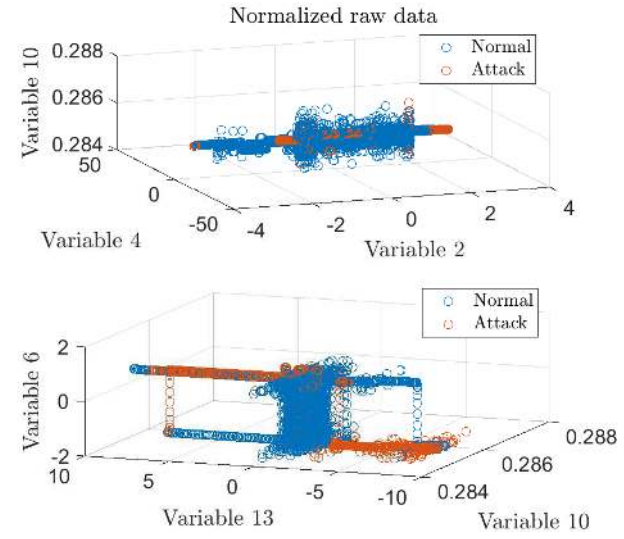

FIGURE 4. Visualizations of random subsets of the SWaT normalized raw dataset.

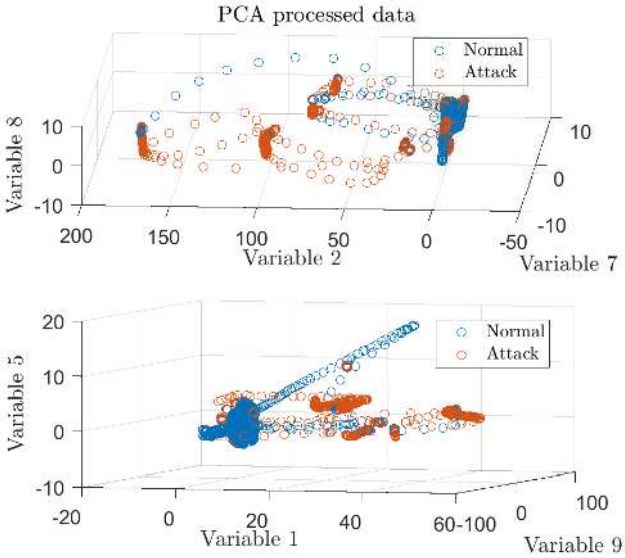

FIGURE 5. Visualizations of random subsets of the SWaT PCA-processed dataset.

TABLE 6. Details of the isolation forest models.

Dataset	Model	$n_{\text{estimators}}$	$n_{\max}$	$m_{\max}$
SWaT	IF-1 model	100	15	5,000
	IF-2 model	250	5	5,000
WADI	IF-1 model	100	45	9,000
	IF-2 model	150	5	31,000

determined using Equation (5). This is demonstrated using the ROC curves shown in FIGURE 8 to FIGURE 11 on the SWaT dataset as well.

Sometimes the system behavior under some attacks is indistinguishable from the ones during the normal operation. Hence, the benefit of using the dual examination of the raw dataset with the original interdependency between the

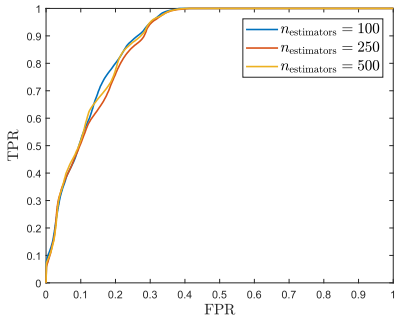

FIGURE 6. ROC curves for models of Isolation Forest-1 with $m_{\max} = 10000$, $n_{\max} = 10$, and varying $n_{\text{estimators}}$.

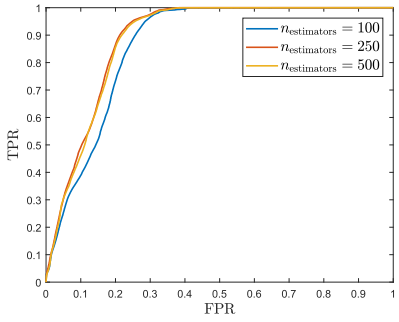

FIGURE 7. ROC curves for models of Isolation Forest-2 with $m_{\max} = 5000$, $n_{\max} = 10$, and varying $n_{\text{estimators}}$.

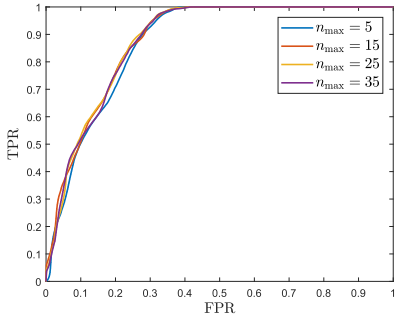

FIGURE 8. ROC curves for models of Isolation Forest-1 with $n_{\text{estimators}} = 250$, $m_{\max} = 5000$, and varying $n_{\max}$.

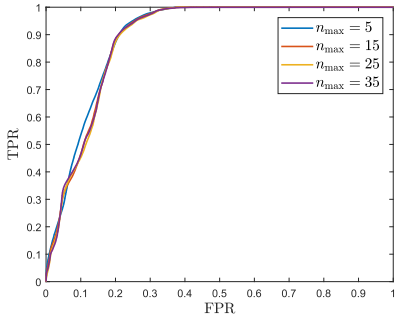

FIGURE 9. ROC curves for models of Isolation Forest-2 with $n_{\text{estimators}} = 250$, $m_{\max} = 5000$, and varying $n_{\max}$.

different variables of the system and comparing it with a cleaner version of the dataset after extracting the uncorrelated components, and removing the redundancy in the data is to help extract additional attacks. FIGURE 12 and FIGURE 13 demonstrate the performance of the IF-1 and the IF-2 models

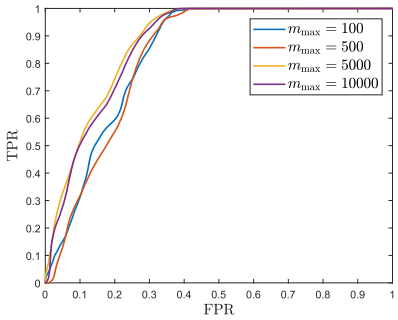

FIGURE 10. ROC curves for models of Isolation Forest-1 with $n_{\text{estimators}} = 250$, $n_{\max} = 5$, and varying $m_{\max}$.

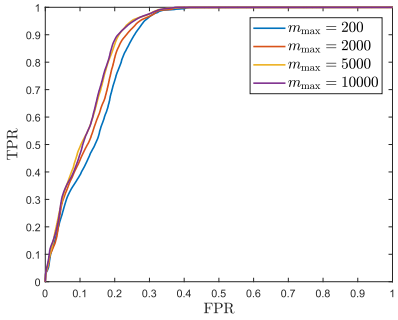

FIGURE 11. ROC curves for models of Isolation Forest-2 with $n_{\text{estimators}} = 250$, $n_{\max} = 10$, and varying $m_{\max}$.

in detecting the SWaT attacks and the WADI attacks, respectively, such that the y-axis represents the recall value and the x-axis is the attack index. For the SWaT testbed, it can be seen that some attacks are detected using the IF-2 model that cannot be detected by the IF-1 model such as Attacks 3, 5, 6, 8, and 9 and vice verse. Again, as demonstrated in FIGURE 4 and FIGURE 5, the performance of the IF-2 is better since after performing data dimensionality reduction using PCA, the redundancy and the uncorrelated components in the data are removed, and the data is less fused such that it is easier to isolate away anomalies. Similarly for the WADI dataset, some attacks are detectable by the IF-1 but are not by the IF-2 such that Attacks 3, 4, 5, 10, 11, 12, 13, 14, and 15 are detected by the IF-1 model while Attacks 2, 7, and 9 are detected by the IF-2 model. It is worth noting that the selection of the detection thresholds for the two models is based on a maximum of 5% false alarm rate. That is, the score values in the scenarios that the attacks do not reflect on the system variables are expected to be comparable. When the detection threshold is set, those score values might fall below this threshold, and hence, be considered as attack incidents and vice versa.

D. COMPARISON WITH OTHER APPROACHES

We compared the proposed method with the other applied approaches in the literature that have been developed using the SWaT and WADI datasets. It is worth noting that the followed data pre-processing procedure in most of the previous works presented in this comparison utilized the datasets in a similar way as in our work, i.e., the number and type of variables used, the use of the normal observations from the two logs for the training and validation while the attack log was used for testing, the use of the steady-state data for the training and validation phase, the consideration of the attack recovery period, etc. The performance evaluation results for

TABLE 7. Comparison between the different detection methods on the second log of the SWaT dataset.

Method	F1-score	Precision	Recall
NN [12]	0.812	0.976	0.696
SVM [15]	0.796	0.925	0.699
1D-CNN [14]	0.860	0.867	0.854
RNN [15]	0.802	0.982	0.678
TABOR [18]	0.823	0.862	0.788
KNN [17]	0.350	0.348	0.348
FB [17]	0.360	0.358	0.358
AE [17]	0.520	0.516	0.516
EGAN [17]	0.510	0.406	0.677
GAN [17]	0.810	0.700	0.954
DIF	0.882	0.935	0.835

TABLE 8. Comparison between the different detection methods on the second log of the WADI dataset.

Method	F1-score	Precision	Recall
PCA [17]	0.250	0.504	0.166
SVM [17]	0.510	0.512	0.512
KNN [17]	0.300	0.299	0.299
FP [17]	0.340	0.336	0.336
AE [17]	0.520	0.520	0.520
EGAN [17]	0.340	0.345	0.345
GAN [17]	0.620	0.538	0.749
DIF	0.656	0.765	0.574

TABLE 9. Recall values for the different detection approaches on the attack subset of the SWaT dataset.

Attack	NN	RNN	SVM	TABOR	1D-CNN	DIF
1	0	0	0	0.05	**0.99**	0.01
2	0	0	0	0.93	**1.00**	0.29
3	0	0	0	0	0.23	**1.00**
4	0	0	0.04	**0.33**	0	0
5	0.95	0.72	0.72	0.99	0.90	**1.00**
6	0.91	0	0.89	0	1.00	**1.00**
7	0.98	0.93	0.92	0.61	1.00	**1.00**
8	0.98	1	0.43	0.99	1.00	**1.00**
9	0.99	0.98	1	0.99	1.00	**1.00**
10	0	0	0	0	0	0
11	0	0	0	0	0	**0.06**
12	**0.60**	0	0	0.24	0.24	0.55
13	0	0	0	0.60	0.63	**0.64**
14	**0.97**	0.12	0.13	0	0	0.45
15	0	0.85	0.85	0.99	**1.00**	0.45
16	**0.98**	0	0.02	0.08	0.91	0
17	0.98	0.99	1	0.99	1	**1**
18	0.71	0.88	0.88	0	**1**	0.82
19	**0.92**	0	0	0	0.17	0.34
20	0.29	0	0.01	0	0.02	**1**
21	0.99	0	0	0.99	**1.00**	0.17
22	0	0	0	**0.20**	0.06	0
23	0.03	0.94	0.94	1.00	1.00	**1.00**
24	0.87	0	0	0	0	**1.00**
25	0.83	0	0	0.99	**1.00**	0
26	0.78	0	0	0	0.30	**1.00**
27	0.33	0	0.91	0	0.94	**1.00**
28	0.84	0	0	0.88	**0.89**	0.43
29	0	0	0	0.60	**0.99**	0
30	0	0	0	0.26	0	**0.95**
31	0.81	0	0.12	0.89	0.88	**0.93**
32	0.84	1	1	0.99	0.90	**1.00**
33	0.77	0.92	0.93	0.99	0.86	**1.00**
34	0.84	0.94	0	0.40	0.91	**1.00**
35	0.78	0.93	0.93	0.99	1.00	**1.00**
36	0	0	0.36	0	**0.64**	0.63

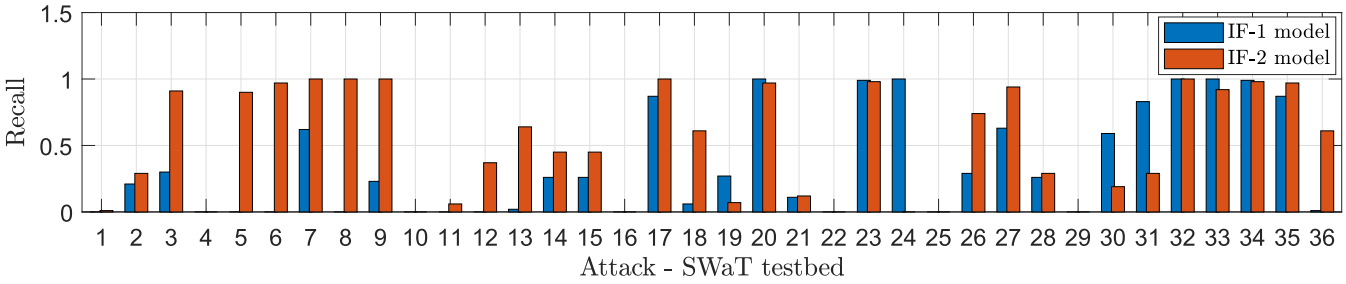

FIGURE 12. The recall values of the two isolation forest models on the SWaT attack log.

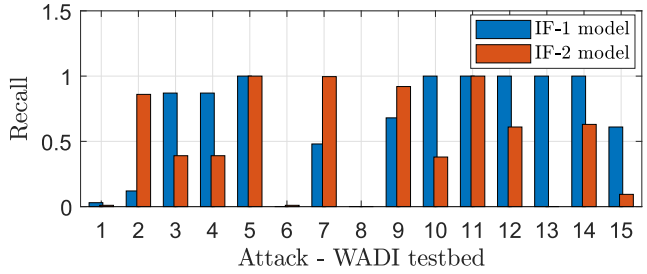

FIGURE 13. The recall values of the two isolation forest models on the WADI attack log.

the second log of the two datasets for the different approaches are summarized in TABLE 7 and TABLE 8, considering that the observation window used for the DIF-based detection method is $w = 120$ seconds that is evaluated every 30 seconds. The evaluation results of additional approaches using PCA, K-Nearest Neighbour (KNN), Feature Bagging (FB), Auto-Encoder (AE), Efficient GAN (EGAN), and SVM that were presented in [12], [17] for comparison are listed as well.

The DIF-based attack detection system achieves an improved F1-score of 88.2% and 65.6% on the SWaT and WADI datasets, respectively. For the SWaT dataset, it was found that the achieved improvement in the F1-score value is up to about 7% for the approach with comparable computational complexity over the the NN-based, SVM, and the TABOR-based approaches. However, it is as minimal as 2.2% for the 1D-CNN-based approach, which is far higher in the computational requirement. In addition, for the WADI dataset when comparing the proposed approach with the GAN-based approach, it was found that the improvement in the F1-score is about 4% while the precision is improved by 23%. However, the recall is less by 18%. There will be a trade-off in terms of the different aspects of the used algorithms, as demonstrated in the former analysis. Moreover, it is assumed that the previous works results, which are summarized in TABLE 7 and TABLE 8, represent the best performing models as per the authors of the original work.

The total number of detected WADI attacks was 12 out of 15, representing 80% of the attack scenarios, namely the undetected attacks were 1, 6, and 8, as demonstrated in FIGURE 13. The common factor between these attacks is that they were conducted by changing the states of at most two valves from OFF to ON, aiming to overflow tanks or interfere with the water distribution process. It seems that the impact of those attacks on the process is insufficient for the proposed approach to detect them.

In terms of the SWaT attack log, the dual isolation-forest-based detection framework was capable of detecting Attacks 3, 20, and 30, unlike the other approaches, as shown in TABLE 9. FIGURE 14 to FIGURE 19 demonstrate the attack indicators for the SWaT attack scenarios with the low recall values reported in TABLE 9, which were detected after a time delay noting that the start of the attack is at the beginning (time = 0 min). As demonstrated in FIGURE 14 and FIGURE 15, the detection delay for Attacks 14, 15, and 28 is less than 1 minute, while, FIGURE 16 and FIGURE 17 show that the time delay in detecting Attacks 12 and 18 is around 2 minutes. In addition, the detection delay for Attacks 13 and 36 is 4 minutes and 10 minutes, respectively, as shown in FIGURE 18 and FIGURE 19. These results indicate that the proposed DIF attack detection approach can eventually

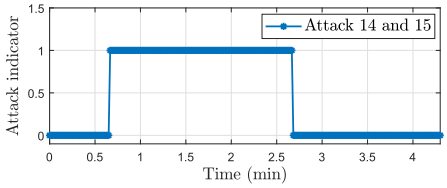

FIGURE 14. Detection of SWaT testbed Attacks 14 and 15 using the DIF-based approach.

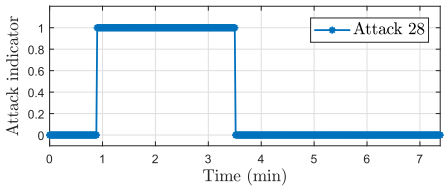

FIGURE 15. Detection of SWaT testbed Attack 28 using the DIF-based approach.

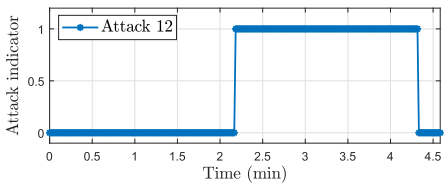

FIGURE 16. Detection of SWaT testbed Attack 12 using the DIF-based approach.

TABLE 10. Analysis of the performance of the proposed approach on some of the S^3 attack scenarios.

Event	S^3 Attack identifier	S^3 Attack description	SWaT attack identifier with similarity	Detected?
S^3 - 2016	1	Close MV-101 and stop P-101 and P-102	30	Yes
	5, 13, 18	Change reading of LIT-101	3, 28, 31	Yes
	14	Manipulating P-205 status	19	No
S^3 - 2017	1	Change reading of LIT-401	26	Yes
	4	Change reading of LIT-301	6	Yes
	10, 16, 21, 29	Change reading of LIT-101	3, 28, 31	Yes
	18	Manipulating DPIT-301 reading and MV-301 to MV-304 statuses	18	Yes
	20	Disrupt Pump P-501 operation	32	Yes

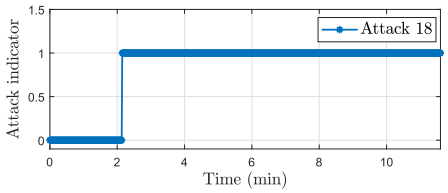

FIGURE 17. Detection of SWaT testbed Attack 18 using the DIF-based approach.

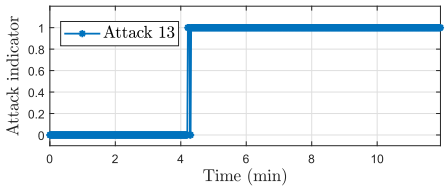

FIGURE 18. Detection of SWaT testbed Attack 13 using the DIF-based approach.

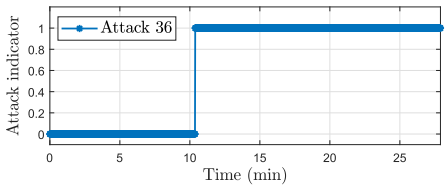

FIGURE 19. Detection of SWaT testbed Attack 36 using the DIF-based approach.

detect Attacks 12, 13, 14, 15, 18, 28, 36, and the low recall values reported in TABLE 9 are mainly due to the detection time delay.

It is worth noting that Attacks 16, 19, 21, and 25 were detected by the NN-based attack detection approach, and additionally, Attacks 1, 2, 25, and 29 were detected by the 1D-CNN-based detection method, but they were not detected by the proposed DIF-based framework. This can be explained by the fact that the scores of those attack (anomalous) points are comparable to others under normal operation, and hence, they cannot be detected without compromising the false alarm rate. In addition, the DIF-based detection method has a relativity higher false alarm rate, which is reflected in the precision metric that was found to be lower when compared with the other approaches. This is due to the fact that the system behavior under some attacks is similar to the ones during the normal operation, and hence, it may be falsely regarded as an attack incident given the threshold setting.

E. CASE STUDIES

This subsection presents two case studies regarding the performance of the proposed approach under the SWaT Security Showdown (S^3) attacks on SWaT testbed, and under adversarial attacks.

1) SWaT SECURITY SHOWDOWN (S^3)

This section presents a qualitative analysis of the performance of our proposed approach on the attack scenarios implemented in the SWaT Security Showdown event, which was held twice in 2016 and 2017 in which independent attack teams were invited to design and lunch real-time attacks on the SWaT testbed. There was a total of 49 different attacks injected targeting the Human Machine Interface (HMI), SCADA, PLCs, historian, sensors, and actuators, and the details of the attacks can be found in [32]. The aim of the event was to enable assessing the effectiveness of detection approaches, namely, the Water Defense (WD) approach, which is a model-based detection method. We were unable to use the S^3 dataset for testing our proposed approach since not all the system variables were recorded/available during the attack injection, but rather only the particular variables of interest for the used detection approach.

As presented in TABLE 10, we qualitatively analyzed the effectiveness of our approach on the S^3 dataset by studying the type of the injected attack in relation to the original attack log provided in the second log of the SWaT dataset. For example, S^3-2016 Attack 1 aimed to underflow a tank by closing valve M-V101 and stopping the pumps P-101 and P-102. Its effect on the process is similar to SWaT Attack 30 in which pumps P-101 and P-102 were both forced to stop to achieve the same attacker aim. In addition, the goal of S^3-2017 Attack 20 is to disrupt the operation of pump P-501, which matched SWaT Attack 32 description in which pump P-501 was forced to turn OFF, and the reading of FIT-502 was tampered with in an attempt to deviate the pump operation from the normal condition. It was found that 13 attacks were of the same characteristics as the SWaT attacks and were all detected expect 1 attack, which is S^3-2016 Attack 14.

2) ANALYSIS OF ADVERSARIAL ATTACKS

Adversarial attacks are crafted attacks by adversaries with the intent of leading the machine learning model to misclassify [37]. Concerns about those types of attacks have raised after the increased deployment of machine learning-based

approaches for cybersecurity applications. Isolation forests are less prone to such attacks because of their working principle, which is - as mentioned previously- is based on isolating anomalies. That is, the aggregated predictions of the different isolation trees in the forest while examining the system variables from several aspects - based on the isolation forests specifications, such as $n_{\max}$- promote the isolation forest to be resilient to these kinds of attacks. On the other hand, deep-learning-based approaches such as CNN, RNN, etc., are more prone to adversarial attacks as they aim to extract patterns or features from the input to make the prediction and a designed attack by adding perturbations to the original input can cause the network to misclassify the adversarial input.

V. CONCLUSION

A dual isolation forests-based attack detection system was developed using the system's process data for the Secure Water Treatment and the Water Distribution testbeds, which are down-scale versions of popular industrial control systems. The working principle of the proposed approach is identifying and separating away anomalies from the normal observations using the concept of isolation after analyzing the data in the original and the PCA-transformed representations.

The DIF-based attack detection framework was compared with other approaches in terms of precision, recall, and F1-score. For the SWaT testbed, it was found that the attack detection using the proposed approach was improved by up to 7% in terms of the F1-score value. In addition, a total of 19 SWaT attacks were detected with a minimum recall of 80%, 6 attacks were detected after a time delay of up to 40% of the attack duration, and 11 attacks were undetected. For the WADI testbed, 80% of the attacks were detected, and the performance of the proposed approach was improved in terms of the precision by 23% when compared to the GAN-based approach at the cost of the number of attacks that were detected, which was reflected on the recall value that was decreased by about 18%.

Future work would be as follows:

1) improving the performance of the detection approach by means of feature extraction,
2) and extending the proposed approach to a hybrid detection system using both process and network traffic data of the system to improve the detection capability of stealthy attacks.

REFERENCES

[1] J. M. E. Colbert and A. Kott, *Cyber-Security of SCADA and Other Industrial Control Systems*. Berlin, Germany: Springer, 2016.
[2] (Jan. 2017). *Ukraine Power Cut 'Was Cyber-Attack'*. Accessed: May 10, 2019. [Online]. Available: https://www.bbc.com/news/technology-38573074
[3] J. Summers and M. Walstrom. (Aug. 2018). *Cyberattack on Critical Infrastructure: Russia and the Ukrainian Power Grid Attacks*. Accessed: May 10, 2019. [Online]. Available: https://jsis.washington.edu/news/cyberattack-critical-infrastructure-russia-ukrainian-power-grid-attacks/
[4] P. Hafezi. (Nov. 2010). *Iran Admits Cyber Attack on Nuclear Plants*. Accessed: May 10, 2019. [Online]. Available: https://www.reuters.com/article/us-iran/iran-admits-cyber-attack-on-nuclear-plants-idUSTRE6AS4MU20101129
[5] L. A. Maglaras and J. Jiang, "Intrusion detection in SCADA systems using machine learning techniques," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 626–631.
[6] L. A. Maglaras and J. Jiang, "OCSVM model combined with K-means recursive clustering for intrusion detection in SCADA systems," in *Proc. 10th Int. Conf. Heterogeneous Netw. for Qual., Rel., Secur. Robustness*, Aug. 2014, pp. 133–134.
[7] P. Nader, P. Honeine, and P. Beauseroy, "l_p-norms in one-class classification for intrusion detection in SCADA systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2308–2317, Nov. 2014.
[8] W. Gao, T. Morris, B. Reaves, and D. Richey, "On SCADA control system command and response injection and intrusion detection," in *Proc. eCrime Res. Summit*, Oct. 2010, pp. 1–9.
[9] I. Kiss, P. Haller, and A. Beres, "Denial of service attack detection in case of tennessee eastman challenge process," *Procedia Technol.*, vol. 19, pp. 835–841, Oct. 2015.
[10] X. Clotet, J. Moyano, and G. León, "A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of critical infrastructures," *Int. J. Crit. Infrastruct. Protection*, vol. 23, pp. 11–20, Dec. 2018.
[11] W. Aoudi, M. Iturbe, and M. Almgren, "Truth will out: Departure-based process-level detection of stealthy attacks on control systems," in *ACM Conf. Comput. Commun. Secur.*, 2018, pp. 817–831.
[12] D. Shalyga, P. Filonov, and A. Lavrentyev, "Anomaly detection for water treatment system based on neural network with automatic architecture optimization," 2018, *arXiv:1807.07282*. [Online]. Available: http://arxiv.org/abs/1807.07282
[13] Y. Intrator, G. Katz, and A. Shabtai, "MDGAN: Boosting anomaly detection using multi-discriminator generative adversarial networks," 2018, *arXiv:1810.05221*. [Online]. Available: https://arxiv.org/abs/1810.05221
[14] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proc. Workshop Cyber-Physical Syst. Secur. PrivaCy (CPS-SPC)*, 2018, pp. 72–83.
[15] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 1058–1065.
[16] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, Jan. 2017, pp. 140–145.
[17] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds. Cham, Switzerland: Springer, 2019, pp. 703–716.
[18] Q. Lin, S. Adepu, S. Verwer, and A. Mathur, "TABOR: A graphical model-based approach for anomaly detection in industrial control systems," in *Proc. Asia Conf. Comput. Commun. Secur.*, New York, NY, USA, May 2018, pp. 525–536.
[19] Y. Chen, C. M. Poskitt, and J. Sun, "Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 648–660.
[20] H. R. Ghaeini and N. O. Tippenhauer, "HAMIDS: Hierarchical monitoring intrusion detection system for industrial control systems," in *Proc. 2nd ACM Workshop Cyber-Phys. Syst. Secur. Privacy*, New York, NY, USA, 2016, pp. 103–111.
[21] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, "Model-based security analysis of a water treatment system," in *Proc. 2nd Int. Workshop Softw. Eng. Smart Cyber-Phys. Syst. (SEsCPS)*, May 2016, pp. 22–28.
[22] S. Adepu and A. Mathur, "Distributed attack detection in a water treatment plant: Method and case study," *IEEE Trans. Dependable Secure Comput.*, to be published.
[23] K. Tidriri, N. Chatti, S. Verron, and T. Tiplica, "Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges," *Annu. Rev. Control*, vol. 42, pp. 63–81, Jan. 2016.
[24] Z. Ye, A. Gilman, Q. Peng, K. Levick, P. Cosman, and L. Milstein, "Comparison of neural network architectures for spectrum sensing," 2019, *arXiv:1907.07321*. [Online]. Available: http://arxiv.org/abs/1907.07321
[25] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, "Map-reduce for machine learning on multicore," in *Proc. 19th Int. Conf. Neural Inf.Process. Syst.* Cambridge, MA, USA: MIT Press, 2006, pp. 281–288.

[26] G. Gowrison, K. Ramar, K. Muneeswaran, and T. Revathi, "Minimal complexity attack classification intrusion detection system," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 921–927, Feb. 2013.
[27] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2765–2777, Oct. 2019.
[28] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect Web attacks using stacked auto-encoder," in *Proc. 6th Iranian Joint Congr. Fuzzy Intell. Syst. (CFIS)*, Feb. 2018, pp. 131–134.
[29] J. Goh, S. Adepu, K. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.*Oct. 2016, pp. 88–99.
[30] 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.
[31] 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.*, New York, NY, USA, 2017, pp. 25–28.
[32] S. Adepu and A. Mathur, "Assessing the effectiveness of attack detection at a hackfest on industrial control systems," *IEEE Trans. Sustain. Comput.*, to be published.
[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008, pp. 413–422.
[34] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 1, p. 39, Mar. 2012.
[35] K. M. Ting, *Confusion Matrix*. Boston, MA, USA: Springer, 2010, p. 209.
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[37] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.

MARIAM ELNOUR received the B.Sc. and M.Sc. degrees from Qatar University, Doha, Qatar, in 2015 and 2019, respectively. From September 2016 to January 2018, she was a Graduate Assistant at Qatar University, where she is currently a Research Assistant. Her research interests include machine learning, signal processing, and fault diagnosis.

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 at Qatar University, Doha, and an Adjunct Associate Professor at Concordia University. He has published more than 190 refereed journal and conference papers, and he is a coauthor (with K. Khorasani) of the book *Fault Detection and Isolation* (FDI): *Multi-Vehicle Unmanned Systems* (Springer, 2011). 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 second bachelor's degree from the University of Dhaka, and the Ph.D. degree in computing from Monash University, Australia. He has 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. He is currently an Associate Professor with the Department of Computer Science and Engineering, Qatar University. Prior to these, he served for Western Sydney University (Australia), as a Senior Lecturer, and was 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 cyber security, secure software engineering, cloud computing, measuring security, and trust in computer software. He has secured over U.S. $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 currently the Editor-in-Chief Emeritus of the 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 math/computer science from Harvard University, in 1978. He was one of the Co-Founders of Nayna Networks, Inc., San Jose, CA, USA—a next-generation telecommunications systems company. He was a Senior Consulting Engineer with Digital Equipment Corporation, Littleton, MA, USA, and then a Professor of computer and information sciences with The Ohio State University, Columbus, OH, USA. He is currently the Barbara J. and Jerome H. Cox, Jr. Professor of computer science and engineering at Washington University in St. Louis. He holds 14 patents and has written or edited 12 books, 16 book chapters, more than 65 journal and magazine papers, and more than 105 conference papers. He is a Fellow of ACM and AAAS. He received the ACM SIGCOMM Award, in 2017, the ACM SIGCOMM Test of Time Award, in 2006, the CDAC-ACCS Foundation Award, in 2009, and ranks among the top cited authors in computer science.