IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 6, 2020, accepted November 3, 2020, date of publication November 10, 2020,
date of current version November 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037093

An IIoT Based ICS to Improve Safety Through Fast and Accurate Hazard Detection and Differentiation

AZIN MORADBEIKIE[1], KAMAL JAMSHIDI[1], ALI BOHLOOLI[1], JORDI GARCIA[2], AND XAVI MASIP-BRUIN[2], (Member, IEEE)

[1]Faculty of Computer Engineering, University of Isfahan, Isfahan 81746-73441, Iran
[2]Advanced Network Architectures Lab (CRAAX), UPC BarcelonaTech, 08800 Vilanova i la Geltrú, Spain

Corresponding author: Kamal Jamshidi (jamshidi@eng.ui.ac.ir)

This work is supported by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund under Contract RTI2018-094532-B-I00 (MINECO/FEDER).

ABSTRACT Safety and security of Industrial Control Systems (ICS) applied in many critical infrastructures is essential. In these systems, hazards can be due either to system failure or cyber-attacks factors. Accurate hazard detection and reducing reconfiguration time after hazard is one of the most important objectives in these systems. One of the procedures that can reduce the reconfiguration time is determining the cause of hazards and, based on the aforementioned factors, adopting the best commands in reconfiguration time. However, it is difficult to differentiate between different types of hazard because their effects on the system can be similar. With the advent of IoT into ICS, known as IIoT, it has become possible to differentiate the hazards through the adoption of data from different IIoT sensors in the environment. In this article, we propose a risk management approach that identifies hazards based on the physical nature of these systems with the support from the IIoT. The identified hazards fall into four categories: stealthy attack, random attack, transient failure, and permanent failure. Then, the reconfiguration process is run based on the proposed differentiation, which provides a better performance and reconfiguration time. In the experimental section, a fluid storage system is simulated, showing 97% correct differentiation of hazards and reducing in 60% the lost time in the system recovery reconfiguration.

INDEX TERMS Industrial control systems (ICS), IIoT, fault and attack detection, resilient control, system reconfiguration.

I. INTRODUCTION

Industrial Control System (ICS) refers to the systems that are applied to improve control, instrumentation, monitoring, as well as increasing production in different industrial infrastructures [1]. Due to the widespread application of these systems in critical infrastructures, their security and safe operation are essential. In general, the architecture of the ICS is divided into three layers, named the field layer, the control layer, and the enterprise layer [2]. Each layer has different features and functionalities, which make them vulnerable to different types of attacks. Various incidents have proven that these systems are more susceptible to cyber-attacks and physical destruction in the field layer, and they are prone to cyber-attacks in the control layer, and the enterprise layer. The field layer is the most vulnerable region of the system due to receiving data and executing control commands where many nodes lack monitoring and control from the environment; thus they are more exposed to attacks and failures [3], [4]. Furthermore, the real-time constraints and limitations in communication and data processing capabilities make it impossible for many existing security control measures to be applied directly to this layer. Given these issues, it is necessary to manage the hazards of security threats and failures in accordance with the capabilities at this layer.

Risk management that manages hazards in the system consists of three major components, such as risk detection, risk analysis, and system reconfiguration. Based on the values received from the physical environment by the sensors, as the input ($y_i(t)$), the controller of ICS sends the appropriate

The associate editor coordinating the review of this manuscript and approving it for publication was Edith C.-H. Ngai.

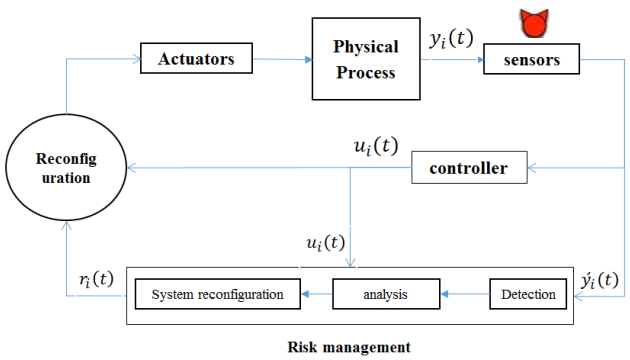


FIGURE 1. Risk management structure in ICS.

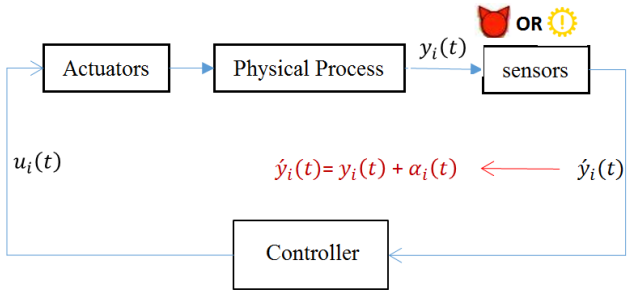


FIGURE 2. Functionality of the system in presence of a failure or an attack.

commands to the actuator ($u_i(t)$). The structure of risk management in ICS is framed in Fig 1 [5].

Hazard detection in the scope of ICS is usually performed through an estimator [6], predicting the next expected value by using different types of machine learning methods. The estimator compares the predicted values with the sensor new values and, if the difference between them is greater than some predefined threshold, then it is determined as a hazard in the system [3], [4]. As to risk analysis, the probability of risk incident and the damage level therefore is computed according to the risk propagation level in the system. Then, the risk damage level is compared with the tolerable threshold of risk in the system and, when necessary, some actions are taken in system reconfiguration to reduce the harmful effects of such risks.

Hazards detected in the system can be due to an attack or a failure. A hazard (either attack or failure) is always reflected in the form of change in the received information ($y_i(t)$). Unfortunately, in many different scenarios, the functionality of the system in presence of a failure or an attack is similar. So it is difficult to differentiate these two issues [6]. Fig 2 illustrates a hazard in the sensor that can be caused by an attack, or a failure, where an attacker or faulty sensor manipulates the real physical value.

As shown in Fig 2, data of a sensor under hazard (fault or attack) similarly seems like a modification in the original data of sensor ($\acute{y}_i(t) = \alpha_i(t) + y_i(t)$), where $\acute{y}_i(t)$ is the new value generated by the sensor under hazard for the sensor i at time t and $\alpha_i(t)$ represents the volume added to the sensor by the attacker or faulty sensor. Given this issue, the same operation of the sensor in presence of a failure or an attack makes differentiation difficult.

Different types of attacks and failures can have different occurrence probabilities and propagation levels [7]. For this reason, the analysis component needs to differentiate as soon as possible hazards after detection based on attack and failure. In addition, hazards differentiation is necessary to select the appropriate action in the reconfiguration component, because tacking appropriate measures in dealing with each of these two cases can be different. On this basis, the lack of proper identification of the cause of the hazards leads to wrong estimation of the likelihood of occurrence and the level of propagation in the system and, therefore, taking wrong control commands in the reconfiguration component. These wrong control commands can lead to a hazard recurrence in the system or, in many cases, transfer the system into a more critical situation. This phenomenon will increase the system reconfiguration time. Due to strict real-time limitations in ICS, reducing recovery time in critical infrastructures is essential to reduce serious system safety harms and, thus, becomes a vital measure. For this reason, fast and accurate hazard detection is important as it leads to correct risk management operations.

Some studies identified attacks like the random attack or stealthy attack [8]–[10], but the identified hazards are considered only as hazards, and no distinction is made between the identified hazards based on their causation. Authors in [5] attempt to use a Kalman filter in the reconfiguration component. However, there is no distinction between identified hazards, so one-way examination and treatment between hazards is run, which makes the system analysis and reconfiguration components inefficient. Authors in [11] deploy multiple sensors in the system which measure the same variable and detect the attack in the presence of transient failures. In their article, differentiation of a transient failure from attack is possible on certain sensors which are supported with other sensors to measure the same variable in the system. But their results have no appropriate performance for the entire ICS.

Differentiation of attack from failure without having additional data from the system sensors is not possible. However, with the advent of the IoT in the ICS as Industry Internet of Things (IIoT), a condition emerges to provide the required redundant data allowing the application of ad-hoc sensor data fusion and thus reducing the dependency on sensors. This makes it possible to identify the hazards in the system and differentiate them based on attack and failure.

By increasing the numbers of IIoT devices within a factory, several challenges arise. The volume of the generated data will be too large to be stored in a dataset for further processing, so the amount of data being generated by IIoT devices and collected could easily saturate network and storage infrastructures. Industry moves to assume fog computing as a platform to solve these problems.

In a previous work [12], in line with this article, we proposed a three-tier architecture using fog computing and the

IIoT for the differentiation of attack from failure. Different from our previous work, in this article, a risk management approach is proposed in correspondence with the nature of ICSs using fuzzy clustering, timed automata, and redundant data from the IIoT. Note that the IIoT can be used to gather broader and more accurate information about our surroundings and it can help to detect some hazards in the system (for example the stealthy attacks) which are not detectable by the traditional detection component. In addition, one of the interesting aspects of ICS that need to be considered in the detection component of risk management is the fact that changes measured by the system sensors follow physical laws in nature [6]. The physical law prevailing by the environment leads to uncertainty in predicted values, so in this article, the acceptable changes in the values of the observed property are not within a specific range; instead, the acceptable range can change and adapt dynamically during operation.

In summary, the main contributions of this article are the following: 1) we've been the first to take advantage of the IIoT sensor data to enhance the accuracy of hazards detection, 2) the adoption of IIoT data allows the differentiation of detected hazards based on their causation, 3) based on their causation, the identified hazards can be differentiated into four categories, named transient failure, permanent failure, random attack, and stealthy attack, 4) a dynamic acceptable range based on physical laws is used in the detection component for more accurate detection, 5) our system, composed from [12] and this article, reduces the network load by taking advantage of the fog computing paradigm, 6) the reconfiguration component can adopt the most appropriate strategy based on the hazards causation, requiring lower response and reconfiguration times and, 7) adaptation of fault-tolerant control techniques in the reconfiguration component of the system.

The remainder of this article is organized as follows. The literature review is presented in Section 2. Short overview of major security attacks in ICS, problem formulation of traditional systems, and the proposed hazard detection formulation are described in Section 3. In Sections 4 and 5, the two main components of the proposed method are described. In Section 6, the simulation and evaluation of the proposed method are presented and discussed. Finally, in Section 7, the conclusions of the paper are presented.

II. RELATED WORK

There exist a great number of articles on assessing the detection and even anticipating and isolating sensor failures [13]. Studies in this area are run in two different categories: a set of monitoring techniques where failure symptoms to identify failures are applied in the system [14] and applying monitoring methods to identify failures in system [15]. In each of these categories, different modelling methods are applied to identify and predict failures in the system. For this purpose, many studies seek to provide a fault-tolerant system. Attempts are made in [9] to provide a fault-tolerance method capable of providing tools for attack-resilience. For an efficient application of these methods, first, it is necessary to consider the different nature of these two categories; next, to provide a method for their correct differentiation for proper and effective treatment. The difference in nature of the attack and failure has made researchers to review these two issues. There exist many studies which are run on the two categories of these systems. The first category deals with cyber-attacks [16]–[19], and the second category examines field layer attacks [6]. Attacks on the field layer in different parts of the communication path among the sensor, actuator, and the controller are possible. In many articles, system modelling is run to detect attacks on the field layer due to the physical laws prevailing in nature. Such modelling measures the normal operation of the system, while the attacks to this layer are detected. Authors in [20], [21] explore the Stuxnet attack due to confused values received from the sensors. Various methods have been proposed in [3], [22]–[24] to detect different attacks on the system based on the values of sensors and control signals sent to the sensor. Authors in [25] proposed HAMIDS, a hierarchical monitoring SCADA intrusion detection system for industrial control systems. HAMIDS is able to detect anomalies that have a distributed impact on the cyber-physical process. By extending the HAMIDS framework with added support for log recording, processing, analysis, and anomaly detection, the authors in [26] proposed a state-aware anomaly detection based on CUSUM computation for ICS. As stated, Attacks on different parts of the field layer are possible. An attempt is made in [27] to remotely verify the integrity of the PLC based on the resulting sensor traces. For this purpose, authors present a system that combines remote software attestation with control process validation (PAtt). PAtt leverages operation permutations which do not affect the physical process but yield unique traces of sensor readings during execution. In addition to common attacks in these systems, like the random attack and denial of service, other attacks, known as stealthy attacks, can be more harmful to these systems. These attacks are not detectable because they are aware of the system and the limited changes in sensor variables [28], [29]. Different studies assist these attacks [30]–[33]. Authors in [34] discuss the practical implementation of stealthy attacks on industrial control systems. They offer Zero-Residual Attacks (ZeRA), which allows the attacker to launch stealthy attacks leveraging estimation of the stateful anomaly detector and matching of residuals as a fraction of actual estimation residual. Then, they propose to use a Stateful Detector (SD) to precisely detect such stealthy attacks. In addition to attack identification, an attempt has been made in [5] to use a Kalman filter to provide a resilience system where all identified hazards are considered as an attack; however, no differentiation between attack and failure has been made. It is necessary to differentiate hazards after detection based on attack and failure, so that the reconfiguration process can be run accordingly. An attempt is made in [11] to differentiate the identified hazards into two general categories, such as transient failures and attacks, by applying specific sensors in the system where the features specified

in the sensor are applied for this purpose. There, both the permanent failure and attack types are not specified. Their inaccurate detection of the causes of the hazards leads to incorrect system reconfiguration.

In this article, the objective is to identify the system's hazards according to the physical rules prevailing on the system environment and to categorize risks into the transient and permanent failures and the stealthy and random attacks by using IIoT data. Based on the recognition and accurate differentiation, the best process can be selected for immediate system reconfiguration.

III. PROBLEM FORMULATION AND PROPOSED HAZARD DETECTION FORMULATION

In this section, a short overview of major security attacks in ICS, problem formulation of traditional systems, and the proposed hazard detection formulation to improve traditional methods are presented.

A. SHORT OVERVIEW OF MAJOR SECURITY ATTACKS IN ICS

The field layer contains sensors, actuators, and other types of equipment, primarily responsible for the perception of physical world data and the execution of control commands from the controller. Safety and Security issues of the field layer involve physical security of the infrastructure of each node. Security measures for this layer are expected to ensure the safety of the sensors, actuators, and other types of equipment [35]. Security of the field layer is the basis of the security of ICS. The main types of security attacks to the field layer of ICS are Eavesdropping, Man-in-the-Middle Attack, and denial-of-service attack. Eavesdropping attacks can set the stage for Man-in-the-Middle attacks or denial-of-service attacks on the system.

The focus of this work is on so-called ''man-in-the-middle'' attacks on the sensor of the system. In the ''man-in-the-middle'', attacker can intercept communications between components in the system and manipulate or corrupt the values of measurements or Commands being sent and received. Random attack and stealthy attack can be considered as a subgroup of man-in-the-middle attack. There are studies that consider several attack scenarios where the adversary's goal is to drive the system to an unsafe state without triggering any alarm. These attacks are called stealthy attacks [29], [36]. In this work, stealthy attacks refer to an attacker who is able to corrupt sensor measurements in a manner that they exactly correspond to a valid physical state of the system.

B. PROBLEM FORMULATION IN TRADITIONAL SYSTEMS

According to [11], [37], for each sensor i in the system, a transient failure model (e_i, w_i) is determined. This model specifies that the sensor i has a maximum threshold transient failure of e_i at time window w_i in (1).

$$if\,(\alpha_i(t) > 0)\ then\ F(t) = 1$$
$$(\sum_{T_f - W_i}^{T_f} F(t)) < e_i \tag{1}$$

where T_f is the failure time and $\alpha_i(t)$ is the volume added to the correct value by the sensor i at time t. In order to enable the system to differentiate between the identified hazards and adopt the best practice in dealing with them in the reconfiguration component, based on (1), the false alarm rate in the system should be in accordance with the (2) to be acceptable in the system.

$$acceptable\ false\ alarm\ rate = (\frac{e_i}{w_i}) \tag{2}$$

In traditional hazard detection components of risk management, an estimator compares the predicted values with the new values of the sensor. If the difference between them is greater than a specified value, it is determined as a hazard in the system. In a threshold hazard detection, the next value is predicted based on the previous values according to (3).

$$dif\,(t+1) = a \times (y(t) - y(t-1)) + (1-a) \times dif\,(t)$$
$$\grave{y}_i(t+1) = y(t) + dif\,(t+1) \tag{3}$$

where $dif\,(t+1)$ is the predicted change of the next value with respect to the previous values, $\grave{y}_i(t+1)$ is the predicted next value, and a is the impact factor of variables that determine the confidence level of each variable. To determine the confidence level of each variable, the system operation is simulated several times and the best confidence level for each variable is determined.

The integration of anomaly detection systems with Machine Learning (ML) methods allow obtaining greater performance and accuracy compared to traditional techniques.

In [38], the authors compare and evaluate different Machine Learning (ML) algorithms for anomaly detection. Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) have the best accuracy, but false alarm rate of these systems is not acceptable in ICS. ICS is in direct interaction with the environment. So, hazard detection in ICS should be done based on physical laws in nature.

C. PROPOSED HAZARD DETECTION FORMULATION

As mentioned, ICS is related to the physical environment and based on the values received from the environment by the sensors as the input $(y_i(t))$, ICS sends the appropriate commands to different controllers $(u_i(t))$. Accordingly, the control system can be considered as a state machine with different discrete states based on the commands sent to the operators. The properties of the physical environment (temperature, volume, pressure, etc.) change continually according to the issued control commands and, therefore, the control system has different discrete states determined by $u(t)$. In each one of these states, the value of properties reported from the physical environment by sensors continuously changes following a specific trend. This trend is in accordance with the physical laws prevailing by the environment. The continuous changes in sensor values correlate with the previously observed values leads to a minimum and maximum stability time in each state of the system. A transition occurrence in state stability time indicates a normal transition. The lack of transition in state's

stability time indicates an anomaly in the system. Different states of the automata are determined based on different states of the controllers. In each of these states, the reported value by the sensor is within a specific range. Transitions between different system states are possible as follows. A time slot is assigned to each state, which indicates when a transition in the system can occur, or how long the system can stay in a state. Switching between states is possible in any of the following two cases: 1) when the value of the input $y(t)$ reaches the defined threshold specified and the time is within a stable time interval. Thus the change of controller values indicate a transition to another state and, 2) when the state is not sustainable, and the reported value by the sensor does not reach the specified threshold. To display this state, the implemented automata has an additional state, meaning an instability risk in the system. For example, a liquid tank volume control system has two main states. In the first state, the input valve of the tank is open ($y_1(t) = 1$) and the evacuation hole is closed ($u_2(t) = 0$). As a result, in this case, the input values from the sensor indicate an increase in the volume of the liquid ($y_1(t) < y_1(t+1)$). In the second state, the input valve of the tank is closed ($u_1(t) = 0$) and the evacuation hole is open ($u_2(t) = 1$). In this case, the input values from the sensor indicate a decrease in the liquid volume ($y_1(t) > y_1(t+1)$). In this example, based on $u(t)$, the system has two states, wherein each one the value of $y(t)$ changes continuously in a specific range following a specific trend.

In those systems, the process and possibility of displacement between states are determined, indicative of the fact that any unexpected displacement is a hazard in the system. Matrix $A = (a_{ji})_{n\times n}$, which is filled with 0 and 1, indicates the possibility or absence of displacement among different states.

The timed automaton is determined by the tuple $< N, L_0, E >$, where N, L_0 and E constitute the set of states, initial states and edges, respectively [39]. The displacement among states occurs when the values of the sensors are greater or less than the specified threshold for that state or the time exceeds the stability time of the state. The timed automata in this article is confirmed through (4).

$$E \subseteq N \times B(C) \times N$$
$$B(C): a \leq x \leq b \quad for\ a, b \in N \tag{4}$$

where $B(c)$ indicates the displacement condition and x is the value of the sensor. As stated, hazards are determined by the volume of mismatching between predicted and observed values. This volume varies in different systems and different periods of time. Such different variations exist in continuous changes in sensor values and are due to physical laws. The physical law prevailing in the environment leads that, at any given time, changes in the value of an observed property are not equal to a certain value, but rather within a specific range. The predicted value for the next entry must be within a specified range and this specific range can change in different periods of time. For example, if a predicted value for the next time period is equal to X in one period, the next value can be within the range $[X - a, X + a]$ and, furthermore, in the next period a can change. The value of a is determined by the physical environment of the system. Note that this change within a specific range does not mean that there is a risk in the system, but it is a result of the prevailing natural laws. Dynamic acceptable range determination will improve the performance of the hazard detection component over the time. According to this, in this article, fuzzy clustering is applied to distinguish the normal changes from hazards in ICSs. Fuzzy clustering is a type of clustering where each sample can belong to all clusters with different membership degrees. In the process of clustering, the changes in relation to the previous data are considered as input data. Each cluster has a center and radius that represents the range of normal variations for the property. Due to the continuity of the sequential data, the membership degree of sequential data changes in different clusters with specific variants.

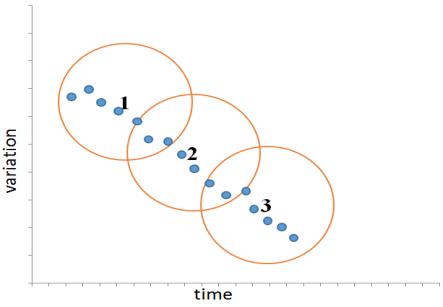


FIGURE 3. Fuzzy clustering.

Based on physical laws in these systems, a sudden change in the value of a feature is not possible.

As an example, a fuzzy clustering with three clusters and time sequential data is shown in Fig 3. In this example, data have three different variations (a) and, in sequential data, acceptable variations can be determined by the membership degree of sequential data. Consequently, in addition to specifying the range of acceptable variations for the expected values, the ability to check continuity in sequential changes is evident.

By adopting this method, it is possible to estimate the acceptable range for the next data in the system based on the sensor data from the beginning up to now. Equation (5) is applied for fuzzy clustering [40].

$$m_{ik} \in [0, 1], \quad \forall k: \sum_{k=1}^{K} m_{ik} = 1 \tag{5}$$

where m_{ik} is the membership of $y_i(t)$ in cluster k, and K is the number of clusters. Here, the timed automaton and fuzzy clustering are applied together to identify hazards in ICS sensors in a precise manner.

IV. SUGGESTED ARCHITECTURE FOR CONTROL SYSTEM

An ICS is divided into two main sections, the field and the control layers. The field layer consists of sensors and operators, and the control layer acts as the decision-maker

TABLE 1. Variable of ICS features.

Variable	Describe
$t \in T$	The specific moment of the system operation
$y_p(t)$	The measured value of sensor p at instant t where $p \in 1, \cdots, P$
$u_m(t)$	The exported command to controller$m$ at instant t where $m \in 1, \cdots, M$
$x(t)$	The state of the system at time t where the system has N state $1, \cdots, N$
$A = (a_{ij})_{n \times n}$	The possibility (1) or impossibility (0) of transition from states i to j
$B = (b_{ijk})_{p \times n \times 2}$	Indicates the minimum and maximum expected input values of the sensors in different states.
c_i	The cluster center i in the system
$K = \{c_1, c2, \cdots, c_k\}$	The K is cluster centers and k is the number of clusters

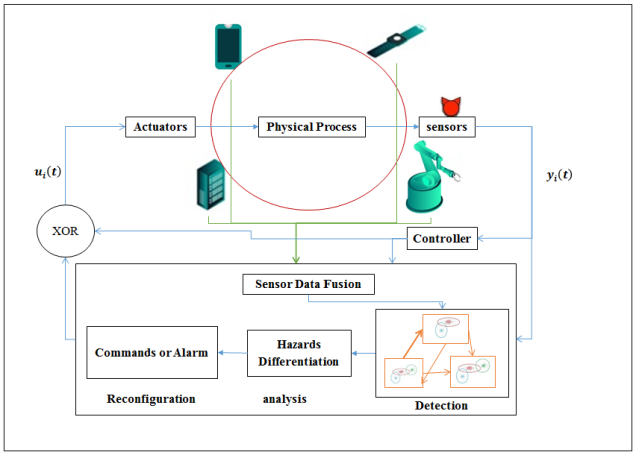


FIGURE 4. Functionality of the system.

of the system. The sensors in the field layer first receive information from the environment ($y_i(t)$) and send it to the control layer. Then, the control layer processes the received information from the environment and obtains an estimate of the system state ($x_i(t)$). Based on the estimated system condition, the corresponding commands are sent to the operators by the control layer ($u_i(t)$). Actuators in the field layer affect the environment based on those commands. The variables of ICS features are expressed in Table 1.

The overall architecture of this proposed ICS is framed in Fig 4. After implementing the timed automata in this system, the data variations are applied as the input data in the clustering. In each time automata state, the value of the property read from the input changes follow a particular trend, and these changes are placed in different clusters. According to the state in which the system is located and the previous values, each change is expected to be in the clusters with a specific percentage of membership, which makes it possible to predict the new value and the range of permissible variables based on clustering.

The functionality of this architecture is described in three stages: First, the system predicts the next state according to (6).

$$x(t+1) = Ax(t) + Bu(t) \tag{6}$$

where B is a $m \times n$ matrix and B_{ij} indicates the activity or inactivity of operator j in state i. Second, based on the performed clustering, through (7) the expected value of $y(t+1)$ is calculated. In (7), the membership percentage of the next value in each cluster ($p_{(t+1)k}$) is predicted based on the previous values.

$$p_{(t+1)k} = a \times p_{tk} + (1-a) \times m_{tk} \tag{7}$$

where p_{tk} represents the predicted percentage of membership for the value of sensor at time t in the cluster k. The symbol a is the impact rate of both the previous and the new values.

Third, based on this prediction and (7), the input value of the sensor for the next time unit is predicted through (8). This predicted value is applied in the system reconfiguration component.

$$y(t+1) = y(t) + (\sum_{k=1}^{k} p_{(t+1)k} \times c_k) + e \tag{8}$$

where e is the error rate. By receiving a new value of the sensor as $\grave{y}(t+1)$ at time $t+1$, the membership percentage of $\grave{y}(t+1)$ is computed in different clusters and the volume of non-subscription membership with the expected value in different clusters is computed as $r(k)$. Based on the computed value, $S(k)$ is calculated through (9):

$$S(k+1) = S(k) + r(k) \tag{9}$$

After a specified time period, if the value of $S(k)$ is greater than the specified value, it is determined as a hazard in the system.

In ICSs, some data injection attacks operate based on information of the attacker gained from the system in a manner that they are not detectable by the system hazard detection. These attacks with T_a duration modify the original data over time and collect the sensor's data after adding some arbitrary value, as shown in (10).

$$y_i^a(t) = \alpha_i(t) + y_i(t) \tag{10}$$

where $y_i^a(t)$ is the new value generated by the attacker for the sensor i at time t and $\alpha_i(t)$ represents the volume added to the sensor by the attacker. In this case, the attack is not detectable by $S(k)$. To detect these types of attacks and to evaluate and distinguish among different types of hazards in the system, some data from objects are integrated. In the ICS environment, there exist different objects consisting of devices inside the factory or of the worker's belongings. Due to the introduction of the IoT paradigm to the industry, known as IIoT, and the objects equipment with different sensors, many objects in the environment can measure different features and send them to the control center. IIoT is the future of industry and its purpose is to provide useful data about our surroundings. Each factor measured by a sensor has a

sensible range. When IIoT sensors are inside of a sensible range of factor, they can estimate the value of the factor based on their distance from intended environment. In industry 4.0 we can use the set of IIoT sensors to gather more accurate information about factor. IIoT sensors gather information about factors when they inter to the factor sensible range. For this purpose, the existing IIoT objects in the environment will receive information from the environment and send them to the center for sensor data fusion. The center integrates information from different objects through (11).

$$x_i = \frac{\sum_{r=1}^{R} b_r + n_i}{R} \tag{11}$$

Information from each sensor is considered with error rate n_i. The R is the count of objects that sent information to the center from ad-hoc sensor for data fusion. R can vary, due to the ad-hoc sensors.

Through (11), the obtained x_i is applied to measure differences between the data obtained from the sensor of the system. If this difference is greater than the specified amount, it is identified as a hazard. That is, this hazard is recognized by this part of the system, otherwise, this component recognizes the system as being safe. Because the data from the environment ad-hoc sensors probably have an uncertain error rate, this part of recognition should be integrated with the control system recognition. To integrate these two parts of recognition, two situations are considered: 1) when the frequency of mismatch between two parts is lower than the acceptable false alarm rate in the system where the weighted integration is applied, with this weighing determined based on each one of the two parts confidence level during the runtime, and 2) when the above frequency is higher than the acceptable false alarm rate of the system, indicating that the system is determined as being hazardous. To determine the confidence level of each part, the system operation is simulated many times and the best confidence level for each section is determined.

One important issue influences the efficiency of this newly proposed method.

Q.1: According to the acceptable range of variation in the system, this question constitutes of two parts: a) How is determined the minimum count of the system clusters? b) How different would the predicted membership percentage in the clusters be if the existing and the previous input value difference is higher than that of the predetermined range?

A.1.a: Based on $u(t)$ the system has different states, where the value of the sensors changes over a certain interval following a certain trend, that represents the measured property of the physical environment. For example, consider a temperature sensor, where its controller can increase, decrease, or remain constant, as shown in Fig 5.

In each of these states the process of variation is different, subjected to a certain range displayed as follows:

$$\min_{ij} \leq y_i(t) \leq \max_{ij} \quad \text{st. } i \in 1, \cdots, p \text{ and } j \in 1, \cdots, n$$

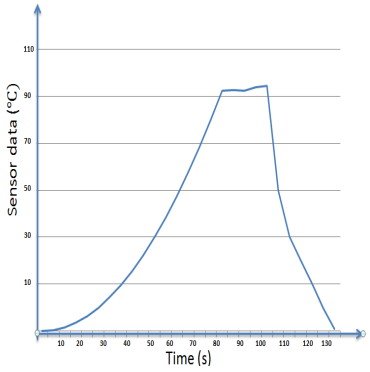


FIGURE 5. Values of the sensor.

Each state in a system has a stability time, where the time that the system can remain in each state is equal to the normal time of the state, so that $\max_{ij} - \min_{ij}$ occurs at a certain time. If the acceptable range for variation of two sequential values is expressed as Δl, the count of acceptable clusters for each state is equal to:

$$k = (\max_{ij} - \min_{ij}/t)/\Delta l$$

A.1.b: The total number of clusters needed in the system is equal to the sum of the different system state clusters. After clustering, by entering a new value from a sensor for hazard identification, it is necessary to determine what is the difference between the membership percentages of the input and the predicted membership percentage in different clusters. If this difference is greater than the defined threshold, it should be declared as a hazard in the system.

The membership percentage of input data from the sensor i in the cluster j is computed through the (12) [40].

$$m_{ij} = (d_{ij}^{\frac{-2}{m-1}})/(\sum_{l=1}^{K} d_{il}^{\frac{-2}{f-1}}) \tag{12}$$

where d_{ij} is the distance from i to the center of the cluster j, and f is the fuzzifier factor. If the predicted value for the input from the sensor i at time t is equal to a and the received value is equal to $a+1$, the membership percentages of both values in the clusters are computed through the following (13)

$$m_{ij} = ((c_j - a)^{\frac{-2}{m-1}})/(\sum_{l=1}^{K}(c_l - a)^{\frac{-2}{f-1}})$$

$$m_{ij} = ((c_j - (a+l))^{\frac{-2}{m-1}})/(\sum_{l=1}^{K}(c_l - (a+l))^{\frac{-2}{f-1}}) \tag{13}$$

To compute the minimum subscription at membership percentage of two sequential values, the sum of $m_{ij} - \dot{m}_{ij}$ for clusters related to this state must be computed by computing this value, the minimum subscription at percentage membership of two sequential values can be computed.

V. EVALUATION OF IDENTIFIED HAZARDS AND SYSTEM RECONFIGURATION

The failures considered in this article are of two types, transient and permanent, which are identified as the two common attacks of random attack and stealthy attack in ICSs. In each of these hazards, a certain value $\alpha_i(t)$ is added to the actual value, so the new computed value obtained through (13) is sent as the result of the sensor measurement. Here a brief explanation of each of these hazards is presented, followed by the process of hazard analysis and system reconfiguration.

`Random attack`: It is assumed that in an attack, the attacker can intercept communication between the sensor and the controller and manipulates the sensor's values. In random attack, the attacker has little information on the structure of the system, so at a particular moment, it receives the reported value of the sensor, adds significant value to it, and sends the new value to the control center.

`Stealthy attack`: In this attack, the attacker's purpose is to drive the system to an unsafe state without triggering any alarm. A stealthy attack manipulates the sensor's values in a manner that they correspond to a valid safe physical state of the system. The attacker has acceptable information from the system, so the attacker attempts to inject data into the measured data obtained by sensors over time in an indistinguishable manner for a detection component.

`Transient failure`: A transient fault is the one that causes a component to malfunction for a limited period of time. In transient failures, the sensor changes the correct value and generates a new value in a random manner.

`Permanent failure`: It means that the component works permanently wrong. In permanent failure, the sensor changes the correct values and sends wrong values in a continuous manner.

A. ANALYSIS OF IDENTIFIED HAZARDS

Based on the number of identified hazards in the system at a specific time interval w_i, the hazards are divided into two categories, transient and permanent, where each one is a product of attack or failure. Transient failures are placed in the transient hazards category of the system, whereas the stealthy attack, random attack, and permanent failure are placed in the permanent hazards category.

1) TRANSIENT HAZARDS

In this category, the count of identified hazards in the specified time interval w_i is less than of e_i, leading to a transient hazard which can only be the output of a transient failure.

$$\left(\sum_{T_f-w_i}^{T_f} F(t)\right) < e_i$$

2) PERMANENT HAZARDS

Permanent hazards are classified into two subcategories, attack and failure, where both the computed correlation values are greater than the established threshold value in the system. The differentiation process of attack and failure is expressed as:

Random attack: In this case, both conditions $(\sum_{T_f-w_i}^{T_f} F(t)) > e_i$ and $S(K) > \tau$ are confirmed.

Permanent failure: In this case, both conditions $(\sum_{T_f-w_i}^{T_f} F(t)) > e_i$ and $S(K) > \tau$ are confirmed, while transient failures are recorded at previous intervals as $(\sum_{T_f-2w_i}^{T_f-w_i} F(t))$.

Stealthy attack: In this case, both conditions $(\sum_{T_f-w_i}^{T_f} F(t)) > e_i$ and $S(K) > \tau$ are established. The correlation value is applied to compute these attacks, where the correlation computed rate for the specified time interval is greater than the specified volume.

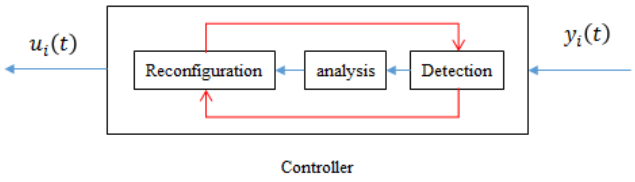


FIGURE 6. Actuator hazard.

B. SYSTEM RECONFIGURATION

There exist four common manners of system reconfiguration, consisting of non-consideration of the hazard in the system, replacement of the sensor, system reboot, or performing a suitable patch in the system. Adopting the most appropriate manner for the identified hazard would lead to a better and faster system reconfiguration. Attack and failure differentiation is highly contributive in system reconfiguration. This procedure specifies the need to correctly identify the hazard causation. In system reconfiguration, for each scenario, a specific procedure tabulated in Table 2 should be of concern.

Each adopted manner has a normal time to perform. After manner adoption, the reconfiguration component sends feedback to the hazard detection about the required normal time of adopted manner to performed. If after the procedure normal time, the hazard existence is being still diagnosed by the detection component, then the detection component sends a ''continues hazard'' command to the reconfiguration component. It can be due to a hazard (fault/attack) in the actuator that is caused by a manually user mistake, or an adversary user. In this case, the system reconfiguration component causes an alert. An actuator hazard is shown in Fig 6.

With respect to the concepts discussed here, the correct determination of hazard causation in the system would lead to an appropriate reconfiguration manner selection for the system and, thus, a better and faster system reconfiguration.

VI. SIMULATION AND SYSTEM ASSESSMENT

The system and simplified data presented in [41] are used to assess this proposed method. The data set is the result of an ICS implementation to a liquid tank volume control system. The volume of the liquid inside the subjected tank is adjusted within 2000 and 8000 liters range. When the

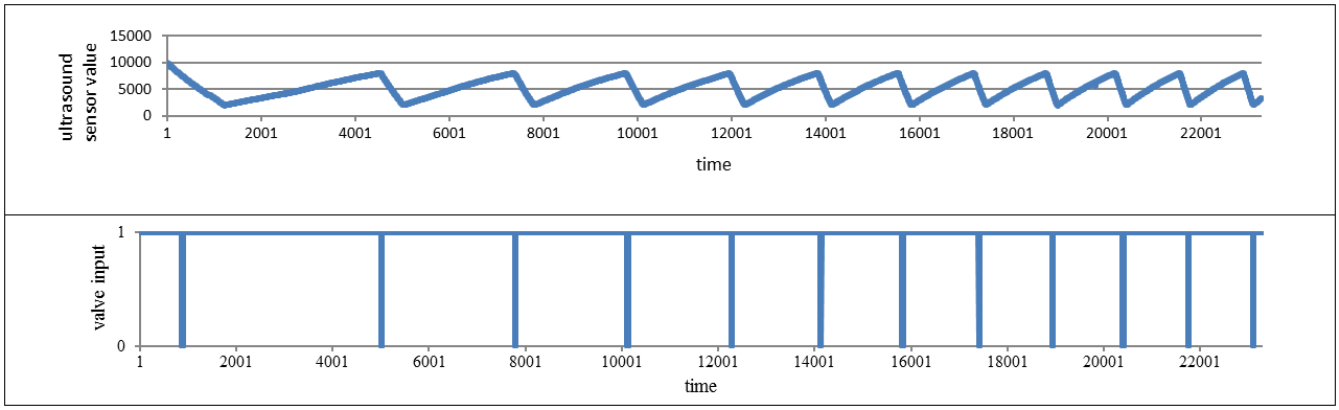


FIGURE 7. Sensor data and control commands in system normal operation.

TABLE 2. System reconfiguration procedure.

	Hazard types	Hazard causation	Reconfiguration procedure
Identified hazards	Transient	Transient failure	Regardless of the value observed by the sensor, it is sufficient to replace the sensor value by $y(t+1)$ predicted through fuzzy clustering.
	Permanent	Permanent failure	Same as above, but also register an alert in the system indicative of the need to repair or replace the sensor in the system.
		Random attack	The value $y(t+1)$ predicted by fuzzy clustering is replaced with the sensor observed value and an alert is registered in the system. To fix the drawback, patch or reboot manners must be implemented.
		Stealthy attack	Same as above, but also patch manners must be implemented to fix the drawback.

volume of the tank reaches its lowest range, a pump at the end of the tank will be activated. An evacuation hole is located at the bottom of the tank so that when the volume of the liquid in the tank exceeds the highest range, the pump is turned off and the access is discharged through the evacuation hole. An ultrasound sensor is applied to control and keep this volume within the determined range. In this system, the measured volume through this sensor is considered as the input and the command issued to the pump, and the evacuation hole are considered as the outputs. Each of these components is connected to a computer through a PLC controller. The liquid volume measured by the sensor is considered as $y_i(t)$ and the command issued by the system to the pump and the evacuation hole are considered as $u_i(t)$. The collected data reveals the normal system operation to last 7000 seconds, is shown in Fig 7.

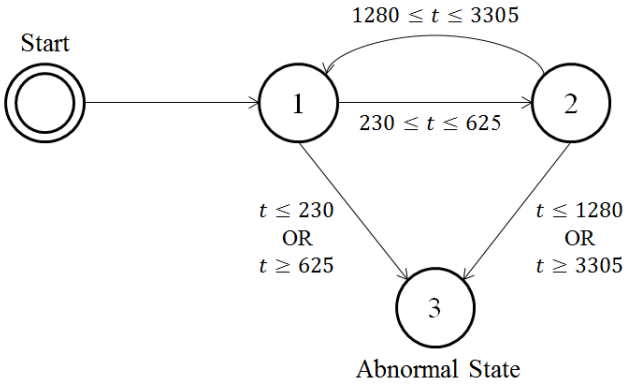


FIGURE 8. Proposed timed automata model for the system.

In this study, the timed automata model has two normal states, where in the first the liquid volume follows an ascending trend while in the second the same is descending. Each state has a specified time factor indicating its stability. The change of state is allowed only within the determined time interval and the specified liquid threshold level; any other mode change is considered as an abnormal change, as can be seen in Fig 8. An implementation of this proposed model runs in a Matlab simulator environment. In this system, the acceptable range of changes in determining the values is set within 0-10.

According to our computation, the volume changes of system data over time is assessed using five clusters. Each sensor value belongs to each cluster with a variable percentage based on its deviation from the previous value. To generate data from ad-hoc sensors, a random number is generated within 1 and 10, which represents the count of objects in the environment for data transmission (R). To inject a random error into each sensor's data, first, 10 figures with a normal distribution with zero mean and a certain Standard Deviation (SD) are generated; next, the Rs of these figures are added to the real value of the subjected attribute and, finally, these computed Rs are considered as the value of the ad-hoc sensors. For example, The Ad-Hoc sensor data and sensor data fusion for

TABLE 3. Example of Ad-Hoc sensor data and sensor data fusion.

SD	Ad-Hoc sensor data					data fusion
6	3215	3206	3198	3211	3194	3205
	3206	3209	3204	3210	3201	3206
	3194	3192	3186	3187	3193	3190
	3211	3207	3208	3214	3215	3211
	3195	3197	3203	3197	3195	3197
5	3194	3210	3212	3218	3214	3210
	3193	3215	3204	3200	3204	3203
	3199	3199	3207	3207	3193	3204
	3204	3194	3195	3196	3185	3195
	3199	3199	3202	3202	3196	3200
4	3197	3200	3195	3196	3200	3198
	3196	3200	3202	3204	3206	3199
	3212	3203	3206	3196	3198	3205
	3205	3193	3198	3199	3202	3199
	3201	3204	3204	3197	3201	3201

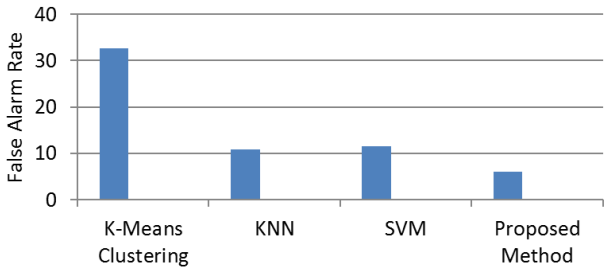


FIGURE 9. Results of false alarm rate on different systems.

a real sensor value of 3200 with different SDs are represented in Table 3.

As it is clear in the table, Ad-hoc sensor data can have a high error rate. But, in most cases, sensor data fusion can lead to an acceptable error rate. For error rate decreasing of sensor data fusion, a confidence level is considered for integration with system sensor data. As it is mentioned, to determine the confidence level of each part, the system operation is simulated many times, and the best confidence level for each section is determined.

To evaluate this proposed method, three different systems are considered where each system uses a different ML method (SVM, KNN, and K-Means Clustering) for hazard detection. Data classification in SVM is achieved by realizing a linear or non-linear separation surface in the input space that separates all data points of one class from the other class. Closest data points to the linear or non-linear separation surface are the support vectors. K-Nearest Neighbour (KNN) algorithm is a non-parametric algorithm [38] used for both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. K-means clustering is a partitioning method. The function k-means partitions data into k mutually exclusive clusters and returns the index of the cluster to which it assigns each observation. To perform the evaluation, we split data of normal operation by 70% (16000 points) for training and 30% (7000 points) for the detection test. We exploit a supervised learning method for training. This means that the models are trained with normal elements. In order to implement the SVM, KNN, and k-means algorithms, we use the Matlab simulator. The KNN and k-means are simulated by different k (4, 5, and 6) and the best k is considered. Then, false alarm rate of these systems is computed as follow:

TABLE 4. False alarm rate.

Detector type	K-means	KNN	SVM	Our method without IIoT	(μ, σ^2) =(0,6)	(μ, σ^2) =(0,5)	(μ, σ^2) =(0,4)
false alarm rate (%)	32	10	11	6	5	4	3

$$False\ Alarm\ Rate = \frac{FP}{FP + TP} \tag{14}$$

where TP = True Positives, and FP = False Positives. In our proposed system, false alarm rate in the absence of data from theintegration of the ad-hoc sensors is computed. Based on the computed rate, in the next step, the data subjected to two different modes of sensor presence are examined. In each one of these two modes, the count of sensors is considered within a range of 1 and 10 in a random manner with a SD of 5 and 6, respectively. The results of simulation on different systems are shown in Table 4. and Fig 9. The results of the simulation of our proposed system are shown in Fig 10, Fig 11, and Fig 12. The implementation result of the system in the 23400 units of time is shown in Fig 10, where the false alarm rate in the 0-2000 interval is higher than the rest of the time intervals. As observed in Fig 7, the value $((\max_{ij} - \min_{ij})/t)$ in different intervals varies. Due to the fact that the count of clusters is constant in the entire system process, it leads to different error rates in different intervals. As a result, the difference in $((\max_{ij} - \min_{ij})/t)$ at 0-2000 interval leads to different false alarm rates in comparison with other intervals. The false alarm count generated by this system without presence of ad-hoc sensors is 1700. This represents 6% false alarm rate in the system. In order to detect it correctly and differentiating attack and failure from the analysis results, if the sensors have up to 5 error count in the time window equal to 100 units of time, it is necessary to reduce the false alarm rate to 5%. The data obtained from ad-hoc sensors with a *SD* of 6 is shown in the upper part of Fig 11. The integration result of the ad-hoc sensors with the result of the control system is shown in the lower part of Fig 11. In this case, the false alarm rate reaches to 5%.

The *SD* of 5 is shown in Fig 12, where the final data integrated false alarm rate of ad-hoc sensors with control system is about 4%.

For the analysis of Table 4, false alarm rate of SVM is equal to 11% while the false alarm rate of our proposed method without IIoT is equal to 6%. This is because of consideration of the fact that changes measured by the system sensors follow physical laws in nature. The decrease in false alarm rate leads to correct hazard differentiation. In addition, in this

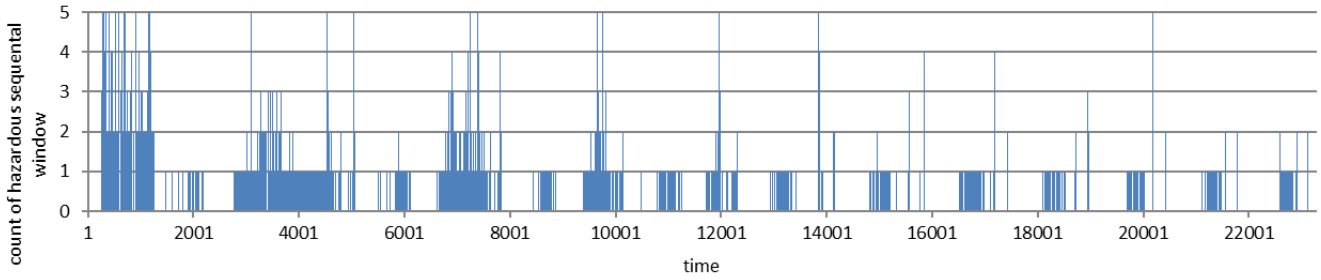


FIGURE 10. Detected hazards during system normal operation.

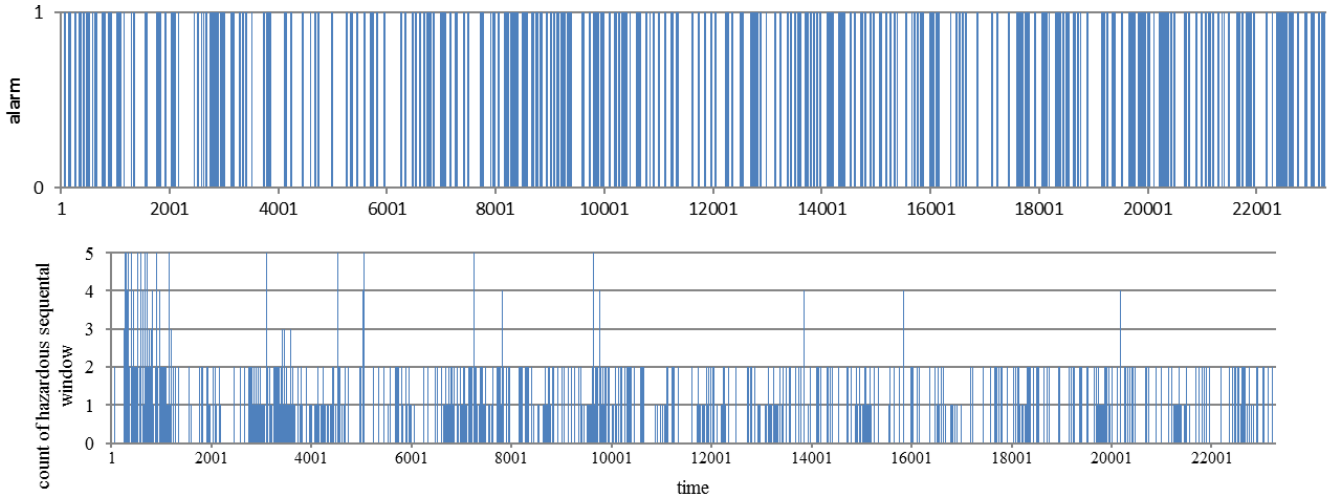


FIGURE 11. Detected hazards in presence of ad-hoc sensors with $(\mu, \sigma^2) = (0, 6)$.

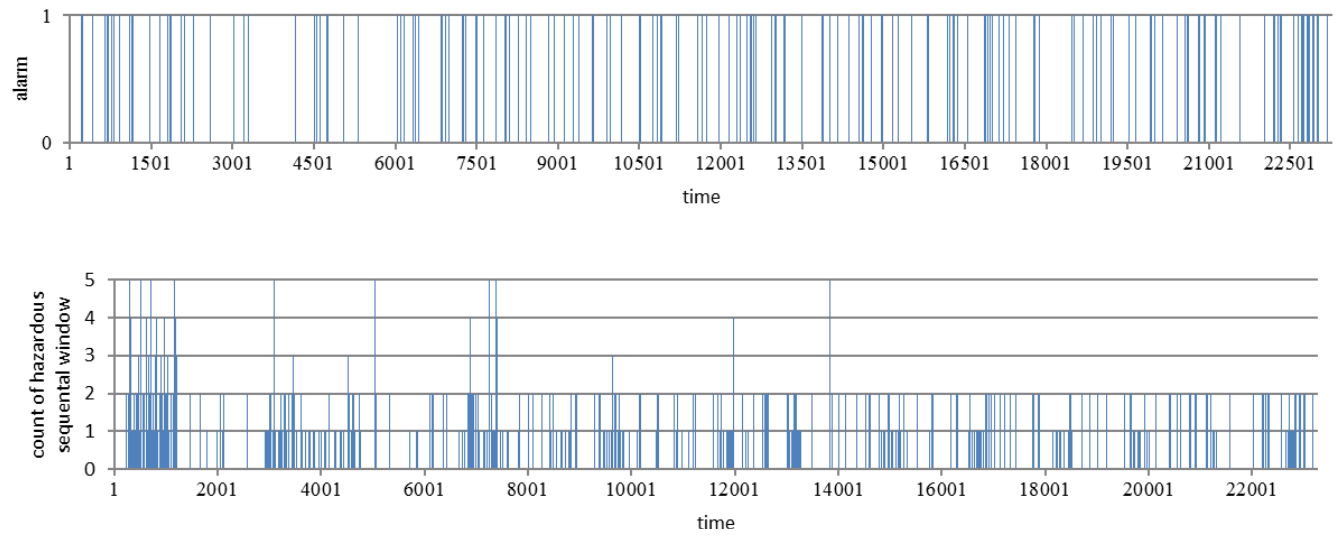


FIGURE 12. Detected hazards in presence of ad-hoc sensors with $(\mu, \sigma^2) = (0, 5)$.

proposed method, the false alarm rate in the absence and presence of the IoT are specified in Table 4, respectively, wherein the proposed system, even of the presence of relatively high sensors error rate improve the system overall performance. In this context, due to the fact that these sensors are in a random and unpredictable manner embedded in the environment, and transmits the necessary information, security and safety of the system is improved. Because the attacker will

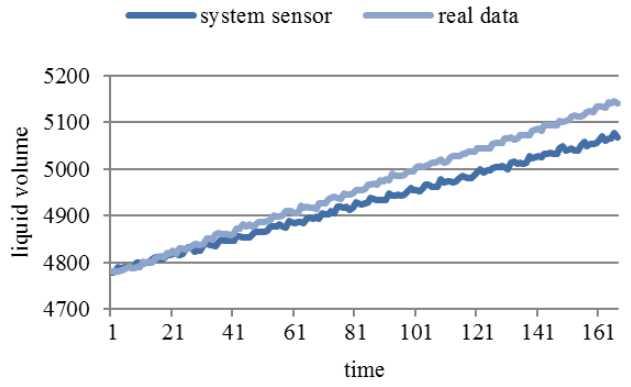


FIGURE 13. Stealthy attack data with small differencer.

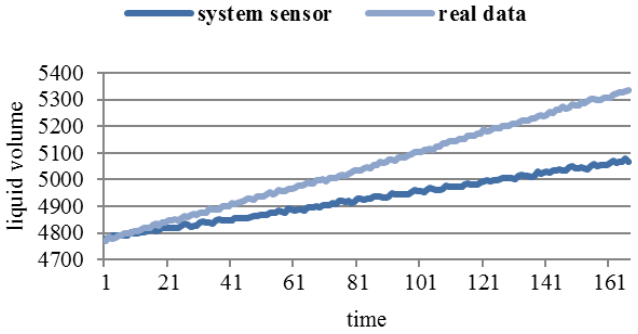


FIGURE 14. Stealthy attack data with large difference.

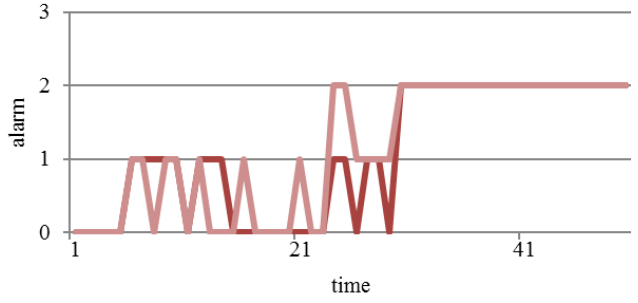


FIGURE 15. Stealthy attack detection with small difference on time.

encounter unpredictable sensors present in the environment and send information, this unpredictability would improve the system performance when facing attacks, thus an increase in the safety and security of the system.

To evaluate this proposed method subject to random attacks, normal system operation (shown in Fig 7) subjected random error are given as the inputs to the system. To perform the evaluation, we split the dataset in 70% for training (16000 points) and 30% for the detection test (7000 points). In the test dataset, 35% of data is randomly modified as random attack. For the evaluation of the ML-based algorithms for anomaly detection, we use the following metrics:

Accuracy: represents the fraction of correct predictions of the model under consideration. In the binary classification case, the accuracy is defined in terms of positives and

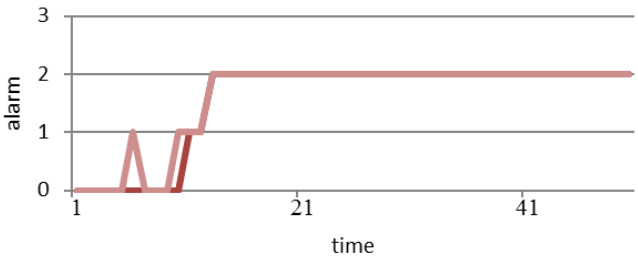


FIGURE 16. Stealthy attack detection with large difference on time.

TABLE 5. Result of ML-based random attack detection algorithms.

Detector type	KNN	SVM	Proposed Method	Proposed Method with IIoT
FPR	0.44	0.44	0.58	0.43
FNR	0.16	0.16	0.02	0.01
Precision	0.51	0.51	0.49	0.57
Recall	0.83	0.83	0.98	0.98
F-measure	0.63	0.63	0.65	0.72
Accuracy	0.65	0.65	0.61	0.72

negatives as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

where TN = True Negatives, and FN = False Negatives.

F-measure: is a metric used to evaluate a classification by taking in consideration both precision and recall as follows:

$$F - measure = 2.(\frac{Precision.recall}{precision + recall}) \tag{16}$$

where:

$$Precision = \frac{TP}{TP + FP} recall = \frac{TP}{TP + FN} \tag{17}$$

False Positive Rate (FPR): wrong identification of hazards in the system. FPR is defined in terms of positives and negatives as follows:

$$FPR = \frac{FP}{FP + TN} \tag{18}$$

False Negative Rate (FNR): is the rate of not retrieved relevant instances in overall relevant instances. FPR is defined in terms of positives and negatives as follows:

$$FNR = \frac{FN}{FN + TP} \tag{19}$$

The simulation results of different ML-based algorithms and our proposed method for the datasets under consideration are shown in Table 5. The KNN algorithm is simulated by different k (4, 5, and 6) and the best k is considered. As it is clear in the table, our proposed method without IIoT has a better FNR but it has higher FPRs. As ICS is used in many critical systems, FNR is more important than FPRs. In addition, our proposed method with IIoT has a better FNR and FPRs.

In stealthy attacks, the sensor's data of the system is manipulated by attacker. So, Stealthy attacks cannot detect by ML

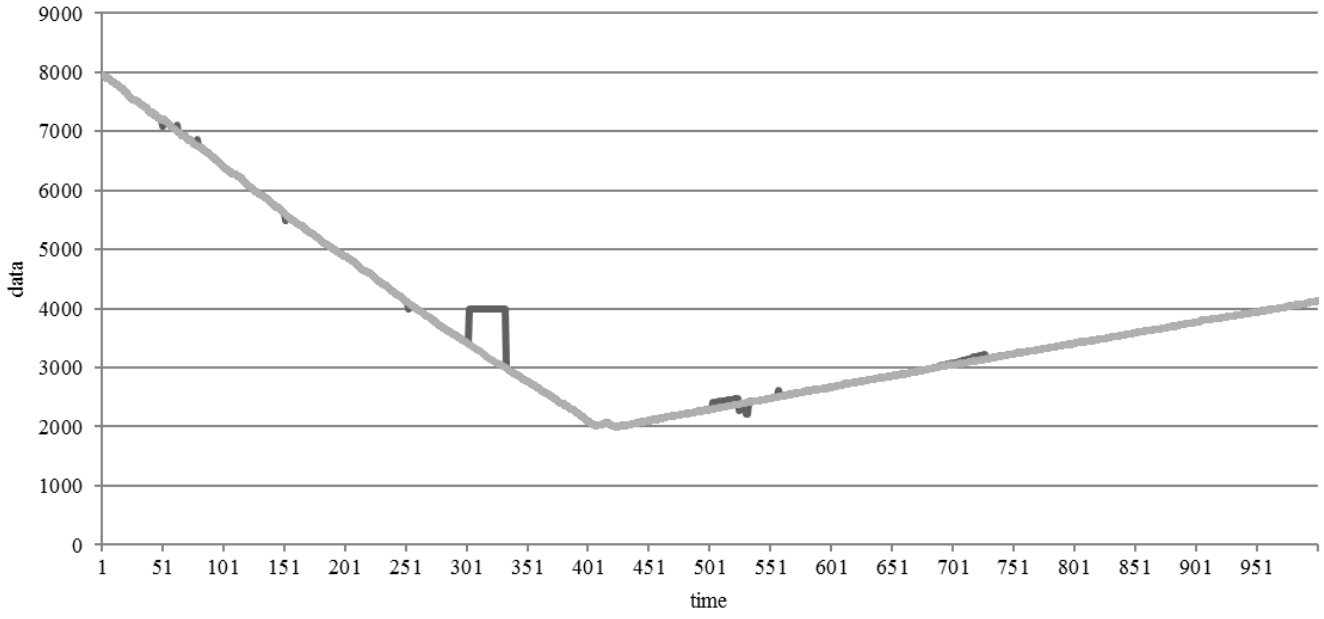


FIGURE 17. Inject of attack and failure data to system normal operation.

methods or our proposed method without IIoT. Because these methods are depending on system sensor data. Our proposed method with IIoT uses redundant data obtained from ad-hoc sensor data for hazard detection.

To evaluate this proposed method subject to stealthy attacks, two different data categories are given as the inputs to the system, where the points marked in dark blue indicate inputs from the system sensor and the points marked in light blue indicate the actual value of the system. Each one of these data varies subject to a different gradient from the original data, as shown in Fig 13 and Fig 14. In each of these cases, the system does not provoke any alarm, thus, the attack can effectively lead to serious damage to the system after a certain time. To identify attacks in the system, the presence of two different modes of IoT is examined and simulated, as shown in Fig 15 and Fig 16.

The result of this proposed method for the data in Fig 13 is shown in Fig 15. The dark red line indicates the IOT present results with a *SD* of 6 and a light red line indicates the IoT present results with a *SD* of 5. The result of this proposed method with two *SD* 5 and 6 on the data in Fig 14 is shown in Fig 16. As observed, this method is able to detect this attack, and the presence of the IoT with different *SD* leads to a slight difference in attack detection time.

To evaluate the hazard analysis of the system, the data obtained from the normal system operation is shown in Fig 17 in gray, whereas the data from normal operation in different sections is changed by injecting certain data into the main data, as can be seen in black in the diagram. These injections occur in four different moments during the 1-200 interval, which is an example of a transient failure in the system sensor; next, it occurs during the 300-330 interval, where the change subject to the random attack, and in the 500-530 and

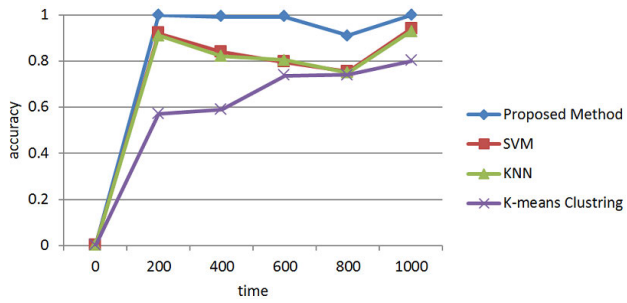


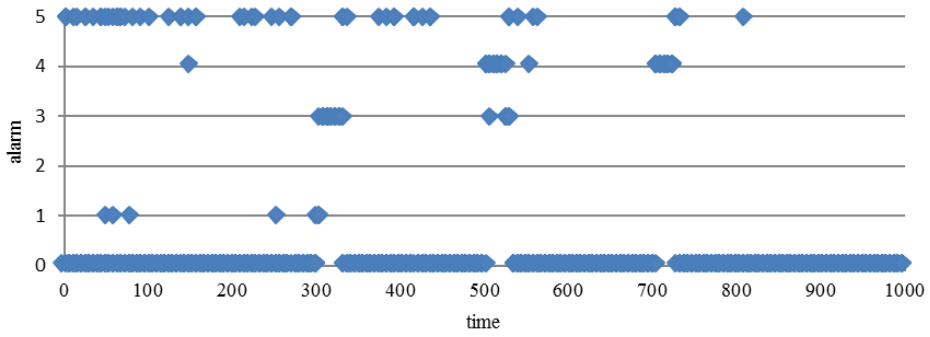
FIGURE 18. Graphical representation of the Accuracy Comparison.

680-730 intervals subject to stealthy attack data change. The hazardous data rate in the system is 10%.

For assessment of our proposed method, three different ML methods (SVM, KNN, and K-means clustering) are simulated. The dataset of normal operation presented in Fig 7 is used for the training step. Then, for the detection test, ML methods are applied in dataset of Fig 17. In order to implement the SVM, KNN, and K-means clustering algorithms, we use Matlab simulation. The number of neighbors of KNN and the number of clusters in K-means clustering, in both cases, is considered equal to 5. In our proposed method, the count of transient failure of the sensor is 5 in the window equal to 100 time units. The SD of the data from the ad-hoc sensors in the environment is considered as 5.

The graphical representation of the accuracy comparison for all the ML algorithms is framed in Fig 18. In Table 6, we present the results obtained for the ML-based anomaly detection algorithms for the datasets under consideration.

In the control system without ad-hoc sensors data fusion, the true positive, false positive, false negative, and true negative in the system are 87, 2, 17, and 893, respectively.



Normal data	0
Transient failure	1
Permanent failure	2
Random attack	3
Stealthy attack	4
False alarm	5

FIGURE 19. Analysis results of detected hazards in the system.

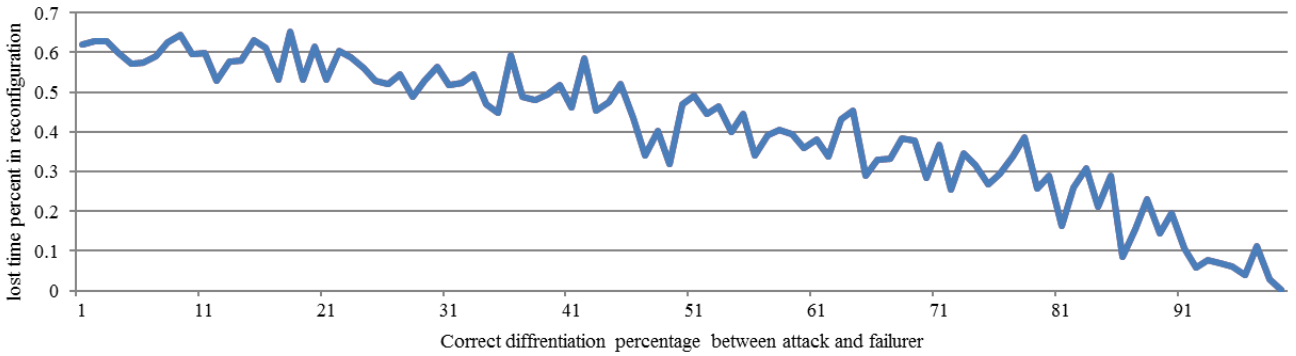


FIGURE 20. Time impact of hazard differentiation.

TABLE 6. Result of ML-based anomaly detection algorithms.

Detector type	K-Means	KNN	SVM	Proposed Method
FPR	0.287	0.061	0.054	0.002
Precision	0.116	0.175	0.169	0.977
Recall	0.365	0.125	0.105	0.836
F-measure	0.176	0.145	0.129	0.901
Accuracy	0.75	0.847	0.852	0.98

TABLE 7. Effective factors on Hazards.

Anomaly	Action	Impact factor	Time
Transient failure	Nothing	1.0	0
Permanent failure	Replace	1.0	100
Random attack	Reboot	0.15	15
	Patching	1.0	50
Stealthy attack	Reboot	0.15	15
	Patching	1.0	50

As shown in Fig 18, the accuracy of ML methods decreases when the system is under random attack and stealthy attack. When the system is under random attack, ML methods do not detect it because the value of changes in the sensor (it is equal to zero) is considered as an acceptable value. As the proposed method considers the physical laws in nature that should be dominant on changes measured by the system sensors, this attack can be detected. In addition, ML methods do not detect stealthy attack either. These attacks can be detected in our system by using IIoT.

The result of system simulation for hazard detection and causation after ad-hoc sensors data fusion is shown in Fig 19, where data are classified into five different categories based on hazard causation. For this purpose, at first, our proposed hazard detection methods applied to the data that leads to normal and false alarm separation from dangerous data. Then, the proposed method for analysis of identified hazards are applied on dangerous data that leads to hazard differentiation based on their cause. According to the Figure, the error rate in the hazards differentiation based on the cause is equal to 3%. As observed, this proposed method detects hazards in the system correctly by reducing false alarm rate to the acceptable rate in the system. By analyzing the detected hazards in accordance with this proposed algorithm, the causation of the hazards in the system is determined. Correct hazards detection and analysis leads to better system performance in the reconfiguration component.

The factors affecting proper differentiation effect of the attack from the failure are tabulated in Table 7, where each hazard is determined through three factors. The first factor determines appropriate actions to reduce damage level of the hazards in the system; the second factor determines the probability percentage of taken actions to resolve the problem. And the last factor specifies the amount of time necessary for the action.

The lost time oscillator due to the deficiency in proper differentiation of the attack from the failure of 100 different samples of the anomaly indicated in the system is shown

in Fig 20, where the deficiency may lead to wrong action that would increase the reconfiguration time in the system; otherwise, leads to a significant reduction in the system's reconfiguration time, thus, a very important tact in ICSs.

As stated, our proposed method is based on using IIoT data to reduce false alarm rate and stealthy attack detection. For doing this, IIoT data and system sensor data must be merged. A confidence level is considered for IIoT and system sensors in fusion. As a result, if IIoT targeted by attackers sends adversary data, IIoT can lead incorrect assessment of system conditions. Incorrect assessment of system conditions can lead to wrong control commands. These wrong control commands can transfer the system into a more critical situation. To accomplish this attack, attacker should access to the majority of IIoT for a specific period of time.

In traditional systems, the attacker only needs access to system sensors. In our proposed method, the attacker will encounter unpredictable sensors present in the environment and send information; this unpredictability would improve the system performance when facing attacks, thus an increase in the safety and security of the system.

As limitations, our proposed method needs an acceptable number of registered IIoT sensors in the factory. These IIoT sensors move in the factory for data collecting. For example, smartphones or smartwatches of the employees.

VII. CONCLUSION AND FUTURE WORKS

A new method is proposed in this article for differentiation of identified hazards based on attacks and failures in ICS. To accomplish this objective, a hazard identification method based on the use of IIoT corresponding to the physical nature of these systems is provided, identifying the hazards in accordance with the system's features in an accurate manner, followed by an accurate categorization of the identified hazards into four: transient failures, permanent failure, random attack, and stealthy attack, and takes the best action in the system reconfiguration component. Unlike the available methods, the hazards here are identified in accordance with the system's characteristics and ad-hoc sensors data integration applying to detect the stealthy attack. This method due to applying ad-hoc sensors has increased its resistance against attacks, as well as reducing the reconfiguration time by adopting the appropriate action because of the enhanced hazard cause identification. The simulation results indicate that by applying the ad-hoc sensors data, even if there is a high error rate, our method leads to a reasonable differentiation between failure and attack. This finding greatly reduces system reconfiguration time. In the future, attempts should be made to distinguish other categories of attacks and failures in the sensor, actuator, control system, and other types of equipment, as well as their corresponding proposals for reconfiguration actions. In addition, we will work to determine a relation between the counts of mobile ad-hoc sensors with trusted mobile ad-hoc sensors. This relation will be useful in trustiness and coverage specification to improve the QoS of the system.

REFERENCES

[1] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017, doi: 10.1109/JIOT.2017.2703172.
[2] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security: Analysis, challenges and solutions," *Comput. Secur.*, vol. 68, pp. 81–97, Jul. 2017, doi: 10.1016/j.cose.2017.04.005.
[3] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. 16th ACM Conf. Trans. Inf. Syst. Secur. (TISSEC)*, New York, NY, USA, 2009, pp. 21–32, doi: 10.1145/1653662.1653666.
[4] R. Tan, V. B. Krishna, D. K. Y. Yau, and Z. Kalbarczyk, "Impact of integrity attacks on real-time pricing in smart grids," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2016, pp. 439–450, doi: 10.1145/2508859.2516705.
[5] K. Paridari, N. O'Mahony, A. El-Din Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proc. IEEE*, vol. 106, no. 1, pp. 113–128, Jan. 2018, doi: 10.1109/JPROC.2017.2725482.
[6] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018, doi: 10.1145/3203245.
[7] H. Abdo, M. Kaouk, J. M. Flaus, and F. Masse, "A safety/security risk analysis approach of industrial control systems: A cyber bowtie–combining new version of attack tree with bowtie analysis," *Comput. Secur.*, vol. 72, pp. 175–195, Jan. 2018, doi: 10.1016/j.cose.2017.09.004.
[8] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Berlin, Germany: Springer-Verlag, 2006, doi: 10.1007/978-3-540-35653-0.
[9] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part II: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015, doi: 10.1109/TIE.2015.2419013.
[10] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015, doi: 10.1109/TIE.2015.2417501.
[11] J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee, "Sensor attack detection in the presence of transient faults," in *Proc. ACM/IEEE 6th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, New York, NY, USA, 2015, pp. 1–10, doi: 10.1145/2735960.2735984.
[12] A. Moradbeikie, K. Jamshidi, A. Bohlooli, J. Garcia, and X. Masip-Bruin, "A fog based approach for hazards differentiation in an IIoT scenario," in *Proc. 5th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Shanghai, China, May 2020, pp. 458–462.
[13] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surveys*, vol. 42, no. 3, pp. 1–42, Mar. 2010, doi: 10.1145/1670679.1670680.
[14] R. Baldoni, L. Montanari, and M. Rizzuto, "On-line failure prediction in safety-critical systems," *Future Gener. Comput. Syst.*, vol. 45, pp. 123–132, Apr. 2015.
[15] F. Salfner, "Event-based failure prediction: An extended hidden Markov model approach," Ph.D. dissertation, Dept. Math. Natural Sci. Fac. II, Humboldt Univ., Berlin, Germany, 2008.
[16] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, vol. 4, Aug. 2011, pp. 447–462.
[17] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for SCADA systems," *Comput. Secur.*, vol. 56, pp. 1–27, Feb. 2016, doi: 10.1016/j.cose.2015.09.009.
[18] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Comput. Secur.*, vol. 70, pp. 436–454, Sep. 2017.
[19] N. R. Rodofile, K. Radke, and E. Foo, "Extending the cyber-attack landscape for SCADA-based critical infrastructure," *Int. J. Crit. Infrastruct. Protection*, vol. 25, pp. 14–35, Jun. 2019, doi: 10.1016/j.ijcip.2019.01.002.
[20] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Secur. Privacy Mag.*, vol. 9, no. 3, pp. 49–51, May 2011, doi: 10.1109/MSP.2011.67.

[21] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White Paper, Symantec Corp., Secur. Response*, vol. 5, no. 6, p. 29, 2011.

[22] 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)*, New York, NY, USA, 2011, pp. 355–366, doi: 10.1145/1966913.1966959.

[23] S. McLaughlin, "CPS: Stateful policy enforcement for control system device usage," in *Proc. 29th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, New York, NY, USA, 2013, pp. 109–118, doi: 10.1145/2523649.2523673.

[24] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava, "PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2015, pp. 1004–1015, doi: 10.1145/2810103.2813679.

[25] 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 (CPS-SPC)*, 2016, pp. 103–111.

[26] H. R. Ghaeini, D. Antonioli, F. Brasser, A.-R. Sadeghi, and N. O. Tippenhauer, "State-aware anomaly detection for industrial control systems," in *Proc. 33rd Annu. ACM Symp. Appl. Comput. (SAC)*, 2018, pp. 1620–1628.

[27] H. R. Ghaeini, M. Chan, R. Bahmani, F. Brasser, L. Garcia, J. Zhou, A. R. Sadeghi, N. O. Tippenhauer, and S. Zonouz, "PAtt: Physics-based attestation of control systems," in *Proc. 22nd Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*, 2019, pp. 165–180.

[28] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 220–225.

[29] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *Proc. 1st Workshop Secure Control Syst. (SCS)*, Stockholm, Sweden, 2010, pp. 1–6.

[30] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 4–13, Mar. 2017, doi: 10.1109/TCNS.2016.2570003.

[31] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sep. 2016, doi: 10.1109/TAC.2015.2498708.

[32] C. Murguia, N. V. D. Wouw, and J. Ruths, "Reachable sets of hidden CPS sensor attacks: Analysis and synthesis tools," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 2088–2094, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.528.

[33] F. Pasqualetti, F. Dorfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013, doi: 10.1109/TAC.2013.2266831.

[34] H. R. Ghaeini, N. O. Tippenhauer, and J. Zhou, "Zero residual attacks on industrial control systems and stateful countermeasures," in *Proc. 14th Int. Conf. Availability, Rel. Secur. (ARES)*, 2019, pp. 1–10.

[35] Y. Gao, Y. Peng, F. Xie, W. Zhao, D. Wang, X. Han, T. Lu, and Z. Li, "Analysis of security threats and vulnerability for cyber-physical systems," in *Proc. 3rd Int. Conf. Comput. Sci. Netw. Technol.*, Oct. 2013, pp. 50–55.

[36] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 220–225.

[37] I. Koren and C. M. Krishna, *Fault-Tolerant Systems*. Amsterdam, The Netherlands: Elsevier, 2010.

[38] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *Proc. Int. Conf. Cyber Secur. Protection Digit. Services (Cyber Secur.)*, Jun. 2020, pp. 1–8.

[39] J. Bengtsson and W. Yi, "Timed automata: Semantics, algorithms and tools," in *Advanced Course on Petri Nets*. Berlin, Germany: Springer, 2003, pp. 87–124.

[40] D. Oliveira, J. Valente, and W. Pedrycz, *Advances in Fuzzy Clustering and Its Applications*. Hoboken, NJ, USA: Wiley, 2007.

[41] P. M. Laso, D. Brosset, and J. Puentes, "Dataset of anomalies and malicious acts in a cyber-physical subsystem," *Data Brief*, vol. 14, pp. 186–191, Oct. 2017, doi: 10.1016/j.dib.2017.07.038.

AZIN MORADBEIKIE was born in Zahedan, Iran. She received the B.Sc. degree in computer engineering from the University of Sistan and Baluchestan, in 2012, and the M.Sc. degree in computer engineering from the International University of Imam Reza, Iran, in 2015. She is currently pursuing the Ph.D. degree with the University of Isfahan, Iran. Her research interest includes industrial control systems and security.

KAMAL JAMSHIDI was born in Isfahan, Iran. He received the M.Sc. degree in electrical engineering from Anna University, India, in 1990, and the Ph.D. degree in electrical engineering from IIT University, India, in 2003. He is currently an Associate Professor with the Faculty of Computer Engineering, University of Isfahan, Iran. His research interests include wireless sensor networks and fuzzy systems, and cyber physical systems.

ALI BOHLOOLI received the B.Sc. and M.Sc. degrees (Hons.) in computer engineering from the Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran, in 2001 and 2003, respectively, and the Ph.D. degree in computer engineering from the University of Isfahan, Iran, in 2011. He is currently an Assistant Professor with the Faculty of Computer Engineering, University of Isfahan. His research interests include computer networks and cyber physical systems.

JORDI GARCIA received the M.Sc. and Ph.D. degrees in computer science from UPC BarcelonaTech, Spain. He was the Vice-Dean of the Barcelona School of Informatics, from 2001 to 2010, the Academic Director of the Center of Cooperation for Development, from 2010 to 2013, and the Rector's Delegate in Shanghai, China, from 2015 to 2016. He is currently an Associate Professor with the Department of Computer Architecture, UPC BarcelonaTech. His current research interests include cloud and edge computing resources management, data management strategies for smart scenarios, and optimization of big data processing.

XAVI MASIP-BRUIN (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Universitat Politècnica de Catalunya (UPC). He is currently a Professor with the Computer Science Department, UPC, where he is also the Director of the CRAAX Laboratory. In 2011 and 2012, he was a Visitor Professor with UPEC, Paris. His current research interests include cloud and fog computing, network management, cybersecurity, the IoT, and particularly on analyzing the benefits brought by combining fog and cloud paradigms. He has been involved in many different research initiatives at national and international level. He has been recognized with the 2016 IBM Faculty Award.