Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system

Elodie Chanthery, Audine Subias

Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system

Elodie Chanthery and **Audine Subias**
LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
e-mail: [elodie.chanthery;audine.subias]@laas.fr

Abstract

Recently, a two-tank benchmark for detection and isolation of cyber-attacks has been proposed to the diagnosis research community to test different diagnosis methods. In this work, we use this benchmark, add some scenarios, and test several diagnosis techniques to evaluate their diagnosability power. We propose to use a well-known model-based diagnosis approach to identify the groups of faults and attacks that are not isolable or detectable. In a second part, we investigate improvements provided by data-based diagnosis techniques and show that they succeed to isolate faults and attack for this benchmark.

1 Introduction

Connected systems and intelligent systems are largely widespread in the majority of industrial domains (production, transport, networks, aeronautics, space,...). Critical systems as power plants, water plants, smart grids also rely heavily on these Information and Communication Technologies (ICT) to control, to monitor and to manage remotely several equipments. If ICT has contributed to improve the control and monitoring of installations it has also exposed critical systems to a new kind of threats by introducing a high degree of connectivity. Consequently, these systems which include both computational resources, communication capabilities and hardware also called cyber-physical systems, are exposed to malicious attacks, known as cyber-attacks. As in any computer system connected to internet the well known CIA (Confidentiality, Integrity, and Availability) triad of a cyber-physical system can be threatened by non-targeted attacks, but it can also fall prey of targeted attacks. In this case the hacker knows that he is targeting a control system and the cyber-attack is a malicious act aimed at degrading or interrupting the operation of an industrial installation or even destroying it [1]. These attacks exploit the vulnerabilities of the system and can have a very important impact not only from a process point of view but also from financial and environmental ones. Obviously, the influence the hacker can have on the system depends on his knowledge about the system. A prominent example is the attack on the water distribution system in Queensland, Australia [2]. The hacker used a laptop and took the control of 150 sewage pumping stations to release one million liters of untreated sewage into a storm-water drain connected to local waterways. Another example is the Stuxnet attack that taking the control of actuators and sensors struck the Iranian nuclear facility causing the failure of the centrifuges. More recently, we can cite the cyber-attack against Ukraine power grid that cut power to many customers for about six hours. Histories and descriptions of cyber-attacks can be found in [3][4][5][6].

In the context of a classical closed-loop control system architecture, the problem of attack diagnosis is getting closer to the problem of fault diagnosis. Nevertheless, the attack diagnosis problem addresses a larger spectrum of feared scenarios. Indeed, attacks as faults can concern sensors, actuators or the controller itself but the hacker is endowed with intelligence and has capabilities that multiply the possibilities of attacks. Whatever the attack type, its effect is at least one of the following: the sensor returns incorrect measurements, the controller sends out incorrect signals, the actuator ignores control input or executes incorrect commands. According to [7] two main types of attacks can be distinguished: those that concern data integrity issued from sensors or controllers (called by the authors *deception attacks*) and those of type *Denial of Service (DoS)* that threat the signals availability preventing the controller from receiving sensor measurements or the actuators from receiving control commands. In the first case, signals are corrupted by the hacker so that the measurement is incorrect, delayed, replayed etc.

The growing interest in this problem of attack detection and diagnosis is reflected in the Diagnosis community. In [8] an observer based method is developed to diagnose stealthy deception attack that can enable remote water pilfering from automated canal systems. The work of [9] define the notions of detectability and identifiability of an attack by its effect on output. [10] proposes an approach to design an input reconstruction filter to identify the effect of random deception attack on the system behavior in the context of limited access to sensor measurements. In [11] the authors propose a frequency based approach to detect replay attacks on cyber-physical systems. [12] propose an approach based on the introduction of a filter between the controller (Programmable Logic Controller) and the sensors/actuators to detect and prevent orders that could damage the system. Additional information issued from a notion of distance between states allow distinguishing cyber-attacks from classical failures. [13] consider the problem of intrusion detection and mitigation in supervisory control systems. Several types of attacks are considered (en-

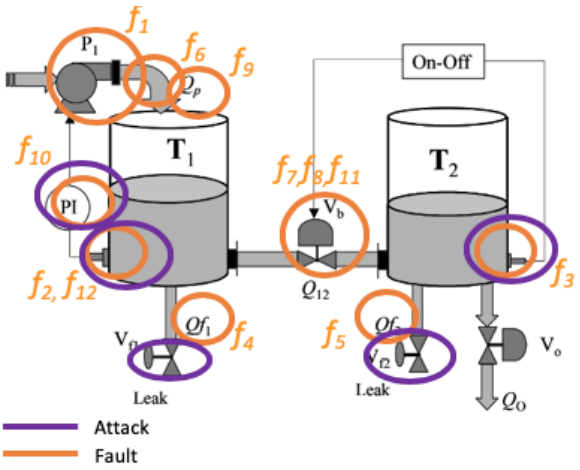

Figure 1: two-tank system

abling/disabling of actuators events, erase/insertion of sensor events).

Recently a two-tank system well known in the diagnosis community [14] has been extended and proposed to the scientific community to test different approaches of detection and isolation of cyber-attacks. To initiate the work the authors considered a classical model-based approach (Analytical Redundancy Relations) [15]. This paper is an answer to this invitation. The contribution of the paper is to show that the initial results obtained from Analytical Redundancy Relations can be improved by the use of data-based methods.

This paper is organized as follows. Section 2 presents the two-tank benchmark. Section 3 describes the different existing scenarios of faults and attacks, as well as some new scenarios proposed in this work. Model-based Diagnosis techniques are applied on the two-tank in Section 4 for detecting and isolating faults and attacks. The use of Data-based methods is developed in Section 5. Section 6 concludes the paper.

2 Benchmark Description

The two-tank benchmark is illustrated in Figure 1. The goal of the system is to provide water to customers with a continuous flow Q_o. A valve V_o simulates the consumer. V_o is opened in nominal mode.

The system consists in two tanks T_1 and T_2 in series. h_1 and h_2 are supposed to be the water level in T_1 and T_2, respectively. T_1 is filled by a pump P_1 controlled by a PI level controller acting on the inlet flow Q_p. It is assumed that Q_p is proportional to the PI controller output U_p^m, that is measured. Note that in the article, the letter "m" is added to a variable when it is measured. The water flow between the two tanks is Q_{12}. It can be controlled by an ON/OFF valve V_b. The water levels in the two tanks may be affected by real leaks but also by attacks simulating a leak. V_{f_1} (resp V_{f_2}) denotes a leakage in T_1, (resp. in T_2) while Q_{f_1} (resp. Q_{f_2}) is an attack. Two sensors provide the measurements of the real water levels of the tanks: h_1^m (for water level in T_1) and h_2^m (for water level in T_2).

A detailed description of the dynamic model of the faultless two-tank system is given in [14].

The input u of the system is supposed to be known.

$$u = \begin{bmatrix} Q_p^m \\ U_b^m \\ U_0^m \\ U_p^m \end{bmatrix} \quad (1)$$

where Q_p^m is the measured flow from the pump P_1, U_b^m represents the position of valve V_b^m (0 if the valve is open, 1 if the valve is closed), U_o^m is the position of the valve V_o, and U_p^m corresponds to the PI controller output.

3 Fault and attack scenarios

The next sections presents the faults and attack scenarios implemented in the two-tank system benchmark.

3.1 Fault Scenarios

A set of fault scenarios has been described in [14].

- f_0 faultless mode: the process runs without fault.
- f_1 Pump fault: from 40s up to 120s the pump is simulated off (like if it is broken).
- f_2 Level sensor mh_1 stuck at zero fault: from 40s up to 120s the output of the sensor is stuck at zero.
- f_3 Level sensor mh_2 fault: from 40s up to 120s the output of the sensor is stuck at zero.
- f_4 Leakage fault in Tank T_1 from 40s up to 120s. $Q_{f_1} = 10^{-4} m^3/s$.
- f_5 Leakage fault in Tank T_2 from 40s up to 120s. $Q_{f_2} = 10^{-4} m^3/s$.
- f_6 Sensor fault for the pump P_1 flow: the sensor is stuck at zero from 40s to up to 120s.
- f_7 Valve V_b stuck-closed fault: v_b is stuck closed from 40s to up to 150s.
- f_8 Valve fault: U_b^m is stuck at zero from 40s to 120s.

Some additional fault scenarios have been defined and tested in this article.

- f_9 Pump sensor fault: Q_p^m is stuck at zero from 40s to up to 120s.
- f_{10} PI controller fault: U_p is always equal to zero from 40s to up to 120s.
- f_{11} Valve V_b stuck-open fault: v_b is stuck open from 40s to up to 150s.
- f_{12} Level sensor mh_1 stuck at non-zero fault: from 40s up to 120s the output of the sensor is stuck at a non-zero value, here $h_1^m = 0.4$.

3.2 Attack Scenarios

In this article we consider as in [15] that the final objective of an attacker is to steal water but we consider also the case where the aim of the attacker is the destruction of the installation or the degradation of the behaviour. We consider an active attacker who has the ability to altering the cyber-physical system by exploiting system security design holes. The attacks can then have some effects on the sensors so that they return incorrect measurements, on the actuators to execute incorrect control actions but also on the controller to perform a physical action on the two-tank system.

We recall here the set of attack scenarios defined in [15].

a_1 Short-term water theft from T_1: this scenario is similar to the scenario with fault f_4. The difference is that it is cast maliciously, with the purpose of stealing water from T_1. $V_{f_1} = 10^{-4} m^3/s$ from 40s up to 80s.

a_2 Short-term water theft from T_1 with hiding signal added to the measurement from 40s up to 80s. The theft is hidden by adding a signal to the output of the level sensor in tank T_1. The PI controller works as if nothing had happened.

a_3 Long-term water theft from T_1 with hiding signal added to the measurement: the attack is the same as a_2 but its duration is extended so that Tank T_1 might become empty, affecting the tank T_2 and Q_0 due to interconnection.

a_4 Long-term water theft from T_1 with small signal added to the measurement: the attack is the same as a_3 but the added signal only compensates 50% of the stolen water, so that the pump will compensate half the theft and the attack will be harder to detect.

a_5 Short-term water theft from T_2 (same as a_1) $V_{f_2} = 10^{-4} m^3/s$ from 40s up to 80s.

a_6 Short-term water theft from T_2 with hiding signal added to the measurement from 40s up to 80s (see a_2).

a_7 Long-term water theft from T_2 with hiding signal added to the measurement (see a_3).

a_8 Long-term water theft from T_2 with small signal added to the measurement (see a_4).

a_9 Replay attack from 160s to up to 200s: the hacker records the measurements coming from the sensors without stealing water from the tanks. Then, when the system has reached its steady-state, the hacker steals water while replacing the sensor values by the recorded ones.

A set of five new attacks has been added to this initial set as follows:

a_{10} Overflow with sensor hack from 40s up to 120s: the objective is not to steal water but to destroy the system. The value of h_1^m is modified to simulate a fake leakage, so that the pump compensates and causes a water overflow in T_1.

a_{11} Overflow with controller hack from 40s up to 120s: the hacker disconnects the PI controller and sets a too high water flow so that there is an overflow in T_1.

a_{12} Sensor hack without theft in tank T_2 from 40s up to 120s: the objective is not to steal water but disturb the system. The sensor value in T_2 is corrupted so that h_2^m remains constant, while the tank is emptied by the customers.

a_{13} Water theft from T_2 when V_0 is closed from 40s to up to 120s: when $Q_0 = 0$, the hacker pretends to be a user but the flows are different.

a_{14} Water theft from T_2 when V_0 is closed from 40s to up to 120s: when $Q_0 = 0$, the hacker pretends to be a user and the flows are the same.

4 Model-based diagnosis techniques on the benchmark

In this article we extend the diagnosis results obtained on the two-tank system based an Analytical Redundancy Relations [16]. Nevertheless, it would be interesting to investigate other model-based approaches notably to take advantage of the dynamical information embedded in the fault and attack profiles. By considering the hybrid nature of the two-tank system, model based approaches developed in the field of switched systems could be investigated as observer based methods.

4.1 Analytical Redundancy Relations

A set of four Analytical Redundancy Relations (ARR) has already been described in [14] and four residuals have been included in the benchmark. We aim to use them for fault detection and isolation, but also for attack detection and isolation.

As a reminder, the ARR are calculated as:

$$\begin{aligned}
r_1(t) = & -C_{vb} U_b^m(t) sign(h_1^m(t) - h_2^m(t)) \sqrt{|h_1^m(t) - h_2^m(t)|} \\
& + Q_p^m(t) - A_1 \frac{dh_1^m}{dt} \\
r_2(t) = & \; C_{vb} U_b^m(t) sign(h_1^m(t) - h_2^m(t)) \sqrt{|h_1^m(t) - h_2^m(t)} \\
& - C_{vo} \sqrt{h_2^m(t)} U_0^m(t) - A_2 \frac{dh_2^m}{dt} \\
r_3(t) = & \; U_p^m(t) - K_P(h_{1,ref} - h_1^m(t)) - K_I \int (h_{1,ref} - h_1^m(\tau)) d\tau \\
r_4(t) = & \; Q_P^m(t) - \begin{cases} U_p^m(t) \text{ if } & 0 < U_p^m(t) < q_{p,max} \\ 0 \text{ if } & U_p^m(t) \leq 0 \\ Q_{pmax} \text{ if } & U_p^m(t) \geq q_{p,max} \end{cases}
\end{aligned} \tag{2}$$

with: C_{vb} the hydraulic flow coefficient of the valve V_b, $A_{i,(i=1,2)}$ the cross-section of the cylindric tank T_i, C_{vo} the hydraulic flow coefficient of the valve V_o, K_P and K_I the coefficients of the PI controller. The numerical values of these parameters are given in [15] and [14].

Two additional residuals have been included in the benchmark, supposing that the pressures in tank T_1 and T_2 may be measured by two new sensors.

We then defined:

$$\begin{aligned}
r_5(t) = & \; P_1^m(t) - \rho.g.h_1^m(t) - P_{atm} \\
r_6(t) = & \; P_2^m(t) - \rho.g.h_2^m(t) - P_{atm}
\end{aligned} \tag{3}$$

4.2 Definition of detectability and isolabity for faults and attacks

Let $F = \{f_1, \ldots, f_{n^f}\}$ be the set of faults of the system, $A = \{a_1, \ldots, a_{n^a}\}$ be the set of attacks. Let f_0 corresponds to the nominal mode without any fault or attack. We define in this section some fundamental definitions for fault diagnosis and attack detection and isolation.

We recall here the definition of a fault signature matrix and extend it for the attacks.

Definition 1 (Fault Signature)**.** *Given a set ARR composed of n^r ARR and F the set of considered n^f faults for the system; consider the function $ARR \times F \longrightarrow \{0,1\}$, then the signature of a fault $f \in F$ is the binary vector $FS(f) = [\tau_1, \ldots, \tau_{n^r}]^T$ where $\tau_k = 1$ if f is a variable of the equation used to form $arr_k \in ARR$, otherwise $\tau_k = 0$.*

The same definition holds for attacks.

Definition 2 (Attack Signature)**.** *Given a set ARR composed of n^r ARR and A the set of considered n^a attacks for the system; consider the function $ARR \times A \longrightarrow \{0,1\}$, then the signature of an attack $a \in A$ is the binary vector $FS(a) = [\tau_1, \ldots, \tau_{n^r}]^T$ where $\tau_k = 1$ if a is a variable of the equation used to form $arr_k \in ARR$, otherwise $\tau_k = 0$.*

We can then define the fault and attack signature matrix as follows:

Definition 3 (Fault and attack signature matrix $FASM$)**.** *The signatures of all the faults in F and of all the attacks in A together constitute the fault and attack signature matrix $FASM$ for the system, i.e $FASM = [FS(f_1), \ldots, FS(f_{n^f}), FS(a_1), \ldots, FS(a_{n^a})]^T$.*

This matrix obviously implies what is usually called the fault signature matrix and what could be called the attack signature matrix.

From the $FASM$ detectability and isolability properties can be defined for faults and attacks.

Definition 4 (Detectable fault)**.** *A fault $f \in F$ is detectable in the system if $FS(f) \neq FS(f_0)$.*

The same definition can be applied for attacks.

Definition 5 (Detectable attack)**.** *An attack $a \in A$ is detectable in the system if $FS(a) \neq FS(f_0)$.*

Definition 6 (Isolability)**.** *Two faults or attacks $fa \in F \cup A$ and $fa' \in F \cup A$ are isolable in the system if they are detectable and if $FS(fa) \neq FS(fa')$.*

Definition 7 (Diagnosability group)**.** *Two faults or attacks $fa \in F \cup A \cup \{f_0\}$ and $fa' \in F \cup A \cup \{f_0\}$ are in the same diagnosability group if they are not isolable from each other.*

In others words, a group of diagnosability is defined as a set of behaviours (nominal, faulty or attacks behaviours) that can not be isolated two by two as their signatures are identical.

4.3 Fault and attack signatures

Some diagnosis results have been provided in [14] but no link between the fault scenarios and the residuals have been clearly established. Here, we aimed at giving the fault and attack signatures for each defined scenario.

Using the four first residuals we succeeded in defining eight diagnosability groups as presented in Table 1.

From Table 1, we can deduce that all faults and attacks are detectable, except the replay attack. Only 3 faults (f_1,f_6 and f_{10}) are isolable from the others and from the attacks. Attacks are always included in diagnosability groups with at least one fault so it is impossible to distinguish them from faults with the ARR approach.

We also tried to take benefit from the two additional residuals obtained from pressure sensors (see section 4.1). From Table 2, we can deduce that all faults and attacks are detectable, except the reply attack, as with 4 residuals. The 2 new residuals increase isolability power: for example diagnosability group G_2 has been split into 3 other diagnosability groups G_{21}, G_{22} and G_{23}, while G_4 is divided into 2 diagnosability groups. The only interesting point is that G_{42} only refers to attack scenarios a_6, a_7, a_8 and a_{12} so that it is possible to distinguish fault from attack in these cases, moreover f_3 is now isolable. However, much ambiguity remains between faults and attacks in 5 diagnosability groups ($G_{21}, G_3, G_{41}G_6, G_8$).

4.4 Results

An HMI has been provided in the matlab benchmark to visualize online the residuals value and the possible diagnoses. For example, we simulated scenario f_9 and found the results illustrated on Figure 2. r_4 is equal to 0, then rises to 1 at 41s.

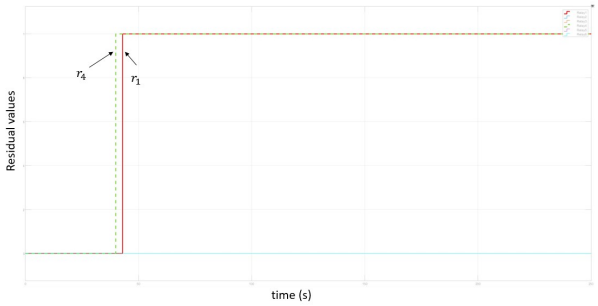

Figure 2: Residual values for scenario f_9

r_1 is equal to 0, then rises to 1 at 44s. The other residuals remain equal to zero. Consequently, the diagnosis hypotheses are the following: from 0s to 41s, the diagnosis hypothesis in $\{f_0, a_9\}$. Between 41s and 44s, it is $\{f_1\}$, then from 44s, the final diagnosis hypothesis is $\{f_9, a_{11}\}$. It can be noticed that the detection delay is then about 1s and the isolation delay is about 4s in this case.

5 Data-based Diagnosis techniques on the benchmark

Since model-based methods are not satisfactory, we decided to consider data-based methods. The final goal is the fusion of model-based and data-based methods, as proposed in [17][18]. At first, we would like to check the efficiency of data-based methods to better isolate faults from attacks for simple cases.

The fault diagnosis and attack isolation problem can be formulated as a classical classification problem. The faults and attacks are considered as different classes in the classification problem. Thanks to simulation, we created a large set of data for each scenario and tried to apply common methods of the machine learning literature, focused on supervised learning with a classification goal, each class corresponding to a fault or an attack. These methods are basically Discriminant Analysis [19], K-Nearest Neighbors [20], decision tree or bagged and boosted decision trees [21]. The goal was then to find a predictive model based on both input and output data.

5.1 Features

The two-tank system has mainly 6 observable variables that have been taken as features and that correspond to the sensor data for the state variables (water level in each tank) and to the inputs. The features are then $h_1^m, h_2^m, Q_p^m, U_b^m, U_0^m$ and U_p^m. Note that the classification methods have been tested without taking into account the additional pressure sensors proposed for deriving residuals r_4 and r_5.

5.2 Training Set

Each scenario has been run 10 times and each run varies because of noises included in the simulation. Each of them has a duration of 250s with a sampling period of 1s, so that each run corresponds to 250 input data. Each input data has been tagged by one of the 17 simplified tags, defined in Table 3. These simplified tags have been used because some scenarios seemed to be exactly the same physically. Future work could investigate the search of differences between them.

As all the scenarios already include some "normal" data (for example, from t=0s up to t=40s, all the data will be

Diagnosability group	FS	Physical explanation
$G_1 : f_1$	$[0001]^T$	Failure pump
$G_2 : f_2, f_3, f_7, f_8, f_{11}, f_{12}, a_2, a_3, a_4, a_{10}$	$[1100]^T$	Impact on T_1
$G_3 : f_4, a_1$	$[1000]^T$	Leakage in T_1
$G_4 : f_5, a_5, a_6, a_7, a_8, a_{12}, a_{13}, a_{14}$	$[0100]^T$	Impact on T_2
$G_5 : f_6$	$[0011]^T$	Sensor fault for P_1 flow
$G_6 : f_9, a_{11}$	$[1001]^T$	Problem on Q_p
$G_7 : f_{10}$	$[0010]^T$	PI controller fault
$G_8 : f_0, a_9$	$[0000]^T$	Reply attack or nominal mode

Table 1: Diagnosability groups with 4 residuals

Diagnosability group	FS	Physical explanation
$G_1 : f_1$	$[000100]^T$	Failure pump
$G_{21} : f_2, f_{12}, a_2, a_3, a_4, a_{10}$	$[110010]^T$	Level sensor mh_1 involved
$G_{22} : f_3$	$[110001]^T$	Level sensor mh_2 involved
$G_{23} : f_7, f_8, f_{11}$	$[110000]^T$	Problem on V_b
$G_3 : f_4, a_1$	$[100000]^T$	Leakage in T_1
$G_{41} : f_5, a_5, a_{13}, a_{14}$	$[010000]^T$	Impact on T_2
$G_{42} : a_6, a_7, a_8, a_{12}$	$[010001]^T$	Attack on T_2
$G_5 : f_6$	$[001100]^T$	Sensor fault for P_1 flow
$G_6 : f_9, a_{11}$	$[100100]^T$	Problem on Q_p
$G_7 : f_{10}$	$[001000]^T$	PI controller fault
$G_8 : f_0, a_9$	$[000000]^T$	Replay attack or nominal mode

Table 2: Diagnosability groups with 6 residuals

scenario	tag	scenario	tag
f_0, a_9	f_0a_9	f_9	f_9
f_1	f_1	f_{10}	f_{10}
f_2, f_{12}	$f_{2,12}$	a_2, a_3, a_4	$a_{2,3,4}$
f_3	f_3	a_6, a_7, a_8	$a_{6,7,8}$
f_4, a_1	f_4a_1	a_{10}	a_{10}
f_5, a_5	f_5a_5	a_{11}	a_{11}
f_6	f_6	a_{12}	a_{12}
f_7, f_{11}	$f_{7,11}$	a_{13}, a_{14}	$a_{13,14}$
f_8	f_8		

Table 3: Simplified tags for classification methods

tagged f_0a_9), we simulate the remaining 25 scenarios (from f_1 to f_{12} and a_1 to a_8) over the 27 initial ones. A total number of 62500 tagged input data have then been stored.

The training was done with Matlab Classification Learner app, that performs supervised machine learning given a known set of input data and known responses to the data (here the tags).

5.3 Validation Scheme and Performances

The chosen validation scheme is the cross-validation method with 5 folds to partition the data set. The app partitions the data into 5 disjoint sets. For each fold, it trains a model using the out-of-fold observations, then assesses model performance using in-fold data.

Table 4 gives the performance results for the main methods included in the Matlab Classification Learner app.

The data-based methods have excellent results for classifying the different scenarios. Fine-kNN has an accuracy of 97.9% when only one neighbor is considered, while the accuracy is 97.4% when 5 neighbors are considered. The Bagged Trees method has an accuracy of 97.5%.

We show on Figure 3 and 4 the confusion matrices for the Fine-kNN and for the Bagged Trees methods. The confusion matrix gives the percentage of true class among the predicted class. As it is possible to see, the worst percentage of true detection is 78% for the Fine-kNN method and 82% for the Bagged Trees method.

From Figure 3 and 4, we can deduce that all the defined tagged cases can be detected by simple data-based methods. Faults and attacks can be isolated, except a_9, a_1 and a_5. These three attacks will be studied in future works. However, the isolation power of data-based methods is much better than model-based methods. For example, with model-based methods, G_6 includes f_9 and a_{11} while data-based methods succeed in isolate both cases. The case f_5a_5 remains the most difficult case to isolate from others.

5.4 New Data Prediction

We exported the models deduced from the Fine-kNN method and from the Bagged-Tree method to the workspace of Matlab and used these trained models to make predictions using new data.

A new scenario involving a failure of the pump has been designed, with different flows. From 0 to 40s, the system has no failure and no attack. From 40s to 120s, the pump is faulty, then after 120s the system again in nominal mode. New data have been tested with the predicted models. Figure 5 shows the prediction results for the Fine-kNN method: each new data has been well classified, except during the transition from normal mode to faulty mode where one data is tagged f_6 instead of f_1. The same problem occurs when using the model trained with Bagged Trees.

Another scenario has been tested, involving an attack corresponding of scenario a_{13}. Figure 6 shows the prediction

method[1]	accuracy (%)	prediction speed (obs/sec)	training time (sec)
Boosted Trees	85.8	21000	88.704
Bagged Trees	97.5	18000	64.451
Medium Tree	77.6	480000	2.826
Fine Tree	88.4	430000	4.4854
Coarse Tree	71.8	500000	2.45
Subspace Discriminant	66.7	12000	36.296
Linear Discriminant	64.6	240000	3.3491
Fine KNN	97.9	47000	8.5107
Medium KNN	96.6	36000	7.143
Coarse KNN	91.2	13000	16.26

Table 4: Performances Results for the main classification included in Matlab

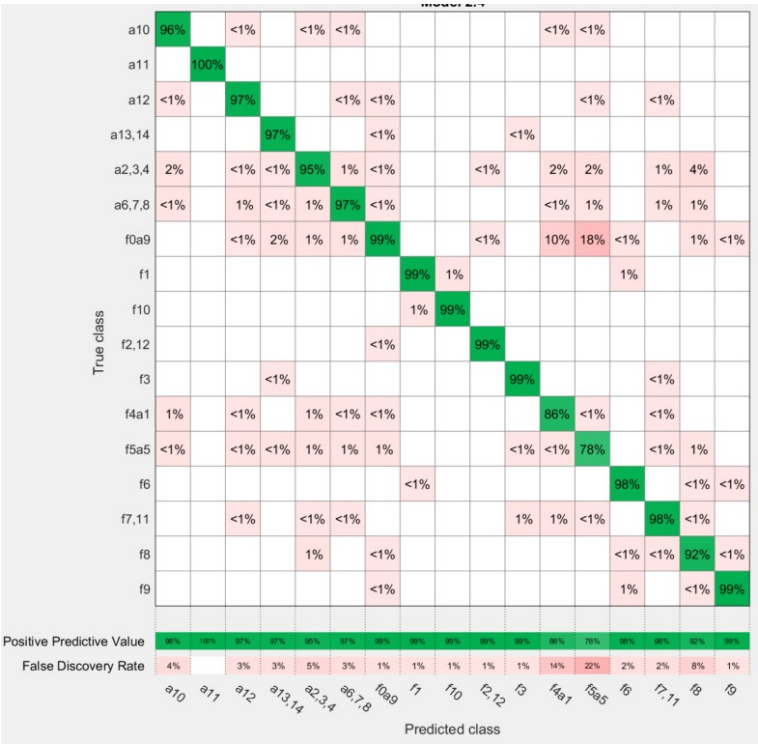

Figure 3: Confusion matrix for the Fine-kNN method

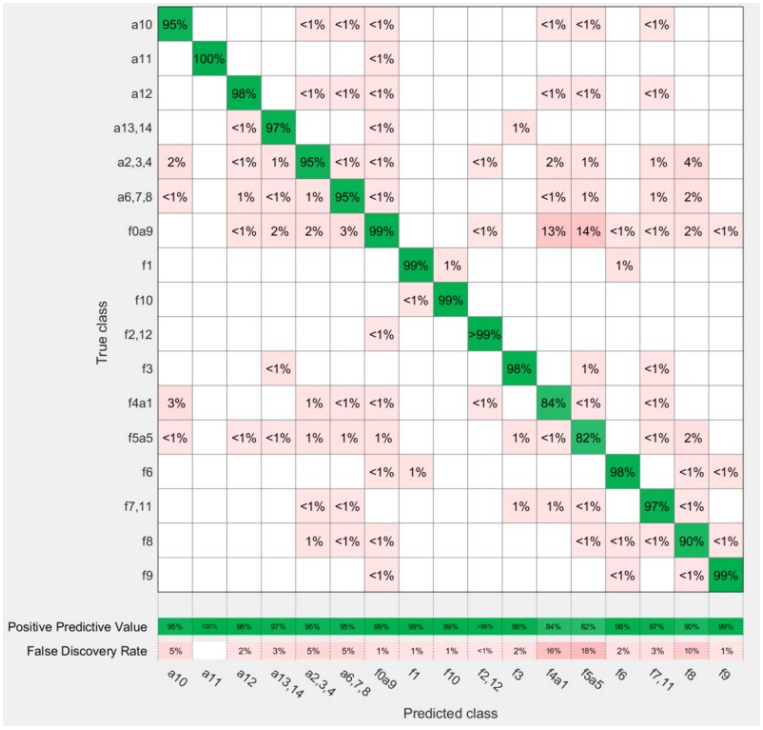

Figure 4: Confusion matrix for Bagged Trees method

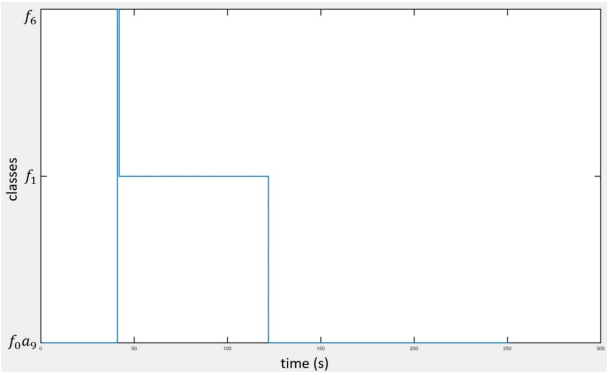

Figure 5: Predicted classes on a scenario of type f_1

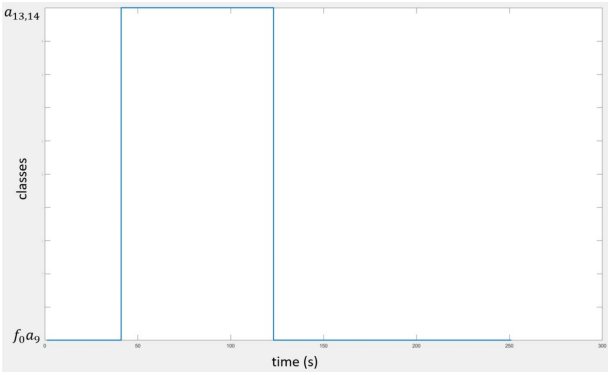

Figure 6: Predicted classes on a scenario of type a_{13}

results for the Fine-kNN method: the attack is immediately detected and tagged $a_{13,14}$ with a Fine-KNN method, while the model-based methods kept the ambiguity with fault f_5. The same results are found with the Bagged Trees method.

6 Conclusion

This article presents the use of model-based diagnosis and data-based techniques for detection and isolation of cyber-attacks and faults on a two-tank system. A set of 12 fault scenarios and 14 attack scenarios was described, 9 of which are new compared to previous works. A first model-based technique was successfully implemented and tested on the benchmark. All faults and attacks are detectable, except replay attack. By introducing 2 new residuals implemented on the benchmark the isolability has been increased so that some attacks can be isolated from faults. However, most of the time the different scenarios have the same signature. In this article the authors propose a first step towards fault and attack detection and isolation. More efficient methods have to be tested to take the fault and attacks dynamic into account. Model-based approaches in the field of switched systems have been mentioned. Another idea could be to take advantage of structural analysis to analyze internal dynamics of the model and residuals.

Data-based diagnosis was performed using supervised machine learning methods. These methods obtained very good results in isolating faults and attacks and could be used in the future either alone, either using a fusion solution. Some aspects of the work however could be consolidated notably the characterization of the faults and attacks. Unsupervised learning methods were also tested. The goal is to find hidden pattern or groupings in data. This approach did not give satisfying results at the moment and should be investigated deeply. Methods for anomaly detection could be beneficial to consider cyberattacks not known in advance. Another research direction is data pre-processing to extract relevant features to fed machine learning methods. Here also dynamical aspects could be better considered.

Acknowledgments

We thank Julie Fouan, Maeva Kleinberg and Florian Valette, 3^{th} year students from INP-ENSEEIHT, Toulouse, who implemented the matlab simulator and the benchmark improvements presented in this paper.

References

[1] Alvaro Cardenas, Saurabh Amin, Zong-Syun Lin, Yu-lun Huang, Chi-Yen Huang, and Shankar Sastry. Attacks against process control systems: Risk assessment, detection, and response. pages 355–366, 01 2011.

[2] J. Slay and M. Miller. Lessons learned from the maroochy water breach. In *International Conference on Critical Infrastructure Protection*, pages 73–82. Springer, 2007.

[3] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A. Sadeghi, M. Maniatakos, and R. Karri. The cybersecurity landscape in industrial control systems. *Proceedings of the IEEE*, 104(5):1039–1057, May 2016.

[4] Yulia Cherdantseva, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby, and Kristan Stoddart. A review of cyber security risk assessment methods for scada systems. *Computers Security*, 56:1 – 27, 2016.

[5] Bill Miller and Dale Rowe. A survey scada of and critical infrastructure incidents. pages 51–56, 10 2012.

[6] F. Khorrami, P. Krishnamurthy, and R. Karri. Cybersecurity for control systems: A process-aware perspective. *IEEE Design Test*, 33(5):75–83, Oct 2016.

[7] Alvaro A. Cárdenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. In *Proceedings of the 3rd Conference on Hot Topics in Security*, HOTSEC'08, pages 6:1–6:6, Berkeley, CA, USA, 2008. USENIX Association.

[8] Saurabh Amin, X Litrico, Shankar Sastry, and Alexandre Bayen. Cyber security of water scada systems—part i: Analysis and experimentation of stealthy deception attacks. *Control Systems Technology, IEEE Transactions on*, 21:1963–1970, 09 2013.

[9] F. Pasqualetti, F. Dörfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, Nov 2013.

[10] Mohamed Amine Sid, Shaikshavali Chitraganti, and Karim Chabir. Medium access scheduling for input reconstruction under deception attacks. *Journal of the Franklin Institute*, 354(9):3678 – 3689, 2017. Special issue on analysis and synthesis of control systems over wireless digital channels.

[11] Helem Sabina Sánchez, Damiano Rotondo, Teresa Escobet, Vicenç Puig, Jordi Saludes, and Joseba Quevedo. Detection of replay attacks in cyber-physical systems using a frequency-based signature. *Journal of the Franklin Institute*, 356(5):2798 – 2824, 2019.

[12] Franck Sicard, Éric Zamai, and Jean-Marie Flaus. *Critical States Distance Filter Based Approach for Detection and Blockage of Cyberattacks in Industrial Control Systems*, pages 117–145. Springer International Publishing, Cham, 2018.

[13] Lilian Kawakami Carvalho, Yi-Chin Wu, Raymond Kwong, and Stéphane Lafortune. Detection and mitigation of classes of attacks in supervisory control systems. *Automatica*, 97:121 – 133, 2018.

[14] B. Ould Bouamama, R. Mrani Alaoui, P. Taillibert, and M. Staroswiecki. Diagnosis of a two-tank system. *Intern Report of CHEM-project, USTL, Lille, France*, 2001.

[15] J. Quevedo, H. Sánchez, D. Rotondo, T. Escobet, and V. Puig. A two-tank benchmark for detection and isolation of cyber attacks. *IFAC-PapersOnLine*, 51(24):770–775, 2018.

[16] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5):687 – 699, 2001.

[17] A. Slimani, P. Ribot, E. Chanthery, and N Rachedi. Fusion of Model-based and Data-based Fault Diagnosis Approaches. In *10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 2018*, Varsovie, Poland, August 2018.

[18] Khaoula Tidriri, Teodor Tiplica, Nizar Chatti, and Sylvain Verron. A generic framework for decision fusion in fault detection and diagnosis. *Engineering Applications of Artificial Intelligence*, 71:73 – 86, 2018.

[19] Jerome Friedman. Regularized discriminant analysis. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 84:165–175, 03 1989.

[20] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[21] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.