IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 29, 2021, accepted August 23, 2021, date of publication August 31, 2021, date of current version September 10, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3109465*

# A Hardware-in-the-Loop Water Distribution Testbed Dataset for Cyber-Physical Security Testing

**L. FARAMONDI[1], F. FLAMMINI[2], (Senior Member, IEEE), S. GUARINO[1], (Graduate Student Member, IEEE), AND R. SETOLA[1], (Senior Member, IEEE)**

[1]Unit of Automatic Control, University Campus Bio-Medico of Rome, 00128 Rome, Italy
[2]School of Innovation, Design and Engineering, Mälardalen University, 63220 Eskilstuna, Sweden

Corresponding author: S. Guarino (s.guarino@unicampus.it)

This work was supported by the Programma Operativo (POR) Fondo europeo di sviluppo regionale (FESR) Lazio Region Project piattafoRma per la valutazionE SperIMentale delle resilienza in infrastrutture critiche (RESIM) under Grant 36673.

**ABSTRACT** This paper presents a dataset to support researchers in the validation process of solutions such as Intrusion Detection Systems (IDS) based on artificial intelligence and machine learning techniques for the detection and categorization of threats in Cyber Physical Systems (CPS). To this end, data were acquired from a hardware-in-the-loop Water Distribution Testbed (WDT) which emulates water flowing between eight tanks via solenoid-valves, pumps, pressure and flow sensors. The testbed is composed of a real subsystem that is virtually connected to a simulated one. The proposed dataset encompasses both physical and network data in order to highlight the consequences of attacks in the physical process as well as in network traffic behaviour. Simulations data are organized in four different acquisitions for a total duration of 2 hours by considering normal scenario and multiple anomalies due to cyber and physical attacks.

**INDEX TERMS** Artificial intelligence, cyber-physical systems, dataset, intrusion detection, machine learning, water distribution, security, testbed, threat recognition.

## I. INTRODUCTION

Industrial Control Systems (ICS) are composed of physical and cyber components used to control industrial processes such as in the case of manufacturing, production, and distribution scenarios [1]. These key elements are also known as Cyber-Physical Systems (CPS), which enable the connection between the operations of the industrial physical plant and the computing and communication infrastructure [2]. They have a crucial role in an ICS because they define both the correct behaviour of the physical process and the correct communication with the Supervisory Control and Data Acquisition (SCADA) systems. CPSs are widely employed in different fields such as smart grids [3], [4], oil and natural gas pipelines [5] and water treatment [6]. Because of their critical role, physical faults, such as broken valves or pumps and cyber attacks can lead to dangerous consequences which can vary from simple changes in network traffic behaviour, such as scanning attacks, to catastrophic events such as loss of service and kinetic effects with dangerous consequences in terms of injury to people, environmental pollution, and physical damage to equipment [4].

In particular, during the last few years, cyber-security has become a critical concern in ICSs due to the widespread usage of wireless networks as well as the opening of industrial networks to the Internet. Despite the benefits of such strategies, such as remote maintenance, simpler adjustment of machines and a constant flow of information, the number of attacks against ICS networks has significantly increased, as reported by Kaspersky in its ICS-CERT annual report [7].

For these reasons, different types of testbeds are needed to measure the effects of cyber and physical attacks on industrial processes and to assess security countermeasures, as witnessed by the results reported in recent scientific literature [8]–[10].

Among the most widespread solutions to secure CPSs, we can mention Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) [13], for network

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

**TABLE 1.** Datasets comparison in terms of network data, physical data, attack types and testbed structure.

| Dataset | Network data | | | Physical data | | |
|---|---|---|---|---|---|---|
| | Available | N° of samples | N° of features | Available | N° of samples | N° of features |
| [11] | X | X | X | ✓ | 66.893 | 7 |
| [12] | ✓ | ∼ 400 M | 18 | ✓ | ∼ 1 M | 51 |
| Our solution | ✓ | ∼ 20 M | 15 | ✓ | 9206 | 41 |

| | Attack types | | | | Testbed | | | |
|---|---|---|---|---|---|---|---|---|
| | DoS attack | MITM attack | Scanning attack | Physical attacks | N° of PLC | N° of sensors | N° of tanks | N° of actuators |
| [11] | ✓ | ✓ | X | ✓ | 1 | 5 | 2 | 2 |
| [12] | X | ✓ | X | X | 6 | 24 | 3 | 27 |
| Our solution | ✓ | ✓ | ✓ | ✓ | 4 | 12 | 8 | 28 |

monitoring (NIDS) and host monitoring (HIDS). In particular, recent scientific literature is addressing areas such as artificial intelligence and machine learning for IDSs and IPSs [14], [15], which seem to be particularly effective in recognizing unforeseen attacks [16]–[18]. A crucial point is the evaluation of these systems in order to assess their ability to detect attacks: to this aim, realistic and sufficiently complex datasets are needed.

The Scientific Literature provides some datasets such as KDD-99 [19] with its updated version NSL-KDD99 [20], UNSW-NB15 [21] and CTU-13 dataset [22]; however, all of them present Information Technology (IT) network traffic without any reference to physical plants. Therefore, there is a need for new datasets with traffic taken from Operational Technology (OT) networks where hardware and software are used to monitor and control physical processes, devices and infrastructure [23]. Moreover, in addition to the network traffic, data taken from Programmable Logic Controllers (PLC) are necessary in order to inquire the effects of cyber and physical attacks against the physical plant [24].

In [11], the authors provide a CPS dataset composed of two tanks, two pumps, one ultrasound sensor, four liquid level sensors and one PLC. Data consist of PLC register values which are reported in 15 `csv` different files. Each of these refers to normal traffic and different types of attacks both physical, such as a person hitting a tank and cyber such as Denial of Service (DoS). The excessive simplicity and the lack of network traffic make this dataset insufficient to guarantee a realistic evaluation of IDSs or IPSs.

On the other hand, the dataset described in [12] is more complex and sophisticated: it is provided by iTrust, the Centre for Research in Cyber Security at the Singapore University. The dataset refers to a Secure Water Treatment (SWAT) testbed consisting of six different stages each of which characterized by a particular physical process controlled by one PLC. Data are reported in `csv` files: one refers to the physical variables read from PLCs while other 784 files report MODBUS-only network traffic. Attacks are launched against the physical elements such as pumps or valves or against the communication network between sensors, actuators and PLCs in order to corrupt the information exchanged. Thus, there is no reference to different types of cyber attack such as DoS and scanning attacks which are typically launched against ICS networks, as described in [25] and in [26]. Moreover, the authors do not consider attacks against the communication network between the SCADA, which acquires the data, and the PLCs. Another possible issue of this dataset is the size; specifically, authors provide about 1 million samples for the physical dataset and a total of about 400 million samples for the network one. This characteristic leads researchers to adopt small and random subsets of the dataset causing serious difficulties in comparing results of different research works, as happened in [27] and explained in [20].

This paper aims to overcome these limitations by providing a hardware-in-the-loop cyber-physical dataset obtained from a Water Distribution Testbed (WDT) [28]. The testbed is partially simulated thanks to the *minicps* tool in order to represent a more complex scenario by increasing the number of tanks and PLCs connected to the ICS network [29]. Data are both physical measurements taken from PLCs and network traffic presenting normal and malicious packets under different types of attacks. Moreover, the limited number of samples makes it convenient to test different IDS solutions on the complete set without the need to select a small random partition. In fact, even if the complexity of a dataset is important in order to faithfully emulate a real industrial plant, a too large dataset is not properly managed by machine learning algorithms reducing its usability [30]. Thus, evaluation results of different papers could be effectively compared in order to identify the best algorithms without any influence from the selected random data partitions.

Therefore, the main contributions are as follows:

- An ICS dataset providing both physical and network data in order to highlight the relations between cyber and physical aspects of the system.
- A balanced complex dataset that can provide more types of cyber and physical attacks and more realistic scenarios while keeping, at the same time, a small number of samples. In this way, we provide a reduced sized dataset that ensures a good trade-off between complexity and usability.

Table 1 summarizes the key features of our dataset in relation to those described in [11] and [12].

The remainder of the paper is organised as follows. Section 2 describes the water distribution testbed and network topology. Section 3 describes the data acquisition and the attacks launched against the testbed. Section 4 describes the organization of the dataset. Section 5 provides some

**FIGURE 1.** **Real subsystem of the WDT.**

preliminary results by applying four machine learning algorithms; while Section 6 concludes the paper.

## II. WATER DISTRIBUTION TESTBED

### A. PHYSICAL CHARACTERISTICS

The WDT is composed of two main subsystems: a real one and a simulated one. As illustrated in Figure 1, the real subsystem consists of 5 tanks ($T_1^r \dots T_5^r$), 20 solenoid valves ($V_1^r \dots V_{20}^r$), 4 pumps ($P_1^r \dots P_4^r$) and 5 pressure sensors ($S_1^r \dots S_5^r$) under each tank. In addition, 8 manual valves are provided in order to simulate water leaks from tanks or pipes. Specifically, tanks are made of polyurethane and are characterized by the following dimensions:

- $S_3^r$ and $S_4^r$: height = 36 *cm*, circumference = 70 *cm*
- $S_1^r$ and $S_2^r$: height = 45 *cm*, circumference = 90 *cm*
- $S_5^r$: height = 40 *cm*, circumference = 100 *cm*

Solenoid valves are Evian©Series 263-Model D263DVP powered at 24V. Each tank has a multiple number of outlet valves in order to modulate the output flow. Specifically, as shown in Figure 2, $T_1^r$ is equipped with outlet valves $V_1^r$, $V_2^r$, $V_3^r$ and $V_4^r$; $T_2^r$ with $V_5^r$, $V_6^r$, $V_7^r$ and $V_8^r$; $T_3^r$ with $V_{10}^r$, $V_{11}^r$ and $V_{12}^r$; $T_4^r$ with $V_{13}^r$, $V_{14}^r$ and $V_{15}^r$ and $T_5^r$ with $V_{19}^r$ and $V_{20}^r$.

Pressure sensors are WIKA©S-11, with a measurement range of 0 . . . 0.1 bar.

Pumps $P_1^r$, $P_2^r$ and $P_3^r$ are Mini-Type Pipe Pump 151410 with a maximum flow of 20 $l/min$ while $P_4^r$ is a EK-DCP 2.2 with a maximum flow of 6 $l/min$.

Tanks are connected by pipes in cross-linked multi-layer polyurethane (PE-Xb) with an external diameter of 7/8″.

The simulated subsystem was implemented by using the *minicps* tool, a lightweight simulator for accurate network traffic in industrial control systems, with basic support for physical layer interaction. It was installed on an Ubuntu machine with the following characteristics: Intel® Xeon® CPU E5-2620 v2 @ 2.10 GHz with a RAM of 16 GB. As illustrated in Figure 2, the simulated environment adds complexity to the real testbed with the addition of 3 tanks ($T_6^s \dots T_8^s$), 2 pumps ($P_5^s$, $P_6^s$), 4 flow sensors ($F_1^s \dots F_4^s$), 2 solenoid valves ($V_{21}^s$, $V_{22}^s$) and 3 pressure sensors ($S_6^s \dots S_8^s$) for each tank. Specifically, tanks are modelled with a circumference of 100 *cm* and a height of 40 *cm*. Pipes are modeled with an external diameter of 7/8″ while pumps are characterized by a flow of 4 $l/min$.

The two subsystems form a water distribution testbed in a hardware-in-the-loop fashion where water flow goes from the real subsystem to the simulated one and vice versa.

### B. PROCESS DETAILS

For the sake of clarity, we now describe in detail the nominal behaviour of the process. According to the scheme represented in Figure 3, the process consists of four stages each of which is controlled by a specific PLC. The first stage $S1$, which is controlled by the real PLC, $PLC_1^r$, starts with pumping the water from the reservoir towards two different paths:

- **Path 1**: The water is pumped by $P_1^r$ towards $T_2^r$. Then, thanks to $V_{17}^r$, it starts to fill up $T_3^r$. When the water level reaches a specific threshold, $V_{10}^r$, $V_{11}^r$ and $V_{12}^r$ are activated in order to get water back to the reservoir.
- **Path 2**: The water is pumped by $P_2^r$ towards $T_1^r$. Then, $P_4^r$ is activated in order to fill up $T_5^r$. When the water level reaches a specific threshold, $P_4^r$ is deactivated. As a result, the remaining water in $T_1^r$ is drained towards $T_4^r$ thanks to the opening of $V_{18}^r$ and then through valves $V_{13}^r$, $V_{14}^r$ and $V_{15}^r$ towards the reservoir.

The second stage $S2$ starts when water level in $T_5^r$ reaches the predefined threshold. $PLC_2^s$, simulated in *minicps*, opens solenoid valve $V_{20}^r$ and starts to fill up $T_6^s$: its water level increases as much as water level in $T_5^r$ decreases. Thus, even if $V_{20}^r$ drains the water towards the reservoir, it is virtually deviated towards $T_6^s$ in order to start the simulated physical process in *minicps*. The water then reaches stage $S3$ thanks to $P_5^s$ controlled by $PLC_3^s$: $T_7^s$ starts to fill up while $F_1^s$ records water flow downstream of the pump.

The last stage $S4$ is controlled by $PLC_4^s$ which defines water flowing from $T_7^s$ towards $T_8^s$ thanks to $P_6^s$. Also in this case the water flow is measured by $F_2^s$. When the water level in $T_8^s$ reaches a specific threshold, $PLC_4^s$ opens solenoid valve $V_{22}^s$ in order to virtually drain the water towards the reservoir.

### C. NETWORK ARCHITECTURE

Network architecture is consistent with the typical three-layer SCADA architecture defined in [31] and shown in Figure 4. The adopted communication protocol is MODBUS TCP/IP which is the de-facto standard used in industrial networks [32].

The first layer is the Field Instrumentation Control Layer which consists of sensors and actuators connected to the PLCs via wired links. All of them are connected to the I/O analog or digital module of the PLCs with the exception of flow sensors $F_1^s$ and $F_2^s$ which are MODBUS TCP/IP sensors with their own IP addresses.

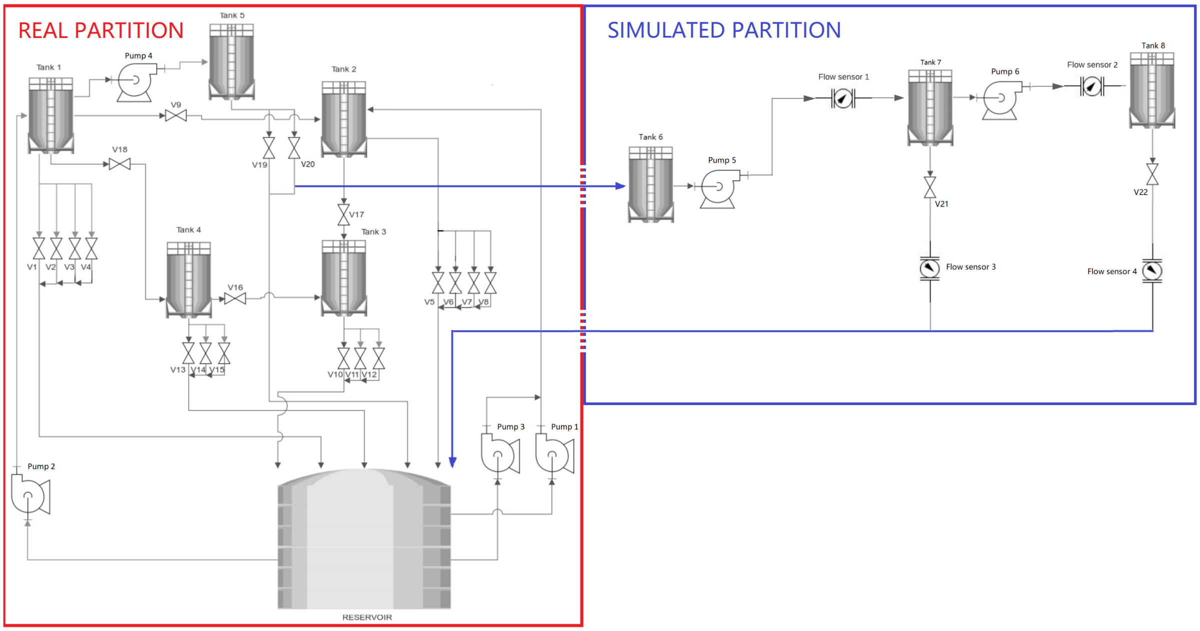



**FIGURE 2.** WDT schematic: the left red rectangle represents the real subsystem while the right blue rectangle the simulated subsystem of the water testbed. Blue rows represent virtual water flowing between the two subsystems.

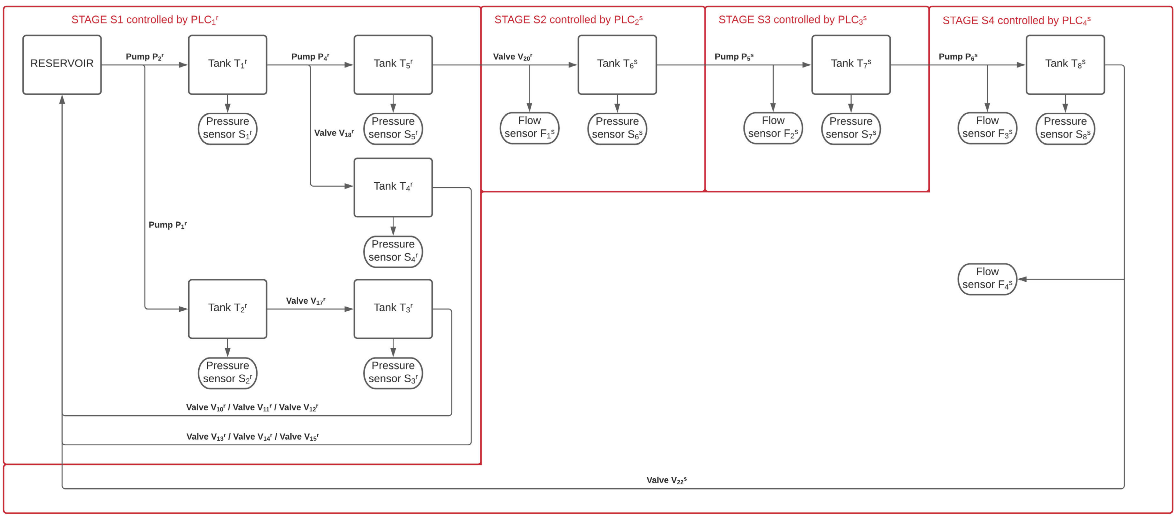



**FIGURE 3.** WDT physical process divided into 4 stages: the first is controlled by the real PLC, $PLC_1^r$, while the remaining ones are controlled by the simulated PLCs, $PLC_2^s$, $PLC_3^s$ and $PLC_4^s$.

The second layer is the Process Control Layer which consists of the four PLCs. In particular, the real one is a Modicon M340 equipped with BMX P342020 processors, DDM16025 discrete I/O and AMM0600 mixed analog I/O modules.

The third and last layer is the Process Control Layer which consists of the Supervisory Control and Data Acquisition system Movicon 11.6 installed on a Windows Server 2012 machine with the following characteristics: Intel® Xeon® CPU E5-2620 v2 @ 2.10 GHz with a RAM of 16 GB. The SCADA includes the Human Machine Interface (HMI) and a Historian which reads and stores data from PLCs.

As shown in Figure 5, the communication network consists of four PLCs, 2 MODBUS TCP/IP flow sensors, the SCADA workstation and an additional host, a Kali Linux machine, which was used to launch cyber attacks, described in detail in Section III.

## III. ATTACKS AGAINST THE TESTBED

As mentioned in Section I, in this work, we considered two different types of attack:

**TABLE 2. Cyber and physical attacks.**

| Type | Class | Subclass | Description |
|---|---|---|---|
| *Cyber* | Man-In-The-Middle (MITM) attack | ARP poisoning | Attack against MODBUS communication protocol with the intention to modify packets data payload |
| *Cyber* | Denial of Service (DoS) attack | TCP flood<br>ICMP flood<br>Land attack | Attack against single hosts with the intention to saturate their resources and to disconnect them from the network |
| *Cyber* | Scanning attack | SYN scan<br>FIN scan<br>Null scan<br>XMAS scan | Attack against hosts with the intention to gather network information about these devices |
| *Physical* | Water leak | Opening of manual valves | Attack consisting in opening manual valves in order to generate water leaks from tanks |
| *Physical* | Sensors and pumps breakdown | | Attack consisting in blocking sensors and pumps |

**TABLE 3. Attack scenarios per each acquisition.**

| # scenario | Start time | End time | Type of attack | Effects on physical process | Effects on network traffic | Elapsed time | Cycle |
|---|---|---|---|---|---|---|---|
| 1.1 | 09/04/2021 18:25:48 | 09/04/2021 18:28:14 | *Cyber*: MITM attack against $PLC_2^s$ and $PLC_3^s$. Affected sensor value: $S_6^s$. | ✓ | ✓ | 2'20" | I |
| 1.2 | 09/04/2021 18:30:08 | 09/04/2021 18:31:14 | *Physical*: water leak from $T_2^r$ towards $T_3^r$ and from $T_2^r$ towards reservoir | ✓ | X | 1'21" | II |
| 1.3 | 09/04/2021 18:34:11 | 09/04/2021 18:35:38 | *Cyber*: MITM attack against $PLC_1^r$ and $PLC_2^s$. Affected sensor value: $S_5^r$. | ✓ | ✓ | 1'14" | III |
| 1.4 | 09/04/2021 18:38:38 | 09/04/2021 18:39:50 | *Physical*: $P_2^r$ breakdown and water leak from $T_1^r$ towards $T_4^r$ | ✓ | X | 12" | IV |
| 1.5 | 09/04/2021 18:43:52 | 09/04/2021 18:45:54 | *Cyber*: MITM attack against $PLC_3^s$ and $PLC_4^s$. Affected sensor value: $S_7^s$. | ✓ | ✓ | 5'26" | IV |
| 1.6 | 09/04/2021 18:49:02 | 09/04/2021 18:51:18 | *Physical*: water leak from $T_2^r$ towards $T_3^r$ | ✓ | X | 14" | VI |
| 1.7 | 09/04/2021 18:58:05 | 09/04/2021 18:59:32 | *Cyber*: MITM attack against $F_1^s$ and $PLC_3^s$. Affected sensor value: $F_1^s$. | ✓ | ✓ | 3'59" | VII |
| 1.8 | 09/04/2021 19:00:40 | 09/04/2021 19:02:07 | *Cyber*:MITM attack against $F_2^s$ and $PLC_3^s$. Affected sensor value: $F_2^s$. | ✓ | ✓ | 1'49" | VIII |

**Second acquisition** *attack_2.csv*, *phy_att_2.csv*

| # scenario | Start time | End time | Type of attack | Effects on physical process | Effects on network traffic | Elapsed time | Cycle |
|---|---|---|---|---|---|---|---|
| 2.1 | 19/04/2021 15:38:52 | 19/04/2021 15:38:52 | *Cyber*: SYN scan against $PLC_1^r$ | X | ✓ | 1'40" | I |
| 2.2 | 19/04/2021 15:40:09 | 19/04/2021 15:40:09 | *Cyber*: FIN scan against $PLC_2^s$ | X | ✓ | 2'57" | I |
| 2.3 | 19/04/2021 15:41:10 | 19/04/2021 15:41:10 | *Cyber*: XMAS scan against $PLC_3^s$ | X | ✓ | 3'58" | I |
| 2.4 | 19/04/2021 15:42:03 | 19/04/2021 15:42:03 | *Cyber*: Null scan against $PLC_4^s$ | X | ✓ | 0'0" | II |
| 2.5 | 19/04/2021 15:43:46 | 19/04/2021 15:44:22 | *Cyber*: ICMP flood attack against $PLC_1^r$ from spoofed HMI IP address | ✓ | ✓ | 1'42" | II |
| 2.6 | 19/04/2021 15:48:15 | 19/04/2021 15:48:56 | *Physical*: breakdown of $P_4^r$ | ✓ | X | 1'2" | III |
| 2.7 | 19/04/2021 15:51:16 | 19/04/2021 15:51:16 | *Cyber*: Null scan against $PLC_3^s$ | X | ✓ | 4'3" | III |
| 2.8 | 19/04/2021 15:54:39 | 19/04/2021 15:55:59 | *Physical*: breakdown of $V_{20}^r$ | ✓ | X | 1'58" | IV |
| 2.9 | 19/04/2021 15:58:40 | 19/04/2021 15:58:40 | *Cyber*: SYN scan against $PLC_1^r$ | X | ✓ | 5'59" | IV |
| 2.10 | 19/04/2021 15:59:57 | 19/04/2021 16:00:10 | *Cyber*: ICMP flood against $PLC_1^r$ | X | ✓ | 1'12" | V |
| 2.11 | 19/04/2021 16:04:39 | 19/04/2021 16:04:39 | *Cyber*: FIN scan against $PLC_2^s$ | X | ✓ | 1'40" | VI |
| 2.12 | 19/04/2021 16:08:26 | 19/04/2021 16:10:01 | *Cyber*: MITM attack against $PLC_3^s$ and $PLC_4^s$. Affected sensor value: $S_7^s$. | ✓ | ✓ | 4'27" | VI |
| 2.13 | 19/04/2021 16:11:46 | 19/04/2021 16:12:14 | *Cyber*: TCP flood against $PLC_1^r$ from spoofed $PLC_2^s$ IP address | ✓ | ✓ | 2'58" | VII |

**Third acquisition** *attack_3.csv*, *phy_att_3.csv*

| # scenario | Start time | End time | Type of attack | Effects on physical process | Effects on network traffic | Elapsed time | Cycle |
|---|---|---|---|---|---|---|---|
| 3.1 | 09/04/2021 19:43:17 | 09/04/2021 19:44:08 | *Physical*: breakdown of $P_4^r$ | ✓ | X | 1'5" | I |
| 3.2 | 09/04/2021 19:45:52 | 09/04/2021 19:46:21 | *Cyber*: ICMP flood against HMI from its spoofed IP address | X | ✓ | 3'40" | I |
| 3.3 | 09/04/2021 19:49:54 | 09/04/2021 19:50:48 | *Physical*: breakdown of $V_{20}^r$ | ✓ | X | 2'1" | II |
| 3.4 | 09/04/2021 19:54:00 | 09/04/2021 19:54:45 | *Physical*: breakdown of $P_2^r$ | ✓ | X | 28" | III |
| 3.5 | 09/04/2021 19:55:29 | 09/04/2021 19:56:15 | *Cyber*: ICMP flood against $PLC_1^r$ with huge payloads | ✓ | ✓ | 1'58" | III |
| 3.6 | 09/04/2021 19:58:02 | 09/04/2021 19:59:09 | *Cyber*: MITM attack against $PLC_3^s$ and $PLC_4^s$. Affected sensor value: $S_7^s$. | ✓ | ✓ | 4'30" | III |
| 3.7 | 09/04/2021 20:01:18 | 09/04/2021 20:02:03 | *Cyber*: MITM attack against $PLC_3^s$ and $PLC_4^s$. Affected sensor value: $S_7^s$. | X | ✓ | 2'20" | IV |

- **Physical attacks:** they are defined as attacks against the physical elements such as sensors and actuators. Some examples are leaks from tanks and pipes, sensors or actuators failures.
- **Cyber attacks:** they are defined as attacks against hosts (SCADA, PLC, and flow sensor) or communication links. Some examples are Denial of Service (DoS) attacks, scanning attacks and MITM attacks.

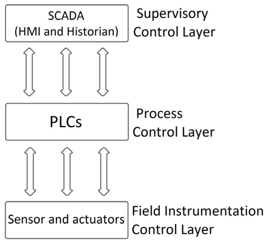



**FIGURE 4.** SCADA architecture.

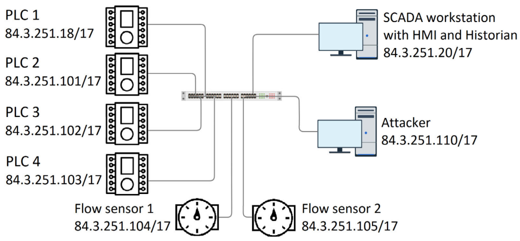



**FIGURE 5.** SCADA network.

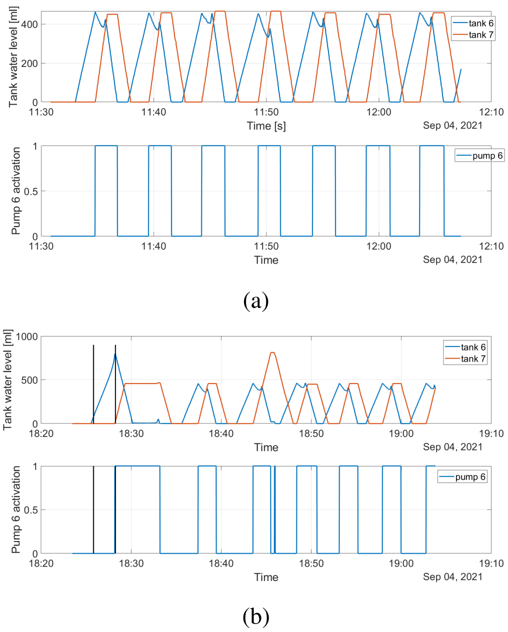



**FIGURE 6.** Effect of a MITM attack against $PLC_2^s$ and $PLC_3^s$ on physical process (Scenario 1.1). The attack changes the water level value of $T_6^s$ requested by $PLC_3^s$ to $PLC_2^s$. (a) shows the normal scenario while (b) the attack effect. Black lines indicate the start and the end of the MITM attack.

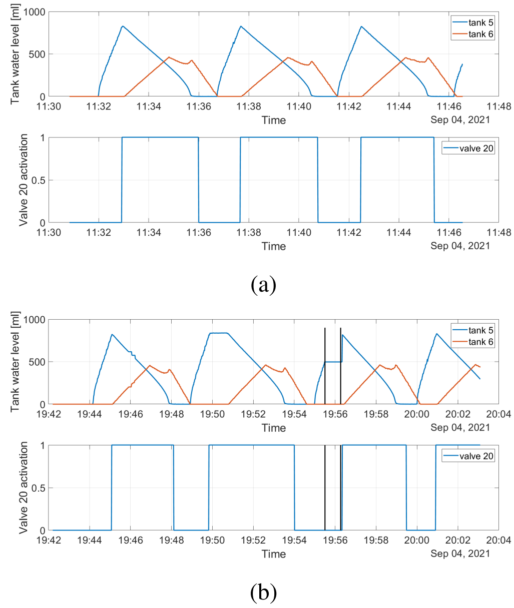



**FIGURE 7.** Effect of a DoS attack against $PLC_1^r$ on physical process (Scenario 3.5). The attack causes the disconnection of $PLC_1^r$ from the network causing a delay in the filling of $T_6^s$. (a) shows the normal scenario while (b) the attack effect. Black lines indicate the start and the end of the DoS attack.

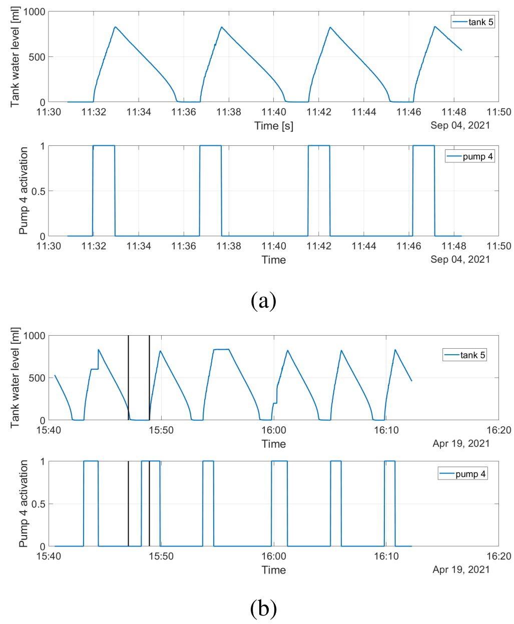



**FIGURE 8.** Effects of a physical attack against $P_4^r$ on the physical process (Scenario 2.6). The attack causes the breakdown of $P_4^r$, which stops water flow towards $T_5^r$. (a) shows the normal scenario, while (b) the effect of the attack. Black lines indicate the start and the end of the physical attack.

According to the ontology provided in [25] and [26], each type of attack is classified and described in Table 2. Attacks can be divided into five different classes and, for each of them, we considered specific subclasses such as SYN scan and FIN scan for scanning attacks [26]. All attacks are launched against both real and simulated subsystems of the WDT. In particular, cyber attacks are carried out thanks to a Kali Linux machine with the following hardware configuration: Intel R, Core(TM) i7-8750H CPU @2.20GHz (1CPU), 4GB RAM.

## A. ATTACK SCENARIOS

Considering the different attacks described in Table 2, we have defined 28 attack scenarios by varying the start time and the specific target. As summarized in Table 3, the effects of such attacks scenarios are collected in three of the four different acquisitions that will be described in-depth in Section IV-A. Table 3 shows the scenarios specifying whether a particular attack has an impact on the physical process or on the network traffic. In particular, as we expected, physical attacks have no effect on the network traffic because they are focused only on the physical components of the testbed. On the other hand, all cyber attacks have effects against the network traffic but not necessarily on the physical process. This behaviour depends on three factors: the time when a specific attack is launched, the current process state, and the specific target. In this perspective, notice that despite Scenario 3.6 and Scenario 3.7 are characterized by the same type of attack (MITM), only the first one has an impact on the physical process. Specifically, in our dataset, a MITM attack fixes the required sensor value to the last value read by the victim before the attack. Thus, in these two scenarios, the attack fixes the water level of $T_7^s$ to the last not impaired value required by $PLC_4^s$ to $PLC_3^s$.

The presence or absence of attack effects against both physical and network behaviour makes the classification task of machine learning algorithms more complex and challenging, as will be described in Section V-D.

Figures 6, 7 and 8 show three different attack scenarios against the physical process; specifically, they refer to Scenario 1.1, Scenario 3.5 and Scenario 2.6 respectively.

Figure 6 shows the effects of a MITM attack against $PLC_2^s$ and $PLC_3^s$. The attacker fixes the water level of $T_6^s$ at the last value required by $PLC_3^s$ to $PLC_2^s$ before the attack. In this way, $PLC_3^s$ will receive always the same compromised value for the entire duration of the attack. Thus, $PLC_3^s$ does not activate $P_5^s$ causing an abnormal increase in the water level of $T_6^s$ while $T_7^s$ remains empty until the attack ends.

Figure 7 shows the effects of a DoS attack against $PLC_1^r$. The attack causes the disconnection of $PLC_1^r$ from the network while $T_5^r$ is still filling up. As a result, $PLC_2^s$ is not able to read the actual value of water level in $T_5^r$ delaying the filling of $T_6^s$.

Figure 8 shows the effects of a physical attack against $P_4^r$. The attack causes the breakdown of $P_4^r$ which stops water flow towards $T_5^r$.

## IV. DATASET ANALYSIS

### A. DATA ACQUISITION

With the aim of reducing the total size of the dataset, we provide four different acquisitions characterised by an overall duration of about 2 hours. Each acquisition consists of a certain number of cycles of the physical process in order to ensure a sufficient knowledge about the normal operation and to define the 28 attack scenarios described in Section III. Specifically, the first acquisition lasts 1 hour and shows a total of 12 process cycles: it refers to the WDT while working in normal conditions without any attack. On the contrary, the remaining three acquisitions, which last 60 minutes, provide 8, 7 and 4 process cycles respectively. They report data about the attacks described in Section III which cause different effects on the physical process or on the network behaviour. These effects depend on the type of attack, the time at which the attack was launched and on the particular target. Consecutive attacks were avoided if both of them caused significant variations in the physical process or network traffic: in these cases, the time between two attacks is at least as long as the time needed to bring WDT back in a safe and normal condition. As shown in Table 2, attack scenarios are distributed along the different cycles and are temporally separated in order to reduce mutual influence. In particular, the column *Cycle* defines the specific physical cycle that is affected by the attack scenario; while, the column *Elapsed time* defines the time elapsed since the beginning of the same cycle.

The acquisitions started with all tanks empty.

For each acquisition, we provide two different datasets: a physical one, which reports the physical measurements of sensors, solenoid valves and pumps taken from PLCs and saved by the historical data recorder (Historian), and a network one, which reports packets features about the traffic exchanged in the SCADA network.

In Figures 9 and 10, the total number of samples for network and physical datasets are reported.

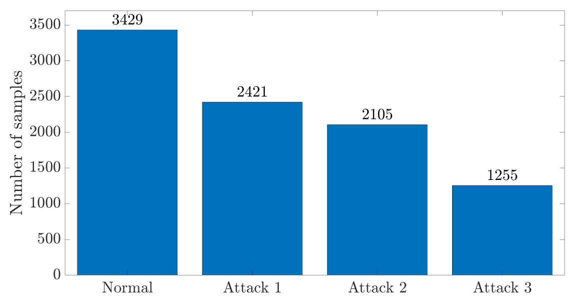



**FIGURE 9. Number of samples for physical dataset reported for each acquisition.**

### B. PHYSICAL DATASET

The Historian recorded the physical data every second in a `csv` file. Thus, each record represents sensors, pumps and solenoid valves states taken from the four PLCs at a particular time. Samples are defined by 41 features which are reported in Table 4.

### C. NETWORK DATASET

Network traffic of all network segments was captured thanks to the *Wireshark* software. Features were extracted from the outgoing `pcap` file using *Python*. Specifically, features were selected by considering that ICS networks are more deterministic and static than IT networks where, on the contrary, changes in terms of network topology and network traffic are more frequent, as described in [33]. Taking this into account, features were selected according to [34] where the authors studied which attributes best differentiate between anomalous and normal behaviour in ICS networks. We considered

**TABLE 4.** Features of physical dataset.

| № | Features | Description | № | Features | Description |
|---|---|---|---|---|---|
| 1 | Time | Datetime of acquisition | 22 | Valv_3 | State of solenoid valve 3 |
| 2 | Tank_1 | Pressure sensor value of tank 1 | 23 | Valv_4 | State of solenoid valve 4 |
| 3 | Tank_2 | Pressure sensor value of tank 2 | 24 | Valv_5 | State of solenoid valve 5 |
| 4 | Tank_3 | Pressure sensor value of tank 3 | 25 | Valv_6 | State of solenoid valve 6 |
| 5 | Tank_4 | Pressure sensor value of tank 4 | 26 | Valv_7 | State of solenoid valve 7 |
| 6 | Tank_5 | Pressure sensor value of tank 5 | 27 | Valv_8 | State of solenoid valve 8 |
| 7 | Tank_6 | Pressure sensor value of tank 6 | 28 | Valv_9 | State of solenoid valve 9 |
| 8 | Tank_7 | Pressure sensor value of tank 7 | 29 | Valv_10 | State of solenoid valve 10 |
| 9 | Tank_8 | Pressure sensor value of tank 8 | 30 | Valv_11 | State of solenoid valve 11 |
| 10 | Pump_1 | State of pump 1 | 31 | Valv_12 | State of solenoid valve 12 |
| 11 | Pump_2 | State of pump 2 | 32 | Valv_13 | State of solenoid valve 13 |
| 12 | Pump_3 | State of pump 3 | 33 | Valv_14 | State of solenoid valve 14 |
| 13 | Pump_4 | State of pump 4 | 34 | Valv_15 | State of solenoid valve 15 |
| 14 | Pump_5 | State of pump 5 | 35 | Valv_16 | State of solenoid valve 16 |
| 15 | Pump_6 | State of pump 6 | 36 | Valv_17 | State of solenoid valve 17 |
| 16 | Flow_sensor_1 | Flow sensor value 1 | 37 | Valv_18 | State of solenoid valve 18 |
| 17 | Flow_sensor_2 | Flow sensor value 2 | 38 | Valv_19 | State of solenoid valve 19 |
| 18 | Flow_sensor_3 | Flow sensor value 3 | 39 | Valv_20 | State of solenoid valve 20 |
| 19 | Flow_sensor_4 | Flow sensor value 4 | 40 | Valv_21 | State of solenoid valve 21 |
| 20 | Valv_1 | State of solenoid valve 1 | 41 | Valv_22 | State of solenoid valve 22 |
| 21 | Valv_2 | State of solenoid valve 2 | | | |

**TABLE 5.** Features of network dataset.

| № | Features | Description |
|---|---|---|
| 1 | Time | Date of acquisition |
| 2 | Src IP address | Source IP address |
| 3 | Dst IP address | Destination IP address |
| 4 | Src MAC address | Source MAC address |
| 5 | Dst MAC address | Destination MAC address |
| 6 | Src Port | Source port |
| 7 | Dst port | Destination port |
| 8 | Proto | Protocol |
| 9 | TCP flags | CWR ǀ ECN ǀ URG ǀ ACK ǀ PSH ǀ RST ǀ SYN ǀ FIN flags |
| 10 | Payload size | Size of packet payload |
| 11 | MODBUS code | MODBUS function code |
| 12 | MODBUS value | MODBUS response value |
| 13 | num_pkts_src | Number of packets of the same source address in the last 2 seconds |
| 14 | num_pkts_dst | Number of packets of the same destination address in the last 2 seconds |

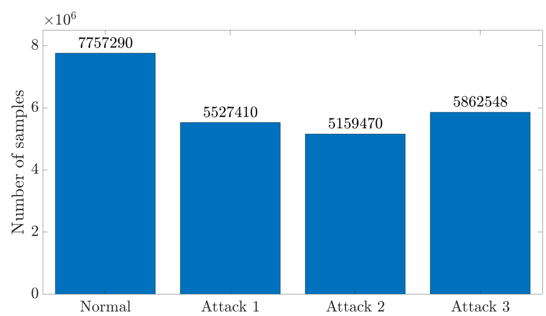



**FIGURE 10.** Number of samples for network dataset reported for each acquisition.

packet-based features, which help with the examination of packets payload in addition to the headers. This choice is justified by the presence of attacks that affect exclusively packets payload such as the MITM attack [35], [36]. Specifically, we analyzed the effectiveness and the applicability of the following features:

- *Src IP address*: Source IP address. In ICS networks, IP addresses are statically assigned; moreover, the number of hosts is static and well-defined. Thus, the appearance of new devices has to trigger an event.
- *Dst address*: Destination IP address. As for the source, also destinations in ICS networks are fixed and well known.
- *Src MAC address*: Source MAC address. Changes in MAC to IP mapping is very infrequent. Thus, the use of ARP messages to resolve MAC addresses of unknown IP addresses has to be notified. Changes in this feature could be the consequence of malfunctions or of ARP-poisoning MITM attack.
- *Dst MAC address*: Destination MAC address. As for the source, also the destinations are well-defined. Any unknown and additional MAC address indicates the presence of malicious hosts connected to the network.
- *Src Port*: Source port. In ICS networks, ports are standard and related to the configuration of hosts and to the protocols adopted.
- *Dst Port*: Destination port. As for the source, also the destination ports are static and well-defined. Unknown ports may indicate the use of protocols that are not allowed in the specific ICS network.
- *Proto*: Protocol. Protocols in ICS networks are limited and well-defined. Thus, the appearance of new protocols must be reported as a network modification.

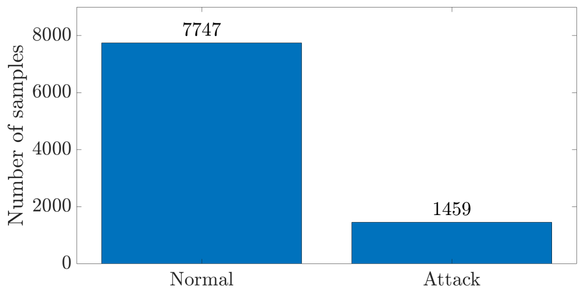



**FIGURE 11.** **Total number of samples divided into normal and malicious.**

- *TCP flags*. In general, TCP flags are used to indicate a specific state of a TCP connection. An attacker can vary these protocol settings in order to gather information on the networked devices as in the case of scanning attacks.
- *Payload size*. Packets exchanged in an ICS network are well-defined and without extra buffering in order to provide real-time requirements. Thus, anomalous packet size could be the consequence of malfunctions or malicious activity.
- *MODBUS code*: MODBUS function code. In MODBUS protocol, the code specifies the type of PLC memory address which is requested. Unusual read requests must be notified as a consequence of unauthorised PLC access.
- *MODBUS value*. Abnormal payload data could be the sign of misconfiguration or malicious actions such as MITM attacks. Changes in MODBUS values could cause a significant impact on the physical process.
- *num_pkts_src*: number of packets of the same source address in the last 2 seconds. In ICS networks, the number of connections between hosts is quite always static and constant. Any variation of this value may be the consequence of malfunctions and DoS or DDoS attacks. This feature captures an anomalous number of connections from one specific host.
- *num_pkts_dst*: number of packets of the same destination address in the last 2 seconds. This feature captures an anomalous number of connections towards one specific target.

Table 5 summarizes the list of network features we considered in our dataset. In addition to those already described, all the samples are identified by the date of acquisition.

### D. LABELLING

To label the samples for each acquisition, we used attack logs focusing in particular on the starting time, the ending time and the type of attack. Each record is characterized by two different labels: the first one defines the type of attack while the second one is either 0 if the record is normal and 1 if the record is attack. Figure 11 shows the total number of samples divided into normal and malicious.

### E. FINAL SHAPE OF DATASETS

We provide the two datasets in 8 different `csv` files. In particular, *attack_1*, *attack_2* and *attack_3* refer to normal and malicious network traffic while *phy_att_1*, *phy_att_2* and *phy_att_3* refer to the corresponding physical values of the WDT. Files *normal* and *phy_norm* refer to legitimate network and physical data.

Moreover, we provide raw network traffic packets in four *pcap* files: *attack_1.pcap*, *attack_2.pcap*, *attack_3.pcap* and *normal.pcap*).

The list of the events is defined in the file *README.xslx*.

### F. USE OF THE WDT DATASET

The WDT dataset is available at the link[1] and can be used free of charge for research and study applications (non-commercial activities) as long as it is reported in the bibliography with reference to this article.

## V. MACHINE LEARNING PERFORMANCE EVALUATION

As described in Section I, our dataset aims at supporting researchers in the validation of artificial intelligence and machine learning algorithms. In this section, we show some preliminary results by applying four different supervised machine learning algorithms to network and physical datasets.

### A. CLASSIFICATION TECHNIQUES

We adopted the following machine learning algorithms: K-Nearest-Neighbor (KNN), Naïve Bayes (NB), Support Vector Machine (SVM) and Random Forest (RF).

KNN is one of the simplest classifiers [37]. It is based on the distribution of training samples in the so-called feature space; a test sample is classified with the most represented class by the k-nearest training samples.

NB is a class of probabilistic classifiers based on the Bayes' theorem which requires the strong assumption of independence between the features. It computes the a-posteriori probability of samples to belong to one of the different classes knowing the likelihood of the features [38].

SVM is one of the best classifier algorithms [39]. It computes a separating hyperplane that divides samples belonging to the two classes in the best way.

RF is an ensemble learning classification algorithm [40]. It computes a predefined number of decision trees at training time; then it returns the most represented class by computing the mode of the classes for each individual tree.

### B. EVALUATION SETUP

Considering both network and physical data, samples from all four acquisitions were merged in order to obtain only two different datasets: one for network traffic and one for PLC data.

[1] https://ieee-dataport.org/open-access/hardware-loop-water-distribution-testbed-wdt-dataset-cyber-physical-security-testing

Before applying machine learning classifiers, we standardized and removed identical records. Specifically, we scaled all features by removing the mean and the variance in order to make data normally distributed. Then, we removed identical records in order to reduce possible biases towards the more representative classes. Datasets were divided into training and test sets using a K-Folds cross-validation. Feature standardization was performed on training set and, subsequently, the mean and variance of training data were used to normalize the test set.

Hyperparameters of classifiers were set as follows: k=10 for the KNN and 100 trees for RF. SVM was applied with a Radial Basis Function (RBF) Kernel and, for Naïve Bayes, the Gaussian version was used.

In order to implement KNN, RF, SVM and NB, we used the Python *Scikit-learn* library [41].

### C. PERFORMANCE METRICS

Performance of machine learning algorithms were computed with the following metrics:

- Accuracy: is the fraction of samples classified correctly

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{1}$$

- Recall: is the fraction of actual positive samples identified correctly

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

  where, TP = True Positive and FN = False Negative. In particular, we considered attack samples as positive and normal samples as negative.
- Precision: is the fraction of positive identifications predicted correctly.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

  where, FP = False Positive.
- F1-score: is the harmonic mean of precision and recall.

$$\text{F1-score} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{4}$$

### D. EVALUATION RESULTS

Table 6 summarizes results in terms of Accuracy, Recall, Precision and F1-score for both physical and network datasets. Regarding the physical dataset, we obtained performance close to 100% for both RF and KNN; while NB and SVM returned lower performance for Precision and Recall.

On the other hand, machine learning applied to the network dataset shows worse results. RF and KNN have better performance than those provided by SVM and NB; but, in all cases, the accuracy does not exceed 75%. Moreover, NB shows a poor ability to correctly detect true positive samples, as reported by the recall value which is less than 20%. On the contrary, SVM is prone to assign as anomalous the majority of samples, as reported by the recall which is close to 100% and the precision which is just 10%.

**TABLE 6. Machine learning evaluation results.**

| Algorithm | Performance metric | Physical dataset | Network dataset |
|---|---|---|---|
| **KNN** | Accuracy | 0.98 | 0.77 |
| | Recall | 0.95 | 0.44 |
| | Precision | 0.95 | 0.68 |
| | F1 score | 0.95 | 0.53 |
| **RF** | Accuracy | 0.99 | 0.75 |
| | Recall | 0.98 | 0.53 |
| | Precision | 0.95 | 0.56 |
| | F1 score | 0.97 | 0.54 |
| **SVM** | Accuracy | 0.93 | 0.69 |
| | Recall | 0.92 | 0.99 |
| | Precision | 0.64 | 0.10 |
| | F1 score | 0.75 | 0.20 |
| **NB** | Accuracy | 0.93 | 0.75 |
| | Recall | 0.92 | 0.15 |
| | Precision | 0.66 | 0.90 |
| | F1 score | 0.77 | 0.27 |

Finally, we can conclude that machine learning algorithms singularly applied to network dataset are not sufficient in order to separate malicious samples from normal samples acquired from an ICS network. This behaviour is linked to the intrinsic inability of network data to report the current state of physical process which is essential in order to identify deviations from the correct dynamics. Thus, taking into account the physical data from PLCs is necessary in order to properly recognize cyber attacks that have an impact against the physical process.

Moreover, as explained in Section III, our dataset provides some attack scenarios that have no influence on network traffic, as in the case of physical attacks. During these scenarios, network data have no information about the attack in progress resulting in the inability to recognize such attack.

On the other hand, we considered some cyber attacks, such as scanning attacks, which only affect network traffic. In these cases, physical data do not provide any discriminating features causing performance penalty.

Thus, in order to get better performance and in order to recognize all types of attacks, it is necessary to consider both the network and the physical data in the classification task.

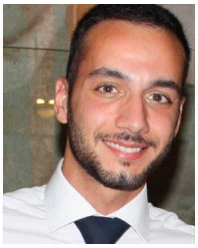
## VI. CONCLUSION

In this paper, we provided a new hardware-in-the-loop cyber-physical dataset obtained from a water distribution testbed. The testbed is composed of a real subsystem and a simulated one, which was used in order to add complexity by increasing the number of tanks, valves, pumps and PLCs for control. The dataset consists of both physical measurements and network traffic in order to overcome well-known limitations of the existing datasets providing enough complexity and a more realistic network traffic with modern attack scenarios. Physical data was extracted by using a Historian machine, while network traffic was captured using the *Wireshark* software. Such attacks were implemented in 28 different attack scenarios considering both the cyber and the physical attacks. Their effects against physical and network dynamics can vary depending on the time, the type of attack, the specific target and the current physical process. There are 41 features for the physical dataset and 14 features for the network one; in the latter, *Python* was used to extract and select features that best differentiate between normal and anomalous network packets.

Finally, we evaluated four machine learning algorithms, KNN, RF, NB and SVM, which were applied to both network and physical datasets. Results showed that classification algorithms cannot detect all the attacks types if they are applied separately on physical and network datasets. Thus, in order to get better performance, both the network and the physical data need to be considered.

## REFERENCES

[1] National Institute of Standards NIST and Technology. *Information Security*. Accessed: Apr. 15, 2021. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-3% 9.pdf
[2] N. Jazdi, "Cyber physical systems in the context of industry 4.0," in *Proc. IEEE Int. Conf. Autom., Qual. Test., Robot.*, May 2014, pp. 1–4.
[3] M. M. Rana, L. Li, and S. W. Su, "Cyber attack protection and control of microgrids," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 602–609, Mar. 2018.
[4] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 446–464, 1st Quart., 2017.
[5] M. R. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, "Applications of wireless sensor networks in the oil, gas and resources industries," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 941–948.
[6] J. Weiss "Industrial control system (ICS) cyber security for water and wastewater systems," in *Securing Water and Wastewater Systems*. Cham, Switzerland: Springer, 2014, pp. 87–105.
[7] *Threat Landscape for Industrial Automation Systems*, Kaspersky, 2019. Accessed: Sep. 1, 2021. [Online]. Available: https://ics-cert.kaspersky.com/media/KASPERSKY_H22019_ICS_REPORT_FINAL_EN.pdf
[8] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *Int. J. Elect. Power Energy Syst.*, vol. 99, pp. 45–56, Jul. 2018.
[9] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A survey of industrial control system testbeds," in *Secure IT Systems*, S. Buchegger and M. Dam, Eds. Cham, Switzerland: Springer, 2015, pp. 11–26.
[10] Q. Qassim, N. Jamil, I. Z. Abidin, M. E. Rusli, S. Yussof, R. Ismail, F. Abdullah, N. Ja'afar, H. C. Hasan, and M. Daud, "A survey of SCADA testbed implementation approaches," *Indian J. Sci. Technol.*, vol. 10, no. 7, pp. 1–8, 2017.
[11] J. Goh, S. Adepu, K. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security*. Cham, Switzerland: Springer, 2017, pp. 88–99.
[12] 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.
[13] D. Ding, Q.-L. Han, Y. Xiang, C. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, Jan. 2018.
[14] S. Ghosh and S. Sampalli, "A survey of security in SCADA networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.
[15] Kunal and M. Dua, "Machine learning approach to IDS: A comprehensive review," in *Proc. 3rd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Jun. 2019, pp. 117–121.
[16] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Proc. IEEE 15th Int. Symp. Intell. Syst. Informat. (SISY)*, Sep. 2017, pp. 000277–000282.
[17] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[18] S. D. Anton, S. Kanoor, D. Fraunholz, and H. D. Schotten, "Evaluation of machine learning-based anomaly detection algorithms on an industrial Modbus/TCP data set," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, New York, NY, USA, Aug. 2018, pp. 1–9.
[19] A Özgür and H Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Preprints*, vol. 4, Apr. 2016, Art. no. e1954v1.
[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.
[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.
[22] S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.
[23] E. Perkins. (2014). *Operational Technology Security Focus on Securing Industrial Control and Automation Systems*. [Online]. Available: https://blogs.gartner.com/earl-perkins/2014/03/14/operational-technology-security-focus-on-securing-industrial-control-and-automation-systems/
[24] S. Choi, J.-H. Yun, and S.-K. Kim, "A comparison of ICS datasets for security research based on attack paths," in *Critical Information Infrastructures Security*, E. Luiijf, I. Žutautaitÿe, and B. M. Hämmerli, Eds. Cham, Switzerland: Springer, 2019, pp. 154–166.
[25] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Proc. Int. Conf. Internet Things 4th Int. Conf. Cyber, Phys. Social Comput.*, Oct. 2011, pp. 380–388.
[26] L. Cazorla, C. Alcaraz, and J. Lopez, "Cyber stealth attacks in critical information infrastructures," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1778–1792, Jun. 2018.
[27] G. Bernieri, M. Conti, and F. Turrin, "Evaluation of machine learning algorithms for anomaly detection in industrial networks," in *Proc. IEEE Int. Symp. Meas. Netw. (M&N)*, Jul. 2019, pp. 1–6.
[28] G. Bernieri, E. E. Miciolino, F. Pascucci, and R. Setola, "Monitoring system reaction in cyber-physical testbed under cyber-attacks," *Comput. Elect. Eng.*, vol. 59, pp. 86–98, Apr. 2017.
[29] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A toolkit for security research on CPS networks," in *Proc. 1st ACM Workshop Cyber-Phys. Syst.-Secur. and/or PrivaCy*, Oct. 2015, pp. 91–100.
[30] J. Prusa, T. M. Khoshgoftaar, and N. Seliya, "The effect of dataset size on training tweet sentiment classifiers," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 96–102.
[31] M. Endi, Y. Z. Elhalwagy, and A. Hashad, "Three-layer PLC/SCADA system architecture in process automation and data monitoring," in *Proc. 2nd Int. Conf. Comput. Automat. Eng. (ICCAE)*, vol. 2, Feb. 2010, pp. 774–779.
[32] *Modicon Modbus Protocol Reference Guide*. Pi–mbus–300 rev. j. Accessed: Mar. 16, 2021. [Online]. Available: https://modbus.org/docs/PI_MBUS_300.pdf
[33] M. Mantere, I. Uusitalo, M. Sailio, and S. Noponen, "Challenges of machine learning based monitoring for industrial control system networks," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 968–972.
[34] M. Mantere, M. Sailio, and S. Noponen, "Network traffic features for anomaly detection in specific industrial control system network," *Future Internet*, vol. 5, no. 4, pp. 460–473, Sep. 2013.
[35] P. Gogoi, H. M. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Packet and flow based network intrusion dataset," in *Contemporary Computing*, M. Parashar, D. Kaushik, O. F. Rana, R. Samtaney, Y. Yang, and A. Zomaya, Eds. Berlin, Germany: Springer, 2012, pp. 322–334.
[36] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 343–356, 3rd Quart., 2010.
[37] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN model-based approach in classification," in *On The Move to Meaningful Internet Systems* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2003.
[38] I. Rish, "An empirical study of the naive Bayes classifier," in *Proc. IJCAI Workshop Empirical Methods Artif. Intell.*, 2001, pp. 41–46.
[39] S. W. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, pp. 1565–1567, 2006.
[40] L. Breiman, "(IMPO )random forests(book)," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.



**L. FARAMONDI** received the Laurea and Ph.D. degrees in computer science and automation from Roma Tre University, Rome. He is currently an Assistant Professor Fellow with the Complex Systems and Security Laboratory, University Campus Bio-Medico of Rome, in 2013 and 2017, respectively. He is involved in several national and European projects about the critical infrastructure and indoor localization. His research interests include the identification of network vulnerabilities, cyberphysical systems, and optimization at large. He has been a member of the IEEE SMC Technical Committee on Homeland Security, since 2017, and IEEE RAS Technical Committee on Digital Manufacturing and Human-Centered Automation, since 2020. He won the IEEE Systems, Man, and Cybernetics Society TCHS Young Researcher Award, in 2021.

**F. FLAMMINI** (Senior Member, IEEE) was born in Formia, Italy, in 1978. He received the master's degree *(cum laude)* in computer engineering and the Ph.D. degree in computer and automation engineering from the University of Naples Federico II, Italy, in 2003 and 2006, respectively. From 2003 to 2007, he was a Software V&V Engineer with Ansaldo STS (now Hitachi Rail), where he was a Senior Innovation Engineer, from 2007 to 2016. From 2016 to 2017, he worked as an Information Security Compliance Manager with the Italian State Mint and Polygraphic Institute. Since 2018, he has been working as an Associate Professor with Linnaeus University, Sweden, where he has chaired the cyber-physical systems (CPS) environment. Since 2020, he has also been a Professor of computer science with Mälardalen University, Sweden. He had leadership roles in more than ten research projects. He has edited or authored more than ten books and 130 publications. His research interests include resilient cyber-physical systems and trustworthy autonomy. He is an ACM Distinguished Speaker and the Chair of the IEEE SMC Technical Committee on Homeland Security.

**S. GUARINO** (Graduate Student Member, IEEE) received the B.S. degree *(cum laude)* in industrial engineering and the master's degree *(cum laude)* in biomedical engineering from the University Campus Bio-Medico of Rome,in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the Complex Systems and Security Laboratory. His scientific research interests include prevention, identification, and mitigation of cyber-attacks against SCADA systems.

**R. SETOLA** (Senior Member, IEEE) received the Laurea degree in electronic engineering and the Ph.D. degree in control engineering from the University of Naples Federico II, in 1992 and 1996, respectively. He was responsible for the Italian Government Working Group on Critical Information Infrastructure Protection (CIIP) and a member of the G8 Senior Experts' Group for CIIP. He is currently a Full Professor with the University Campus Bio-Medico of Rome, where he directs the Automation Research Unit and the Master Program in homeland security. He has been the coordinator of several EU projects. He has authored nine books and more than 250 scientific papers. His main research interests include simulation, modeling and control of complex systems, and critical infrastructure protection.