A Survey on Industrial Control System Testbeds and Datasets for Security Research

Mauro Conti, *Senior Member, IEEE*, Denis Donadel, and Federico Turrin

***Abstract*—The increasing digitization and interconnection of legacy Industrial Control Systems (ICSs) open new vulnerability surfaces, exposing such systems to malicious attackers. Furthermore, since ICSs are often employed in critical infrastructures (e.g., nuclear plants) and manufacturing companies (e.g., chemical industries), attacks can lead to devastating physical damages. In dealing with this security requirement, the research community focuses on developing new security mechanisms such as Intrusion Detection Systems (IDSs), facilitated by leveraging modern machine learning techniques. However, these algorithms require a testing platform and a considerable amount of data to be trained and tested accurately. To satisfy this prerequisite, Academia, Industry, and Government are increasingly proposing testbed (i.e., scaled-down versions of ICSs or simulations) to test the performances of the IDSs. Furthermore, to enable researchers to cross-validate security systems (e.g., security-by-design concepts or anomaly detectors), several datasets have been collected from testbeds and shared with the community. In this paper, we provide a deep and comprehensive overview of ICSs, presenting the architecture design, the employed devices, and the security protocols implemented. We then collect, compare, and describe testbeds and datasets in the literature, highlighting key challenges and design guidelines to keep in mind in the design phases. Furthermore, we enrich our work by reporting the best performing IDS algorithms tested on every dataset to create a baseline in state of the art for this field. Finally, driven by knowledge accumulated during this survey's development, we report advice and good practices on the development, the choice, and the utilization of testbeds, datasets, and IDSs.**

***Index Terms*—Cyber-physical systems, industrial control systems, security, intrusion detection systems, dataset, testbed.**

I. Introduction

CRITICAL infrastructures and the emerging Industry 4.0 are increasingly using more advanced technologies such as computers, electrical and mechanical devices to monitor the physical processes. The networks resulting from smart computing integration for the processes monitoring are called Industrial Control Systems (ICSs) or, sometimes, SCADA systems.

ICSs are composed of two macro areas. The Operational Technology (OT) network includes hardware and software used to monitor and manage industrial equipment, assets, processes, and events (e.g., Programmable Logic Controllers (PLCs), sensors, actuators). On the other side, the traditional Information Technology (IT) network contains workstations, databases, and other classical machines used to manipulate information. IT and OT networks were originally disconnected. However, due to the so-called IT/OT Convergence [1], the two networks have been interconnected to facilitate the digitization of processes, opening new vulnerability surfaces.

For Cyber-Physical Systems (CPSs), which also contains ICSs, the classical CIA triad (Confidentiality, Integrity, Availability) is considered reversed, in order of importance, as Availability, Integrity, and Confidentiality [2], [3]. In this context, reliability becomes the most critical request since, differently from IT systems where the main concerns are about the confidentiality of the data, for an ICS instead, the availability is fundamental since it can guarantee human safety and fault tolerance [4]. For instance, in a nuclear plant environment, data availability (e.g., the temperature of the core) is more important than its confidentiality [5].

Since these systems control physical and sometimes dangerous processes, security is a fundamental need. However, in recent years several viruses attempting ICSs were identified. One of the first cyberattacks targeting SCADA systems dates back to 1982 [6] when a trojan targeting the Trans-Siberian pipeline causes a massive explosion. In successive years, many incidents exposed the security weaknesses of ICSs. Stuxnet [7], [8] is probably the most famous malware discovered in this field. Stuxnet was a worm discovered in 2010 targeting Programmable Logic Controllers (PLCs) used in gas pipeline and power plants. It was able to cause self-destruction of 984 centrifuges in a uranium-enrichment plant in Iran. In 2014, the third version of a known trojan family, BlackEnergy [9], was developed to target ICSs. In the following years, this trojan was spread mainly inside a Microsoft Word document that, once open, request to activate macros that hide the virus. Victims of these attacks are media and energy companies, mining industries, railways, and airports in Ukraine. On December 23, 2015, an attack employing BlackEnergy3 caused a three-hour disconnection of 30 substations in the Kyiv Power Distribution company, leading to several hours of blackouts in the area. More recently, in 2017, another important malware, TRITON [10], was identified after an unscheduled shutdown of a Saudi Arabian petrochemical processing plant. TRITON reprograms some special PLCs used for safety purposes, causing them to enter a failed state.

According to a report of Kaspersky Lab [11], in the second half of 2016 the 39.2% of the industrial machines secured

Manuscript received February 10, 2021; revised May 27, 2021; accepted June 28, 2021. Date of publication July 2, 2021; date of current version December 8, 2021. The work of Federico Turrin was supported in part by the Cariparo Foundation and in part by Yarix S.r.l. *(Corresponding author: Federico Turrin.)*

The authors are with the Department of Mathematics, University of Padova, 35131 Padua, Italy (e-mail: conti@math.unipd.it; denis.donadel@studenti.unipd.it; turrin@math.unipd.it).

Digital Object Identifier 10.1109/COMST.2021.3094360

by Kaspersky's products have been attacked, a clear sign that threats to ICS are a growing problem nowadays. The vulnerabilities affecting these systems are also reported in recent studies on real ICS traffic over the Internet [12], [13], showing a dramatic lack of security features on the communication.

A successful attack on ICS implied a huge economic impact on the organization. These consequences include operational shutdowns, damage to the equipment, business waste, intellectual property fraud, and significant health and safety risks. Nozomi Network reports that known shutdown events of an ICS [14] due to an attack cost from 225K$ up to 600M$. An increasing attack trend against ICS is Ransomware, which aims to obtain economic rescue [15]. According to Coveware [16], in Q4 of 2019, the average ransom payment increased by 104% to 84,116$, up from 41,198$ in Q3 of 2019. One of the most recent Ransomware is EKANS, which was discovered targeting 64 ICS [17].

To prevent such catastrophic events, it is fundamental to implement novel security-by-design approaches, and where it is impossible to apply them, prevention or mitigation techniques must be integrated. However, to develop a new security-by-design concept, it is required a complete testing infrastructure. Generally, researchers rely on scaled-down versions of a real ICS, created ad-hoc to reproduce real-world systems but in a controlled environment, called *testbed*. Testbeds could be based on physical devices to provide reliable data at the cost of being more expensive or virtual if the application does not require exact measures. However, the development of a new testbed is not straightforward, instead it is challenging from different points of view, ranging from implementation costs, sharing capability, and fidelity (Section VI).

To develop prevention and mitigation techniques, nowadays, researchers involve machine learning techniques that exploit big amounts of data to train classification algorithms to detect misbehavior or potential attacks. The straightforward approach to collect data is to record and provide to researchers data from real ICSs. However, since these systems are generally critical and fundamental for society, this strategy can be challenging in many aspects. For instance, it is difficult, if not impossible, to deploy attacks in a real environment because they can damage the physical process or some devices. Moreover, privacy is a problem: private companies could be reluctant to share system data from their ICSs. In fact, disclosing this data can cause intellectual property theft or reveal the infrastructure's vulnerabilities, attracting malicious attackers' attention. From a testbed, it is possible to generate data and share them with other researchers to compare and improve different detection algorithms' results. These captures are called datasets and can be composed of physical measures (i.e., data from OT sensors) and/or network traffic (i.e., data from network communications). Datasets are an excellent testing solution due to their simplicity and availability. However, they are also challenging from many points of view, for instance, in the generation process and lack modularity (Section VII).

A. Contribution

In this paper, we present a comprehensive survey specifically targeting the security research platform in the ICS field. This work aims to collect all the information related to the testbed and dataset to support future research and studies on this sector. Furthermore, for each dataset identified, we report the best score achieved by an Intrusion Detection System (IDS) in terms of F1-Score, Accuracy, and Precision, which are the most common metrics. We have accurately analyzed all the testbeds, datasets, and IDSs to provide the readers with an exhaustive overview of the current ICS state of the art. The paper aims to assist interested readers: (i) to discover the different testbeds and datasets which can be used for security research in ICS with a description of the design key points, (ii) to have a clear baseline when developing an IDS on a particular dataset, and (iii) to understand the challenges and the good practices to keep in mind when designing an ICS testbed or dataset.

We summarize our main contributions as follows:

- We provide a comprehensive background on ICS, which offers an overview on the reference architecture and the main components characterizing such systems;
- We present the most employed industrial communication protocols with a particular focus on the intrinsic security features and proposed security expansions of each protocol;
- We provide an exhaustive overview of the current ICS state of the art by analyzing different testbeds, dataset, and IDS related to the ICS field available on literature to provide the reader with key points design concepts;
- We offer the reader an exhaustive survey of the different testbeds and datasets which can be used for security research in ICS;
- We describe the best performing IDS developed for the presented datasets. During the development of this work, we noted a lack of a defined methodology to test the detection frameworks (e.g., testing the IDS on the single attacks or the whole dataset) and a defined baseline to compare the developed IDS. We believe that this baseline can offer a starting point for future researchers to begin working on ICS security having a clear idea of the current state of the art direction and trend;
- Finally, we offer a review of the challenges and the good practices to keep in mind when designing an ICS testbed, dataset, or IDS, with some insight into the future directions useful to fill the field gaps.

To continue collecting the testbeds and datasets in the future and sharing them with the community, we also developed a website (Section VI) to support the resource sharing among the researchers in this field.

B. Survey Organization

The remainder of this paper is organized as follows. In Section II we provide an overview of the previous survey on this field, highlighting how we differ from them. In Section III we provide background on Industrial Control Systems describing the reference architecture and the common devices employed. In Section IV we describe the most typical protocols for ICS communication, highlighting the main characteristics, security features and offering an analysis of their diffusion in the market. In Section V we briefly recall the

concept of Intrusion Detection System, also describing conventional attacks implemented on ICSs. Then, in Section VI and Section VII we describe and analyze respectively the different testbeds and datasets present in the literature. In Section VIII, some advice and good practices are illustrated both for researchers that use these technologies both for institutions that want to create brand new datasets or testbeds. Finally Section IX concludes the paper.

II. Related ICS Surveys

Literature includes different surveys comparing testbeds and datasets created for applications in the ICS field. However, to the best of our knowledge, no detailed analysis gathers and describes both the ICS datasets and testbeds, but also the main IDSs implemented on them. For every dataset, we also report the algorithm with the best performances and the most interesting and innovative detection approaches. We believe that this could be useful for future research in this field and set a baseline to compare the different detection results.

In 2015, Holm *et al.* [18] proposed a complete revision of several papers related to ICS testbeds. The authors then focused on the objective and the component's implementation of 30 different testbeds. Furthermore, the authors provided an analysis of each testbed's main requirements (i.e., fidelity, repeatability, measurement accuracy, and safe executions). The paper's main scope was to provide an overview of the actual state of such systems' development without detailing each single testbed composition. In fact, except for a table that indicates each testbed's location, all the others show only aggregated information. Moreover, since the paper is not recent, some of the presented testbeds are quite old and not widely used nowadays, while others that were only designed have never been made (e.g., [19]).

McLaughlin *et al.* [20], in 2016, presented a complete survey on state of the art in ICS security. The paper briefly introduces the ICS operation's key principles and the history of cyberattacks targeting ICSs. The authors then addressed the vulnerability assessment process, outlining the cybersecurity assessment strategy advised for ICS and providing a list of steps to study the security and the vulnerabilities of an industrial system. In the end, the authors focused on new attacks and mitigation techniques. Moreover, the paper briefly presents a small list of some testbeds which can be used for security research in this field. Differently, our work is less focused on providing a complete landscape on ICS security, while it offers a more in-depth analysis and review of all the most used testbeds and datasets for ICS security research.

In 2017, Cintuglu *et al.* [21] presented a comprehensive survey focused on smart grid testbeds, providing a systematic study with a particular focus on their domains, research goals, test platforms, and communications infrastructures. There are some intersections between smart grid and ICS fields, such as some used components and protocols. Nevertheless, some different concepts require a separate ICS analysis, like the specific applications and sensors used, combined with the complexity of smart grids. To classify smart grid testbeds, the authors provide different possible taxonomy. Some of them can be applied to the entire ICS field (e.g., platform type). Instead, some others are specific to Smart Grid (e.g., NIST grid domain). The employed testbed classification in [21] is mainly based on the research area, which motivates the development of each system. In this work, instead, we classify testbed mainly based on the platform type, providing the reader an overview of the most suited ICS testbeds and datasets for his research.

Recently in 2019, Geng *et al.* [22] presented a survey on ICS testbeds based on the same four requirements of [18]. Besides analyzing different ICS datasets, the authors also present the different techniques that can be employed to build a testbed, including application scenarios, the main challenges, and future development directions. However, the authors' only introduced an analysis of each testbed's structure without going into details or providing comparison tables.

In the same year, Choi *et al.* [23] gathered and analyzed datasets for ICS security research, providing different comparison tables to understand the most suitable dataset depending on the case study. The authors based the comparison on the attack vector strategy. The paper includes 11 commonly used datasets. Some existing datasets not widely used or without attacks data are intentionally not considered. However, even if not suited for anomaly detection tasks, the latter could be useful to study the ICS environment's behavior. Also, one of the presented dataset (i.e., DEFCON23) is no longer available. Gosh and Sampalli [24] in 2019 presented a survey that classifies the security threats and the existing security schemes in industrial systems. Particular interest was given to the threats and the corresponding security measures related to Quantum Computing. However, differently from our work, the survey [24] does not focus on the testbeds and datasets available in the literature.

In 2020, in [25] the authors present an exhaustive survey with guidelines and good practices to help the building of an ICS testbed, highlighting the main challenges and the results of a focus group involving security experts to identify relevant design factors and guidelines. In the same years, Green *et al.* published another survey in this direction [26] with interesting guidelines for each ICS layer and a set of characteristics to consider when outlining testbed objectives, architecture, and evaluation process. While these works are interesting and give a comprehensive insight into the process of designing and evaluating a testbed, they do not consider datasets and IDSs, their requirements, and relationships.

Another interesting survey presented in 2020 by [27] discusses the transition of ICS stand-alone systems to cloud-based environments. The authors presented in detail the benefits, the main security challenges, and different case studies related to cloud-based ICSs. However, differently from our work, the authors mainly focused on reviewing the current ICS cloud transition state. Instead, we are interested in providing an overview of the testbed and datasets designed for security research. Pliatsios *et al.* [28] in 2020 present an exhaustive survey on SCADA systems with a particular focus on the threats and vulnerabilities rising from the insecure design of the industrial protocols. The authors presented in detail the security issues of the protocols in terms of CIA Triad and how an

attacker can exploit such vulnerabilities. However, few spaces on the survey were dedicated to the testbed, while dataset and the most recent Intrusion Detection techniques were not discussed.

Differently from previous works, our survey aims to collect all the platforms (i.e., testbeds and datasets) useful for ICS security research. We base the existing literature to provide a detailed analysis of the current research issues, challenges, and future directions characterizing this field. We also report the best performance of the IDS on every dataset, which can be helpful for future IDS research baseline.

III. Industrial Control Systems

In this section, we offer a background on ICSs, useful when start approaching this field and to understand the remainder of this paper. Firstly, in Section III-A, we focus on the architecture of such systems compared to the classical systems architecture. Then we present a summary of the widely used ICS components in Section III-B.

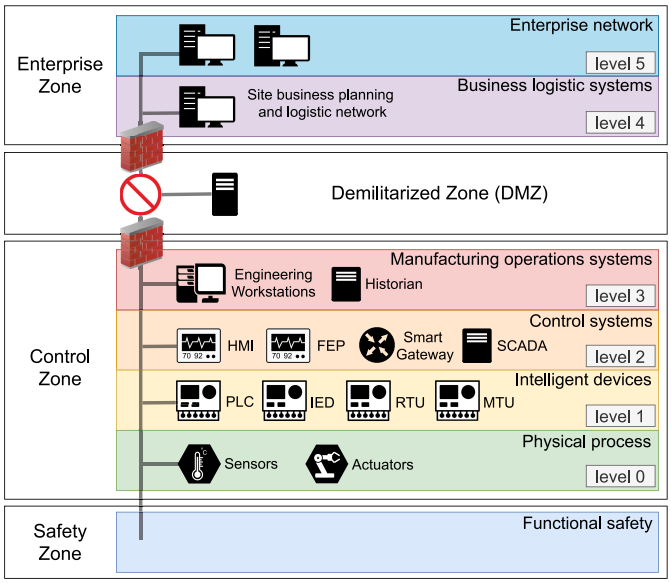


Fig. 1. ICS Purdue Model architecture and corresponding level of the different industrial devices.

A. ICS Architecture

Industrial Control Systems (ICSs) are composed of the interconnection of different computers, electrical and mechanical devices used to manage physical processes. These systems are usually very complex and include heterogeneous hardware and software components such as sensors, actuators, physical systems and processes being controlled or monitored, computational nodes, communication protocols, Supervisory Control And Data Acquisition (SCADA) systems, and controllers [29]. Control can be fully automated or may include a human in the loop that interacts via a Human Machine Interface (HMI). ICSs are widespread in modern industries (e.g., gas pipeline, water treatments) and critical infrastructures (e.g., power plant and railway).

Unlike classical IT systems, ICSs are composed of standard network traffic over TCP/IP stack and data from physical processes and low-level components. This interconnected and intertwined nature can open a wide space for new generation attacks exploiting new vulnerabilities surfaces. Several protocols are used in ICS, based on the specific purpose of each system. Industrial protocols are specifically designed to deal with real-time constraints and legacy devices in an air-gap environment. Many protocols do not implement any encryption or authentication mechanism due to these constraints, opening several vulnerabilities surfaces. Moreover, sometimes, the industrial protocols are customized from the company opening, again, many documentation and vulnerability issues.

The reference architecture of the ICS is the Purdue Model [22], [30]. As depicted in Figure 1, the Purdue module divides an ICS network into logical segments with similar functions or similar requirements:

1) *Enterprise Zone*, or IT network, includes the traditional IT devices and systems such as the logistic business systems and the enterprise network.
2) *Demilitarized Zone (DMZ)* controls the exchange of data between the Control Zone and the Enterprise Zone, managing the connection between the IT and the OT networks in a secure way;
3) *Control Zone*, sometimes also referred to as OT network, includes systems and equipment for monitoring, controlling, and maintaining the automated operation of the logistic and physical processes. It is divided into four sub-levels:
   - *Level 0* includes sensors and actuators that act directly on the physical process;
   - *Level 1* includes intelligent devices such as PLC, Intelligent Electronic Device (IED), and Remote Terminal Units (RTU);
   - *Level 2* includes control systems such as Human Machine Interfaces (HMI), alarms, and control room workstations;
   - *Level 3* includes manufacturing operation systems that are often responsible for managing control plant operations to produce the desired end product;

   Level 2 and Level 3 devices can communicate with the Enterprise Zone through the DMZ.
4) *Safety Zone* includes devices and systems for managing ICS security by monitoring for anomalies and avoiding dangerous failures;

The role of the DMZ is to filter the internal communication of the network. In fact, according to the Purdue model, all the traffic Exchanged between OT and IT networks must pass through the DMZ. However, this is rarely respected in the real-world, mainly due to the implementation difficulty or, more generally, the companies' insufficient attention to the industry building phase's security aspects. This condition exposes the critical part of the system (i.e., OT network) to potential attacks.

Compared with the classical IT environment, ICSs need a different risk handling strategy. The reliability is fundamental, and outrages are not tolerated due to the critical nature of the

processes monitored, unlike IT, where occasional failures are acceptable. The risk impact is also different: in the IT environment, the principal risk is the compromising of privacy and confidentiality (e.g., loss or unauthorized alteration of data). Instead, in the OT environment, a data compromise can cause a loss of production, equipment, and, in the worst case, a loss of lives or environmental damage.

Another difference with respecting traditional IT systems relies on information handling performance: in an IT environment, the throughput must be high enough, while delays and jitter are accepted. On the other hand, in the industrial field, communication is defined with a regular polling time. Generally, this polling time is in second or millisecond orders, but delays are serious concerns. Finally, in IT systems, recovery can be made by rebooting. In contrast, in the OT system, fault tolerance is essential since a reboot would imply shutting down the entire industry and can lead to enormous economic losses [19].

For all these reasons, and considering that nowadays most of the ICS are connected with the Enterprise zone, it is essential to protect them using new and precise technologies.

B. ICS Components

Industrial Control Systems are composed of a wide range of heterogeneous devices and components with a specif role in the system. In this section, we briefly introduce the most common devices in the ICS fields. These devices are generally installed or simulated in the testbed to replicate the ICS environment. We reported in Figure 1 the level of the Purdue Model on which each device is installed.

Programmable Logic Controller (PLC): PLC is a microprocessor-controlled electronic device that reads input signals from sensors, executes programmed instructions using these inputs and orders from supervisory controllers, and creates output signals that may change switch settings or move actuators. PLC is generally the boundary between the OT network and the physical process. It is often rugged to operate in critical environmental conditions such as very high or low temperature, vibration, or in the presence of big electromagnetic fields. As with most ICS components, PLCs are designed to last more than 10-15 years in continuous operations. The Real-Time Operations System (RTOS) installed in each PLC makes it suited for critical operations. The time to read all inputs, execute logic, and write outputs must last only a few milliseconds. A PLC has a power supply, central processing unit (CPU), communications interface, and analog/digital input/output (I/O) modules that can be connected to sensors (input) or actuators (outputs). These components are generally connected to the local network to communicate with supervisory processes. Based on the manufacturing company and the user requests, these communications can happen through different mediums (e.g., serial, fiber optic, wireless) using different protocols. Modern PLCs can use a UNIX-derived micro-kernel and present a built-in Web interface that makes the management more simple but exposing the device to new vulnerabilities. The most common practice to program PLCs is the ladder logic programming language, which allows programming the hardware logic of the PLC through graphical blocks representing the internal logic circuits of the device.

Remote Terminal Unit (RTU): An RTU is a microprocessor-controlled electronic device. Like PLC, it is designed for harsh environments and is generally located far from the control center, for instance, in voltage switch-gear. There are two types of RTUs: station and field RTUs. Field RTU receives input signals from field devices and sensors and then executes programmed logic with these inputs. It gathers data by polling the field devices/sensors at a predefined interval. It is an interface between field devices/sensors and the station RTU, which receives data from field RTUs and orders from supervisory controllers. Then station RTU generates outputs used to control physical devices like actuators. Both field and station RTU has a power supply, CPU, and digital/analog I/O modules. For communication with the control center, RTU uses WAN technologies such as satellite, microwave, unlicensed radio, cellular backhaul, GPRS, ISDN, POTS, TETRA, or Internet-based links.

Intelligent Electronic Device (IED): An IED is a device containing one or more processors that can receive or send data from an external source. Examples of IEDs are electronic multi-function meters, digital relays, and controllers. Thanks to the higher complexity compared to a PLC and RTU, IED can perform more operations. An IED can be used for protection functions like detecting faults at a substation or for control functions such as local and remote control of switching objects and provide a visual display and operator controls on the device front panel. Other functions can be related to monitoring (for instance, a circuit breaker condition), metering (e.g., tracking three-phase currents), and communications with supervisory components.

Engineering Workstation: The Engineering Workstation is generally a desktop computer or server running a standard operating system hosting the various software for controller and applications. Engineers use this platform to manage the controllers.

Human Machine Interface (HMI): The HMI is a software installed on desktop computers, tablets, smartphones, or dedicated flat panel screens that permit operators to check and monitor the automation processes. As illustrated in Figure 2, the HMI shows the state of a plant operator, such as process values, alarms, and data trends. An HMI can monitor multiple process networks and several devices. An operator can use the HMI to send manual commands to controllers, for instance, to change some values in the production chain. Generally, the HMI shows a diagram or plant process model with status information to facilitate such a job.

Data Historian: A Data Historian is a software application used to collect real-time data from the processes and aggregate them into a database for analysis. Data Historian mainly collects the same information shown in an HMI. The database and the hardware, generally a desktop workstation or a server, is designed for a very fast ingest of data without dropping data and uses industrial interface protocols.

Front End Processor (FEP): The FEP is a dedicated communications processor used to poll status information from

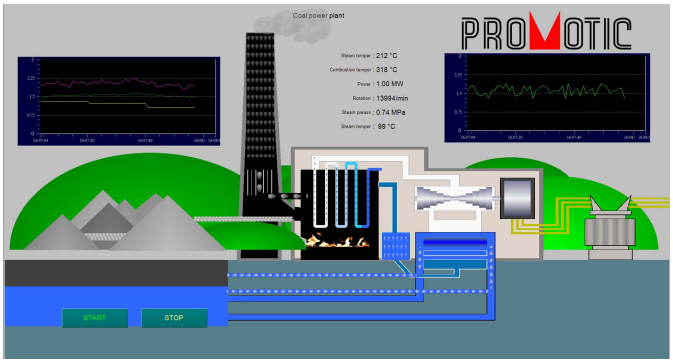


Fig. 2. An example of HMI interface generated with Promotic Open-Source Tool [31].

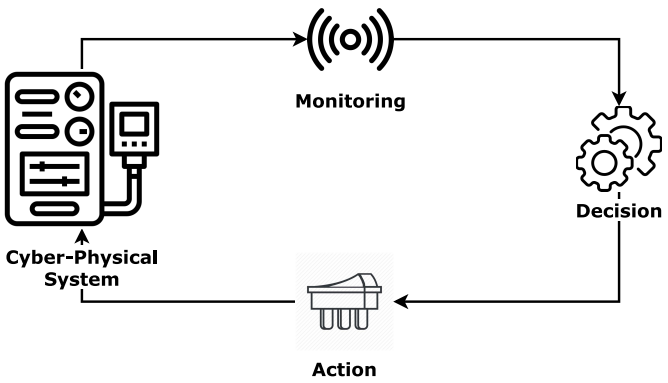


Fig. 3. The CPS close loop.

multiple devices to give operators the possibility to monitor the system's overall status.

Communications Gateways: A Communications Gateway is essential to make communications possible between devices from different manufacturers that use different protocols. Gateways can translate packets from a sending system to the receiver protocol.

Master Terminal Unit (MTU): MTUs manage the communication with the RTU or the PLC, gathers data from the PLCs, and process them. The communication between the MTU and the PLCs is bidirectional, but only the MTU can initiate the communication. Therefore, the MTU uses a master-slave communication where the MTU is the Master and PLCs are the slaves. Messages from the MTU to the PLCs can be triggered by an operator or be automatically triggered. These messages can either read memory parts representing current values like water flow, oil pressure, the temperature of a tank, or either write values in the memory and modify the configuration.

Supervisory Control and Data Acquisition (SCADA): SCADA devices are placed on the higher level of the ICS hierarchy and are used to monitor and control centralized data acquired from different field sites. Furthermore, they manage the communication between the various devices and represent the remote connection point for the remote operators with the OT network. Over the year, SCADA systems protocols moved from proprietary standards towards open international standards, resulting in attackers knowing precisely the protocols. That is why there is a gain of interest in reinforcing industrial control systems security.

ICS Field Devices: Field devices include all the components that are in direct contact with the physical process. The controllers can use them to get information regarding the physical process (e.g., the measure of temperature or pressure using sensors). Instead, actuators can interact with a physical process following commands from a controller (e.g., control motors, pumps, valves, turbines, agitators). The communication with the controllers is generally performed via I/O modules. Field Devices are implemented in the so-called CPS closed-loop to perform the three CPS main functions: monitoring using sensors, making decisions using PLCs, and applying actions using actuators. These three functions operate within a feedback loop covering, as shown in Figure 3.

To sum up, controllers such as PLC, RTU, and IED are mainly used to interact with the Field Devices that can instead directly operate on the processes. HMI, Front End Processor, Engineering Workstation, and Data Historian are used to control and manage the system data. Instead, SCADA and Gateways Communications are used [32] to set up all the connections between different components.

IV. Industrial Protocols Security

With the growth of the ICSs, several new protocols have been developed to support the specific requirements of the OT environment, like fault tolerance and reliability. The majority of these protocols were designed to operate in an air-gapped environment. Therefore originally, less importance was given to the security aspects with respect to the real-time constraint. Some of them have no security features at all (e.g., Authentication, Encryption). However, after the IT and OT convergence, they have still been used in practice [12].

This section reports the main industrial protocols focusing on the security properties initially and currently implemented. Standards like PowerLink Ethernet, EtherCAT, Constrained Application Protocol (CoAP), Message Queue Telemetry Transport (MQTT), ZigBee PRO, WirelessHART, or ISA100.11.a are not used in the datasets and testbeds identified in this paper. Therefore we decided not to report them. However, some of these protocols are widely used in different fields, for instance, in Industrial Internet of Things (IIoT) scenarios.

Table I summarized the main information related to the protocols presented in this section. It includes:

- Name of the **Manufacturer**;
- Standard **Ports** according to IANA [33];
- Information related to the **Original** protocol:
  - **Name** of the protocol;
  - **Year** of release;
  - A tick if **Encryption**, **Integrity**, or **Authentication** are available;
- Information related to the **Enhancement** version with security measures offered by the manufacturer:
  - Name of the new **Version** of the protocol;
  - **Year** of release;

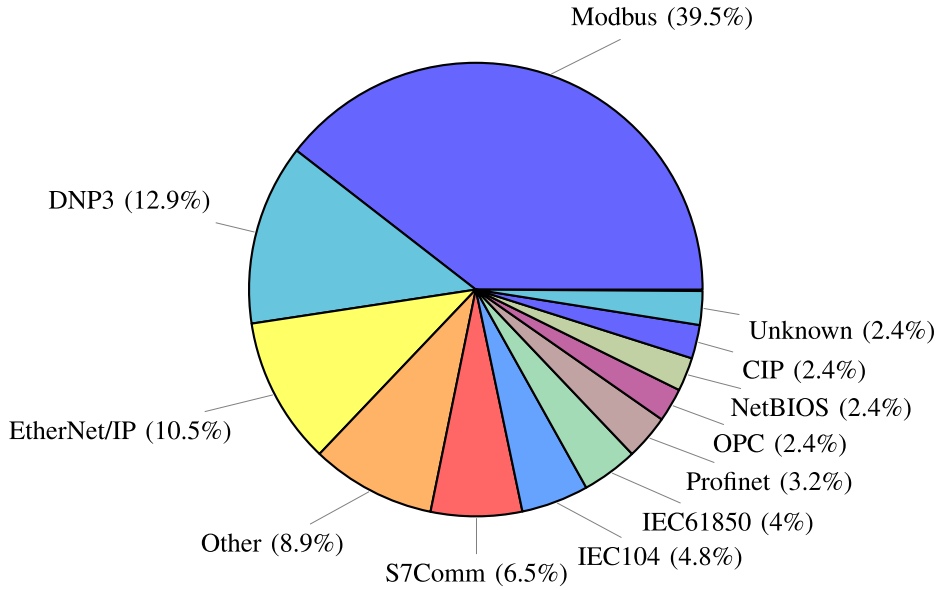


Fig. 4. Percentage distribution of different protocols in the datasets and testbeds analyzed.

TABLE I
SUMMARY OF THE MAIN PROTOCOL'S CHARACTERISTICS AND THEM SECURITY MEASURES IMPLEMENTED IN THE ENHANCEMENT EXTENSIONS. IN PARTICULAR, THE TABLE SHOWS **E**: ENCRYPTION; **I**: INTEGRITY; AND **A**: AUTHENTICATION. THE ENHANCEMENT PART REFERS TO THE VERSION PROPOSED BY THE MANUFACTURER

		Original					Enhancement				
Manufacturer	**Ports**	**Name**	**Year**	**E**	**I**	**A**	**Version**	**Year**	**E**	**I**	**A**
Schneider Electric	502/802	Modbus	1979				Modbus/TCP Security	2008	✓	✓	✓
GE Harris	20000	DNP3	1990			✓	DNP3-SA	2020	✓	✓	✓
Siemens	102	S7Comm	1994				S7CommPlus	2014	✓		✓
PROFINET Int.	-	PROFINET	2003				PROFINET Security Classes	2019	✓	✓	✓
ODVA	44818	EtherNet/IP	2000				CIP Security	2015	✓	✓	✓
OPC Foundation	4841	OPC	1996				OPC-UA	2006	✓	✓	✓
IEC	2404	IEC 104	2000				IEC 62351	2007	✓	✓	✓
IEC	102	IEC 61850	2003				IEC 62351	2007	✓	✓	✓

– A tick if **Encryption**, **Integrity**, or **Authentication** are available.

A. Industrial Protocols

Modbus: Modbus is a serial communication protocol initially published by Modicon (now Schneider Electric) in 1979 for use with its programmable logic controllers (PLCs). Today Modbus [34] is one of the most used and famous protocols in the ICSs. During the years, various versions of Modbus have been released. The first version was thought for serial communications, allowing to establish asynchronous serial communications on RS-232 and RS-485 interface. Modbus is also adapted to transmission means other than copper, such as optical fiber and radio links. A typical communication via Modbus consists essentially of three stages: the formulation of a request from one device to another, the execution of the actions necessary to satisfy the request, and the resulting information's return to the initial device. This approach's main advantage lies in the interaction mode between the various network nodes: being a client-server type, each server device can exchange data simultaneously with more than one client. The Modbus/TCP variant is substantially identical to the original serial version, but with the addition of a TCP/IP encapsulation module.

Security: Implementations of serial Modbus use both RS232 and RS485, which are physical layer communication protocols. It makes no sense to speak of security on this layer, as these are functionalities developed on higher layers. Modbus was designed to be used in environments isolated from the Internet regarding the application layer's security. Therefore it does not include any security mechanism on this layer. These deficiencies are magnified by the fact that Modbus is a protocol designed for legacy programming control elements like remote terminal units (RTUs) or PLCs making the injection of malicious code into these elements easier. Modbus

TABLE II
PROTOCOL USED IN THE PRESENTED TESTBEDS AND DATASETS. • INDICATES THAT THE PROTOCOL IS SUPPORTED/AVAILABLE. IN SOME WORK IT IS NOT INDICATED THE VERSION OF MODBUS (TCP, RTU, OR ASCII) ADOPTED. IN SUCH CASES, A • IS USED TO INDICATE A GENERAL VERSION OF THE PROTOCOL

Name	Modbus	S7Comm	EtherNet/IP	DNP3	Logs	Phy.	Others
4SICS	TCP	•	•	•			
Aghamolki et al.	•			•			IEEE-C37.118
Ahmed et al.	•		•				Profinet
Alves et al.	TCP						
BATADAL						•	
Blazek et al.							IEC61850
BU-Testbed			•				
CockpitCI	TCP						
CyberCity Dataset	TCP		•				NetBIOS
CyberCity Testbed	TCP		•				NetBIOS
D1: Power System					•	•	
D2: Gas Pipeline	TCP						
D3b: Water Storage Tank	•						
D4: New Gas Pipeline	•						
D5: Energy M. S. D.					•		
Davis et al.	TCP						Custom TCP
DVCP	•						Profinet
Electra Modbus	•						
Electra S7Comm		•					
EPIC (IPSC)	TCP						
EPIC (iTrust)	TCP						IEC61850
EPIC Dataset	TCP					•	
EPS-ICS							Unknown
Farooqui et al.							Unknown

Security [35] offers a Modbus/TCP version enhancement focused on using the port 802. This new version enables TLS to provide confidentiality, integrity, and authentication using x.509v3 certificates. Moreover, it specified certificate-based authorization using role information transferred via certificate expansions. Researchers have also proposed different modifications to introduce confidentiality [36] or authentication [37] via covert-channel on Modbus.

DNP3: Westronic, Inc. (now GE Harris) designed DNP in 1990. In January 1995, the *DNP Users Group Technical Committee* was formed to review enhancements and recommend them for approval to the general Users Group. One of the most important tasks of this body was to publish the "DNP Subset Definitions" document, which establishes standards for scaled-up or scaled-down implementations of DNP3 [38]. DNP3 is an open, intelligent, robust, and efficient SCADA protocol organized into four layers: physical, data link, pseudo-transport, and application. In serial implementations, commands are issued broadcast. DNP3 contains significant features that make it more robust, efficient, and interoperable than older protocols such as Modbus, at the cost of higher complexity. The protocol's primary goal is to maximize system availability by putting less care into confidentiality and data integrity factors. DNP3 organizes data into data types such as binary inputs/outputs, analog inputs/outputs, counters, time and date, file transfer objects.

Security: As previously mentioned, DNP3 is a protocol designed to maximize system availability by putting less care into confidentiality and data integrity factors. Data link level includes the detection of transmission errors through Cyclical Redundancy Check (CRC) calculation. However, CRC is not a proper security measure since if an attacker can modify a packet, he/she can also change the CRC. At the application level, some efforts have been made to provide a safe authentication standard in DNP3. While in the beginning, pre-shared keys were used to authenticate, according to the standard IEEE 1815-2010 (deprecated), the latest versions implement Public Key Infrastructure (PKI) with remote key changes (standard

TABLE II
(Continued.) PROTOCOL USED IN THE PRESENTED TESTBEDS AND DATASETS. ● INDICATES THAT THE PROTOCOL IS SUPPORTED/AVAILABLE. IN SOME WORK IT IS NOT INDICATED THE VERSION OF MODBUS (TCP, RTU, OR ASCII) ADOPTED. IN SUCH CASES, A ● IS USED TO INDICATE A GENERAL VERSION OF THE PROTOCOL

Name	Modbus	S7Comm	EtherNet/IP	DNP3	Logs	Phy.	Others
Gas Pipeline testbed	TCP						
Genge et al.	●			●			Profinet
Giani et al.	●			●			
Gillen et al.			●				CIP
GRFICS	●						
HAI Dataset						●	
HAI Testbed							Fieldbus
Hui Nuclear		●					Profinet, Custom TCP
HVAC_Traces		●					DCE/RPC, NetBIOS
HYDRA	●						
Jarmakiewicz et al.							IEC61850, IEC104
Jin et al.				●			
Kaouk et al.	TCP						
Kim et al.							Variable
Koganti et al.	●						
Koutsandria et al.	●						
KYPO4INDUSTRY	●			●			
Lancaster's testbed							Converted to IP
Lee et al.				●			IEC61850
LegoSCADA	●			●			
Lemay Covert	●						
Lemay SCADA	●						
LICSTER	TCP						
Maynard SCADA							IEC104, OPC
Microgrid							Unknown
MiniCPS	TCP		●				
Mississipi Ethernet							Ethernet
Mississipi Serial	RTU, ASCII			●			
Modbus SCADA #1	TCP, RTU						
MSICST	TCP	●	●				
NIST	TCP		●				DeviceNet, OPC
PNNL							Not specified
PowerCyber				●			IEC61850
Queiroz et al.	TCP						
QUT_DNP3				●	●		
QUT_S7 (Myers)		●				●	
QUT_S7Comm		●					
Reavers & Morris	TCP, RTU						
RICS-el							IEC104
S4x15 ICS	TCP						BACnet
Sayegh et al.							FINS

TABLE II
PROTOCOL USED IN THE PRESENTED TESTBEDS AND DATASETS. ● INDICATES THAT THE PROTOCOL IS SUPPORTED/AVAILABLE. IN SOME WORK IT IS NOT INDICATED THE VERSION OF MODBUS (TCP, RTU, OR ASCII) ADOPTED. IN SUCH CASES, A ● IS USED TO INDICATE A GENERAL VERSION OF THE PROTOCOL

Name	Modbus	S7Comm	EtherNet/IP	DNP3	Logs	Phy.	Others
SCADA-SST	●						
SCADASim	TCP			●			
SCADAVT	TCP						Custom TCP
SGTB							Unknown
Singhet et al.				●			IEC104
SNL Testbed	TCP			●			IEC104 (partially)
SWaT Dataset			●			●	CIP
SWaT			●				CIP
T-GPP	TCP			●			
TASSCS	TCP						
Teixeira et. al	●						
TETRIS							
Turbo-Gas Power Plant	TCP		●				
VPST			●				
VTET	●	●					OPC
WADI Dataset						●	
WADI	TCP						
Wang et al.	TCP			●			
WUSTL-IIOT-2018	●						
Yang et al.							IEC104
Zhang et al.							Unknown

IEE 1815-2012). Recently, in 2020, GE Harris presents DNP3 version 6, introducing DNP3-SA [39], a separate protocol layer that supports Message Authentication Codec (MACs) to provide secure communication sessions, including authentication and integrity. Moreover, this version supports encryption to offer data confidentiality by using the AES-256 algorithm. Some other solutions have been proposed in the literature to implement cryptography protections, such as end-to-end encryption [40] and VPN for IP networks [41].

S7Comm: Introduced in 1995, S7comm (S7 Communication) [42] is a Siemens proprietary protocol that runs between standard PLCs of the Siemens S7-200/300/400 family and new generation PLCs like S7-1200/1500. It is a proprietary and closed standard without significant literature related to it. Siemens has a proprietary HMI software for the SIMATIC products and an Ethernet driver that provides connectivity to devices via the Siemens TCP/IP Ethernet protocol. In addition to this driver, there are also 3rd-party communication suites for interfacing and exchanging data with Siemens S7 PLCs.

Security: S7Comm is a closed protocol, so there is no related documentation. However, as various works underline, the base version of S7Comm does not include security features, and it is vulnerable to replay attacks [43]. However, in 2010 Stuxnet exploited the security vulnerabilities of S7Comm to compromise a Nuclear Plant in Iran. As a result of this incident, Siemens has developed a new version of the protocol, called S7CommPlus, with replay-attack protection. It has been proven that this version is also vulnerable to reverse debugging attacks [43].

PROFINET: Developed by PROFIBUS & PROFINET International (PI), PROFINET [44] is an open standard for Industrial Ethernet standardized in IEC 61158 and IEC 61784. Introduced in 2003, it is an evolution of the PROFIBUS standard, whose lines can be integrated into the PROFINET system via an IO-Proxy. This protocol follows the provider-consumer model for data exchange in a cascading real-time concept. It is compatible with Ethernet thanks to its flexible line, ring, star structures, and copper and fiber-optic cable solutions. It is also compatible with radio communications such as WLAN and Bluetooth. Thanks to Ethernet-based communication, it provides a direct interface to the IT level. The primary functions include a cyclic exchange of I/O data with real-time properties, acyclic data communication for reading and writing of

demand-oriented data, including the identification and maintenance function, and a flexible alarm model for error signaling with three alarm levels.

Security: PROFINET is a protocol operating in the application, link, and physical layers. The link layer in this protocol uses FDL (Fieldbus Data Link) to manage access to the medium. FDL operates with a hybrid access method that combines master-slave technology with the passing of a token, indicating who can initiate communication and occupy the bus. These measures ensure that devices do not communicate simultaneously. However, FDL constitutes any safety mechanism and may be susceptible to attacks involving traffic injection or Denial Of Service (DoS). In 2019, PI introduced three Security Classes to offer a way to select security measures based on the consumer needs [45]. Class 1 improves robustness through a digital signing of General Station Description (GSD) files using a PKI infrastructure, an extended Simple Network Management Protocol (SNMP) configuration, and a DCP in read-only mode. Class 2 expands the previous class by offering integrity and authenticity via cryptographic functions and confidentiality only of the configuration data. Instead, Class 3 offers all the previous characteristics and the confidentiality of all the data. Furthermore, it is worth mention that PROFIBUS offers some services that can use TCP/IP as a transport protocol, but only during an initial phase for device assignment. It is possible to add some of the classical TCP/IP cryptography and authentication security elements in these services.

ODVA's Networks: Founded in 1995, ODVA [46] is a global association whose members comprise the world's leading automation companies with the mission of developing advance open and interoperable communication technologies for industrial automation. The primary interest is developing the Common Industrial Protocol (CIP), supporting the various network adoptions such as DeviceNet, CompoNet, ControlNet, and the widely used EtherNet/IP. CIP encompasses a comprehensive suite of messages and services to collect industrial automation applications such as control, safety, energy, synchronization, motion, information, and network management. This protocol allows users to integrate these applications with the IT Ethernet networks and the Internet. The protocol follows a model for objects: each one is made up of attributes (data), services (commands), connections, and behavior (the relationships between data and services). CIP also defines device types, with each device type having a device profile. The device profiles indicate which CIP objects must be implemented, what configuration options are possible, and the formats of I/O data.

EtherNet/IP is an adaption of CIP to the Ethernet TCP/IP stack, while DeviceNet provides a way to use CIP over the CAN technology. ControlNet uses CIP over a Concurrent Time Division Multiple Access (CTDMA) data link layer, and CompoNet implements CIP on a Time Division Multiple Access (TDMA) data link layer.

Security: Recently, in 2015, ODVA introduced the CIP Security framework [47] to provide security measures to CIP protocol. Since different systems might need different security levels, CIP Security provides different security specifications profiles to help users configuring inter-operable devices. On EtherNet/IP, it enables TLS and DTLS to secure the TCP and UDP transport layer protocols. TLS and DTLS provide authentication of the endpoints using X.509 certificates or pre-shared keys, message integrity and authentication employing TLS message authentication code (HMAC), and optional message encryption.

Open Platform Communications (OPC): The classic OPC [48], developed in 1996, was designed to provide a communication protocol for personal computer-based software applications and automation hardware. It was based on Microsoft's distributed component object model, making them platform-dependent and not suitable for cross-domain scenarios and the Internet. Nowadays, the classic OPC is no anymore developed. In 2006 a new version, OPC United Architecture (OPC-UA), was released as an operational framework for communications in process control systems. It provides greater interoperability, eliminating MS-Windows dependency, but maintaining retro compatibility with its predecessor. This specification is built around Service-Oriented Architecture (SOA) and is based on Web services, making it easier to implement OPC connections over the Internet. The general layout of the communication is simple: the hardware devices (e.g., PLC, Controller) act as data sources, and the software applications (e.g., SCADA, HMI) play the role of data consumers, whereas the OPC interface acts as connectivity middleware, enabling the data flow. Using the OPC, the client applications access and manage the field information without knowing the physical nature of data sources. With OPC-UA improvements, the protocol is widely used in critical and industrial fields such as energy automation, virtualized environment, and building automation.

Security: The use of Distributed Component Object Model (DCOM) and Remote Procedure Call (RCP) make OPC very susceptible to different attacks [49]. Since it is inherently difficult to apply patches to industrial control systems, many discovered vulnerabilities with available patches continue to be potentially exploitable industrial control networks. Instead, OPC-UA implements a security model and five security classes, bringing greater security to the architecture at the cost of slightly higher complexity [50]. It is also possible to implement only a fraction of the security measures by using one of the five security classes provided. The security model allows generating a secure channel that provides encryption, signatures, and certificates at the communication layer. Furthermore, a session in the application layer is used to manage user authentication and user authorization. Thanks to these security measures, it is advisable to deploy OPC-UA rather than the classic version of OPC and to update the already deployed versions wherever possible.

IEC 60870-5-104 (IEC 104): Released in 2000, IEC 60870-5-104 (IEC 104) protocol [51] is an extension of the IEC 101 protocol with the changes in transport, network, link, and physical layer services to suit the complete network access. The standard uses an open TCP/IP interface to network to connect to the LAN (Local Area Network), and routers with different facilities can be used to connect to the WAN (Wide Area Network). There are two different methods of transporting messages. The first provides bit-serial communications

over low-bandwidth communications channels. In the second, introduced with IEC 104, the protocol's lower levels have been completely replaced by the TCP/IP transport and network protocols. Thanks to the IEC 104 simple structure in terms of its data types and data addressing options, it is possible to quickly achieve interoperability with other protocols.

Security: IEC 104, has been proven to be vulnerable to different types of attacks, such as man-in-the-middle and replay attacks [52]. A more recent and secure standard of the IEC family is IEC 62351. This version implements end-to-end encryption to prevent attacks such as replay, man-in-the-middle, and packet injection. However, due to the higher complexity, industries rarely upgrade IEC 104 to IEC 62351.

IEC 61850: Like IEC 104, IEC 61850 [53] was originally designed to enable communications inside substations automation systems. In recent versions, an extension of IEC 61850 allows substation-to-substation communication and provides tools for translation with other protocols such as IEC 60870-5, DNP3, and Modbus. The protocol is devised using an object-oriented design suited for communication between devices of different vendors. To provide long-term stability, IEC 61850 divides the information model and communication protocols. For not time-critical applications, the protocol uses MMS via TCP/IP as the communication protocol. Instead, GOOSE can be employed over Ethernet if the time constraint is critical. In the case of voltage and current sample information transportation, SV over Ethernet is generally used. However, recent versions of the standard provide GOOSE/SV mapping over TCP/IP using UDP packets at the transport layer for inter-substation information exchange.

Security: IEC 62351 standard provides various security measures, offering guidelines and developing a secure operation framework. Since in time-critical application encryption is not suitable due to the 3ms delivery overhead, the standard recommends using digital signature generated by SHA256 and RSA public key algorithms. For MMS communications instead, TLS is recommended, with optional end-to-end encryption of all the packets exchanged [54]. Furthermore, the employment of IEC 61850 in heterogeneous networks exposes the system to protocol mapping vulnerabilities. It is possible to prevent these vulnerabilities by developing ad-hoc security by design architectures [55].

Other Protocols: In addition to previously presented protocols, some datasets contain few packets related to other generic protocols used in a wide range of applications. Published in 1995, BACnet is a data communication protocol for building automation and control networks supported by some HVAC components but not widely used [56]. Distributed Computing Environment/Remote Procedure Calls (DCE/RPC) is a remote procedure call system that allows programmers to write distributed software as if it were on the same computer. One of the datasets presented in this paper includes DCE/RPC, together with NetBIOS, a networking protocol allowing applications on separate computers to communicate over a LAN. Other generic packets are visible in some datasets like Address Resolution Protocol (ARP) and Domain Name System (DNS) requests but are generally not related to the industrial field.

B. Industrial Protocols Employment

In Table II, we provide the complete list of the datasets and testbeds analyzed in this work, together with the protocol used in the specific platform. In detail, the table associates to each testbed the protocols supported and to each dataset the protocols available. Moreover, it indicates if data logs and physical measures are provided in the datasets. As previously described, there are several different protocols employed in the ICS field. In Figure 4 we reported the percentage of usage of each protocol in the testbeds and datasets investigated in this survey. Modbus, and its different versions (i.e., TCP, RTU, ASCII), are the most used protocols, while EtherNet/IP, DNP3, and S7Comm follow with a lower but significant employments.

Since the testbed and dataset employed should represent an approximation of real-world scenarios, it is interesting to compare if the distribution of the protocols implemented in the different datasets is similar to the protocol's distribution in the real industrial system. Verifying this claim is a challenging task due to the various privacy concerns of companies in disclosing information. Various works tried to deal with this problem by measuring the industrial traffic present on the Internet. Although with limitations, the traffic measurement can represent a reasonable estimate of the most popular industrial protocols currently used. By leveraging Censys search engine, Xu *et al.* [57] scanned the Internet for about two years (from 2015 to 2017), examining for industrial devices exposed. In particular, they focused on five protocols: Modbus, S7Comm, DNP3, BACnet, and Tridium Fox. Results show a significant prevalence of Modbus and Tridium Fox devices (with more than 20K devices found), a middle spread of BACnet (about 11K devices) and S7Comm (about 4.5K devices), and a lower number of devices using DNP3 (less than 1K). Furthermore, the authors noticed an increasing number of Modbus and S7Comm devices during the two years of recording, while the number of DNP3 devices decreased. A similar study, presented by Barbieri *et al.* [12], leveraged Shodan and an Internet Exchange Point (IXP) in Italy to measure ICS host exposure. They discover many devices using Modbus, MQTT, and Niagara Fox. Furthermore, the authors also identified EtherNet/IP, S7Comm, and BACNet devices but with significantly lower samples.

In addition to measurement works, we can also rely on market analysis. According to an HMS report [58], the overall market share of Industrial Ethernet protocols increased in 2020. In particular, EtherNet/IP and Profinet obtained first place with 17% of market share, while in third place there is EtherCat with a share of 7%. On the other side, Fieldbus protocols such as Profibus and DeviceNet showed a decrease of 5% on the market share with respect to the previous year. Interestingly that Modbus protocol (TCP and RTU variants), despite being heavily employed in testing, results in a 10% of market share (5% RTU, 5% TCP).

Based on these findings, Modbus/TCP results as the most employed protocol in testbeds, datasets, and Internet measurements. Nevertheless, it obtains a low market share in the last year (i.e., 2020), outranked by EtherNet/IP, which is also employed in a significant part of the testbeds and datasets presented in

this survey, and Profinet, which instead is used in only the 3.2% of the testing system analyzed in this work. Therefore, Profinet can be an interesting protocol to introduce in future testbeds and datasets to follow the market trend. Other protocols that an increasing market share are BACnet, TridiumFox, and NiagaraFox, which are not present in the testing platforms, except for one dataset that contains BACnet packets. Finally, another protocol that could be interesting to include in testing platforms is EtherCAT, which has a 7% of market share. However, no testing system currently supported it.

V. ICS Attack and Defence

In this section, we offer an overview of the various attacks and defense mechanisms in ICSs. In particular, in Section V-A we present an overview of the typical attacks which can target ICSs and are implemented in the different testbeds and datasets. Instead, in Section V-B we proposed a brief overview of the techniques which can be employed to detect and mitigate cyberattacks in this field.

A. Typical Attacks

ICSs are extremely complex systems, which connect IT components with sensors, actuators, and other OT devices. Such an interconnected scenario with a wide variety of various components may hide attack surfaces caused, for instance, by device-specific vulnerabilities or misconfigurations.

Having a clear idea of the different attack typologies is essential in building and testing defenses. Based on that, testbeds should be capable of simulating verisimilar attacks, while datasets should include not only normal operation data but also attack data. Reproducing attacks is a challenging task because the simulation should precisely emulate a realistic abnormal operating condition. However, it is impossible to replicate every type of attack due to the devices' potential damages. Some attacks could also shift the testbed's operating behavior in a dangerous state and seriously damage the machines. Furthermore, the limited class of attacks implemented could raise a generalization problem of the detection strategy, not transferable to novel and unknown attacks.

In a CPS scenario, by definition, there are two possible attacks vector surfaces on the system. *Network-based* attacks, targeting the networking part of the network such as packets, protocols or routing policies, and *Physical-based* attacks, aimed at corrupting the physical process of the devices. Sometimes, these two attack categories' goals may also converge or combine to reach a specific target.

Network Attacks: The most common attack models include the Control Zone network access by the attacker to compromise an ICS. An attacker can obtain the network control through a phishing attack to the site operators [10] or by exploiting the security lack of the legacy devices connected to the Internet [12]. There are different actions that malicious actors can perform, but it is possible to categorize the main ones into five different classes [59], [60] of network attack. These attacks are also implemented in the testbeds to generate abnormal operating conditions.

- *Reconnaissance Attack* aims at the identification of potential victims within a network. Usually, this class of attack is used to plan other moves, such as identify other vulnerable devices. These attacks can be passive (e.g., port mirror) or active (e.g., nmap). Reconnaissance Attack can be performed to understand the topology of the ICS with the consequently vulnerable devices or to identify the physical process involved.
- *Man-in-the-Middle (MitM) Attack* allows an attacker to sit in the middle of communicating parties. The attacker is then able to read or modify the communications, inject commands, or drop packets. As introduced in Section IV-A, most industrial protocols suffer from insecurity by design and, therefore, an attacker can perform malicious protocol exploitation. In [28] the authors reported a detailed taxonomy of attacks against Modbus and DNP3 protocols. The final aims of this type of attack can vary from the control of some devices to the disruption of the ICS's normal state to damage the system's owner or the system itself. MitM attacks on ICS can intercept wired communication, which requires the installation of network tap or wireless by using antennas.
- *Injection Attack* aims at supplying untrusted and malicious inputs to a system. Typically, in an ICS, an attacker can inject data such as false measures from sensors or actuators (Data Injection Attack) or command (Command Injection Attack). Often a compromised node launch this type of attack, but, in some cases, the injected data can originate from other sources (e.g., a new entry point for the network). This type of attack also includes the injection of Malware (e.g., Worm [7], Ransomware [17]) or Backdoors in the devices, which allow the attacker to drive the system to an unsafe state.
- *Replay Attack* is based on the retransmission of a valid message that has been previously seen in the network. This attack is difficult to be detected, and it can lead to malfunctions of the system. For example, in a nuclear plant context, the attacker could retransmit a message with a low temperature of the reactor instead of rising, inhibiting the activation of safety measures.
- *Denial of Service (DoS) Attack* is used to make devices unavailable by overloading the system resources to disrupt the communication between machines in the system. Usually, a common technique is packet flooding and, if packets are generated from many different sources, it is called Distributed Denial of Service (DDoS). This attack can stop some devices, making them unavailable and lead to unpredicted behaviors in the ICS. As previously explained, industrial devices are generally legacy and have low computational power, therefore even a low amount of packets can stop their normal functioning.

Physical Process Attacks: This class of attacks aims to alter the physical process and the complex relations of the system to manage it. Cyber-Physical Systems enable such attack surfaces due to the field device (i.e., sensors and actuators), sometimes in remote places. To achieve these attacks, the attacker could have previously obtained access to the system with one or more of the network attacks previously described. Generally,

physical process attacks represent the final attack chain goal, which started with the network as an entry point.

- *Stealth Attack* generates small perturbations in the system process to create long term damages (e.g., loss in production terms or the devices' degradation). The stealth attack can use a static perturbation, by introducing a constant error in the physical measure (e.g., increasing or decreasing the production), or dynamic, by rapidly oscillating between upper and lower measurement bounds (e.g., causing turbulence in the flows). This class of attacks is generally difficult to detect since it maintains the process inside its limits.
- *Ladder Logic Modification:* as introduced in Section III-B, the control logic of the PLC is generally programmed in ladder logic. An unauthorized modification of the internal logic aims at modifying the physical condition monitored by the device. According to [61] there can be two types of modification: logic modification and function modification. Logic modification aims to modify the internal boolean logic of the device to bypass the control condition, while function modification attempts to change the internal parameters updates. Although this attack requires a high level of access to the target device, it can cause dangerous consequences to the system by driving it to unstable conditions or damaging the equipment.
- *Device Manumission* is achieved by physically tampering with the field device to compromise the data recorded. This attack aims to induce wrong measurements in the system exploiting the distributed and, therefore, less monitored nature of these systems.
- *Direct Damage Attacks* aims to disrupt and damage the entire process or physical equipment by introducing significant process variations that bring the system into an unsafe state. This attack may also have severe consequences on the population or the environment around the site.

B. ICS Defence Techniques

It is possible to enforce ICS security by implementing security-by-design architectures. For instance, it is possible to use DMZ as specified in the Purdue Model (Figure 1), enforcing network separation and segregation. Furthermore, boundary protections and firewalls with ICS-specific rules help protect an ICS from external attacks. The National Institute of Standard and Technology (NIST) proposed a complete guide explaining how to set up a secure network to protect an ICS [2]. Another good practice is to implement the secure version of the industrial protocols. However, security-by-design can be challenging to consider in ICSs, due to implementation constraints. Sometimes, it could also happen that companies consider the security aspects after the construction phase, and this can rise problem in the integration of measures in the infrastructure. Detection mechanisms can solve this limitation and be integrated into the system after the construction phase, for instance, in central nodes or with network tap.

The most widely adopted technique to secure an ICS is represented by Intrusion Detection Systems (IDSs). IDSs are algorithms designed to detect attacks by passively or actively monitoring the system. If the IDS is passive, it will only raise passive alerts in case of an anomaly. Instead, if the IDS is active, it will also take active response action in case of an anomaly (e.g., shutting down part of the system). IDSs represent a cost-effective solution since they can be installed without changing the system topology or substituting all network devices.

In the following, we briefly report the two main categories of IDS, which employ two different approaches to detect attacks or domain drifts.

Knowledge-based intrusion detection (also called *misuse-based*) focus on looking for runtime features that match a specific pattern of misbehavior. This method aims to exploit the stationary of ICSs, which, unlike IT systems, are characterized by control loop operation regulated by a constant polling time communication. The most famous misuse-based solutions include Snort [62] and Suricata [63] which has also been extended to include industrial protocols rules. The knowledge-based approach requires low computational power and has a low false-positive ratio since the system reacts only to known threats. However, it has the disadvantage of offering no protection against zero-day vulnerabilities. For this reason, the research community is focusing on developing a dynamic mechanism that can identify domain shifts without the need for signatures [64].

The current research trend focuses on the *anomaly-based* intrusion detection, which looks for runtime features that differ from normal behavior. The normal behavior pattern can be defined using unsupervised approaches training the model with live data or semi-supervised utilizing a set of truth data. This approach is called *behavior specification-based* intrusion detection. It represents a suitable ICS solution since it aims to dynamically learn the regular behavior model of network traffic and physical models. Again, ICS systems are generally characterized by constant time communication, thus helping the definition of a more robust model.

This last method is promising thanks to modern machine learning and deep learning techniques that can be used to automatize the anomaly detection classification process. A common requirement of these algorithms is the need for a considerable quantity of data: generally, the more data you provide to the training phase, the more precise your detection will be.

IDS are also classified according to the data source. *Network-based* IDS uses network adapters to collect and analyze packets in real-time. On the contrary, *host-based* IDS monitors the documents, processes, and other information specific to a particular device to identify. The disadvantage is that monitoring regard only one node in the network, while with the former approach, all the network is under control. On the other hand, *host-based* can detect also threat coming from sources other than the network (e.g., USB sticks) [65].

The basic idea behind IDS is to exploit the massive amount of data collected from the sensors and predict an ICS's operations. Many features can be extracted from network traffic

(e.g., timings of packets, bytes transmitted) and from the physical process (e.g., sensor measurements, actuator statuses). This information can be employed in different ways to detect abnormal behaviors and attacks in an ICS.

A novel detection design concept that exploits the correlation of multiple ICS points was proposed by Bernieri *et al.* [66]. In this work, the authors proposed a distributed detection approach to consider the different information characterizing ICSs to identify more complex vulnerabilities. Similarly, Ghaeini and Tippenhauer [67] proposed a hierarchical IDS which is able to combine data from different levels to detect attacks having a distributed effect on the system. Another novel approach has been proposed by Caselli *et al.* [68] who designed a sequence-aware IDS which is based on the monitoring of sequences of events instead of single ones.

VI. ICS Testbeds

In this section, we present a comprehensive analysis of the various ICS testbeds available in the literature. Firstly, in Section VI-A we introduce the classification method we employ in this work. Instead, in Section VI-B and Section VI-C we recall, respectively, the requirements for an effective testbed and the main challenges in developing an ICS testbed. Then, we propose a detailed description of a set of interesting testbeds we choose, dividing them into the three categories that we design. In particular, Section VI-D contains physical testbeds, Section VI-E presents virtualized testbed, and Section VI-F describes hybrid testbeds which are a conjunction point of the other two categories.

A. Testbeds Classification

There are different possible classifications of a testbed, basing on its sector, construction methodology, or the process involved. In this survey, and particularly in this section, we consider the functional elements involved in the testbed, classifying them as *Physical*, *Virtual*, or *Hybrid* testbed. The different testbed categories are illustrated in Figure 5, even if sometimes the difference between can be minimal. For instance, many of the virtual testbeds presented can be interconnected with physical devices or wholly virtualized. Instead, Hybrid systems were designed with some real components and, without them, they could not work correctly.

Physical testbeds use real hardware and software to configure both the network and physical layers. They are a suitable approach when researchers need a solution to collect realistic measurement variation and latencies. Furthermore, it is possible to exploit the vulnerabilities of a specific device. On the other hand, physical testbeds are expensive both in construction and maintenance. They generally have a long building time, and they may not provide a safe execution of dangerous physical processes (e.g., nuclear sector).

On the contrary, virtual testbeds leverage software simulations and emulations with single or multiple programs to reproduce the entire network and all the different components. A virtual testbed represents a low-cost solution, but it is not easy to simulate high fidelity physical processes due to the virtualized environment. Despite this lack of precision, dangerous and risky processes (e.g., Nuclear sector) can be, in this way, simulated in a laboratory. MATLAB, Modelica, Ptolemy, and PowerWorld are software used in the process simulation phase. Other tools are used to model control center communication networks (e.g., DETER, Emulab, CORE, ns3) and other devices used in the system such as PLCs (e.g., STEP7, RSEmulate, Modbus Rsim, Soft-PLC). Despite not generating data with perfect fidelity, these approaches are easy to update and upgrade, which gives them good flexibility and extendibility.

A widely diffused approach is developing testbeds composed of both physical devices and software simulations. This approach represents a good trade-off between physical and virtual solutions and is called a hybrid testbed. The main difference between the complete physical testbeds is that part of the components is simulated using specialized software. This solution can reduce the system's fidelity, but on the other hand, it permits to contain the cost and development time. However, as stated before, the separations between Virtual and Hybrid testbed is not always well defined. Sometimes virtual testbeds can be modified to work as a hybrid testbed by supporting physical devices. For example, VTET [69] can be deployed using physical PLCs to replace the simulated ones. In this work, we consider as Hybrid a virtualized testbed composed of at least one real industrial device (e.g., PLC, IED, actuator, sensor).

In Figure 7, we reported the geographic distribution of the Physical and Hybrid around the world. We think that this representation could help the reader see the current research trend in this sector in the world. In particular, Figure 6 provides a high level view of where the testbeds are placed in the world, while Figures 7(d), 7(c), and 7(b) show close-up of the countries with more than one testbed. In these figures, the marker size represents the estimated cost of the testbed. Simultaneously, the color indicates the Citations of the associated reference according to Google Scholar at the writing time. Furthermore, we developed a website with an interactive map to collect and also provide information about future ICS testbeds and datasets.[1] Our goal is to continue to update this collection in the future. Moreover, in Table III we reported a brief comparison between the testbed presented in this paper, highlighting their main information and features. In particular, for every testbed we reported the following information.

- *Name* of the testbed (or of the authors if a name is not provided);
- *Institution* in which the testbed has been developed;
- *Country* The country on which is based the testbed or the institution of the first author;
- *Sector* indicates the field of the represented process;
- *Category* of the testbed. It can be *Physical*, *Virtual*, or *Hybrid*;

[1] https://spritz.math.unipd.it/projects/ics_survey/

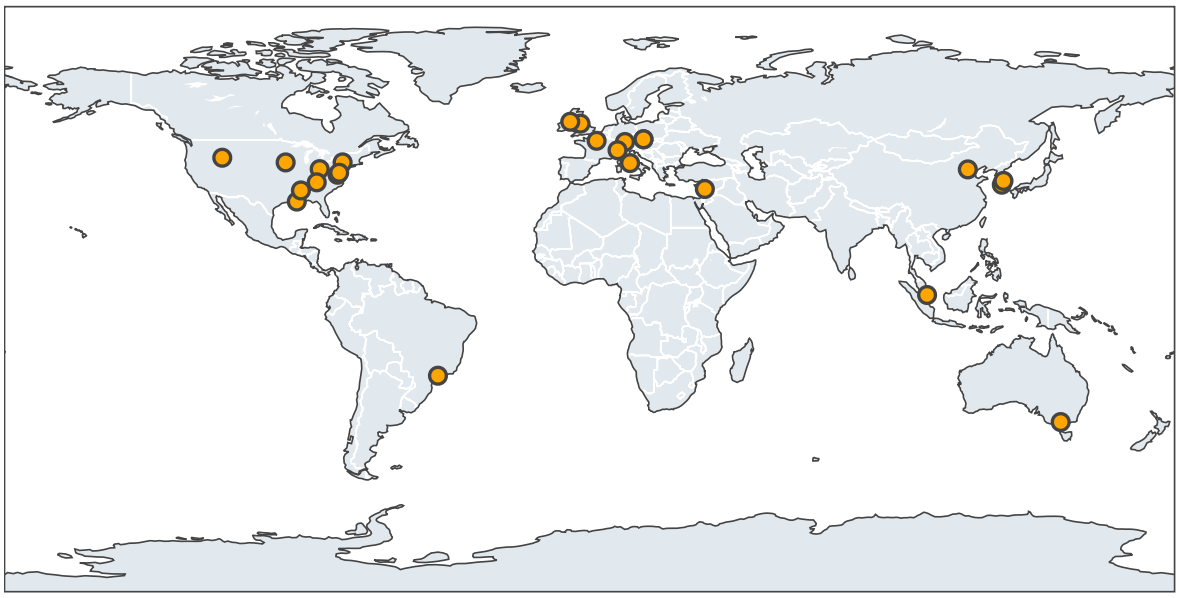
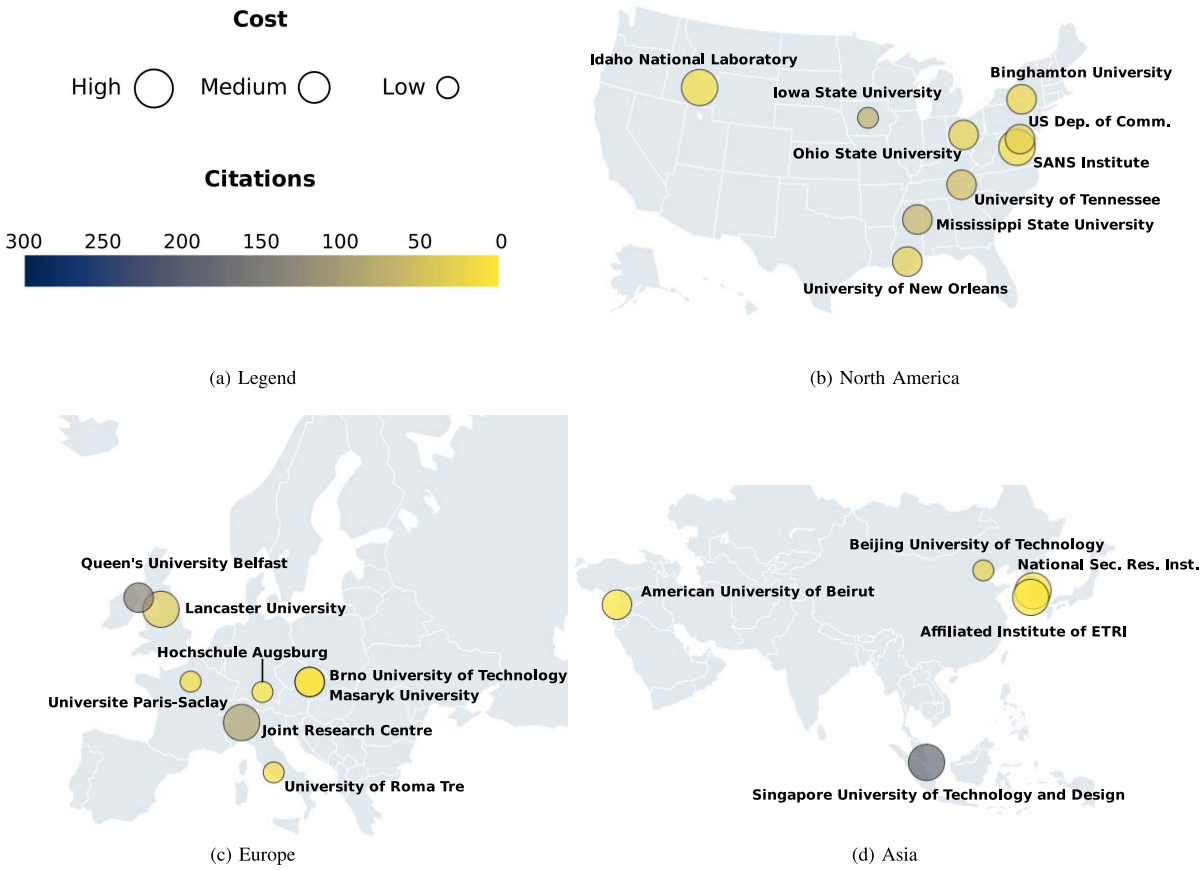


Fig. 6. Physical and hybrid with physical process testbeds distribution around the World.



(a) Legend

(b) North America

(c) Europe

(d) Asia

Fig. 7. Physical and hybrid with physical process testbeds distribution on the continents with more than one testbed: North America, Europe, and Asia. If there is more than one dataset in a place (e.g., Singapore SUTD), we aggregated the information.

- *Physical Process* indicates how is implemented the physical level. It can be *Simulated* with a software or *Real* if consists of a physical implementation;
- *License* of the testbed. It can be:
  - *Open-source* if the source code is freely available;
  - *Open description* if, despite the source code is not provided, the description is sufficiently detailed to allow a reader developing a similar copy;
  - *Education* if it is maintained by an university and open to collaborator;

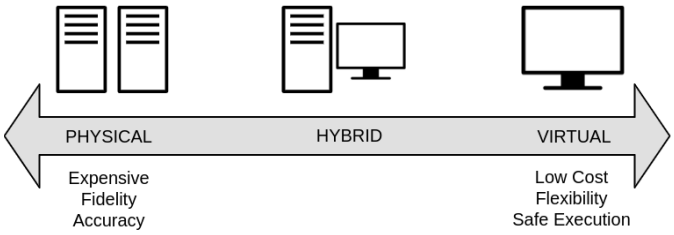


Fig. 5. A summary of differences between testbed types.

  - *Collaborations* if it is maintained by an institution which can accept collaborations;
  - *Not available* if it is owed by a private company and so not accessible or not available online.
- *Scope* indicates the applications of the testbed. It can be:
  - *Security* if the main scope is related to cybersecurity research;
  - *Forensic* if the target scope is to provide a way to perform forensics research;
  - *Pedagogy* if the main scope is to provide education to students;
  - *General* if a precise scope is not specified.
- *Cost* estimated for the testbed implementation. When available, the cost estimation is based on our analysis of the documentation of the testbed. Furthermore, we consider the testbed's size, and the number and type of devices employed (e.g., a single professional PLC cost at least 300$).
  It can be:
  - *Low* for a cost estimated $<$500$. These testbeds are generally composed of a single PLC or different low-cost devices (e.g., Raspberry or Arduino).
  - *Medium* for a cost estimated between 500$ and 10$K$$. These testbeds are composed of a discrete number of devices and hardware not extremely expensive, but at the same time, they have a reasonable degree of complexity.
  - *High* for a cost estimated $>10K$$. This category is conceived for testbeds with a high degree of complexity and a large number of devices employed.
- *Reference* includes a reference to a description of the testbed.
- *Resource*, if available, indicates a resource for the download.

This information was not always available or easy to retrieve; therefore, the degree of detail may vary according to the specific dataset.

B. Testbeds Requirements

When researchers need to work with a real-world ICS environment, the proper solution is to build a testbed for conducting rigorous, transparent, and replicable testing of new technologies. The different testbeds vary in dimension, complexity, or sector. According to [22], an effective testbed needs to satisfy four main requirements: i) *Fidelity*, ii) *Repeatability*, iii) *Measurement Accuracy*, and iv) *Safe execution*. Sometimes, it could be challenging to satisfy all these requirements together; therefore, it is important to determine an optimal trade-off based on the research needs during the design phase.

A testbed should be developed to achieve good *fidelity* by accurately replicate the devices and processes from a real-world ICS. This is an expensive and space-consuming task, making it difficult for other researchers without much funding to replicate the same environment to verify the results. In these cases, mathematical models can be employed to virtualize physical processes in a cheap but less accurate way.

Repeatability is an essential property for a testbed: it allows other researchers to reproduce the findings and compare other solutions on the same system. This property can be easily achievable for completely simulated testbeds, while it can be extremely challenging for ICSs that employ physical components or processes.

A testbed should monitor a physical process and take *accurate measurements* without interfering with it. Sensors must be placed smartly, and if different points of measures are available, they must be carefully synchronized to provide accurate and reliable data.

Often ICSs are used to manage critical physical processes (e.g., chemical reactions, nuclear plants). If under attack, these kinds of *processes can be dangerous* and can cause physical damage to the system itself. Since researchers need to study countermeasures' effect and effectiveness to attacks, testbeds must be provided with safe execute risky processes. This design challenge can be mitigated by employing simulations at the cost of a loss of accuracy. In other cases, processes are instead less critical. However, they can have an expensive or time-consuming recovery after an attack (e.g., after an attack completely empties a container into a water treatment system, it will take time to refill it again). In these scenarios, a virtual approach can be an excellent alternative to the physical replication [69].

C. Challenges in Developing a Testbed

The development of an industrial testbed is challenging from several points of view. Different works analyze the challenges in developing a well-designed ICS testbed [70], [71]. Based on the existing literature, in the following, we present the main problems related to the development of such a testbed.

- *Design Guidelines:* When a research group decides to venture into building a testbed, it is fundamental to have a clear idea of the architecture. A clear and defined architecture can be useful in the development phases and to project further expansions. Moreover, it can guide other groups in building their own testbeds and therefore enabling the experiment repeatability. However, it is difficult to identify clear guidelines that help design a testbed from the engineering perspective.
- *Real Word Representation:* An industrial system must represent a real-world industrial scenario, including all the physical processes related to the environment.

TABLE III
SUMMARY OF TESTBEDS PRESENTED IN THE LITERATURE. WE DENOTE **CATEGORY** AS H: HYBRID, P: PHYSICAL, AND V: VIRTUAL. TO SPECIFY THE ICS PROCESS **PHYSICS** WE USE S: SIMULATED, R: REAL, M: MIXED, AND NO: IF THERE IS NOT A PHYSICAL PROCESS. TO DENOTE THE **LICENSE**, WE USE OD: OPEN DESCRIPTION, E: EDUCATION, OS: OPEN-SOURCE, C: COLLABORATION, AND NA: NOT AVAILABLE. TO DENOTE THE **SCOPE**, WE USE S: SECURITY, G: GENERAL, P: PEDAGOGY, AND F: FORENSIC. TO DENOTE THE **COST** WE USE L: LOW, M: MEDIUM, AND H: HIGH. THE ENTRIES IN YELLOW INDICATES THAT THE TESTBED IS HYBRID, THE ENTRIES IN BLUE MEAN THAT THE TESTBED IS PHYSICAL, WHILE ENTRIES IN GREEN CORRESPOND TO VIRTUAL TESTBEDS

Name or authors	Category	Institution	Country	Sector	Physics	License	Scope	Cost	Reference	Resource
Aghamolki et al.	H	USF	Florida, US	Power Grid	S	OD	S	M	[72]	-
Alves et al.	H	UAH	Alabama, US	Gas Pipeline	S	OD	S	L	[73]	-
CockpitCI	H	University of Coimbra	Portugal	Power Grid	S	OD	S	M	[74]	-
CyberCity	H	SANS Institute	-	City	M	E	S, P	H	[75]	-
EPIC (IPSC)	H	JRC Ispra	Italy	General CPS	S	OS	S	L	[76]	[77]
EPS-ICS	H	BIT	China	Generic ICS	R	NA	G	L	[78]	-
Gillen et al.	H	ORNL	Tennessee, US	Cooling System	S	OD	S	M	[79]	-
Hui Nuclear	H	Queen's University	UK	Nuclear Plant	S	OD	S	M	[80]	-
HYDRA	H	University of Roma Tre	Italy	Water Distribution	R	OS	S	L	[81]	[82]
Jarmakiewicz et al.	H	MUT	Poland	Power Grid	S	OD	G	M	[83]	-
Kaouk et al.	H	University of Grenoble	France	Generic ICS	S	OD	S	L	[84]	-
Kim et al.	H	NSRI	South Korea	6 Different ICS	R	E	S, P	H	[85]	-
Koutsandria et al.	H	Sapienza University	Italy	Power Grid	S	OD	S, F	M	[86]	-
KYPO4INDUSTRY	H	Masaryk University	Czech Republic	Linear Motor	R	OD	P	M	[87]	-
LegoSCADA	H	Universite Paris-Saclay	France	Vehicular	R	OS	S	L	[88]	[89]
Microgrid	H	OSU	Ohio, US	Power Grid	M	OD	G, P	M	[90]	-
MSICST	H	-	China	4 Different ICS	S	OD	S	H	[91]	-
NIST	H	USDOC	US	4 Different ICS	M	C	S	M	[92]	-
PNNL	H	PNNL	Washington, US	Generic CPS	S	OD	S	L	[93]	-
Queiroz et al.	H	RMIT University	Australia	Water Distribution	R	OD	S	L	[94]	-
SNL Testbed	H	SNL	New Mexico, US	Generic ICS	No	OD	S	L	[95]	-
VPST	H	University of Illinois	Illinois, US	Power Grid	S	OD	S, G	L	[96]	-
Ahmed et al.	P	UNO	Louisiana, US	3 Different ICS	R	OD	S, F, P	M	[97]	-
Blazek et al.	P	BUT	Czech Republic	Power Grid	R	OD	S	M	[98]	-
BU-Testbed	P	Binghamton University	New York, US	Power Plant	R	OD	S	M	[99]	-
EPIC (iTrust)	P	SUTD	Singapore	Electric Power	R	E	S	H	[100]	[101]
HAI Testbed	P	ETRI	South Korea	Power Plant	R	OD	S	H	[102]	-
Lancaster's	P	Lancaster University	UK	Generic ICS	R	C	G, P	H	[103]	-
LICSTER	P	HS-Augsburg	Germany	Generic ICS	R	OS	S, P	L	[104]	[105]
Mississipi Ethernet	P	MSU	Mississippi, US	2 different ICS	R	E	S, P	M	[106]	-
Mississipi Serial	P	MSU	Mississippi, US	Industrial op.s	R	E	S, P	M	[106]	-
PowerCyber	P	Iowa State University	Iowa, US	Power Grid	R	OD	S, P	L	[107]	-
Sayegh et al.	P	AUB	Lebanon	Generic ICS	No	OD	S	M	[108]	-
SGTB	P	INL	Idaho, US	Power Grid	R	NA	S	H	[109]	-
SWaT	P	SUTD	Singapore	Water Treatment	R	C	S	H	[110]	[111]
Teixeira et. al	P	IFET	Brazil	Water Distribution	R	OD	S	M	[112]	-
T-GPP	P	JRC Ispra	Italy	Power Plant	R	OD	S	H	[113]	-
WADI	P	SUTD	Singapore	Water Distribution	R	C	S, P	H	[114]	[115]
Yang et al.	P	QUB	Irland	Power Grid	R	OD	S	M	[116]	-
Zhang et al.	P	University of Tennessee	Tennessee, US	Nuclear Plant	R	OD	S	M	[117]	-
Davis et al.	V	University of Illinois	Illinois, US	Power Grid	S	OD	S	L	[118]	-
DVCP	V	TUHH	Germany	Chemical Process	S	OS	S, F	L	[119]	[120]
Farooqui et al.	V	NUST	Pakistan	Generic CPS	S	OD	S	L	[121]	-
Gas Pipeline	V	MSU	Mississippi, US	Gas Pipeline	S	NA	S	L	[122]	-
Genge et al.	V	JRC Ispra	Italy	Generic ICS	S	OD	S	L	[123]	-
Giani et al.	V	UC Berkeley	-	Generic ICS	S	OD	S	L	[124]	-
GRFICS	V	Georgia Tech	Georgia, US	Chemical Process	S	OS	S, P	L	[125]	[126]

TABLE III
(Continued.) SUMMARY OF TESTBEDS PRESENTED IN THE LITERATURE. WE DENOTE **CATEGORY** AS H: HYBRID, P: PHYSICAL, AND V: VIRTUAL. TO SPECIFY THE ICS PROCESS **PHYSICS** WE USE S: SIMULATED, R: REAL, M: MIXED, AND NO: IF THERE IS NOT A PHYSICAL PROCESS. TO DENOTE THE **LICENSE**, WE USE OD: OPEN DESCRIPTION, E: EDUCATION, OS: OPEN-SOURCE, C: COLLABORATION, AND NA: NOT AVAILABLE. TO DENOTE THE **SCOPE**, WE USE S: SECURITY, G: GENERAL, P: PEDAGOGY, AND F: FORENSIC. TO DENOTE THE **COST** WE USE L: LOW, M: MEDIUM, AND H: HIGH. THE ENTRIES IN YELLOW INDICATES THAT THE TESTBED IS HYBRID, THE ENTRIES IN BLUE MEAN THAT THE TESTBED IS PHYSICAL, WHILE ENTRIES IN GREEN CORRESPOND TO VIRTUAL TESTBEDS

Name or authors	Category	Institution	Country	Sector	Physics	License	Scope	Cost	Reference	Resource
Jin et al.	V	UIUC	Illinois, US	Generic ICS	S	OD	S	L	[127]	-
Koganti et al.	V	University of Idaho	Idaho, US	Power Grid	S	OD	S	L	[128]	-
Lee et al.	V	Ajou University	Korea	Power Plant	S	OD	S	L	[129]	-
Maynard SCADA	V	Queen's University	UK	Generic ICS	S	OS	S	L	[130]	[131]
MiniCPS	V	SUTD	Singapore	Generic CPS	S	OS	S	L	[132]	[133]
Reavers & Morris	V	Georgia Tech	Georgia, US	Generic ICS	S	OD	S	L	[134]	-
RICS-el	V	FOI	Sweden	Power Grid	S	OD	S	L	[135]	-
SCADA-SST	V	KFUPM	Saudi Arabia	2 different ICS	S	OS	S	L	[136]	[137]
SCADASim	V	RMIT University	Australia	Generic ICS	S	OS	S	L	[138]	[139]
SCADAVT	V	RMIT University	Australia	Water Distribution	S	OD	S	L	[140]	-
Singhet et al.	V	C-DAC	India	Power Grid	S	OD	S	L	[141]	-
TASSCS	V	University of Arizona	Arizona, US	Power Grid	S	OD	S	L	[142]	-
VTET	V	SKL-MEAC	China	Chemical Process	S	OD	S	L	[69]	-
Wang et al.	V	Tsinghua University	China	Generic ICS	S	OD	S	L	[143]	-

Furthermore, the Industrial testbed must include the most common industrial devices installed in the real world ICSs and supporting the most used protocols. Also, it is crucial to consider different versions of devices, knowing their different security features [93], [144]. The testbed should also include the different vulnerabilities that could, however, lead to a bias in the attack strategy vector.

- *Replication in Safety:* The physical processes controlled by ICSs are wide different, ranging from manufacturing processes to critical nuclear plants. The most delicate processes cannot always be replicated in a scaled-down version inside a laboratory. Furthermore, during attacks targeting the process's stability, even the less critical operation can express important safety issues [144].
- *Complexity:* Industrial systems devices can be hard to configure and maintain due to their specificity and because they are designed to perform a precise and unique task. It is also challenging to find IT experts who have the needed knowledge to manage and maintain a complex ICS containing several OT specifications. The maintenance requirements must be considered from the early design stages since the increasing complexity can become even more expensive and difficult to manage [144].
- *Cost:* To build physical industrial testbeds, research groups have to deal with building and maintenance costs. Expenses are one of the main reasons why there are not many testbeds available for research, and the ones that exist are generally not easily accessible by everyone. To overcome this problem, virtualized and emulated solutions are relatively diffuse in the field, even if they cannot provide the same fidelity and replication accuracy.
- *Lack of Documentation:* Another challenge in ICS research is the lack of documentation of the existing systems. Companies do not share internal information related to their system's architecture, the devices implemented, or the devices' software version. This is primarily due to the companies' privacy concerns, protection of intellectual proprieties, and security reasons. In fact, if a company discloses the presence of legacy devices with well-known vulnerabilities, it can attract several malicious actors' attention. This absence of documentation made the implementation of effective real-word testbeds difficult. Furthermore, the lack of documentation can be problematic for the in-laboratory testbed. If poor documentation is provided, new researchers who start to work on a testbed might spend much time understanding the system's behavior and components and have a concrete idea. To provide exhaustive documentation, it is essential to write it step-by-step during the testbed building process, avoiding writing it after the testbed is entirely built, which can be difficult and not cost-effective [103].
- *Reproducibility:* Due to the complexity of an ICS, it could be challenging to reproduce the experimental conditions of another research to replicate the results or test other solutions. The differences between the original conditions and the reproduced one can be minimal but, in some cases, can be sufficient to lead to different results. To facilitate the deployment, experiment-management systems can help researchers with the setup and the management of a testbed (e.g., [145]) by using a template or code generation. Moreover, scripts for auto-configuration of an emulated testbed can be offered by developers (e.g., [132]) to simplify the sharing process. However,

suppose the testbed is composed of physical processes and components. In that case, it could be difficult to perfectly replicate them since many external variables can influence the system behavior (e.g., the temperature, the pressure) [146].

- *Scalability:* If expanding a simulated or emulated testbed is generally straightforward, doing it with a physical testbed can be challenging. Real devices are expensive, and researchers are not always able to afford them. Alternatives to expand physical processes are Hardware-In-the-Loop (HIL), i.e., mathematical representations of physical processes inserted in the chain. HILs offer great scalability of the system even if generally they are not advisable due to the lack of accurate mathematical models. A cheap way to add new devices is to employ software simulations. Software simulations are cost-effective solutions with the drawback of less precise and reliable physical representation. To provide system scalability and intelligent reconfiguration of all the physical devices implemented, virtualization and VLANs can be an excellent solution to be implemented in ICS without any substantial disadvantages [144].
- *Data Collection:* A not trivial aspect of building a testbed is the data access and recording. It is generally a manual process, but it is vital to develop strategies to automate the collection precisely, providing reliability and synchronization between the different data collection points, for example, by introducing a central historian server.

D. Physical Testbeds

Ahmed et al. [97] presented a physical testbed built at the University of New Orleans, which models three industrial processes on a small scale but by employing real-world equipment such as transformers and PLCs. A small gas pipeline that transports compressed air was built using a pipe fed with an air compressor. A valve regulates the other end of the pipe. Instead, the second system is a power transmission and distribution that carries electricity from power generation sources to individual consumers. This system is composed of a power station and four substations. Finally, the third system developed is a wastewater treatment system composed of sedimentation, aeration, and clarification processes. All the systems are installed at the top of a trolley, making the testbed easily transportable and particularly suitable for pedagogy and research. Each system is controlled by one PLC connected through a switch to a historian and an HMI. This last device makes it possible to visualize and control the systems. The industrial protocols employed are Modbus, EtherNet/IP, and PROFINET.

Electrical Power and Intelligent Control (EPIC) [100], [101] is a high-cost 72kVA electric power testbed that mimics a real-world power system in small scale smart-grid, and it is available for rent. The testbed is shown in Figure 8 and it is composed of four stages, namely: Generation, Transmission, Micro-grid, and Smart Home. Each stage is controlled by PLCs connected to a master PLC using switches and then to a SCADA gateway. The physical process is entrusted to two motor-driven generators, photovoltaic panels, and a battery. Communications occur using the IEC 61850 standard protocol for the electrical substation and automation system that runs over TCP/IP stack. The authors also present false data injection attacks, malware attacks, power supply interruption attacks, and physical damage attacks, together with possible mitigation techniques. The testbed resides at the Singapore University of Technology and Design (SUTD), and it is used to supply power to two other testbeds inside the same institution (i.e., SWaT [110], and WADI [114]) to create also the possibility for research related to a cascade-connected ICSs. The authors also shared a related dataset, which will be analyzed in Section VII.

HAI Tesbted (HIL-based Augmented ICS) [102], [147] is an extensive and expensive interconnection of three independent real ICSs coordinated by a real-time Hardware-in-The-Loop (HIL) developed at The Affiliated Institute of ETRI, Republic of Korea. Emerson's boiler control system, GE's turbine control system, and FESTO's water treatment control system are built-in small-sized by employing components used in industrial environments. The HIL is used to simulate the power plant to combine the three control systems and form an integrated power generation system. The interconnection employs Ethernet at Level 2, while different proprietary Fieldbus versions are used to communicate with the field devices [148]. The authors' developed a tool to schedule HMI tasks for long periods without human intervention. This tool also helps to schedule attacks (e.g., MitM attacks) only when a particular ICS state occurs. In [102] the authors present various physical attacks targeting the pump and the pressure of the boiler system. An expansion of the testbed [147] was built to make it possible to launch also network attacks using tools like Nessus or Acunetix.

BU-Testbed [99] is a physical reproduction of two power generation systems developed at Binghamton University. The first one is composed of an AC motor directly coupled to a permanent magnet DC motor, generating up to 400V. The other one instead contains an AC motor used to drive a 12-volt DC blower motor used to generate electricity. The testbed also includes two types of Alley Bradley PLCs and a private computer with an LCD monitor used as HMI. The communication uses the EtherNet/IP protocol. Furthermore, the authors explain some cyber-physical attacks which are practicable on the testbed. These attacks regard different categories: 1) attacks on networks (i.e., MitM, DNS poisoning); 2) network congestions and delay (i.e., DoS); 3) attacks on controllers, sensors, and drivers (i.e., malicious software injection and firmware modification); and 4) attacks on HMI and programmable stations (malware injection). In another work [149], Korkmaz *et al.* presented a similar testbed in which the vulnerability to time delay attacks has been evaluated. Results show the feasibility of such attacks, which can stop the power generation process and shut down the testbed.

Lancaster's testbed by Green *et al.* [103] at the Lancaster University is a big physical scaled-version of a generic industrial ICS (the testbed does not explicitly the physical processes involved). It is composed of six Manufacturing Zones, a DMZ, and an Enterprise Zone. Each core zone is split at the network level using VLANs. The legacy serial-based communications

Fig. 8. EPIC Testbed by iTrust in Singapore.

Fig. 9. SWaT Testbed by iTrust of Singapore.

have been upgraded to IP to reduce the complexity and allow communications with a vast number of ICS devices. The connections are almost all physical, apart from two manufacturing zones connected using 3G, 4G, and satellite communications. To account for a changing landscape and to add flexibility, all the desktop and server-based software applications run inside a VMWare vSphere server as virtual machines. The authors are continuously improving the testbed to make it more usable and more complete. Students and researchers of the university use the testbed, but the authors also plan to make it more available for external researchers.

Morris *et al.* [106] at the Mississippi State University built seven different small physical testbeds for security research and pedagogy purposes. Five of them have communications based on Modbus/ASCII, Modbus/RTU, and DNP3 (henceforth called **Mississipi Serial**) and represent respectively: 1) a gas pipeline used to move petroleum products to market; 2) a storage tank used in the petrochemical industry; 3) a raised water tower used to provide pressure in the water distribution system; 4) a factory conveyor belt control system, and 5) an industrial blower used to force air through an exhaust system. These five systems are controlled by the same HMI but on different screens. It enables the control of all the systems from the same point and simulates a more extensive system by making them operate simultaneously. The remaining two testbeds are connected through an Ethernet network (and then are called **Mississipi Ethernet**) and include: 1) a steel rolling operation; and 2) a smart grid transmission system. The authors also use the testbeds to generate datasets that are freely available online [150].

Smart Grid Test Bed (SGTB) [109], [151] deployed by Idaho National Laboratory is the world's first full-scale replication of a smart grid, and it is part of the United States National SCADA Test Bed Program. It is a 61-mile transmission massive testbed connecting twelve facilities with power distribution networks that can selectively operate at various voltages (12.47kV, 24.9kV, and 34.5kV). Portions of the power loop can be isolated and reconfigured for independent, specialized testing. As planned in 2017, the authors obtain more funds to expand SBTB with a SCADA testbed to be installed in the command and control shelter to allow operators to observe, manage, and manipulate test line configurations and record testbed operating parameters. However, to the best of our knowledge, the authors never release updates about the project. This testbed is not an ordinary scaled-down version of real systems. Instead, SGTB is a full-size plant. Even if it represents an impressive and valuable work, unfortunately, students and researchers have limited access to such a facility [107].

SWaT [110], [152] is a six-stage water treatment plant developed by the Singapore University of Technology and Design (SUTD) represented in Figure 9. One PLC (plus one for backup) controls each stage, and the overall testbed leverages a distributed control approach. Furthermore, through a Human-Machine Interface (HMI), an operator can manually control all the system components. Communication between PLCs and sensors/actuators is based on Ethernet ring topology, while PLCs communicate with each other through a separate network based on an Ethernet star topology. The protocols implemented in the systems are EtherNet/IP and Common Industrial Protocol (CIP). In the paper, the authors implemented various attacks to manipulate plant operations. The different attacks leverage different assumptions on the attack model. In particular, the attacks are categorized as single-stage attacks, targeting a specific stage process and multi-stage attacks, which combine the compromising of various stages. Furthermore, each attack may target a single system point or multiple system points. The attacks include different scenarios (e.g., an attacker with access to the local plant communication network or an attacker who is on-site and has physical access to the device) and different types of attacks (e.g., MitM, eavesdrop, or packets modification). The testbed is accessible only for collaborations or by renting it. Recently, a python-based software simulation of the testbed was developed and released with open-source code [153], [154]. Also, datasets based on different data collection are openly available upon request. These datasets contain both network and physical packets in normal behavior and with the system under attacks [155]. We present the dataset in Section VII.

Teixeira *et al.* [112] implemented an ICS testbed to model a simple water storage tank's control system. The storage tank is equipped with two-level sensors to control the water level. When it reaches the maximum level, the upper sensor sends a signal to the PLC, which turns off the water pump used to fill the tank. At the same time, another pump is activated to draw water from the tank. When the water reaches the lower sensors, a signal is sent to the PLC, which will reverse the two

pumps' state to fill up the tank again. The SCADA system gets data from the PLC using the Modbus protocol and displays them to the system operator through the HMI interface. To complete the study, the authors tested some attacks such as scanning, device identification, and not authorized read of actuators. By recording SCADA network traffic for 25 hours, a dataset has been released [156] and will be presented in Section VII. In 2019, minor improvements of the testbed had been presented [157], such as embedding a turbidity sensor and a turbidity alarm to add analog input to the system.

Turbo-Gas Power Plant (T-GPP) testbed [113] is an experimental platform presented by Fovino *et al.* at the Joint Research Centre of Ispra (Italy) to perform security research on a SCADA system. It is a physical testbed that replicates a power plant's dynamics process and its control systems providing additional mechanisms for running and analyzing the system. The testbed is composed of seven different functional elements: 1) Field Network, used to link PLCs with the SCADA servers, actuators, and sensors; 2) Process Network, that interconnects the different physical subsystems; 3) Intranet, the internal private network connecting PCs and server of the company; 4) Demilitarized Zone, used to separate IT area from OT components; 5) External Network, such as the Internet; 6) Observer Network, a network of meshed sensors to gather a massive quantity of raw data useful for the analysis; and 7) Horizontal Services Network, used for the management of the laboratory. The paper profoundly analyzes such systems' vulnerabilities, highlighting those related to the protocols implemented (i.e., Modbus/TCP and DNP3), and describes various attacks deployed on the testbed: DoS, worm, and malware infection on the process network, phishing attack, and local DNS poisoning. Finally, the authors propose different countermeasures to the attacks.

WADI [114], [158] is a scaled version of a water distribution testbed build by the Singapore University of Technology and Design (SUTD) to perform security researches. It consists of five stages controlled by three PLC and two RTU, which can supply 10 US gallons/min of water. The communication happens using Modbus/TCP protocol at Layer 0, while at Level 1 network between PLCs uses TCP over Ethernet instead of RTUs that exploit High-Speed Packet Access (HSPA) using GPRS modem to generate a precise real-world scenario. The authors also implemented different attacks against the testbed by manipulating data from sensors to cut off the consumer tank's water supply. The system is physically connected to SWaT, and it can be used to generate a more accurate scenario and study the cascade effects of a cyber attack on connected ICS. Furthermore, WADI is available to organizations for joint research programs and usage, but a dataset generated upon request is available. We will analyze the dataset in Section VII.

In 2014, Yang *et al.* [116] proposed a physical SCADA power grid testbed specifically designed to test their detection approach. At the control network level, the testbed is composed of an HMI, a database to log events and data, a host used to perform the attacks, and different networking components (e.g., protocol gateway, switch, firewall, router). Instead, the physical network is composed of various IED simulated, connected to a real photovoltaic system. The connections between the Gateway and the IEC devices are based on the IEC 60870-5 series protocol, and then the Gateway translates the IEC 60870-5 to allow the communication with the HMI station. The IDS proposed by the authors was installed between the HMI and the Protocol Gateway. It monitors all the incoming connections to the substation and the LAN network through port mirroring.

Zhang *et al.* [117] presented a security research on a physical process ICS testbed which simulates a two-loop nuclear power system. The primary loop includes a 9kW heater representing the reactor core, controlled by the SCADA master through an open-loop controller. It also contains a variable speed coolant pump, upper and lower delay tanks, and other instrumentation such as a flow meter and temperature detectors. The secondary loop is composed of valves, a magnetic flow meter, and two temperature detectors. The SCADA system consists of an engineering workstation as the SCADA master and a National Instruments chassis used to read data and control signal output modules as SCADA slave. The system is completed with data storage and an attacker machine with Kali Linux. LabVIEW was installed on the engineering workstation to record sensor data and send control commands to actuators. In the same paper, the authors proposed some attacks to the testbed (e.g., MitM, DoS). Furthermore, they implemented some intrusion detection mechanisms based on Random Forest (RF), k-Nearest Neighbors (KNN), and Auto-Associative Kernel Regression (AAKR).

E. Virtual Testbeds

Davis *et al.* [118] is a power grid simulated testbed based on a client-server paradigm. The client mimicked a control room's graphical interface containing SCADA data and used it to control power elements. Each client can switch between different servers to monitor several systems from the same machine. The most common operating systems support client software. On the other hand, the server is based on the PowerWorld [159] simulator and can model a complex power grid. The server sends the process data to the client via a custom TCP/IP protocol, converted to Modbus/TCP using an integrated protocol converter. Furthermore, the simulator can connect and interact with real hardware devices, but it is not mandatory. The network is emulated using RINSE [160] which allows clients to launch different commands to simulate attacks (e.g., DoS attacks), defense techniques (e.g., filtering), diagnostic tools, device controls, and simulator data. The authors present various attacks, such as DDoS and network overload, comparing the results with and without security measures. To the best of the authors' knowledge, the testbed is not available online.

In [119] the authors present **Damn Vulnerable Chemical Process (DVCP)**, an open-source framework developed for cyber-physical security experimentation based on two models of chemical processes. In particular, the framework includes **DVCP-TE** and **DVCP-VAC**, two simulated ICS testbed based respectively on Tennesse-Estman [161] and Vacuum-assisted closure (VAC) [162] chemical processes simulated with MATLAB. The authors use these simulation models in

hybrid scenarios with the simulated process and real industrial hardware (i.e., SIMATIC S7-1200+KTP400 Starter Kit). Furthermore, the Modbus and PROFINET protocols were implemented to enable communication between the simulated process, the PLC, and the HMI. However, for this implementation, the authors did not share any code or further implementation information.

Genge *et al.* [123] proposed a framework based on Emulab [163] for the emulation of the components and to Simulink [164] for the physical processes simulation. The architecture comprises three layers: the cyber layer containing the regular emulated ICT components used in SCADA systems, the physical layer providing the simulation of physical processes, and the link-layer to connect the cyber and physical layers through the use of a shared memory region. The paper provides a qualitative comparison with other works, comparing the testbed with other related projects. Results show high performances in all the functionalities considered (e.g., repeatability, safe execution), except for the physical layer fidelity, where physical testbeds perform better. Furthermore, estimating the cost to build and maintain a physical testbed is compared with the predicted expense related to the presented framework, showing considerable savings through the years. A peculiarity of this framework is the possibility to attack the different components using specific malware. For instance, as a case study, the authors present Stuxnet [7] on a boiling water power plant, showing its effectiveness. Another attack example targets a chemical process by deleting and delaying some packets and changing the process parameters to reach their shut-down safety limits. There are many supported protocols such as Modbus, Profinet, and DNP3. The testbed, implemented in C#, is not available online to the best of the authors' knowledge.

Giani *et al.* [124] developed a virtual SCADA testbed for security-related researches purposes. However, this work represents a preliminary study presenting the testbed at a high-level, but without a practical implementation description. At the center of the architecture, there is the SCADA master station containing the SCADA server and the HMI. The SCADA master station containing the SCADA server and the HMI is placed in the architecture center. SCADA master servers run the server-side applications that communicate with the RTUs using different strategies: dial-up modems, private leased line, wireless or radio channel, and LAN/WAN links. The most used protocols for these communications are Modbus and DNP3. The SCADA server is also connected to the corporate network, connected in turn to the Internet, exposing the system to vulnerabilities, such as unauthorized remote access. The authors planned to employ a single simulation-based instantiation to build all the testbed elements in the same machine using software like Simulink [164]. Other implementation strategies for the system architecture are possible, like the federated simulation-based, where a different machine simulates each element. Moreover, emulation-based and implementation-based instantiations that use actual commercial SCADA devices along with simulation and emulation of software modules, networks, and physical processes are depicted but not implemented. Finally, the authors depict various possible attacks (e.g., DoS, integrity attacks, phishing attacks) and suggestions about security mechanisms. To the best of our knowledge, the testbed is not publicly available online.

GRFICS [125] is a graphical and open-source [126] ICS simulation tool based on the Tennessee Eastman process (Figure 10). Currently, the testbed is designed for educational purposes and allows only the use of pre-defined functions. The ICS devices are simulated. In particular, the OpenPLC [165] is used for the PLCs, and the HMI Virtual Machine simulated an HMI using AdvancedHMI [166] software. The testbed allows running many pre-defined attacks such as MitM, Command Injection, False Data Injection, Reprogramming of PLCs (i.e., Stuxnet), Loading Malicious Binary Payload (i.e., TRITON), and Common IT attacks (i.e., password cracking, buffer overflow). Once the attacks are launched, the interface allows monitoring the testbed attacks' consequences, log the process information, and how much cost is wasted through the purge. Finally, the testbed allows the installation of the Snort detector [62] and to customize it with new rules. The communications on the testbed are based on Modbus protocols.

Maynard *et al.* [130] proposed **Maynard SCADA**, an open-source, scalable framework for deploying a replication of a SCADA network. The testbed is composed of a collection of scripts used to build and configure virtual machines that, by default, are emulated using Oracle VirtualBox [167]. The resulting network can also support and integrate the connection with physical devices. Maynard SCADA supports IEC 60870-5-104 (IEC104) and OPC Unified Architecture (OCP-UA) to support additional industrial protocols such as Modbus or IEC 61850. The framework implements two types of profiles: an operation profile, which defines the deployment of nodes, simulators, and configuration of the network; and a configuration profile to configure nodes to represent specific industrial devices (e.g., HMI, RTU). Such profiles can be developed by the community, adding new use cases and simplifying the testbed's deployment. The framework does not consider the physical process simulations, but it can be easily integrated using third-parties software (e.g., Simulink [164]). Furthermore, the paper [130] shows a common metering application using seven virtualized nodes with detailed instructions to replicate it. The instructions also include an accurate description system's requirements and a comparison between some other testbeds in the same document. The framework is entirely open-source, and it is accessible on GitHub [131], where are also available some datasets.

MiniCPS [132] by Antonioli and Tippenhauer is a toolkit used to create an extensible and reproducible research environment for network communications, control systems, and physical layer interactions in CPS. MiniCPS is an extension of Mininet [168], a widespread network simulator built around the Software-Defined Networking paradigm that exploits lightweight system virtualization using Linux containers. Connections between simulated devices are emulated using virtual Ethernet links with an easy drag and drop interface. These connections can be configured through Linux Traffic Control to emulate link performance such as delay, loss rate, and bandwidth. MiniCPS extends the classic Mininet by

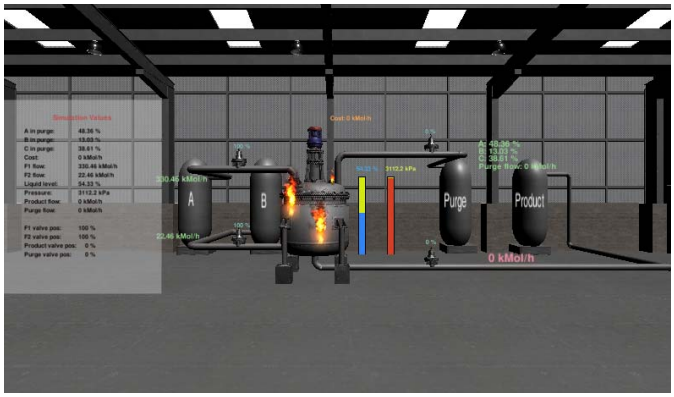


Fig. 10. Example of GRFICS simulator rendering.

implementing ICS components such as PLCs and allowing the connection with real physical devices. The testbed is not focused on the physical process simulation that can be implemented using third-parties process simulation engines (e.g., Simulink [164]). The reproducibility is a significant advantage of this testbed: it is possible to write Python scripts that generate a complete ICS environment easily exportable and shareable. On the top of the emulated Ethernet network, the testbed includes different industrial protocols, in particular, using the *CPPPO* Python library, MiniCPS implement, for example, EtherNet/IP and Modbus/TCP. The paper [132] accurately describes all the design decisions and the consequent strengths and drawbacks of the testbed. The authors present an attack scenario of a MitM attack on a replicated model of the SWaT testbed [110], also providing different countermeasures based on a custom SDN controller. The testbed and its documentation are open-source and available on Github [133].

Morris *et al.* [122] presented a virtual gas pipeline system (called **Gas Pipeline testbed**) that is a simulation of a testbed previously built. The testbed consists of four components running in different virtual machines: a virtual physical process, a Python-based PLC simulation, a network simulation, and an HMI. The various components communicate through Modbus/TCP over a virtual network and may be connected to real devices. The virtual system allows modeling a pump, a valve, a pipeline, a fluid, and a fluid flow. The models are based on a previous physical testbed [106], allowing to compare measures from the two testbeds. The virtual testbed mimics the physical device's behavior but with some difference in pressure change frequency. Also, the startup process is similar but not identical. The authors present a command injection attack to the virtual testbed, but the resulting behavior is not compared with the physical testbed. To the best of our knowledge, this virtual simulator is not publicly available online.

Reavers and Morris [134] develop an open and complete platform for creating virtual testbeds. The resulting system is highly scalable, and it is possible to install plenty of different virtual devices. The testbed's main components are process simulators, data loggers, and configuration files used to configure virtual devices and connections among them. All the simulations are implemented with Python without adopting off-the-shelf network simulation tools. The process simulator includes four components: 1) a simulator module, 2) a communication interface, 3) an update queue, and 4) configuration files. The simulator communicates directly with the virtual test devices via a "backchannel" to transmit measurements and inputs. The virtual devices supported are RTU, MTU, IED, PLC, repeaters, and Programmable Automation Controllers (PAC), which can run as standalone processes or inside virtual machines. It is possible to connect physical devices such as wireless radios and HMI. There are two main protocols for the communications: Modbus/TCP, which can be logged using standard applications such as Wireshark or tcpdump; and Modbus/RTU, which instead needs a PortLogger, a class of proxy that reads the communication and then resends it to the channel. In the paper [134], the authors present two testbed applications. The first represents a gas pipeline, while the second models a water storage tank control system. To verify the simulated data's consistency, the authors implement these two testbeds as physical ones. Furthermore, to obtain more relevant results, the authors decided to compare the virtual and physical testbeds' behaviors during different attacks (e.g., data injection and DoS). Results show that the attacks are effective in both scenarios, with some slight variations on the time needed for the attack to succeed. The paper also compares both the virtual and physical systems' normal behavior discovering many similarities, but with some detectable differences. The study's conclusion states that the virtual testbed is good for proof-of-concept, but a physical testbed is needed in some cases. To the best of our knowledge, the virtual platform is not publicly available online, but some datasets are available [150]. Starting from this work, Thornton and Morris in 2015 [169], deployed a similar platform that permits the usage of Simulink [164] instead of Python to simulate the physical processes.

RICS-el testbed [135] is a virtual testbed representing a power system built on top of the Cyber Range And Training Environment (CRATE) infrastructure at the Swedish Defence Research Agency (FOI) [170]. All the hosts of the testbed are run on virtual machines using VirtualBox [167]. Researches and vendor experts designed the OT segment. It is divided into the OT DMZ, the OT LAN, the substation communication WAN, and the power grid simulator, including all the RTUs. In the OT DMZ, there are the FTP server, the HMI, and the historian. The WAN is used to enable communication between the 15 hosts. Three of these hosts are RTUs that communicate with the front-end through the IEC 60870-5-104 (IEC104) protocol. The power grid simulator is the key component of the architecture: it can generate realistic traffic and event in the whole RICS-el environment. In detail, this testbed simulates a backbone high voltage 400kW grid with twenty substations and some medium voltage transmission. Finally, to add more realist to the environment, the system is connected through another DMZ to an office IT segment. It contains a LAN to interconnect 17 office workstations, nine sales workstations, and some other servers. Ongoing work is focusing on adding realistic traffic to each segment by emulating users and different scenarios.

SCADASim [138] is a simulator for SCADA systems created on top of OMNET++ [171]. The testbed is developed to

satisfy specific requirements: 1) it allows plug-n-play to create simulations to allow system experts to set up the software; 2) allows connectivity to multiple external hardware or software that can be used to expand the simulator; and 3) supports multiple industry-standard protocols such as Modbus/TCP, DNP3, and the integration of proprietary protocols. The simulator contains modules for the emulation of ICS devices (e.g., RTU, PLC, MTU, HMI) and components to implement different attacks (DoS, MitM, spoofing, eavesdropping). SCADASim architecture includes three components: 1) a real-time scheduler; 2) a communication port implementing protocols for communication to the external environment; and 3) a simulation object that models external components within the simulation environment. For the evaluation, the authors present two simulations: a smart meter and a wind power plant. A DoS attack and a spoofing attack are also deployed on the systems and analyzed in the paper. SCADASim is an open-source project available on Github [139].

SCADAVT [140] is a framework to build a virtual SCADA model-based testbed designed for research in security field purposes. The framework is developed on top of the CORE emulator [172] by integrating the Modbus/TCP communication protocol between master, slave, and HMI server. Simulations of I/O modules are also integrated into the CORE emulator, which acts as a server, receives input data from the external environment, and sends output data when requested using a simple custom TCP-based protocol. The physical process is modeled using an EPANET server [173] which provides a graphical interface to reproduce water distribution systems. Two attack scenarios are also presented: a DoS and manipulation of command messages. The framework, which supports real devices' connection, is described in detail, but the source code is not publicly available.

TASSCS (Testbed for Analyzing Security of SCADA Control Systems) [142] was developed by the University of Arizona mainly to test a novel technique to protect SCADA systems from attacks, which the authors called Autonomic Software Protection System (ASPS). The testbed architecture is composed of different components. The Control HQ is the central command and control for all the resources and services offered. It contains the HMI, the control server, the data storage, and the engineering LAN. Through a WAN, the Control HQ is connected to a large scale electric grid modeled using the PowerWorld simulation tool [159]. Finally, a device is used to monitor all the ingress and egress communications that pass through the WAN to feed the ASPS. This last device acts as an active anomaly detector: it can identify attacks and stop them. To enable communication, Modbus/TCP is used. The Modbus Server is simulated using Modbus RSim (the official website is not anymore available) and connected to an Opnet-based network simulator [174]. The authors present various attacks, including spoofing, MitM, DoS, and data injection. Two of these attacks scenario are also implemented: a DoS attack and a compromised HMI scenario to force a complete network blackout. The protection system under testing was able to detect the two launched attacks.

Virtual Tennessee-Eastman Testbed (VTET) [69] is a simple virtual testbed that simulates a chemical ICS with MATLAB.

Fig. 11. A figure representing CyberCity testbed.

The architecture is based on four components: a physical PLC, a PC used for network communication, and two other PCs simulating the physical process and a PLC. The process is the Tennessee-Eastman (TE) [161], a nonlinear and continuous process widely used in the chemical field. VTET can work in two different modes. In the full-virtualization mode, the physical PLC is disconnected, and the testbed is completely virtual and simulates the controller using NetToPLCSim [175] and PLCSim [176], the official simulator of Siemens PLC. Instead, the semi-virtualization mode allows replacing the simulated PLC with the real one. VTET supports three standard ICS protocols for network communication: Open Platform Communications (OPC), Modbus, and S7Comm. The authors present and test five attacks, mainly using MitM and jamming techniques to disturb or disrupt the physical process. Unfortunately, the testbed is not available online to our knowledge, but the description is quite complete on the paper.

F. Hybrid Testbeds

CyberCity Testbed [75], [177] is a physical representation of an entire city (Figure 11) developed by the SANS Institute to test security measures on the ICS field. It includes a bank simulation, a hospital, a power plant, a train station, a water town, and many other available infrastructures. Furthermore, 15k "people" who have e-mail accounts, work passwords, and bank deposits are generated to create a complete environment. A tabletop scale model of the town was built to visually show the effects of attacks on the electric train, the water tower, and the traffic light. Although the lack of official documentation, Hink and Goseva-Popstojanova in [75] recover some details about the components of the testbed by studying a dataset generated from CyberCity, which is available online [178]. They discover a wide variety of components ranging from Web servers emulated using VMWare to physical Siemens PLCs, Cisco routers, and NetDuino+ controllers. The protocols used by the ICS components are mainly Modbus/TCP, EtherNet/IP, and NetBIOS. Nowadays, the testbed is mainly used to teach cybersecurity on ICS as part of the SANS Institute courses and federal agencies to perform security research.

Experimentation Platform for Internet Contingencies (EPIC) by Siaterlis *et al.* [76] is an innovative hybrid testbed to simulate CPSs based on Emulab [163]. It is developed by the Joint Research Center at the Institute for the Protection and Security of the Citizen in Ispra, Italy. The testbed architecture comprises two control servers, a pool of physical resources used as experimental nodes (e.g., PCs, routers), and a set of switches employed to interconnect the nodes. Every configuration step uses a Web interface where a user can create a customized network. The EPIC setup phases require a detailed description of the required topology using a formal language (i.e., an extension of Network Simulator (NS) language). The experiment is then instantiated by using Emulab [163] which can automatically configure network switches to recreate the desired virtual topology by connecting nodes using multiple VLANs. Finally, experiment-specific software can be launched through events defined in the setup script or manually by logging in to each station. Physical processes are simulated using Simulink Coder [179] and managed by a software simulation unit. For communications, EPIC provides tools to generate latencies for the simulation of different network types and integrate realistic background traffic datasets. It also supports industrial protocols such as Modbus through proxy units that translate calls between the simulation unit and other SCADA devices. Furthermore, after a theoretical comparison between fidelity, repeatability, and measurement accuracy between EPIC and other popular testbeds, these characteristics are analyzed on EPIC with a deep testing phase. The software part is open-source and freely available online [77] with complete documentation.

EPS-ICS [78] is a framework to implement a hybrid testbed, principally developed by the Technical Assessment Research Lab (CNITSEC) in Beijing, China. The testbed implements a multi-level design approach where Level 3, the corporate network, and Level 2, the supervisory control LAN, are emulated. Instead, Level 1 devices, including Distributed Control Systems (DCS) controllers, PLCs, and RTUs, are real physical devices. Finally, a mathematical model is used to simulate the physical process at Level 0, and it is implemented with Simulink [164]. This approach allows replicating the interactions between the ICS components. The communication interface between network testbed and physical devices is implemented through layer three switches with an IP routing. However, the industrial protocols used are not mentioned in the relative paper.

Gillen *et al.* [79] presented a hybrid replication of the cooling system for Oak Ridge National Laboratory's 200-petaflop Summit supercomputer, currently declared the fastest open-science computer in the world [180], [181]. Summit consists of over 4600 nodes and has a peak power draw of 13MW. The cooling system cycles through over 4000 gallons of water each minute. The developed replica is based on the same controller, an Allen Bradley Control- Logix PLC with 34 I/O modules distributed over six chassis connected with an Ethernet/IP ring-topology backbone. Furthermore, the HMI, the historian, the industrial switches, and the power supplies are perfect physical replicas. An engineering workstation is connected to the system to configure the different components. On the other side, the over 500 sensors and actuators employed in the cooling system are instead emulated by using over 40 Raspberry PIs and 200 daughter boards. All the sensors and actuators communicate with the PLC using hard-wire electrical signals or an Ethernet-based signal line. For this last case, raw traffic from the production environment has been recorded. A software-based model of the protocol, traffic rate, and handshakes of the real cooling system was computed and employed by the Raspberry PIs to emulate the entire communication. The authors collected 30 days of data from the real Summit cooling system historian to correctly emulate sensors and actuators. Then they used emulation scripts to generate data from the devices. Each sensor and each actuator is connected to an independent display used to verify the correct measures, despite the HMI values. This is useful in the case of attacks targeting data visualization (e.g., replay attack). To validate the testbed, the authors compared its behavior with the real Summit supercomputer cooling system. Considering the alerts, logs, and historian data, all the data are replicated accurately. Instead, concerning the network traffic consistency, results show an hour-to-hour average variance under 0.01% for the majority of the properties. Therefore, the fidelity is adequately accurate to simulate the original system properly.

Henry *et al.* [80] introduce **Hui Nuclear**, a hybrid testbed modeling a nuclear reactor built at the Center for Secure Information Technologies (CSIT) at the Queen's University of Belfast. The testbed's scope is to generate a realistic network interaction and a simple way to collect network data to be used in the CPS security field. The testbed implements four main sub-process controlled by four PLCs. The main reactor sub-process, the heat exchanger sub-process, and the heat exchanger sub-process controlled by physical Siemens PLCs. Instead, the generator sub-process is monitored by a Schneider PLC. The inter-communications between sub-processes are enabled by physical interactions or IP network communications through S7Comm protocol, Profinet, and a custom protocol based on TCP. For practical and safety reasons, the heating process and the turbine are simulated by two Raspberry PIs. The network architecture is exhaustive and contains all the Purdue model areas, together with firewalls, IDS, and logging services.

HYDRA [81] is a low-cost and open-source physical emulator for critical infrastructures developed at the Université Roma Tre in Italy. It can be used for investigating fault diagnosis, cybersecurity strategies, and testing control algorithms. The testbed is designed to emulate a simple water distribution system's behavior. It employs seven tanks at the physical level deployed vertically. Each tank can be easily unconnected or moved to another position giving the testbed high modularity and flexibility. The communications between sensors and actuators implement the Modbus protocol on a Local Area Network (LAN) to PLCs and RTUs simulated using Arduino Nano and Galileo. The authors also present an attack scenario of a data modification attack. The code and all the testbed technical details are open-source and available on Github [82].

Kim et al. [85] proposed a platform to perform cybersecurity exercises for national critical infrastructure protection. The testbed was designed to replicate a realistic ICS environment

that matches the characteristics of the Cyber Conflict Excercise (CCE). CCE is an annual national real-time attack-defense battlefield competition organized in South Korea and Locked Shields (LS). It is the world's largest international technical live-fire cyber defense exercise. The platform can scale and provide dozens of identical ICS setups to satisfy an increasing number of participants. With respect to standard testbeds, this project required a visualization layer representing the physical facilities and the damage caused by the attackers. To make it possible, a diorama city was considered the most cost-effective and modular approach. It contains symbolic structures representing the critical infrastructures, surrounded by residential and commercial buildings, and tri-color LED lights to introduce a physical representation of attacks' effects. The paper [85] describes an implementation of the proposed platform, which includes six different critical infrastructures: a power grid, a nuclear plant, a water purification plant, railroad control, airport control, and traffic light control. The system contains two PLCs of different vendors that control some typical actuators (e.g., mechanical relay, magnetic switch, motor). Furthermore, a platform with 255 LED lights was built to illustrate the state of the critical infrastructures. The control network layer is hosted by remote cloud servers and contains HMI, an engineering workstation, a historian DB, a patch management system, and office computers. The protocol adopted depends on the selected PLCs.

In [86], Koutsandria *et al.* presented a hybrid testbed for testing a real-time Network IDS. To simulate the ICS environment, the authors employ a combination of simulated and real devices. The testbed is based on MATLAB Simulink to simulate the physical and control networks. In particular, the authors model the physical system with Simulink by simulating IED and field devices controlled by a PLC via Modbus in a master/slave communication model. In this setting, the authors describe the implementation of the master devices both with a SIMATIC S7-1200 PLC and a simulated PLC. The network communication and information exchanged by the different devices are collocated through a network tap implemented with a central hub and a Raspberry PIs [182] running a packet dissector. The authors gave particular attention also to the data management and visualization part. All the traffic collected is saved in a historian server and managed with OSIsoft [183] PI System. Then the historian information is continuously analyzed and monitored by an IDS based on rules and behavior analysis. To validate this architecture and its capabilities, the authors also present three attacks (i.e., two network communication alterations and a physical behavior violation) scenario showing the effectiveness of the detection rules.

KYPO4INDUSTRY [87] is a training facility for students based on open-source hardware and software, built at Masaryk University in the Czech Republic. This testbed consists of a laboratory room designed to help computer science students to learn cybersecurity in a simulated industrial environment. The laboratory is divided into different tables to split the students into groups and give everyone the possibility to have hands-on experience on the entire system. Tables can be moved and rearranged around the room to generate a flexible environment for every possible activity, ranging from team assignments to student presentations. A control panel exposes the I/O modules on each table, and the touchscreen is used to interact with PLCs (simulated using Raspberry Pi [182]), linear motor, and communication gateway. The software stack includes the Linux OS, Docker ecosystem, and on-premise OpenStack cloud environment to achieve an automated orchestration. Thank the open-source hardware and software used in the system, and different industrial protocols can be implemented, such as Modbus or DNP3. Finally, the paper introduces the university's course syllabus that employs the facility, showing the arguments addressed on each of the 13 weeks of the course.

LegoSCADA [88], [89] is a cost-effective hybrid testbed developed at the Universite Paris-Saclay in France. The testbed's conceptual architecture is based on three block elements: the controller, the system, and the sensors. The controller reads data from the sensors, computes new information, and transmits new commands to the actuators. Many RTU and PLCs can be connected to the controller based on the system that we want to represent. The protocols supported are Modbus and DNP3. To test the architecture, the authors have developed a test scenario based on Lego Mindstorms EV3 brick [184] which emulates a PLC on a car, a Raspberry Pi [182] to emulate an RTU connected to the vehicle, and a personal computer as a controller. The controller is always correcting the car speed and polling the distance between the car and an obstacle. Furthermore, a single RTU and a single controller can control more PLCs, and, therefore, more cars can be connected to the testbed. MitM attacks are deployed on the developed testbed, in particular replay attacks and injection attacks. Moreover, a watermark authentication technique has been tested to stop the attacks with interesting results.

LICSTER [104], [105] is an open-source and open-hardware testbed presented at the Hochschule Augsburg in Germany. Its main target is to give students and researchers an affordable system to perform security research with an expense of about 500 euros. The system is composed of an OpenPLC [165], an HMI built using a Web server and a SCADA system. Each of these components is loaded on three dedicated Raspberry PIs. The physical process implemented is a representation of an industrial process provided by Fischertechnik [185]. A conveyor belt is used to move a plastic cylinder to a punching machine, which is then activated. In the end, the cylinder is taken back to the original position. The process can be easily substituted with others. Modbus/TCP is the protocol used to enable communication between components. Different attack scenarios on LICSTER are presented and tested. The authors cover widely used threats to levels 0, 1, and 2, such as passive/active sniffing, Dos, MitM, and manipulation over the network. For each attack, an evaluation is presented containing useful information (e.g., impact, skill level, detection difficulty). Scripts and instructions on the implementation are available on the Github repository [105].

Microgrid [90] is a flexible and adaptable testbed developed by The Ohio State University, composed of a hybrid setup of physical hardware and real-time simulations. The testbed

contains Power Hardware-In-the-Loop (PHIL) able to emulate power hardware not installed in the testbed, along with a real-time SCADA system with an OPNET [174] based real-time System-In-the-Loop (SITL) communication network simulation system. PHIL can emulate several components like stationary battery unit, charging station, renewable energy resources, 9-bus or 14-bus systems. It is also possible to connect physical components to PHIL. The paper presents an implementation of a 5kVA charging system of a simulated electric vehicle, a photovoltaic system, local energy storage, and different power electronic circuits. The three main components of the simulated SCADA environment are the data acquisition, the real-time virtual communication network, and the real-time control center with the HMI. The authors introduce a case study implementation by connecting local energy storage and a second power grid PHIL simulation. Furthermore, the authors validate the case study with experimental results and analysis. The testbed is designed to study topics related to smart grids and provide hands-on experience to students.

MSICST (Multiple-Scenario Industrial Control System Testbed) [91] is a hybrid representation of four different ICS scenarios: a thermal power plant, a rail transit, a smart grid, and intelligent manufacturing. Physical processes are always simulated while the control systems are built using commercial hardware and software. Furthermore, in some scenarios, a combination of software simulation and actual physical equipment is used to build a more realistic scenario. MSICST also contains an attacker model and a monitoring network. The thermal plant comprises four PLCs of different manufacturers used to manage the three simulated physical systems: combustion system, steam-water system, and electrical system. A sand table is synchronized with the simulation to visualize what is occurring using Light Emitting Diode (LED), fans, and smoke generators. The rail transit scenario includes three stations, two trains, and a circular rail transit line. All the components are realized in a sand table as a scaled-down version of a real system. To achieve automatic control of trains and station components, the authors use two Siemens PLCs. Regarding the smart grid, the testbed mainly focuses on the power consumption part. It contains two smart meters, a concentrator, and a station device, which can, for instance, display the power consumption of an area. Finally, the intelligent manufacturing scenario is based on a Computer Numerical Control (CNC) that contains a controller, memory, and HMI. Moreover, a Distributed Numerical Control (DNC) system was developed to improve the manufacturing industry's intelligence level. A DMZ containing the data historian and the HMI server is generated to separate the OT area from the enterprise zone. This latter simulates an office by using a PC with Windows 7. The protocol used for communication in the OT area are mainly Modbus/TCP, S7Comm, and EtherNet/IP, depending on the PLC used. Some vulnerability discovery experiments have been done on MSICST, ranging from discovering vulnerabilities on a specific type of PLC to some attacks to known vulnerabilities of S7Comm and Modbus provided by the lack of encryption and identity authentication. Some security measures are presented as well, like a whitelist-based host protection software and a new IDS solution that combines traditional IT system IDS with behavior-based ICS-specific IDS.

NIST (National Institute of Standards and Technology) developed a cybersecurity testbed for ICS presented in detail in [92]. The testbed is designed to emulate three real-world industrial systems without replicating the entire plant or assembling a complete system. The first system is a Tennessee Eastman (TE) problem [161], a widely used process in the chemical manufacturing field. The TE process is simulated using an open-source code [186], and it is connected to physical devices such as switches, PLCs, HMI, and terminals through different protocols such as OPC, Ethernet/IP, and DeviceNet. The second is an entirely physical cooperative robotic assembly system for smart manufacturing. It contains a PLC, controllers, buttons for emergency stops, HMI, and two robots. These devices are interconnected through Ethernet, EtherCAT, Serial, Modbus, and Analog/Digital signals based on the components' needs. Finally, the third simulates a pipeline network with a Wide Area Network (WAN) SCADA infrastructure and an intelligent transportation system, including public infrastructure components, cooperative real-time embedded components, and wireless components. However, this last testbed was only introduced in the paper and was not implemented at the publication time (i.e., 2015). The testbed is available upon request to academia, government, and industry to analyze new technologies. Based on the research on these testbeds, NIST published a long and complete guide to ICS security in 2015 [2].

PNNL [93] by Edgar *et al.* at Pacific Northwest National Laboratory is a remotely configurable and community-accessible hybrid testbed to support research on cyber-physical equipment. This testbed combines physical, simulated, and virtual components giving considerable implementation flexibility. In fact, the testbed allows simulating from small systems like traffic lights to extremely complex scenarios such as power grids. The testbed is composed of many different back-end functionalities that can be employed in user management. Each user can remotely deploy its own system configuration and manage the operation and data gathering process. Furthermore, users can control different areas of the architecture, including the environment (used to simulate the physical process), devices (e.g., PLCs, RTUs), network communication (representing the backbone communication), simulation, and device integration. The testbed is accessible following the indications provided in the PNNL website [187] and using Arion [188] as modeling software.

SNL Testbed [95] is a complex hybrid testbed built by Sandia National Labs in Albuquerque, USA. It contains simulated components (i.e., represented using a model in OPNET [174]), emulated nodes (i.e., using real software running on an emulated machine), and physical (i.e., real software running in real hardware) devices. The reference paper also includes an accurate explanation concerning the connection between the various components. The testbed is presented as a case study used to model a complex scenario, containing: the corporate network (connected to the Internet), a DMZ, a control system network (containing HMI, the SCADA Server, Engineering Workstation, and Front End Processor),

and the field layer (containing sensors, RTUs, and IEDs). The protocols implemented are Modbus/TCP, DNP3, and IEC 60870. Finally, the authors present a security assessment of the testbed considering different threats and attacks such as reconnaissance, resistance to standard penetration tools (e.g., Metasploit [189]), and MitM.

VPST (Virtual Power System Testbed) [96] of the University of Illinois is designed to be integrated with other testbeds across the country to explore SCADA protocols and equipment's performance and security. Thanks to its easy integration with real devices and testbeds, VPST has the advantage of having actual HIL and a faithful communication system. The architecture is divided into three main subsystems: the first handles electrical simulation using PowerWorld [159], the second simulates the communication systems using RINSE [160], and the third includes all the actual devices. Furthermore, a framework for the Inter-Testbed Connection (ITC) is integrated with VPST. This framework is based on low bandwidth and reliable control plane and a high bandwidth data plane. ITC requires secure connectivity, which is achieved using OpenVPN and IPSec. Moreover, the implementation of performance, reproducibility, and resource allocation properties are addressed in the paper. Fidelity is another essential property achieved by implementing real industrial protocols such as DNP3 or Modbus, leaving the possibility of testing new versions and protocols (e.g., DNP3SA that provides Secure Authentication). The paper presents some example use cases: attack robustness analysis, incremental deployment analysis, and Human-in-the-loop event analysis. However, thanks to its flexibility, the testbed is suited for many different types of research.

VII. ICS Datasets

In this section, we provide a description of the ICS dataset available in the literature, highlighting the key design point and the most interesting and performant IDS applied to them. In Section VII-A we outline the classification method that we use in the following sections, while in Section VII-B we introduce the main requirements and challenges in developing a dataset. In Section VII-C we briefly recall the common evaluation metric for IDS. Then, in Section VII-D we present the datasets offering only physical level data, while in Section VII-E we describe network-level datasets. Finally, in Section VII-F we highlight datasets containing both the information.

A. Datasets Classification

Datasets are a collection of data recorded from a testbed or synthetically produces, which can be used to train and test an IDS. Unlike datasets concerning IT systems, which are composed only of network traffic, to characterize an ICS, a dataset must contain both network traffic, representing the communications between the various devices, and the physical processes' measurements.

Datasets are generally shared as csv, arff, or pcap format files, depending on the typology of data collected. An interesting solution introduced by Morris and Gao [190] consists of providing also some datasets containing only a subset of the data. They can be used, for instance, to quickly look at the data without downloading huge files or training a preliminary algorithm during the early stages of development.

There are many ways to categorize datasets. For example, Choi *et al.* in [23] groups datasets based on attack path. In this survey, we decided to divide datasets based on the typology of the collected data. The capturing can contain data at *physical level*, i.e., field data such as measures from sensors, actuator, and other physical level devices, or *network level* data, i.e., packet or flow sent in the channel under control. However, datasets can contain both the typology of data, and so they are considered both *physical and network level*. Sometimes, it is possible to find other types of data, like device logs, to better understand the ICS's behavior. To perform our study and provide reliable statistics, we downloaded every dataset and analyzed it reporting the main interesting properties.

Table IV summarizes the main features and statistics of the presented datasets. We reported the following features.

- **Name** of the dataset (or of the authors if a name is not provided);
- **Sector** indicates the field of the source ICS;
- **Data** type provided. Can be
  - *Logs* if logging information of the system during the process are available;
  - *Network* if network traffic data are provided;
  - *Physical* if measurements of sensors and actuator states are available;
- **Time** provide an approximation of the duration of the recording;
- **Entries** indicates an approximate number of entries contained in the dataset. In the case of datasets containing different versions, the most used or the most recent is considered;
- **Reference** includes a reference to a description of the dataset;
- **Resource** indicates a webpage in which the dataset is downloadable or information about how to retrieve it are available;
- **Attacks** specified the categories of attacks contained in the dataset, if any. Can be Reconnaissance, Replay, MitM, DoS, Injection, or Others which contains less used categories. More information about the attacks are presented in Section V-A.
- **%** indicates the percentage of data under attack on the total entries, if any;
- **Format** indicates the format of the files containing the capture. Can be:
  - `pcap` is a widely used format containing network packets;
  - `csv` is an extension for files containing Comma Separated Values;
  - `log` contains textual logging of events;
  - `xslx` is a format for spreadsheet files;
  - `arff` is a format used to save data for databases in a textual format. It is generally used with Weka [191];
  - `inp` contains data of emulations. In this context, it is generally used with Epanet [173].

- **IDS** contains a reference to the best IDS available in literature applied on the testbed at the best of authors knowledge;
- **F1-score**, **Accuracy**, and **Precision** represent the evaluation metrics of the IDS specified, according to Section VII-C.

The detection algorithms selected are implemented on the whole dataset and not on a fraction of it. Furthermore, the selection does not take into consideration the rank of the publication venue of the paper. For some datasets and IDSs were not possible to obtain all the information since the related paper does not provide exhaustive information. Thus, the degree of depth of analysis may not be the same for all work.

B. Datasets Challenges & Requirements

There are several challenges in generating a valuable dataset. Therefore it is fundamental to create it by following a suitable methodology and keeping in mind the design requirements. Gómez *et al.* [59] described a framework useful to generate reliable anomaly ICS datasets to be employed in anomaly detection tasks. Firstly, it is important to select a priori, one or more attacks that will be implemented. To do so, researchers must know the main protocols used in the field of interest, discover the related threat, and design attacks according to the related vulnerabilities. Then, attacks can be deployed, carefully choosing the nodes affected, each attack's duration, and its starting time. Finally, it is possible to capture network packets and/or data from sensors and actuators: it is essential to define the data capture duration, the sampling frequency and smartly choose the collecting point. Generally, the latter should be a central node of the system. The last step is the final dataset generation. To generate the dataset to release, it is important to carefully choose the features useful to describe the system under consideration. The behavior of the system can be represented at packet-level, flow-level, or physical-level data.

The **deployment of attacks** in datasets is probably the most challenging phase. In fact, if not accurately performed, the attacks generated can lead to an inaccurate system representation or bias in the detection methodology. There are principally two ways to generate attacks. The first one, and the most accurate one, is to attack the testbed in real-time, recording the corresponding network traffic or the ICS's physical state. Another strategy is to insert synthetic malicious data, a posterior, in a dataset with regular operation. However, this strategy could lead to inaccuracies and may not accurately represent the real system behavior response. In fact, if we want to inject packets on a dataset with normal operations, we must consider all the complex cascade relations of the systems. By breaking these relations, we would leave a trace that an IDS can exploit to detect anomalies, creating detection bias. Since this property is not present in real systems, the IDS will miss most of the attacks in physical environments, reducing the detection generalization in other systems. It is one of the main problems of Lemay and Fernandez dataset [192], which uses tools such as Metasploit [189] to inject malicious traffic.

Another critical concern causing the lack of available datasets from real environments (i.e., ICS of companies) is related to the collected **data's privacy**. In fact, companies may be reluctant to share their internal configurations, intellectual property, or proprietary protocols. Moreover, giving the public access to an industrial site data may allow malicious users to identify vulnerabilities and exploit them to attack the company. As a result, many datasets are generally generated from scale-down testbeds and a few real ICS environments.

Since many intrusion detection techniques are supervised, a complete dataset must provide **labels** indicating normal or abnormal data. Furthermore, labels are essential as ground truth for the evaluation of detection performances during the test phase. However, the labeling process is not always straightforward. For example, some attacks can move the system in abnormal behavior after a long time the malicious packets have been sent. In this scenario, the data labeled as malicious should start when the actual attack starts or when the system's behavior starts to be compromised? An analysis of this problem can be found in [193] and [192]. In both cases, the solution could raise a problem in the ground truth. Therefore, there is no right or wrong answer to this question. It depends on the context and the attack type, but it must be specified in the dataset's documentation to allow researchers to act accordingly.

C. Evaluation Metrics

In this section, we briefly recall the metrics used to evaluate the performances of the detection algorithms. According to the literature, the most common metrics are Accuracy and F1-Score. They are defined as follows.

- *Accuracy:* represents the fraction of correct predictions of the model under consideration. In the binary classification case, the accuracy is defined in terms of positives and negatives samples classified as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

  where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.
- *F1-Score:* is a metric used to evaluate a classification, defined as the harmonic mean between *precision* and *recall* as follows:

$$F1 - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{2}$$

  where the *true negative rate*, or precision is:

$$precision = \frac{TP}{TP + FP}, \tag{3}$$

  while the *positive and negative predictive values*, or recall, is:

$$recall = \frac{TP}{TP + FN}. \tag{4}$$

TABLE IV

SUMMARY OF DATASETS PRESENTED IN THE LITERATURE. THE **DATA** TYPE INDICATED AS L: LOGS, N: NETWORK; P: PHYSICAL. **TIMES** OF RECORDING ARE ESTIMATIONS AND MEASURE UNITS ARE H: HOURS, D: DAYS, M: MONTHS. **ENTRIES** NUMBERS ARE ESTIMATIONS, TOO. WE DENOTE THE **ATTACKS** LAUNCHED DURING THE RECORDING AS RC: RECONNAISSANCE; RP: REPLAY; M: MITM; I: INJECTION; D: DOS; O: OTHERS. THE **%** COLUMN INDICATES THE PERCENTAGE OF DATA UNDER ATTACK WITH RESPECT TO THE WHOLE DATASET. FILE **FORMATS** ARE INDICATED AS P: PCAP; C: CSV; L: LOG; A: ARFF; I: INP; AND X: XLSX. *: THE VERSION OF WADI DATASET CONSIDERED IS THE ONE DATED NOVEMBER 2019; THE ONE OF SWAT DATASET IS INSTEAD A1 DATED 2015, THE MOST USED ONE AS THE BEST OF THE AUTHORS' KNOWLEDGE

Name	Sector	Data	Time	Entries	Ref.	Res.	Attacks	%	Formats	IDS	F1	Acc.	Prec.
D5: Energy M.S.D.	Energy Manag.	L	30d	6M	-	[150]	-	-	C	-	-	-	-
QUT_DNP3	Power Grid	N, L	40d	31M	[60]	[198]	RC, RP, M, I, O	∼0.01	P, L	-	-	-	-
QUT_S7Comm	Mining Refinery	N, L	17.5h	2M	[199]	[200]	M	∼10	C, L, P	-	-	-	-
4SICS	Generic ICS	N	46h	3M	-	[201]	unk	unk	P	[202]	∼1	∼1	-
CyberCity Dataset	City	N	16d	170K	[75]	[178]	I, M, D, RC, O	16.58	P	-	-	-	-
D2: Gas Pipeline	Gas Pipeline	N	-	400K	[203]	[150]	I	0.97	C	[203]	0.75	-	0.75
D3b: Water S. T.	Water Storage	N	-	230K	[190]	[150]	RC, I, D	27	A	[204]	0.981	0.981	0.981
D4: New Gas P.	Gas Pipeline	N	-	270K	[122]	[150]	M, I, O	21.86	A	[204]	0.988	0.988	0.988
Electra Modbus	Power System	N	>12h	16M	[59]	[205]	RC, I, RP	5.2	C	[59]	0.987	-	0.988
Electra S7Comm	Power System	N	>12h	387M	[59]	[205]	RC, I, RP	1.42	C	[59]	0.996	-	1.000
HVAC_Traces	HVAC	N	7d	40M	[206]	[207]	-	-	P	-	-	-	-
Lemay Covert	Breakers	N	6.55h	1.6M	[192]	[208]	Covert Channel	100	P, C	-	-	-	-
Lemay SCADA	Breakers	N	∼6h	900K	[192]	[208]	RC, I, O	3.29	P, C	[209]	1.000	1.000	-
Modbus SCADA #1	Liquid Pump	N	∼24d	41M	[210]	[211]	M, D	4.81	P	[212]	0.775	0.812	0.964
S4x15 ICS	Generic ICS	N	<1h	310K	-	[213]	unk	unk	P	[202]	∼1	∼1	-
WUSTL-IIOT-2018	Water Control	N	25h	7M	[112]	[156]	O	6.07	C	[112]	1.000	1.000	1.000
D1: Power System	Power System	P, L	-	78K	-	[150]	I, O	71.02	C, A	[214]	0.955	0.950	0.980
EPIC Dataset	Generic ICS	P, N	4h	5K	[100]	[215]	-	-	P, C	-	-		
QUT_S7 (Myers)	Generic ICS	P, N	8.5h	15M	[216]	[217]	I, M	<0.001	P, L, X	[216]	0.744	-	0.727
SWaT Dataset*	Water Treatment	P, N	11d	950K	[155]	[215]	I, O	5.76	P, C, X	[218]	0.889	-	0.919
BATADAL	Water Distribution	P	22m	13K	[194]	[196]	RP, M, O	1.69	C, I	[219]	0.970	0.989	0.987
HAI Dataset	Power Plant	P	10d	1M	[147]	[148]	RP, M	1.83	C	[220]	0.780	-	0.950
WADI Dataset*	Water Distribution	P	16d	950K	[114]	[215]	M	1.04	C	[218]	0.804	-	0.908

D. Physical Level

BATADAL (BATtle of the Attack Detection ALgorithms) [194]–[196] was a design challenge aimed at the creation of an attack detection algorithm. Every participant was provided with three datasets containing observations of a simulated C-Town network [197], a real-world, medium-sized water distribution system operated through PLCs and SCADA systems, which allows modeling the hydraulic response of a water distribution network under attack. The dataset, provided in csv format, contains SCADA reading for 43 system's variables and is designed for different purposes: two are thought for training (1 year in normal behavior and 6 months with some partially labeled attacks) and one for testing (4 months with unlabeled attacks). The 14 cyberattacks conducted on the system include malicious activation of actuators, change of actuators settings, replay, and MitM attacks. In the paper [194], the authors present the dataset and the evaluation criteria for the competition (time-to-detection and classification performance). Furthermore, it briefly explains the strategies employed by each participant. The dataset is free and available in csv format. There is available also an inp file that can be used with EPANET2 to simulate the system. A new version of the dataset is also available at [221] contains sensor readings without concealment and is discussed in [222].

The challenges' participants developed different algorithms for intrusion detection ranging from Random Forest to Recurrent Neural Network. Housh and Ohar [219] achieved the best result by proposing a model-based fault detection approach that employs a simulator to generate benign data and then compare them to the available SCADA readings to detect anomalous behaviors. This approach is composed of three main phases: 1) available SCADA data are used in a Mixed-Integer Linear Program to estimate the water demand in each node; 2) EPANET simulator is used to generate reference values, which are used to produce simulation errors when compared to actual readings; and 3) a multi-level classification approach is implemented to classify the obtained simulation errors into events and normal conditions. The result shows a Precision of 0.987, and Accuracy of 0.989, and an F1-Score of 0.970. In [223] Kravchik and Shabtai present a detection approach base on under-complete Autoencoder in the frequency domain, which could reach an F1-Score of 0.937, which is high, considering the simplicity and non-specificity of the used algorithm. The presented paper was applied to the first version of BATADAL. Finally, we must consider that BATADAL is synthetically generated. Therefore this dataset does not suffer from significant noisy problems, making anomaly detection easier.

Morris *et al.* presented different ICS datasets, which are available online [150]. Each dataset's name is labeled with a number from 1 to 5 and the involved industrial sector.

Dataset 1: Power System Datasets are a collection of three datasets provided by Morris *et al.* [150], [224] containing the same data but with various labels. One dataset has binary labels (i.e., Normal and Attack). The second dataset has three-class labels (i.e., Attack, Natural, and No Events), which identity as "Natural" events single line-to-ground (SLG) faults and line maintenance and as "No Events" the normal operations. Finally, the third dataset includes 41 different labels containing more information about the attacks and various events. In particular, one label is reserved for "No Events" (i.e., Normal Operation), eighth labels contain different classes of the "intensity" of the "Natural" samples previously mentioned. The remaining 32 labels identify different attacks such as Data Injection, Command Injection, and Relay Setting Change. All the details about the labeling process are available in the readme file at [150]. Physical measures and logs from the control panel, relays, and Snort captures are collected from a physical testbed containing two power generators, four IEDs that can switch four breakers, all connected through switches and routers. Data are provided as a csv file (for the first two datasets) and ARFF format (for the third dataset).

Different interesting IDSs are implemented on these datasets [214], [224]–[226]. In [214] different machine learning-based anomaly detection algorithms are tested against the three datasets. The most performant method was JRipper algorithm [227] together with Adaboost [228] to improve the performance. The algorithms were trained on voltage and current measurements of the four synchrophasors (29 features each). This information was combined with information on frequencies, impedances, and status flags of relays for a total of 128 features. Results show an F1-score, recall, and precision almost always greater than 0.8, with a peak of F1-Score of 0.955 in the three-class dataset. Also, Accuracy was always greater than 0.85. Even if the authors did not include any numerical results based on common metrics, it is worth mention another approach presented in [224]. The authors presented a specification-based intrusion detection framework, which is tested in the discussed dataset. They implemented a Bayesian network to model different threat scenarios. The authors' purpose was to build a network with a unique path for each threat scenario. In other words, each scenario must be described as a sequence of system states, actions, and events that uniquely identify it. For each threat identified, the system collected related measurable variables and events. Then, each scenario is divided into actions that cause the system state transition. Finally, the Bayesian network is built on an independent path of states, computed for each threat. An IDS was implemented starting from the Bayesian network obtained, which reads states and logs to track the system states. The obtained IDS can classify ten different scenarios containing both faults and cyber-attacks by monitoring the state transitions, with different precision based on the relay location.

Dataset 5: Energy Management System Data [150] is a large anonymized log collected by an Energy Management System (EMS) in a utility in the United States of America.

The dataset's csv contains the timestamp and ID of each event, the SCADA category (i.e., information of the type of event), each device type, the event message, the priority code, the name of the substation, and the area of responsibility (i.e., the controlling authority). Data are collected in a period of 30 days. Since the dataset contains only normal operations, no attacks are provided. For this reason, to the best of the

authors' knowledge, there are not IDS implemented on this dataset.

HAI Dataset (HIL-based Augmented ICS) [147], [148], [229] is a collection of physical data from three physical control systems (a GE's turbine, Emerson's boiler, and a FESTO's water treatment systems) combined through the dSPACE HIL simulator [102]. Data were sampled every second in 59 points representing the variables measured or controlled by the control system. Basing on the GitHub repository [148] (which currently differs from [229]), the data collected contains seven days of normal system behavior, a day with 20 different attack scenarios on each control loop, and two days with 14 attacks on multiple control loops, for a total of 10 days of capturing. Totally, there are around 1 million samples, 1.83% of which are labeled as under MitM attacks, in particular relay and modification attacks. However, all the attacks are deeply explained in [229]. Data are provided in a csv format with a document that accurately depicts the testbed architecture and the dataset's data.

Due to the novelty of the dataset, released in 2020, there is a lack of IDS implemented on this dataset. However, in [220] the authors present an anomaly detection strategy based on clustered deep one-class classification (CD-OCC). It is an unsupervised approach that combines clustering algorithms with deep learning (DL) models. In particular, K-means were applied for clustering on the training set. Then, different types of neural networks (e.g., DNN, CNN, RNN) were implemented to predict the clusters and return softmax values classified with the iForest algorithm. Currently, on the HAI Dataset, the higher precision is achieved using DNN as cluster predictor (0.95) while the overall higher scores are obtained with CNN as cluster predictor (F1-score: 0.78; Precision: 0.78). To complete the research, the same algorithms are tested on another popular dataset, SWaT [155], showing the best results with the same algorithms (i.e., CNN and DNN).

WADI [114], [115] is a dataset with data collected from WADI, a water distribution testbed, created as an extension of the SWaT testbed [110]. The system comprises three subsystems: a primary grid, a secondary grid, and a return water grid. It is also able to simulate water consumption following time-varying demand patterns. The dataset collects 16 days of continuous operation: 14 under regular operation and two days within an attack scenario (a total of 15 attacks). The adversary aimed to cut off the water supply to the consumer tanks. In the attacker model, the adversary has remote access to the SCADA system. The data recorded represent the state of all the 123 sensors and actuators connected using Modbus/TCP protocol. The dataset is free upon request [215], and it is provided as csv files.

There are many IDSs designed and tested on WADI Dataset in literature. MAD-GAN [230] is an unsupervised multivariate anomaly detection method based on Generative Adversarial Networks (GANs). This method uses a generative model to create a fake time series and a discriminator to distinguish between normal and abnormal data. A peculiarity of this work is that, instead of considering each data stream independently, the framework considers the entire variable set concurrently to capture the latent interactions among variables. To do so, the authors implement a sliding-window approach to divide the multivariate time series into sub-sequences. On WADI, MAD-GAN obtains a precision of 0.53 and an F1-Score of 0.62. Better results were achieved by Kravchik and Shabtai [223] which obtain a Precision of 0.83 and an F1-Score of 0.75. They employ an Autoencoder with sequences of length 7 in the time domain. With respect to SWaT [155] and BATADAL [194], the authors also mention that it was impossible to apply the AutoEncoder on the frequency domain because most of the features do not have a clear dominant frequency. However, the best results on WADI were obtained by DAICS [218], a deep learning solution for anomaly detection in ICSs. The authors propose a 2-branch feature extraction framework. The wide branch, containing only one fully connected layer, is used to memorize the normal state of sensors and actuators. Instead, the deep branch comprises two fully connected layers between two convolution layers and provides the generalization degree required to handle events not covered in the training set. Moreover, DAICS introduces the *few-time-steps algorithm* which can be used to efficiently reconfigure DAICS in a production environment when operators encounter false alarms. DAICS can achieve a Precision of 0.919 and an F1-Score of 0.804 on WADI.

E. Network Level

CyberCity Dataset [75], [177], [178] is a dataset collected by the SANS Institute from their own ICS CyberCity testbed. CyberCity testbed is a complete simulation of an entire city containing a bank, a hospital, a power plant, and many other generally available components in a small town. There is also a tabletop scale model of the city, which shows an electric train's behavior, a water tower, and a miniature traffic light. A pcap file is freely downloadable online [178] containing over 170k network packets recorded as a dataset for the Holiday Hack cybersecurity challenge in 2013. The data are unlabeled, but in [75] the authors estimate that about 16% of the data is under attack. Various attacks are included, such as scanning, information disclosure, command injection, MitM, and DoS. The ICS components use Modbus/TCP, EtherNet/IP, and NetBIOS as communication protocols. For each attack presented, some preventative measures are proposed and evaluated. Some examples are awareness training, system patching, IDS, or anti-virus, but it is remarked that neither one is 100% effective. It is worth noting that, at the best of the authors' knowledge, there is no precise and official documentation of the dataset provided by the SANS Institute.

Dataset 2: Gas Pipeline Datasets [150], [203] contains a collection of labeled Modbus/RTU telemetry streams from a gas pipeline system in Mississipi State University's Critical Infrastructure Protection Center [106]. Each stream is composed of some selected features, including, for instance, an identification bit to discriminate between command and responses, states of components, length of data, and physical measurements. The authors include different command injection and data injection attacks, alongside some data in normal behavior. The dataset contains about 397k samples,

divided into csv files with a name indicating the particular attack. The dataset also includes a feature to identify the samples that are effectively part of an attack, with information about the attacker's action in the particular moment. The total percentage of samples with abnormal behavior is 0.97%. Unfortunately, the dataset does not include each sample's timestamps, making it impossible to analyze timing information.

The dataset was used to test different machine learning algorithms as a discriminator of malicious RTU transactions to detect the deployed attacks [203]. Features are derived by analyzing each raw packet individually to extract the protocol command values. K-Nearest Neighbors and Random Forest are the two algorithms that provided better results across all the attacks, with a Recall/Precision of 0.75 or higher for five of the seven attacks. More in detail, the most problematic attacks were burst values (i.e., sending multiple successive pressure values, faster than the data display rate, to the operator interface) and setpoint value injection (i.e., the attacker sends false pressure values equal to the setpoint). Yüksel *et al.* [231] formally describe a user-understandable framework with effective anomaly detection techniques for ICSs. The test implemented using Modbus/RTU employs the Dataset 2: Gas Pipeline by dividing the attacks into scanning, illegal values, timing, and illicit command. The features extracted were only related to the ICS protocol fields of each packet, such as the type of command, location, or address the host is accessing. To improve results, the authors also performed a feature selection by excluding features with low importance (e.g., incremental fields). The results are highly variable depending on the trade-off between the detection rate and the false positive rate. However, by fine-tuning the algorithm, it was possible to achieve a detection rate of 0.9991 and a false positive rate of 0.001.

Dataset 3: Gas Pipeline and Water Storage Tank by Morris *et al.* [150], [190] are two different datasets from physical testbeds containing both physical data field and network traffic. The first comprises data deriving from a gas pipeline, while the second contains data from a water storage tank. Both the datasets come from testbeds at the Mississipi State University's Critical Infrastructure Protection Center [106] and are shared as ARFF files. A bump-in-the-wire approach was used to capture data logs and inject attacks in Modbus communication in both cases. The implemented attacks are reconnaissance, response and command injection, and DoS. They cover around 27% of the total data. The authors also provide two short datasets created using 10% of the complete datasets, suited for rapid tests during the preliminary IDS development phases. As explained in [122], [150], the gas pipeline dataset contains unintended patterns that cause some algorithms to identify attacks and non-attacks in unrealistic ways easily. Therefore, we do not report this work in the corresponding dataset tables. Instead, we consider the second version of this dataset, called Dataset 4: New Gas Pipeline.

Dataset 4: New Gas Pipeline [122], [150] is a new version of the Dataset 3: Gas Pipeline dataset. This version was proposed to fix dataset problems causing machine learning algorithms models that do not match real system behavior and lead to overly optimistic classification accuracy. In this version, the authors implement 35 attacks and precisely document them in the paper and the dataset. The dataset includes different labels for each attack, which cover 21.86% of the capture. Like the previous version, the protocol used is Modbus and data are available as an ARFF dataset containing both physical data and information about the network packets.

D3 and D4 datasets are widely used in the study of IDS for ICS. Feng *et al.* [232] presented a multi-level anomaly detector using package signatures and LSTM networks. The detection architecture provided is composed of two-level. First, a packet-level anomaly detector based on a Bloom Filter is applied; second, the first-level not-anomaly data are used as input to a stacked LSTM neural network model time-series level anomaly detection. The anomaly detector was tested on Dataset 4: New Gas Pipeline using two LSTM layers of 256 nodes, each achieving a Precision of 0.94, Accuracy of 0.92, and an F1-Score of 0.85. The most problematic attack to be detected was the injection of malicious state commands for which a Gaussian Mixture Model performed better. Demertizis *et al.* [204] proposed the Spiking One-Class Anomaly Detection Framework (SOCCADF), which employs the advanced evolving Spiking Neural Netowork (eSNN). eSNN is a modular connectionist-based system that evolves its structure and functionality in a continuous, self-organized, online, adaptive, and interactive way using incoming information. The framework is supervised and was tested on both the Dataset 3: Water Storage Tank (Precision 0.981; Accuracy 0.981; F1-Score 0.981) and the Dataset 4: New Gas Pipeline (Precision 0.988; Accuracy 0.988; F1-Score 0.988). The same authors adopted eSNN on GRYPHON [233], which simplifies the validation mechanisms to work in a semi-supervised way, getting as input only data in standard behavior (i.e., labeled as normal packets). This approach was able to get a Precision of 0.980, an Accuracy of 0.980, and an F1-Score of 0.980 on the Dataset 3: Water Storage Tank, while a Precision of 0.975, an Accuracy of 0.977, and an F1-Score of 0.970 on the Dataset 4: New Gas Pipeline. Another interesting work is the metaheuristic approach by Mansouri *et al.* [234]. In this work, the authors provide an anomaly detector based on neural networks with a pre-processing step able to act with a different algorithm based on the packet's delay to have as little impact as possible on the real-time communications. When computational speed is required, computationally efficient Evolutionary System [235] optimization is used. Instead, a more accurate but computationally expensive Grey Wolf optimizer [236] is used if with higher latency scenarios. A neural network is then used to detect malicious data with an accuracy up to 98% on the Dataset 4 New Gas Pipeline.

Electra [59] dataset was obtained from a real scenario of an electric traction station used in the railway industry. Electra is composed of 5 PLCs, a SCADA system, a switch, and a firewall. All the communications between the components implement Modbus and S7comm over TCP/IP with a master-slave model. There are two different datasets, one for each communication protocol. The implemented and labeled

attacks are false data injection, replay attack, and reconnaissance attacks in both cases. The attacks were deployed with a new device attached to the network with a MitM configuration. In both **Electra Modbus** and **Electra S7comm** datasets, the capture lasts about 12 hours in which 94% and 98% of the data are in normal condition, respectively. The data amount is enormous, containing 387M entries for S7Comm (36.8GB) and 16M for Modbus (1.5GB). The two datasets are freely available on the Web [205] in csv format.

Together with the datasets' presentation, Gómez *et al.* provided an implementation of the main algorithms used for anomaly detection. The authors try both supervised and semi-supervised algorithms (i.e., One-class SVM, Isolation Forest, SVM, Random Forest, and Neural Network). Features were obtained by packet inspection and by considering only the control protocol fields such as MAC and IP addresses, timestamps, errors, and the application data. On Electra Modbus, a simple supervised Random Forest with 200 estimators was sufficient to achieve a Precision of 0.988 and an F1-Score of 0.987, while a single layer supervised Neural Network with 128 neurons was able to reach a Precision of 0.9999 and an F1-Score of 0.996 on the S7Comm version. On the other hand, the semi-supervised OCSVM performed properly on both the dataset, reaching 0.996 of Precision in Electra S7Comm. In the successive year, the same authors proposed SafeMan [237], a framework to manage both cybersecurity and safety in the manufacturing industry. It is composed of a set of applications and services used to monitor and analyze the industrial process in real-time. SafeMan is based on Edge Computing (EC) to achieve low latency and fast deployment of applications and services. Furthermore, EC allows performing the necessary computing tasks close to the manufacturing activity or the network edge. The framework contains several components to assist the deployment, and the risk assessment, together with the cyber threats detection application proposed in [59]. A different and innovative approach was introduced by Li *et al.* [238] who design an anomaly detection method based on cross-domain knowledge transferring. Features are extracted from each packet. In detail, for each frame, the authors derived nine basic features (e.g., connection duration, protocol type, connection status) and 15 content features (e.g., number of failed login attempts, number of access to the control). The authors employ the TrAdaBoost algorithm to train a neural network using not only a part of the data of the Electra Datasets but also employing data from different domains, both from other ICS (e.g., SWaT Dataset [155]) or other CPS fields (e.g., KDDCup99 Dataset [239]). Then, they compared the error rate with respect to a standard SVM and a standard LSTM, showing better results, especially when employing a small fraction ($< 10\%$) of the Electra Dataset in the training phase.

HVAC_Traces by Ndonda and Sadre [206], [207] is a dataset recorded on a Heating, Ventilation, and Air Conditioning (HVAC) system powered by Honeywell and used to provide thermal comfort and acceptable indoor air quality on a university campus. The Building Management System (BMS) is fully automated, and it is suited to monitor from 15 to 20 structures, each containing different PLCs and RTUs. Operators can access the system through the HMI. The protocols implemented are proprietary (e.g., DCE/RPC, NetBIOS, S7Comm) and use TCP/IP at the transport layer. The data capture was produced using *tcpdump*, at two routers via port mirroring. To obtain an accurate timestamp on each packet in two separate recording points, the authors synchronized the clocks using Network Time Protocol (NTP) [240]. However, it was not sufficient to ensure good accurate timing. To overcome this problem, the authors introduced a correction factor calculated using ad-hoc ICMP messages sent periodically on the network. The anonymized dataset is publicly accessible in pcap files, where each file contains one hour of traffic. In total, there are about 7 days of collected data in normal conditions, without any attacks. Since the dataset does not contain attacks and is a novel collection, there are no IDS tested to the best of our knowledge.

Lemay and Fernandez [192] present a dataset of a SCADA network, also called **Lemay SCADA**, virtually implemented with SCADA Sandbox. The simulations contain different MTUs and controllers connected with the Modbus/TCP protocol. The attacks are generated with an infected machine that launches various exploits to infect other devices. Then, the compromised machines launch different attacks by leveraging Metasploit [189] (e.g., Malware Injection, Reconnaissance). The authors give particular attention to the labeling process and to maintain normal intra-packet time properties. The captured data are divided into various collections with an explicit name indicating the types of implemented attacks. The authors also implemented a cover-channel attacks dataset presented as **Lemay Covert**. In these attacks, the least significant bit of the Modbus packets is used to carry information. To the best of our knowledge, this is the only available dataset containing side-channel attacks. Unfortunately, none of the attacks were designed considering Modbus protocol vulnerabilities. Instead, they are implemented with Off-the-Shelf Tools (i.e., Metasploit [189]). Data collection lasts about 6.25h, and the samples labeled as attacks are about 0.15% of the total for the first attack, while the covert channel packets are present in the whole capture. The datasets are shared in both pcap and csv format.

Schneider and Böttinger [241] proposed an unsupervised anomaly detection framework. They employ deep autoencoders with pipelining parallel processing strategies to speed up the training. While the proposed framework performs well on the SWaT dataset [155], it shows very different results depending on the attack type when applied to the Lemay SCADA dataset. In particular, to correctly detect an attack, the framework requires a minimum duration of it. For attacks lasting longer than the minimum threshold, the Precision reaches 100%. Anton *et al.* [209] implemented different standard classification machine learning algorithms on Lemay SCADA. The authors extract 14 basic features from the packets and nine additional features derived from timing and frequency information. Algorithms are tested on three different batches of packets resulted from merging different Lemay SCADA datasets. Both Random Forest and SVM result in an F1-Score and an Accuracy greater than 0.999 with all the batch, while k-means clustering report the lowest results. In a follow-up work of Anton *et al.* [242] data are considered as time

series. Each second of network traffic was aggregated into a single data point. Three different algorithms were implemented to detect anomalies inside the three batches of captures defined in [209]. The first algorithm implemented was Matrix Profiles, and it performs well on data with periodic characteristics, requiring only one hyperparameter. Second, the Seasonal ARIMA-process performed well on periodical data and is more resistant to noise but requires a more complicated tuning of the three hyperparameters. Finally, the authors implemented LSTM, which requires a high training effort compared with the other two light-weight approaches. They tested the algorithms on a subset of the Lemay SCADA datasets containing seven attacks divided into three categories (fake command, executable upload, file moving). The attacks are almost all correctly classified with every algorithm. With LSTM, the accuracy is always greater than 0.90, while the F1-Score is really variable based on the threshold selection methodology.

Modbus SCADA #1 [210], [211] by Cruz *et al.* is a dataset containing data recorded from a small physical testbed simulating a liquid pump. The testbed comprises an HMI, an Adruino-based RTU, a PLC, a Variable-Frequency Drive (VFD), and a 3-phase motor. The protocols used are Modbus/TCP and Modbus/RTU. Data are divided into subfolders based on the attack deployed. Moreover, each pcap file is named with an intuitive strategy that includes the duration of both the capture and the attack. The attacks implemented range from MitM to different flooding types: ping flooding, TCP SYN flooding, and Modbus Query flooding. All these flooding attacks are aimed at the generation of DoS. The data recording lasts for 24 days, containing 4.81% of data flagged as under attack.

This dataset was used by Radoglou-Grammatikis *et al.* [212] to test an IDS. Firstly, the authors present an expansion of Smod [243], a penetration testing tool for Modbus/TCP, to enable the generation of DoS, MitM, and replay attacks. Then, they deployed an IDS to detect DoS attacks and a server for machine learning offloading computation. To train and test the models, the authors employed CICFlowMeter [244] to extract 83 features from each Modbus packets flow. Among the various algorithms tested, AdaBoost [228] and Random Forest achieve the best results with a Precision of 0.96, an Accuracy of 0.81 and an F1-Score of 0.77.

QUT_DNP3 [60], [198] is a dataset presented in the Ph.D. dissertation of the author. The dataset contains data collected from a small section of a transmission substation SCADA network. The testbed involves GOOSE and DNP3 protocols, enabling the communication between the Master, the Slave, the IED, and the attacker machine. All the communications pass through an industrial switch. The attacks are categorized into six categories: Injection, Flooding, Masquerading, Replay, MitM, and all attacks. Each category also contains Reconnaissance packets. The attacks are launched by an attacker machine, which also generates a log providing information about each attack sequence's start and end. Each dataset file has a different duration based on the attack frequency during the capture creation since the authors implement a random time between two attacks. Moreover, the dataset is divided into two categories based on the attack frequency: frequent attacks (i.e., approximately an attack every half an hour) and infrequent attacks (i.e., approximately an attack every random time between one and four hours). For each frequency category, the authors provide two datasets, respectively, for training and testing. Furthermore, a control dataset with only legitimate communications (i.e., without any attacks) is available, and it covers 24 hours of recording. In total, the dataset contains 40 days of recording. It is worth mention that the labeling process was performed with particular care since it was the main topic of the thesis work. The dataset is available on Github [198]. However, to the best of our knowledge, there are no IDS tested on this dataset.

QUT_S7Comm [199], [200] developed by Rodofile *et al.* is an open-source dataset collected in a three time-based subprocesses testbed of a mining refinery plant. The plant testbed is composed of one Siemens PLC acting as Master and three PLCs actings as slaves, all connected with a switch to an HMI and communicating using S7Comm protocol. The attack dataset comprises 9 hours of data and 64 attacks from 13 different possible typologies. Data are provided with pcap files and four process logs: a master log, a conveyor log, a tank log, and a reactor log. The labels of the attack samples are contained in separated csv files. The control dataset comprises 8.5 hours of network traffic and process log data, with 32 different processes. This dataset's peculiarity is the particular division of the network traffic in separate files based on the node capture perspective: a file collected from the attacker's point of view, one from the HMI, and one from the master PLC. This particular composition could be initially complex to use, but on the other hand, it provides higher flexibility with respect to datasets with the entire capture. The dataset is available on Github [200]. To the best of our knowledge, there are no IDSs implemented on this dataset.

4SICS [201] is a pcap dataset collected by Netresec from an ICS lab at the Industrial Cyber Security Conference. At this conference, there was an ICS testbed composed of heterogeneous devices such as PLCs, RTUs, servers, and industrial network equipment (e.g., switch, firewalls). It was available for hands-on testing by the conference attendees and, since the testbed was left almost uncontrolled, the data recorded are not labeled. Furthermore, it is impossible to know what the users have done and, eventually, what kinds of attacks are present. The dataset includes a wide variety of ICS protocol traffic such as S7Comm, Modbus/TCP, EtherNet/IP, and DNP3.

S4x15 ICS Village CTF Dataset [213] provided by Digital Bond, contains network traffic collected during a capture-the-flag (CTF) competition in the ICS Village. The system was composed of different interconnected PLCs, and the dataset contains, without labeling, the attacks launched by the players to the system. The dataset contains pcap files with Modbus/TCP and BACnet packets.

Basing on this dataset, Yu *et al.* [202] proposed an anomaly detection method based on TCP and UDP payload inspection. The detector's architecture comprises an offline module for the expected behavior model and an online module containing the actual anomaly detector and a packet signature generator. In the proposed work [202], the authors use 4SICS [201] dataset to model the normal traffic behavior for Modbus/TCP

protocol, while the normal traffic behavior of BACnet protocol is based S4x15 dataset. Instead, malicious packets are retrieved from Quickdraw-Snort [245], a collection of Snort rules for ICS environments, which also provides some testing packets. Results show Accuracy and Recall close to 100% and a very low false alarm rate.

WUSTL-IIOT-2018 [112], [156] is a dataset recorded from a testbed simulating a water tank control system. Network traffic was monitored for 25 hours, collecting 25 features. Then, the authors performed a data cleaning process to delete corrupted or missing values and outliers. In this phase, about 10k observations were erased, leaving the final version with about 7037k entries. Furthermore, only the six more relevant features are available in the provided csv file, together with a column indicating if the observations are related to an attack. Various attacks have been launched during the capture: port scanning using Nmap [246], address scan attacks, device identification, and unauthorized access to actuators status by using known exploits of the Mobus protocol. The final dataset contains 6.07% of data under attack.

On the same paper [112], the authors developed IDSs employing standard Machine Learning algorithms. The authors selected only six features from the datasets concerning the number of packets, the packets' length, source and destination of addresses, and port numbers. Best results were achieved by Decision Tree and KNN with an accuracy of up to 100% considering the offline evaluation. Instead, regarding the online phase, Decision Tree and Random Forest obtain the best results with an accuracy of 0.999. Furthermore, these last two models performed well in terms of False Alarm Rate (i.e., percentage of the normal flows misclassified as abnormal flows) and Un-Detection Rate (i.e., the fraction of the abnormal flows misclassified as normal flows), which are close to 1.

F. Physical and Network Levels

Electric Power and Intelligent Control (EPIC) is a collection of data from 8 scenarios collected with the EPIC testbed [100], [101]. Each collection scenario lasts for 30 minutes under normal operations, resulting in more than 5000 readings of sensors and actuators, together with the corresponding Modbus network traffic. Blaq_0 [215] is a dataset obtained from the same testbed under different attacks. Blaq_0 contains network-level data collected from a three-day Hackaton 2018 where different teams attack the EPIC testbed. Both datasets are free upon request, but the second one is not widely used. The EPIC dataset contains both pcap and csv files. Both the datasets are free upon request on the iTrust website [215].

QUT_S7 (Myers) [216] is a dataset generated from a small scale ICS testbed. It is composed of a bi-directional conveyor belt system, a water pump system, and a "reactor" pressure vessel system. All these devices are connected to a power meter, and an HMI is used to collect logs. The protocol used is S7 Communication, the standard protocol for Siemens PLCs. The dataset contains device logs with information about each component's state and pcap files with network traffic. Data are divided into two folders containing control data (i.e., data in a normal behavior) and attack data. 21 cyberattacks were launched, consisting of two major types: Injection Attacks and Flooding Attacks. Furthermore, the authors also provide an xlsx file containing pre-processed data. The dataset is freely available for the download [217].

In the same paper, Myers *et al.* [216] proposed a novel process mining based anomaly detection technique. The detector idea is to collect logs in order to produce a record of each device's status. From this data, the authors compute a model containing the expected behavior of the ICS. The model is designed to manage the entire process instance from start to finish with only acceptable events. Finally, process mining is used to perform conformance checking activity, calculating the fit of a given event log by replaying it on the model. Results show that only 16 attacks out of 21 were correctly identified with several false positives (i.e., Precision: 0.727; F1-Score: 0.744; Recall: 0.762). The authors motivate that for most false positives, the starting condition was altered by previous attacks. It is a common problem that can be mitigated with a correct generation of the dataset, as will be discussed in Section VIII, especially taking care of the labeling part.

SWaT [111], [155] is the most popular dataset in the ICS field. It contains monitoring data from a fully operational scaled-down water treatment plant. The testbed contains two separate networks: a level 1 star network that allows the communication between the SCADA system and the six PLCs and a level 0 ring network that transmits sensor and actuator data to the corresponding PLC. The protocols employed for communications are CIP and EtherNet/IP. The dataset received various updates and improvements over the years. More precisely, up to today, there are seven different data collections (the last one is dated June 2020). The first version (December 2015, described in [155], [215]) is the largest and most used. This version includes both network traffic and recordings from all the 51 sensors and actuators for eleven days. Of these eleven days, seven days are under normal operation, and four days contain 36 different attacks, classified into four types based on the attack number of stage and devices affected. The first version of SWaT contains about 944k physical samples (5.76% are labeled as attack samples). The singular network packets are instead unlabelled, with only a flag indicating the presence (or not) of malicious data in each packet's batch. In 2017, a new version was released, which collected about 136 hours of network traffic together with measurements of sensors and actuators provided in csv form. No attacks were performed during the recording phase. In 2019 about 4 hours of recording were saved as a dataset. About one hour contains 6 different attacks like spoofing and tampering with some switch. In this case, physical data are provided as xlsx files. In the same year, another version was released containing both network and physical data of about 3 hours, during which two malware attacks were launched. The first try to exfiltrate historian data, while the second disrupt sensors reading and process. The most recent version is dated 2020 and contains 4 runs, each one lasted 2 or 4 hours. Physical data without any attacks are provided in xlsx form. In 2017 during the SUTD Security Showdown, a competition where researchers could access and attack the SWaT testbed, all the network flows are saved in a dataset called S317 [247]. The dataset records three days of

competition and contains historical data and the description of the attack scenarios. Both datasets are free upon request, but the second one is not widely used. Data are provided in different csv, xlsx, and pcap formats. As reported in [248], the various SWAT releases are very different from the operational point of view, also implementing different actuators control logic. It makes it difficult to transfer the detection framework among the dataset releases. Furthermore, even in the same SWaT version, the systems behave very differently. It is probably due to the testbed recovery time after an attack. To overcome this problem, in more recent versions, the authors restart the testbed after each attack. All the datasets are free upon request at the iTrust datasets webpage [215].

The SWaT dataset is probably the most used dataset on which researchers try their IDSs. In almost all the papers using the SWaT testbeds, there is no explicit mention of the dataset version used. However, from the data description, we can infer that the first version is used in almost all the cases. Several innovative detection methodologies have been tested on this dataset, ranging from sensor noise fingerprint [249], a graphical-based detection approach [250] and a framework to generate invariants with association rules mining [251]. Kravchik and Shabtai [223] employ the SWaT physical data to test different detection approaches such as Dynamic PCA, 1D-CNN, and AutoEncoder, both in the time and frequency domains. Thanks to the linearity of many relations between sensors and actuators in SWaT, PCA performed very well, especially using a sliding-window approach, which results in 0.92 of Precision and 0.879 F1-Score. Also AutoEncoder in the frequency domain reaches similar scores (i.e., Precision: 0.924; F1-Score: 0.873). Similarly to WADI, excellent results on SWaT physical data were achieved by Abdelaty *et al.* [218]. This paper introduces DAICS, 2-branch neural networks with automatic tuning mechanisms to update the system model. DAICS scores 0.9185 of Precision and 0.889 of F1-Score. It is worth noting that, despite a not high Precision (i.e., 0.70) and F1-Score (i.e., 0.81), MAD-GAN [230] on SWaT achieved a Recall of 0.954, the higher one to the best of our knowledge.

Instead, by relying on SWaT network traffic, Schneider and Böttinger [241] implement an autoencoder-based unsupervised anomaly detection framework leveraging pipelining parallel processing strategies to speed up the training. To overcome the problem of unlabelled network packets and generate the ground truth, the authors developed a semi-automatic label estimation mechanism to detect the packets with a higher probability of being anomalous and then use a manual investigation to label them. Results reveal a Precision of 0.998 and an F1-Score of about 0.988 in scenarios like TCP session reset attack and SYN Flood attack, while other types of attacks as duplicate acknowledgments or TCP retransmissions are essentially not detected.

VIII. Lesson Learned: Good Practices

Basing on the knowledge acquired on surveying the different works and analyzing the common mistakes and solutions implemented, in this section, we summarize concepts and good practices to consider when selecting a testing system.

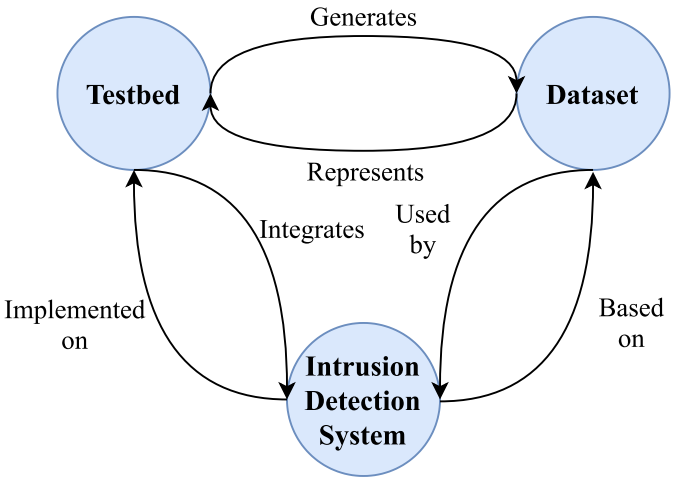


Fig. 12. Relations between Testbed, Dataset, and IDS.

In particular, we summarize the good practices in creating an effective testbed in Section VIII-A and to develop a dataset in Section VIII-B. Furthermore, we also provide additional insight to help the standardization of the IDS results in Section VIII-C. During the designing phase of each of the three resources, the designer must consider the final use of such resources and the other two resources' requirements in future integration. Figure 12 graphically represents the relation between the three resources. More precisely, a testbed should allow an efficient data collection to produce a well representative dataset and integrate IDSs to validate the case studies in a real scenario. A dataset must be designed to be an exhaustive and precise representation of a testbed and easily allow data analysis tasks and the implementation of an IDSs. Finally, an IDS, which represents the higher-level products with respect to the other two resources, should generalize on different datasets to avoid construction biases. Moreover, the design of a dataset should consider an easy integration into a real-world scenario such as a testbed.

A. Good Practices: Testbed

An effective testbed development passes through various steps and challenges, each composed of a notable complexity level. These challenges should be considered during the design phase.

Scope Identification: During the design phase, the designer must consider the final application. The applications of a testbed can be [70]: i) Discovery, to study and obtain knowledge about a particular ICS field or system functioning; ii) Demonstration, to validate or experiment the research findings; and iii) Education, to use the testbed to educate students, researchers, and stakeholders. Every scope implies different requirements to deal with and different funding. For instance, if a testbed is specifically designed for IDS development, the authors must consider developing an attack chain and data collection accurately. On the contrary, the Educational testbeds do not have this requirement. Instead, they should be composed of an easily understandable and representative process. In this

case, water systems are an excellent choice due to their immediate visualization. Once the scope is identified, the designer can give the system's specific layer adequate importance to satisfy the scope.

Fidelity: If the testbed is used for Discovery or Education, data's perfect fidelity is generally not needed. In these cases, Virtual or Hybrid testbeds are the preferable platforms to be used due to their flexibility and cheapness. Contrarily, in the case of validation tests, Physical testbeds are the best solutions since the smallest variation of measures is fundamental for the research. A complete work that can help researchers to identify the correct design criteria is [25]. The U.S. National Institute of Standard and Technology (NIST) has recommended that a SCADA testbed for security assessment should consider four general areas [2]: the control center, the communication architecture, the field devices, and the physical process itself.

Expensiveness: Expensiveness in the construction and the maintenance of Physical testbeds are the first barriers a research group will encounter when deciding to build one. Suppose a research group can deal with this limitation. In that case, it is useful to share with the community datasets collected from the Physical testbed and the related documentation, as the iTrust laboratory of SUTD [195] is doing with SWaT and WADI. Furthermore, provide a simple way for other researchers to access the testbed can be an added value not only for the community, which can take advance of it but also for the owner who can have a more critical view of the system.

Reproducibility and Comparability: Robust and innovative researches need to be reproducible and peer validated. Basing on this, testing on physical testbeds is not recommended since it creates difficulties in reproducibility. An intelligent solution is to create one or more datasets capturing network traffic and physical measures from the testbed and share them with the community. In this way, the reviewer can easily verify the study, while the community will benefit from newly available datasets. On the other hand, if a virtual testbed is used, it is not required to provide a dataset to support the research. Instead, it is possible to provide the software with the entire simulation. However, it is fundamental to precisely indicate the architecture and the state of the ICS at the beginning of the experiment to avoid reproducibility errors due to a different scenario.

Missing Representation: We identified that the most common scenario represented with a testbed is relative to water management (e.g., Water Distribution, Water Treatment). This is probably because Water Systems are the most easier scenario to implement in terms of equipment and maintenance costs. Another very represented scenario is related to the Electric Plant (e.g., Power Grid, Power Power). However, IDS rarely investigates this scenario. We believe this is due to the difficulty in developing detection systems that deal with high-dynamic environments such as Electricity. Also, the majority of the related dataset does not include attacks scenario. For future organizations that want to approach a testbed development, we identified a low contribution in scenarios such as large-scale manufacturing, Nuclear Plants, Transportation Systems, health-care infrastructure, IIoT, or HVAC. Another interesting scenario missing in testbeds analyzed is the cloud-based ICS. As discussed in [27], this scenario is always more adopted in real systems. Therefore, having a testbed related to cloud-based ICS would help secure future ICSs by studying the connection between the plant and the cloud, which inevitably opens new attack entry points.

Standardization: We noticed that a common problem in the analyzed testbeds is the lack of standardization according to the industrial security standards. The most common international standard for cybersecurity in the industrial system is the IEC 62443 [252]. However, different organization releases guidelines and requirements to securing ICSs. These organizations include NIST [2], North American Electric Reliability Corporation (NERC) [253], and European Union Agency for Cybersecurity (ENISA) [254]. In 2013 ENISA published a report providing a list of related works in the field of ICS security [254]. In this report, ENISA lists initiatives and groups working on ICS security worldwide. Among the presented projects, the most relevant and yet active, such as SNL Testbed at Sandia National Labs [95] or Joint Research Center's testbeds in Italy [76], [113], are discussed in this paper. Furthermore, the report describes different standards, guidelines, recommendations, and practices for companies and manufacturers, to help them to secure their installations by exploiting commercial solutions and valuable policies.

B. Good Practices: Dataset

A well-designed dataset should exhaustively represent a testbed's behavior and allow easy implementation of research findings. To do this, the design process of an effective dataset must consider the following points.

Labeling: When designing a dataset, the labeling process must be precisely described in the documentation to allow researchers to process the data accordingly. Packets that are part of attacks must be carefully labeled to provide ground truth to researchers. Furthermore, in a valuable dataset, labels must also contain information about the attack type (e.g., Injection, Replay, DoS) and the attack phase. This last element is essential due to the recovery time of many ICSs: after an attack occurs, the system may need some time to stabilize itself. This behavior can be wrongly considered part of the attack by an inaccurate labeling process. Hence, a good strategy is to flag this kind of packet as *recovery*, leaving the decision on how to consider them to researchers. The work [193] explained that the authors of the SWaT dataset decided to label a process data sample as "Attack" when the attack was launched, instead of when the system behavior started to change. This approach can lead to a ground truth problem if not correctly documented or managed. Furthermore, it is important to consider the label generation methodology accurately. A manual approach to flag each entry of an attack as malicious is costly, and if the data amount is large may be impracticable. On the other hand, fully automatic strategies are possible, and they work quite well if attacks are at the same time automatically generated. However, automatic labeling cannot provide high accuracy in case of complex attacks on highly distributed systems. Semi-supervised approaches provide a trade-off that efficiently

spends an expert's work supported by a visualization platform such as RiskID [255].

Documentation: Many of the datasets surveyed lack in documentation. To allow correct and easy use of the dataset, the designer should include detailed information with the dataset's characteristics or a description of the source testbed design. Exhaustive documentation should include the system's control logic, a description of the implemented attacks, and the configuration settings. In the SWaT case, as reported in [248], the recent versions of the dataset implement different control logic. However, the authors never mentioned such modifications.

Attacks Similarity: A complete dataset should also include attacks. Similarly, in a testbed, a researcher needs to be able to deploy attacks easily. However, the designer must approach this phase with caution. Attacks should be as similar as possible to real cases. If a dataset is collected from a testbed, it is sufficient to launch the attacks following an adversary approach and various system information. The authors should also include a clear and complete analysis of the attacker model. On the other hand, it is important and challenging to capture traffic while the attack is occurring in order to generate the datasets. Adding synthetic packets in the resulting capture, if not accurately managed, could disrupt the fidelity of the dataset, making it unrealistic. Furthermore, if data are captured in different monitoring points inside the ICS, it is required a synchronization mechanism to provide consistent data.

Domain Shift: A common problem in a dataset is the so-called Domain Shift, i.e., the difference between the training entries and the testing data [218], [248]. To support researchers and IDS development, datasets should be released with complete documentation explaining the system's initial state. Another problem observed in [248] is related to the testbed remains unstable for a long time. More precisely, after the end of an attack, a system's behavior may need time to recover, remaining unstable. In this case, its behavior will be identified as anomalous by detectors even if flagged as Normal. To deal with this problem, when designing a dataset, the authors should consider adding another label to classify the dataset, e.g., "System Unstable". If the authors can directly interact with the testbed, another solution could be to restart the system after an attack. An imbalanced dataset is a collection of data that contains a significantly low number of samples from one class with respect to the other [256]. It is a critical issue that can influence the performances of Machine Learning based classifiers. Some datasets provided by researchers contain a low percentage of data classified as under attack, as reported in Table IV. This happens because attacks generally last for seconds or minutes, while the ICS is expected to run for much longer. There are different techniques to get better results from an imbalanced dataset [257]. One solution is to act at the data level by re-balance the data in a pre-processing phase using different sampling strategies (e.g., down-sampling). Another novel solution is Data Augmentation, recently introduced to improve Anomaly detection performances in [258]. This technique leverages generative models, such as GAN, to generate synthetic samples. In [256] the authors performed an experiment to understand the impact of an imbalanced dataset in the ICS security field. Different datasets were obtained from an extensive network traffic capture collected from a water control system testbed. To do this, the authors associated to a fixed number of attack samples a variable number of normal samples to create five different datasets with different imbalance ratios (i.e., the percentage of data under attack over the whole dataset). Ratios span from 0.1% to 10.0%. Results show a high Recall variance with better results on higher ratios (Recall$>$0.99 for ratio$\geq$0.70%; Recall$<$0.12 for ratio$\leq$0.30). At the same time, the Undetected Rate (UR) (i.e., the fraction of the attack samples classified as normal) shows near-zero values for high ratio and large misclassification on low ratio datasets (UR$<$0.01 for ratio$\geq$0.70%; UR$>$0.88 for ratio$\leq$0.30). Basing on these results, the authors conclude that it is advisable to generate datasets with at least 1% of data under attack to reduce imbalance problems when testing IDSs.

C. Good Practices: Intrusion Detection System

Nowadays, the majority of IDS are based on Machine Learning and Deep Learning techniques. To build a model using such techniques is required a notable amount of well-organized dataset (e.g., balanced, labeled). Since these techniques are not always straightforward to understand and implement, researchers should implement a clear and well-approved pipeline. The following aspects can help the development of effective IDSs.

Results Baseline: While reading the different papers concerning the implementation of IDSs, we noticed the absence of an evaluation baseline in many cases. Defining a good baseline could help researchers evaluate if their proposed research is effective and improve the current state of the art. Furthermore, various works base their results on a subset of an available dataset. If not used for a specif reason (e.g., isolate and test a specific attack), this approach could cause a problem in understanding an IDS's effectiveness. For this reason, we believe that our Table IV can support the future evaluation baseline. We also identified issues in the evaluation metrics. Many research not always use the same metrics, making it difficult to compare different approaches. We suggest using as many common metrics as possible, such as F1-Score, Accuracy, Precision, and Recall. Baseline problems are also mentioned in other fields, such as Review Helpfulness predictions [259], where the authors proposed to used the same features on the different models proposed by researchers. In this way, it is easier to compare the efficiency of a prediction model. In the ICS field, IDSs model features can be very different and based on diverse approaches. However, comparing a proposed model with the best IDSs of the art state could help future researchers identify the right research directions. Furthermore, to avoid the effect of design bias of the dataset, an IDS should be validated on multiple datasets.

Data Verification: Generally, researchers rarely investigate the causes of the weak performance of IDSs. Sometimes, the reasons may be due to a problem related to the distribution of the dataset. In [248], the authors showed that SWaT dataset behavior and data distribution between Training Set and Test

Set significantly differ. However, even if the SWaT dataset currently represents the most used dataset to test IDSs, no previous works analyzed the statistical distribution through statistical tests. Finally, to allow the community to validate the research approach and improve it, a good practice is to release the code repository source code (e.g., GitHub, Bitbucket).

IX. Conclusion

In recent years the interconnection between IT and OT networks has opened up modern ICSs to new risks and novel vulnerability surfaces. These vulnerabilities were highlighted in many works but also by the dangerous malware targeting industrial companies. Therefore, it is vital to develop new security mechanisms to protect such systems.

In this paper, we provide a comprehensive overview of the ICS field by presenting the architecture and the typical devices employed. We then proposed an analysis of the industrial protocols used in ICSs, highlighting security measures offered by the protocols, their expansions, and analysis of their diffusion in the real world. Furthermore, we surveyed and analyzed the different platforms to test new security mechanisms in the ICS field. To do this, we categorized the testbeds as Physical, Virtual, or Hybrid based on their functioning and explaining the various challenges and requirements to consider during the development or selection phases. Also, we presented the different ICS datasets dividing them based on the type of data provided and useful information that can help the reader choose the dataset (e.g., attack implemented, format, and various data information). To do this, we accessed every dataset and analyzed it separately. We also reported the IDS with the best performance present in the literature to offer a baseline to further works for each dataset. Finally, we depicted different good practices and suggestions for researchers who want to use this kind of testing method and institutions that want to build testbeds or collect datasets.

We believe this survey can help address future research on this field and new researcher approaching the ICS area. In the future, we aim to continue collecting new testbeds and datasets on the website to create a collection of useful information the research community can exploit for researches and studies on this essential field.

References

[1] C. Alcaraz, "Secure interconnection of IT-OT networks in industry 4.0," in *Critical Infrastructure Security and Resilience*. Cham, Switzerland: Springer, 2019, pp. 201–217.

[2] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Guide to industrial control systems (ICS) security, revision 2," Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. SP 800-82, 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[3] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Proc. 4th Int. Conf. Cyber, Physical Social Comput. Internet Things*, 2011, pp. 380–388.

[4] B. Filkins, D. Wylie, and A. J. Dely, *SANS 2019 State of OT/ICS Cybersecurity Survey*, SANS Inst., Bethesda, MD, USA, Jun. 2019.

[5] L. Neitzel and B. Huba, *Top Ten Differences Between ICS and IT Cybersecurity*, InTech Mag., Research Triangle, NC, USA, Jun. 2014.

[6] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proc. 1st Annu. Conf. Res. Inf. Technol.*, 2012, pp. 51–56.

[7] N. Falliere, L. O. Murchu, and E. Chien, "W32. Stuxnet dossier, symantec security response, version 1.4," in *Symantec Security Response*, vol. 4. Cupertino, CA, USA: Symantec Corp., Feb. 2011, pp. 1–69.

[8] *Israeli Test on Worm Called Crucial in Iran Nuclear Delay*. Accessed: Jan. 25, 2021. [Online]. Available: https://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html

[9] S. Shrivastava, "BlackEnergy—Malware for cyber-physical attacks," iTrust Centre Res. Cyber Security, Singapore Univ. Technol. Design, Singapore, Rep. iTrust-Analysis-001, May 2016.

[10] A. D. Pinto, Y. Dragoni, and A. Carcano, "TRITON: The first ICS cyber attack on safety instrument systems understanding the malware, its communications and its OT payload," in *Proc. Black Hat USA*, 2018, pp. 1–28.

[11] *Threat Landscape for Industrial Automation Systems in the Second Half of 2016*, AO Kaspersky Lab, Moscow, Russia, 2016.

[12] G. Barbieri, M. Conti, N. O. Tippenhauer, and F. Turrin, "Assessing the use of insecure ICS protocols via IXP network traffic analysis," 2020. [Online]. Available: arXiv:2007.01114.

[13] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Uncovering vulnerable industrial control systems from the Internet core," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, 2020, pp. 1–9.

[14] N. Networks. *The Cost of OT Cybersecurity Incidents and How to Reduce Risk*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.nozominetworks.com/solutions/challenge/cost-of-ot-cyber-security-incidents/

[15] IBM. *IBM Study: Businesses More Likely to Pay Ransomware Than Consumers*. Accessed: Feb. 2, 2021. [Online]. Available: https://www-03.ibm.com/press/us/en/pressrelease/51230.wss

[16] Coverware. *Ransomware Costs Double in Q4 as Ryuk, Sodinokibi Proliferate*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.coveware.com/blog/2020/1/22/ransomware-costs-double-in-q4-as-ryuk-sodinokibi-proliferate

[17] Dragos. *EKANS Ransomware and ICS Operations*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.dragos.com/blog/industry-news/ekans-ransomware-and-ics-operations/

[18] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A survey of industrial control system testbeds," in *Proc. NordSec Secure IT Syst.*, vol. 9417, 2015, pp. 213–230.

[19] H. Christiansson and E. Luiijf, "Creating a European SCADA security testbed," in *IFIP Advances in Information and Communication Technology*, vol. 253. Boston, MA, USA: Springer, 2008, pp. 237–247.

[20] S. McLaughlin *et al.*, "The cybersecurity landscape in industrial control systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1039–1057, May 2016.

[21] 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.

[22] Y. Geng, Y. Wang, W. Liu, Q. Wei, K. Liu, and H. Wu, "A survey of industrial control system testbeds," *IOP Conf. Series Mater. Sci. Eng.*, vol. 569, no. 4, 2019, Art. no. 042030.

[23] S. Choi, J. H. Yun, and S. K. Kim, *A Comparison of ICS Datasets for Security Research Based on Attack Paths* (Lecture Notes in Computer Science 11260). Cham, Switzerland: Springer, 2019. doi: http://dx.doi.org/10.1007/978-3-030-05849-4_12.

[24] 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.

[25] U. P. D. Ani, J. M. Watson, B. Green, B. Craggs, and J. R. Nurse, "Design considerations for building credible security testbeds: Perspectives from industrial control system use cases," *J. Cyber Security Technol.*, vol. 5, no. 2, pp. 71–119, 2020.

[26] B. Green *et al.*, "ICS testbed tetris: Practical building blocks towards a cyber security resource," in *Proc. 13th USENIX Workshop Cyber Security Exp. Test*, 2020. [Online]. Available: https://www.usenix.org/conference/cset20/presentation/green

[27] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Comput. Security*, vol. 89, Feb. 2020, Art. no. 101677.

[28] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on SCADA systems: Secure protocols, incidents, threats and tactics," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1942–1976, 3rd Quart., 2020.

[29] F. Khorrami, P. Krishnamurthy, and R. Karri, "Cybersecurity for control systems: A process-aware perspective," *IEEE Design Test*, vol. 33, no. 5, pp. 75–83, Oct. 2016.

[30] L. Obregon, *InfoSec Reading Room Secure Architecture for Industrial Control Systems*, SANS Inst. InfoSec, GIAC (GSEC) Gold Certification, Bethesda, MD, USA, 2014.

[31] *Promotic Software*. Accessed: Feb. 10, 2021. [Online]. Available: https://www.promotic.eu/en/index.htm

[32] D. Sullivan, E. Luiijf, and E. J. M. Colbert, *Components of Industrial Control Systems*, vol. 66. New York, NY, USA: Springer, 2016, pp. 15–28. [Online]. Available: http://link.springer.com/10.1007/978-3-319-32125-7_2

[33] IANA. *Service Name and Transport Protocol Port Number Registry*. Accessed: Feb. 2, 2021. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[34] "Modbus application protocol specification, V1.1b3," Modbus Org., Hopkinton, MA, USA, Rep., 2012. [Online]. Available: http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

[35] "MODBUS/TCP security, protocol specification, V21," Modbus Org., Hopkinton, MA, USA, Rep., 2018. [Online]. Available: https://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf

[36] A. Shahzad *et al.*, "Real time modbus transmissions and cryptography security designs and enhancements of protocol sensitive information," *Symmetry*, vol. 7, no. 3, pp. 1176–1210, 2015.

[37] G. Bernieri, S. Cecconello, M. Conti, and G. Lain, "TAMBUS: A novel authentication method through covert channels for securing industrial networks," *Comput. Netw.*, vol. 183, Dec. 2020, Art. no. 107583.

[38] "DNP3 overview, revision 1.2," Triangle MicroWorks, Inc., Raleigh, NC, USA, Rep., 2002. [Online]. Available: https://www.trianglemicroworks.com/docs/default-source/referenced-documents/DNP3_Overview.pdf

[39] "Overview of DNP3 security version 6," Harris Corp., Melbourne, FL, USA, Rep., 2020. [Online]. Available: https://www.dnp.org/Resources/Public-Documents

[40] S. Bagaria, S. B. Prabhakar, and Z. Saquib, "Flexi-DNP3: Flexible distributed network protocol version 3 (DNP3) for SCADA security," in *Proc. Int. Conf. Recent Trends Inf. Syst. (ReTIS)*, 2011, pp. 293–296.

[41] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera, "DNPSec: Distributed network protocol version 3 (DNP3) security framework," in *Advances in Computer, Information, and Systems Sciences, and Engineering*. Dordrecht, The Netherlands: Springer, 2007, pp. 227–234.

[42] A. Kleinmann and A. Wool, "Accurate modeling of the Siemens S7 SCADA protocol for intrusion detection and digital forensics," *J. Digit. Forensics Security Law*, vol. 9, no. 2, p. 4, 2014.

[43] C. Lei, L. Donghong, and M. Liang, "The spear to break the security wall of S7CommPlus," presented at the DEF CON 25, Jul. 2017. [Online]. Available: https://infocondb.org/con/def-con/def-con-25/the-spear-to-break-the-security-wall-of-s7commplus

[44] "PROFINET system description—Technology and application," Profibus, Karlsruhe, Germany, Rep., Oct. 2014. [Online]. Available: http://us.profinet.com/wp-content/uploads/2012/11/PROFINET_SystemDescription_ENG_2014_web.pdf

[45] "Security extensions for PROFINET—PI white paper for PROFINET, version 1.05," Profibus Int., Karlsruhe, Germany, Rep., Feb. 2019. [Online]. Available: https://de.profibus.com/downloads/pi-white-paper-security-extensions-for-profinet

[46] V. Schiffer, *Common Industrial Protocol (CIP$^{\text{TM}}$) and the Family of CIP Networks*. ODVA, Inc., Ann Arbor, MI, USA, 2016.

[47] ODVA. *CIP Secuirty*. Accessed: Oct. 8, 2020. [Online]. Available: www.odva.org/technology-standards/distinct-cip-services/cip-security

[48] I. González, A. J. Calderón, J. Figueiredo, and J. M. Sousa, "A literature survey on open platform communications (OPC) applied to advanced industrial environments," *Electronics*, vol. 8, no. 5, pp. 1–29, 2019.

[49] B. Rolston, "Security implications of OPC, OLE, DCOM, and RPC in control systems," Idaho Nat. Lab., Idaho Falls, ID, USA, Rep. INL/EXT-05-01005, 2006.

[50] H. Renjie, L. Feng, and P. Dongbo, "Research on OPC UA security," in *Proc. 5th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, 2010, pp. 1439–1444.

[51] G. Clarke, D. Reynders, and E. Wright, *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. London, U.K.: Elsevier, 2004.

[52] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks," in *Proc. ICS-CSR*, 2014, pp. 30–42.

[53] D. Baigent, M. Adamiak, and R. Mackiewicz, *Communication Networks and Systems in Substations: An Overview for Users*, IEC Standard 61850, 2013.

[54] S. M. Hussain, T. S. Ustun, and A. Kalam, "A review of IEC 62351 security mechanisms for IEC 61850 message exchanges," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 5643–5654, Sep. 2020.

[55] H. Yoo and T. Shon, "Challenges and research directions for heterogeneous cyber-physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture," *Future Gener. Comput. Syst.*, vol. 61, pp. 128–136, Aug. 2016.

[56] American Society of Heating and Air-Conditioning Engineers. *BACnet*. Accessed: Oct. 11, 2020. [Online]. Available: http://www.bacnet.org/

[57] W. Xu, Y. Tao, and X. Guan, "The landscape of industrial control systems (ICs) devices on the Internet," in *Proc. Int. Conf. Cyber Situational Awareness Data Anal. Assess.*, 2018, pp. 1–8.

[58] T. Carlsson. *Industrial Network Market Shares 2020 According to HMS Networks*. Accessed: Jan. 29, 2021. [Online]. Available: https://www.hms-networks.com/news-and-insights/news-from-hms/2020/05/29/industrial-network-market-shares-2020-according-to-hms-networks

[59] Á. L. P. Gómez *et al.*, "On the generation of anomaly detection datasets in industrial control systems," *IEEE Access*, vol. 7, pp. 177460–177473, 2019.

[60] N. R. Rodofile, "Generating attacks and labelling attack datasets for industrial control intrusion detection systems," Ph.D. dissertation, School Electr. Eng. Comput. Sci., Queensland Univ. Technol., Brisbane, QLD, Australia, 2013.

[61] N. R. Rodofile, K. Radke, and E. Foo, "Extending the cyber-attack landscape for SCADA-based critical infrastructure," *Int. J. Crit. Infrastruct. Protect.*, vol. 25, pp. 14–35, Jun. 2019.

[62] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proc. LISA*, 1999, pp. 229–238.

[63] Suricata Intrusion Detrction System. Accessed: May 25, 2021. [Online]. Available: https://suricata-ids.org/

[64] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 1–29, 2014.

[65] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *Int. J. Distrib. Sens. Netw.*, vol. 14, no. 8, 2018, Art. no. 155014771879461.

[66] G. Bernieri, M. Conti, and F. Turrin, "KingFisher: An industrial security framework based on variational autoencoders," in *Proc. 1st Workshop Mach. Learn. Edge Sens. Syst.*, 2019, pp. 7–12. [Online]. Available: https://doi.org/10.1145/3362743.3362961

[67] H. R. Ghaeini and N. O. Tippenhauer, "HAMIDS: Hierarchical monitoring intrusion detection system for industrial control systems," in *Proc. 2nd ACM Workshop Cyber Phys. Syst. Security Privacy*, 2016, pp. 103–111.

[68] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware intrusion detection in industrial control systems," in *Proc. 1st ACM Workshop Cyber Phys. Syst. Security*, 2015, pp. 13–24. [Online]. Available: https://doi.org/10.1145/2732198.2732200

[69] Y. Xie, W. Wang, F. Wang, and R. Chang, "VTET: A virtual industrial control system testbed for cyber security research," in *Proc. 3rd Int. Conf. Security Smart Cities Ind. Control Syst. Commun.*, 2018, pp. 1–7.

[70] N. O. Tippenhauer, "Design and realization of testbeds for security research in the industrial Internet of Things," in *Security and Privacy Trends in the Industrial Internet of Things*. Cham, Switzerland: Springer, 2019, pp. 287–310.

[71] M. Almgren, W. Aoudi, R. Gustafsson, R. Krahl, and A. Lindhé, "The nuts and bolts of deploying process-level IDS in industrial control systems," in *Proc. ACM Int. Conf. Series*, 2018, pp. 17–24.

[72] H. G. Aghamolki, Z. Miao, and L. Fan, "A hardware-in-the-loop SCADA testbed," in *Proc. North Amer. Power Symp. (NAPS)*, 2015, pp. 1–6.

[73] T. Alves, R. Das, and T. Morris, "Virtualization of industrial control system testbeds for cybersecurity," in *Proc. 2nd Annu. Ind. Control Syst. Security Workshop*, 2016, pp. 10–14.

[74] T. Cruz *et al.*, "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2236–2246, Dec. 2016.

[75] R. C. B. Hink and K. Goseva-Popstojanova, "Characterization of cyber-attacks aimed at integrated industrial control and enterprise systems: A case study," in *Proc. IEEE Int. Symp. High Assurance Syst. Eng.*, vol. 2016, Mar. 2016, pp. 149–156.

[76] C. Siaterlis, B. Genge, and M. Hohenadel, "EPIC: A testbed for scientifically rigorous cyber-physical security experimentation," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 2, pp. 319–330, Dec. 2013.

[77] *Epic Testbed*. Accessed: May 8, 2020. [Online]. Available: http://sourceforge.net/projects/amici/

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

[79] R. E. Gillen *et al.*, "Design and implementation of full-scale industrial control system test bed for assessing cyber-security defenses," in *Proc. 21st IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 2020, pp. 341–346.

[80] H. Henry, P. Maynard, and K. McLaughlin, "ICS interaction testbed: A platform for cyber-physical security research," in *Proc. ICS-CSR*, 2019, pp. 1–8.

[81] G. Bernieri, F. Del Moro, L. Faramondi, and F. Pascucci, "A testbed for integrated fault diagnosis and cyber security investigation," in *Proc. Int. Conf. Control Decis. Inf. Technol.*, 2016, pp. 454–459.

[82] *Hydra Testbed Repository*. Accessed: Dec. 11, 2020. [Online]. Available: https://github.com/hydra-testbed/Part-list

[83] J. Jarmakiewicz, K. Maślanka, and K. Parobczak, "Development of cyber security testbed for critical infrastructure," in *Proc. Int. Conf. Military Commun. Inf. Syst. (ICMCIS)*, 2015, pp. 1–10.

[84] M. Kaouk, F.-X. Morgand, and J.-M. Flaus. (2018). *A Testbed for Cybersecurity Assessment of Industrial and IoT-Based Control Systems*. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02074654

[85] J. Kim, K. Kim, and M. Jang, "Cyber-physical battlefield platform for large-scale cybersecurity exercises," in *Proc. Int. Conf. Cyber Conflict*, vol. 2019, 2019, pp. 1–19.

[86] G. Koutsandria, R. Gentz, M. Jamei, A. Scaglione, S. Peisert, and C. McParland, "A real-time testbed environment for cyber-physical security on the power grid," in *Proc. 1st ACM Workshop Cyber-Phys. Syst. Security PrivaCy*, 2015, pp. 67–78.

[87] P. Čeleda, J. Vykopal, V. Švábenskỳ, and K. Slavíček, "KYPO4INDUSTRY: A testbed for teaching cybersecurity of industrial control systems," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, 2020, pp. 1026–1032.

[88] J. Rubio-Hernan, J. Rodolfo-Mejias, and J. Garcia-Alfaro, "Security of cyber-physical systems," in *Proc. Int. Workshop Security Ind. Control Syst. Cyber-Phys. Syst.*, 2016, pp. 3–18.

[89] *LegoSCADA Testbed*. Accessed: Jan. 11, 2021. [Online]. Available: http://j.mp/legoscada

[90] F. Guo *et al.*, "Design and development of a reconfigurable hybrid Microgrid testbed," in *Proc. IEEE Energy Conversion Congr. Expo.*, 2013, pp. 1350–1356.

[91] W. Xu, Y. Tao, C. Yang, and H. Chen, "MSICST: Multiple-scenario industrial control system testbed for security research," *Comput. Mater. Continua*, vol. 60, no. 2, pp. 691–705, 2019.

[92] R. Candell, T. Zimmerman, and K. Stouffer, "An industrial control system cybersecurity performance testbed," U.S. Dept. Commer., Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. NISTIR 8089, Nov. 2015.

[93] T. Edgar, D. Manz, and T. Carroll, "Towards an experimental testbed facility for cyber-physical security research," in *Proc. CSIIRW*, 2011, p. 53.

[94] C. Queiroz, A. Mahmood, J. Hu, Z. Tari, and X. Yu, "Building a SCADA security testbed," in *Proc. Netw. Syst. Security*, 2009, pp. 357–364.

[95] V. Urias, B. Van Leeuwen, and B. Richardson, "Supervisory command and data acquisition (SCADA) system cyber security analysis using a live, virtual, and constructive (LVC) testbed," in *Proc. IEEE Military Commun. Conf.*, 2012, pp. 1–8.

[96] D. C. Bergman, D. Jin, D. M. Nicol, and T. Yardley, "The virtual power system testbed and inter-testbed integration," in *Proc. 2nd Workshop Cyber Security Experimentation Test*, Aug. 2009, p. 5.

[97] I. Ahmed, V. Roussev, W. Johnson, S. Senthivel, and S. Sudhakaran, "A SCADA system testbed for cybersecurity and forensic research and pedagogy," in *Proc. ICSS*, 2016, pp. 1–9.

[98] P. Blazek, R. Fujdiak, P. Mlynek, and J. Misurec, "Development of cyber-physical security testbed based on IEC 61850 architecture," *Elektronika Elektrotechnika*, vol. 25, no. 5, pp. 82–87, 2019.

[99] E. Korkmaz, A. Dolgikh, M. Davis, and V. Skormin, "Industrial control systems security testbed," in *Proc. 11th Annu. Symp. Inf. Assurance*, 2016, pp. 13–18.

[100] S. Adepu, N. K. Kandasamy, and A. Mathur, *EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 11387)). Cham, Switzerland: Springer, Nov. 2018, pp. 37–52.

[101] Singapore University of Technology and Design (SUTD). *Electric Power and Intelligent Control (EPIC) Testbed*. Accessed: Jan. 13, 2021. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_epic/

[102] H. K. Shin, W. Lee, J. H. Yun, and H. C. Kim, "Implementation of programmable CPS testbed for anomaly detection," in *Proc. 12th USENIX Workshop Cyber Security Experimentation Test*, 2019, pp. 1–9. [Online]. Available: https://www.usenix.org/conference/cset19/presentation/shin

[103] B. Green, A. Le, R. Antrobus, U. Roedig, D. Hutchison, and A. Rashid, "Pains, gains and PLCs: Ten lessons from building an industrial control systems testbed for security research," in *Proc. 10th USENIX Workshop Cyber Security Experimentation Test*, 2017, p. 4.

[104] F. Sauer, M. Niedermaier, S. Kießling, and D. Merli, "LICSTER—A low-cost ICS security testbed for education and research," in *Proc. 6th Int. Symp. ICS SCADA Cyber Security Res.*, 2019, pp. 1–10.

[105] Hsainnos. *Low-cost ICS Testbed Github Repository*. Accessed: Feb. 10, 2021. [Online]. Available: https://github.com/hsainnos/LICSTER

[106] T. Morris, R. Vaughn, and Y. S. Dandass, "A testbed for SCADA control system cybersecurity research and pedagogy," in *Proc. CSIIRW*, 2011, p. 27.

[107] A. Hahn *et al.*, "Development of the PowerCyber SCADA security testbed," in *Proc. CSIIRW*, 2010, pp. 1–4.

[108] N. Sayegh, A. Chehab, I. H. Elhajj, and A. Kayssi, "Internal security attacks on SCADA systems," in *Proc. 3rd Int. Conf. Commun. Inf. Technol. (ICCIT)*, 2013, pp. 22–27.

[109] *Cx-270322: Idaho National Laboratory (INL) Smart Grid Test Bed Revision 2*. Accessed: Apr. 30, 2020. [Online]. Available: https://www.energy.gov/sites/prod/files/2017/10/f38/CX-270322.pdf

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

[111] Singapore University of Technology and Design (SUTD). *Secure Water Treatment (SWaT) Testbed*. Accessed: Jan. 13, 2021. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/

[112] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, pp. 1–15, 2018.

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

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

[115] Singapore University of Technology and Design (SUTD). *Water Distribution (WADI) Testbed*. Accessed: Jan. 13, 2021. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/

[116] Y. Yang *et al.*, "Multiattribute SCADA-specific intrusion detection system for power networks," *IEEE Trans. Power Del.*, vol. 29, no. 3, pp. 1092–1102, Jun. 2014.

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

[118] C. M. Davis, J. E. Tate, H. Okhravi, C. Grier, T. J. Overbye, and D. Nicol, "SCADA cyber security testbed development," in *Proc. 38th Annu. North Amer. Power Symp.*, 2006, pp. 483–488.

[119] M. Krotofil and J. Larsen, "Rocking the pocket book: Hacking chemical plants for competition and extortion," Black Hat, White Paper, 2015.

[120] *DVCP-TE*. Accessed: Jan. 12, 2021. [Online]. Available: https://github.com/satejnik/DVCP-TE

[121] A. A. Farooqui, S. S. H. Zaidi, A. Y. Memon, and S. Qazi, "Cyber security backdrop: A SCADA testbed," in *Proc. IEEE Comput. Commun. IT Appl. Conf.*, 2014, pp. 98–103.

[122] T. H. Morris, Z. Thornton, and I. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," in *Proc. 7th Annu. Southeastern Cyber Security Summit*, 2015, pp. 1–6. [Online]. Available: http://www.ece.uah.edu/∼thm0009/icsdatasets/cyberhuntsvillepaper_v4.pdf

[123] B. Genge, C. Siaterlis, I. Nai Fovino, and M. Masera, "A cyber-physical experimentation environment for the security analysis of networked industrial control systems," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1146–1161, 2012. doi: http://dx.doi.org/10.1016/j.compeleceng.2012.06.015.

[124] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley, "A testbed for secure and robust SCADA systems," *ACM SIGBED Rev.*, vol. 5, no. 2, pp. 1–4, 2008.

[125] D. Formby, M. Rad, and R. Beyah, "Lowering the barriers to industrial control system security with GRFICS," in *Proc. USENIX Workshop Adv. Security Educ.*, 2018, pp. 1–9.

[126] D. Formby, M. Rad, and R. Beyah. *Grfics*. Accessed: Jan. 11, 2021. [Online]. Available: https://github.com/djformby/GRFICS

[127] D. Jin, D. M. Nicol, and G. Yan, "An event buffer flooding attack in dnp3 controlled scada systems," in *Proc. Winter Simulat. Conf. (WSC)*, 2011, pp. 2614–2626.

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

[129] S. Lee, S. Lee, H. Yoo, S. Kwon, and T. Shon, "Design and implementation of cybersecurity testbed for industrial IoT systems," *J. Supercomput.*, vol. 74, no. 9, pp. 4506–4520, 2018.

[130] P. Maynard, K. McLaughlin, and S. Sezer, "An open framework for deploying experimental SCADA testbed networks," in *Proc. 5th Int. Symp. ICS SCADA Cyber Security Res.*, 2018, pp. 92–101.

[131] P. Maynard, K. McLaughlin, and S. Sezer. *ICS Testbed Framework*. Accessed: May 8, 2020. [Online]. Available: https://github.com/PMaynard/ICS-TestBed-Framework

[132] D. Antonioli and N. O. Tippenhauer, "MiniCPS: A toolkit for security research on CPS networks," in *Proc. 1st ACM Workshop Cyber-Phys. Syst. Security Privacy*, 2015, pp. 91–100.

[133] *MiniCPS: A Framework for Cyber-Physical Systems Real-Time Simulation, Built on Top of Mininet*. Accessed: May 8, 2020. [Online]. Available: https://github.com/scy-phy/minicps

[134] B. Reaves and T. Morris, "An open virtual testbed for industrial control system security research," *Int. J. Inf. Security*, vol. 11, no. 4, pp. 215–229, 2012.

[135] M. Almgren *et al.*, "RICS-el: Building a national testbed for research and training on SCADA security (short paper)," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Security*, 2018, pp. 219–225.

[136] A. Ghaleb, S. Zhioua, and A. Almulhem, "SCADA-SST: A SCADA security testbed," in *Proc. World Congr. Ind. Control Syst. Security (WCICSS)*, 2016, pp. 1–6.

[137] *SCADA-SST—SCADA Security Testbed*. Accessed: Jan. 12, 2021. [Online]. Available: https://sourceforge.net/projects/scada-sst/

[138] C. Queiroz, A. Mahmood, and Z. Tari, "SCADASim—A framework for building SCADA simulations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 589–597, Dec. 2011.

[139] T. Z. Q. Carlos and M. Abdun. *Scadasim on Github*. Accessed: Jul. 26, 2020. [Online]. Available: https://github.com/caxqueiroz/scadasim

[140] A. Almalawi, Z. Tari, I. Khalil, and A. Fahad, "SCADAVT—A framework for SCADA security testbed based on virtualization technology," in *Proc. Conf. Local Comput. Netw.*, 2013, pp. 639–646.

[141] P. Singh, S. Garg, V. Kumar, and Z. Saquib, "A testbed for SCADA cyber security and intrusion detection," in *Proc. Int. Conf. Cyber Security Smart Cities Ind. Control Syst. Commun. (SSIC)*, 2015, pp. 1–6.

[142] M. Mallouhi, Y. Al-Nashif, D. Cox, T. Chadaga, and S. Hariri, "A testbed for analyzing security of SCADA control systems (TASSCS)," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Europe*, 2011, pp. 1–7.

[143] C. Wang, L. Fang, and Y. Dai, "A simulation environment for SCADA security analysis and assessment," in *Proc. Int. Conf. Meas. Technol. Mechatronics Autom.*, vol. 1, 2010, pp. 342–347.

[144] J. Gardiner, B. Craggs, B. Green, and A. Rashid, "Oops I did it again: Further adventures in the land of ICS security testbeds," in *Proc. ACM Conf. Comput. Commun. Security*, 2019, pp. 75–86.

[145] E. Eide, L. Stoller, and J. Lepreau, "An experimentation workbench for replayable networking research," in *Proc. NSDI*, 2007, p. 16.

[146] E. Eide. (May 2010). *Toward Replayable Research in Networking and Systems*. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.170.9948&rep=rep1&type=pdf

[147] S. Choi, J.-H. Yun, B.-G. Min, and H. Kim, "POSTER: Expanding a programmable CPS testbed for network attack analysis," in *Proc. 15th ACM Asia Conf. Comput. Commun. Security*, New York, NY, USA, Oct. 2020, pp. 928–930. [Online]. Available: https://dl.acm.org/doi/10.1145/3320269.3405447

[148] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim. *IL-Based Augmented ICS (HAI) Security Dataset*. Accessed: Oct. 22, 2020. [Online]. Available: https://github.com/icsdataset/hai

[149] E. Korkmaz, A. Dolgikh, M. Davis, and V. Skormin, "ICS security testbed with delay attack case study," in *Proc. IEEE Military Commun. Conf. MILCOM*, 2016, pp. 283–288.

[150] T. Morris *et al. Industrial Control System (ICS) Cyber Attack Datasets*. Accessed: Apr. 27, 2020. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[151] I. Moreno-Garcia, A. Moreno-Munoz, V. Pallares-Lopez, M. Gonzalez-Redondo, E. J. Palacios-Garcia, and C. D. Moreno-Moreno, "Development and application of a smart grid test bench," *J. Clean. Prod.*, vol. 162, pp. 45–60, Sep. 2017.

[152] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Security*, 2016, pp. 88–99.

[153] Y. Chen, C. M. Poskitt, and J. Sun, "Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system," in *Proc. IEEE Symp. Security Privacy (SP)*, 2018, pp. 648–660.

[154] *SWaT_Simulator*. Accessed: Jan. 12, 2021. [Online]. Available: https://github.com/yuqiChen94/Swat_Simulator

[155] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, *A Dataset to Support Research in the Design of Secure Water Treatment Systems* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 10242)). Cham, Switzerland: Springer, Oct. 2017, pp. 88–99.

[156] *WUSTL-IIOT-2018 Dataset for ICS (SCADA) Cybersecurity Research*. Accessed: Jan. 11, 2021. [Online]. Available: https://www.cse.wustl.edu/ jain/iiot/index.html

[157] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of Industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[158] *SUTD-MIT International Design Centre (IDC), Water Distribution (WADI) Testbed*. Accessed: May 6, 2020. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/

[159] *PowerWorld: The Visual Approach to Electric Power Systems*. Accessed: Dec. 15, 2020. [Online]. Available: https://www.powerworld.com/

[160] M. Liljenstam, J. Liu, D. Nicol, Y. Yuan, G. Yan, and C. Grier, "RINSE: The real-time immersive network simulation environment for network security exercises (extended version)," *Simulation*, vol. 82, no. 1, pp. 43–59, 2006.

[161] P. R. Lyman and C. Georgakis, "Plant-wide control of the Tennessee Eastman problem," *Comput. Chem. Eng.*, vol. 19, no. 3, pp. 321–331, 1995.

[162] L. Argenta and M. J. Morykwas, "Vacuum-assisted closure: A new method for wound control and treatment: Clinical experience," *Ann. Plast. Surg.*, vol. 38, no. 6, pp. 563–576, 1997.

[163] *Emulab: A Time- and Space-Shared Platform for Research, Education, and Development in Distributed Systems and Networks*. Accessed: Dec. 14, 2020. [Online]. Available: https://www.emulab.net/

[164] *MATLAB/Simulink: Simulation and Model-Based Design*. Accessed: Dec. 14, 2020. [Online]. Available: https://mathworks.com/products/simulink.html

[165] T. R. Alves, M. Buratto, F. M. De Souza, and T. V. Rodrigues, "OpenPLC: An open source alternative to automation," in *Proc. IEEE Global Humanitarian Technol. Conf. (GHTC)*, 2014, pp. 585–589.

[166] *AdvancedHMI Software*. Accessed: Jan. 11, 2021. [Online]. Available: https://www.advancedhmi.com/

[167] *Oracle VM VirtualBox: A Powerful x86 and AMD64/Intel64 Virtualization Product for Enterprise as Well as Home Use*. Accessed: Dec. 15, 2020. [Online]. Available: https://www.virtualbox.org/

[168] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in *Proc. Int. Conf. Commun. Comput. Syst. (ICCCS)*, 2014, pp. 139–142.

[169] Z. Thornton and T. Morris, *Enhancing a Virtual SCADA Laboratory Using Simulink*. Cham, Switzerland: Springer, 2015, pp. 119–133. [Online]. Available: http://link.springer.com/10.1007/978-3-319-26567-4

[170] *CRATE—Cyber Range and Training Environment*. Accessed: Jan. 11, 2021. [Online]. Available: https://www.foi.se/en/foi/resources/crate-cyber-range-and-training-environment.html

[171] *OMNeT++ Discrete Event Simulator. An Extensible, Modular, Component-Based C++ Simulation Library and Framework, Primarily for Building Network Simulators*. Accessed: Dec. 15, 2020. [Online]. Available: https://omnetpp.org/

[172] *Common Open Research Emulator (CORE)*. Accessed: Dec. 15, 2020. [Online]. Available: https://www.nrl.navy.mil/itd/ncs/products/core

[173] *EPANET: Application for Modeling Drinking Water Distribution Systems*. Accessed: Dec. 15, 2020. [Online]. Available: https://www.epa.gov/water-research/epanet

[174] *OPNET Network Simulator*. Accessed: Dec. 14, 2020. [Online]. Available: https://opnetprojects.com/opnet-network-simulator/

[175] *NetToPLCsim—Network Extension for Plcsim*. Accessed: Dec. 16, 2020. [Online]. Available: http://nettoplcsim.sourceforge.net/

[176] *Getting Started with S7-PLCSIM Advanced and Simulation Tables*. Accessed: Dec. 16, 2020. [Online]. Available: https://cache.industry.siemens.com/dl/files/047/109759047/att_962042/v3/109759047_PLCSIMAdv_SimTable_DOC_V10_en.pdf

[177] *CyberCity Allows Government Hackers to Train for Attacks*. Accessed: Jan. 10, 2021. [Online]. Available: https://www.washingtonpost.com/investigations/cybercity-allows-government-hackers-to-train-for-attacks/2012/11/26/588f4dae-1244-11e2-be82-c3411b7680a9_story.html

[178] *CyberCity SANS Holiday Hack 2013 Dataset*. Accessed: Jan. 10, 2021. [Online]. Available: https://assets.contentstack.io/v3/assets/blt36c2e63521272fdc/bltff8e7c1232f3bcbc/5fbd7be072a3526f28dbed75/sansholidayhack2013.pcap

[179] *Matlab/Simulink Coder: Generate C and C++ Code From Simulink and Stateflow Models*. Accessed: Dec. 14, 2020. [Online]. Available: https://www.mathworks.com/products/simulink-coder.html

[180] *Summit—Oak Ridge National Laboratory's 200 Petaflop Supercomputer*. Accessed: Dec. 22, 2020. [Online]. Available: https://www.olcf.ornl.gov/olcf-resources/compute-systems/summit/

[181] *November 2019—TOP500 Supercomputer Sites*. Accessed: Dec. 22, 2020. [Online]. Available: https://www.top500.org/lists/2019/11/

[182] *Teach, Learn and Make With Raspberry Pi*. Accessed: Dec. 14, 2020. [Online]. Available: https://www.raspberrypi.org/

[183] *OSIsoft*. Accessed: Jan. 12, 2021. [Online]. Available: http://www.osisoft.com/

[184] M. Rollins, *Beginning Lego Mindstorms Ev3*. Berkeley, CA, USA: Apress, 2014.

[185] *Fischertechnik—The Fischertechnik Learning Environment Is Used for Learning and Understanding Industry 4.0 Applications*. Accessed: Jan. 5, 2021. [Online]. Available: https://www.fischertechnik.de/en

[186] NIST. *Tesim on Github*. Accessed: Jul. 31, 2020. [Online]. Available: https://github.com/usnistgov/tesim

[187] *PNNL Control Testbed*. Accessed: Aug. 3, 2020. [Online]. Available: https://controls.pnnl.gov/testbed/

[188] *Arion: Simplifying Models for Controls Research*. Accessed: Dec. 14, 2020. [Online]. Available: https://arion.labworks.org/

[189] D. Maynor, *Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research*. Burlington, MA, USA: Elsevier, 2011.

[190] T. Morris and W. Gao, "Industrial control system traffic data sets for intrusion detection research," in *IFIP Advances in Information and Communication Technology*, vol. 441. Heidelberg, Germany: Springer, 2014, pp. 65–78.

[191] *WEKA—The Workbench for Machine Learning*. Accessed: Jan. 15, 2021. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/

[192] A. Lemay and J. M. Fernandez, "Providing SCADA network data sets for intrusion detection research," in *Proc. 9th USENIX Workshop Cyber Security Experimentation Test*, 2016, p. 8.

[193] 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.*, 2019, pp. 1–6.

[194] R. Taormina *et al.*, "Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks," *J. Water Resources Plan. Manag.*, vol. 144, no. 8, 2018, Art. no. 04018048.

[195] *iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design, iTrust Labs Dataset Info*. Accessed: Apr. 24, 2020. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info

[196] *Batadal Datasets*. Accessed: May 8, 2020. [Online]. Available: http://www.batadal.net/data.html

[197] R. Taormina, S. Galelli, H. Douglas, N. O. Tippenhauer, E. Salomons, and A. Ostfeld, "A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems. environmental modelling software," *Environ. Model. Softw.*, vol. 112, pp. 46–51, Feb. 2019.

[198] N. Rodofile. *SCADA Network Attack Datasets and Process Logs*. Accessed: May 8, 2020. [Online]. Available: https://github.com/qut-infosec/2017QUT_DNP3

[199] N. R. Rodofile, T. Schmidt, S. T. Sherry, C. Djamaludin, K. Radke, and E. Foo, *Process Control Cyber-Attacks and Labelled Datasets on S7Comm Critical Infrastructure* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 10343)). Cham, Switzerland: Springer, 2017, pp. 452–459.

[200] N. Rodofile. *SCADA Network Attack Datasets and Process Logs*. Accessed: May 8, 2020. [Online]. Available: https://github.com/qut-infosec/2017QUT_S7comm

[201] *ICS Lab: 4SICS ICS Lab PCAP File*. Accessed: Apr. 27, 2020. [Online]. Available: https://www.netresec.com/?page=PCAP4SICS

[202] T. Yu, J. Huang, I. Liao, and K. Kao, "Mining anomaly communication patterns for industrial control systems," in *Proc. Aust. Univ. Power Eng. Conf. (AUPEC)*, 2018, pp. 1–6, doi: doi: 10.1109/AUPEC.2018.8757940.

[203] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner, "An evaluation of machine learning methods to detect malicious SCADA communications," in *Proc. 12th Int. Conf. Mach. Learn. Appl.*, vol. 2, 2013, pp. 54–59.

[204] K. Demertzis, L. Iliadis, and S. Spartalis, "A spiking one-class anomaly detection framework for cyber-security on industrial control systems," in *Proc. Eng. Appl. Neural Netw.*, vol. 2, 2017, pp. 122–134. [Online]. Available: http://link.springer.com/10.1007/978-3-319-65172-9_11

[205] *Dataset for Cybersecurity Research in Industrial Control Systems*. Accessed: May 6, 2020. [Online]. Available: http://perception.inf.um.es/ICS-datasets/

[206] G. K. Ndonda and R. Sadre, "Network trace generation for flow-based IDS evaluation in control and automation systems," *Int. J. Crit. Infrastruct. Protect.*, vol. 31, Dec. 2020, Art. no. 100385.

[207] R. Sadre and G. K. Ndonda. *HVAC Traces*. Accessed: May 21, 2020. [Online]. Available: https://github.com/gkabasele/HVAC_Traces

[208] A. Lemay. *Modbus Dataset From CSET 2016*. Accessed: May 8, 2020. [Online]. Available: https://github.com/antoine-lemay/Modbus_dataset

[209] 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. Security*, 2018, pp. 1–9.

[210] I. Frazão, P. H. Abreu, T. Cruz, H. Araújo, and P. Simões, *Denial of Service Attacks: Detecting the Frailties of Machine Learning Algorithms in the Classification Process* (Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics 11260)). Cham, Switzerland: Springer, 2019, pp. 230–235.

[211] I. Frazão, P. Henriques Abreu, T. Cruz, H. Araujo, and P. Simoes. *Modbus TCP SCADA #1 Dataset*. Accessed: Apr. 30, 2020. [Online]. Available: https://github.com/tjcruz-dei/ICS_PCAPS/releases/tag/MODBUSTC%231

[212] P. Radoglou-Grammatikis, I. Siniosoglou, T. Liatifis, A. Kourouniadis, K. Rompolos, and P. Sarigiannidis, "Implementation and detection of modbus cyberattacks," in *Proc. 9th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, 2020, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9200287/

[213] D. Peterson and R. Wightman. *Digital Bond S4X15 ICS Village CTF PCAP Files*. Accessed: Apr. 27, 2020. [Online]. Available: https://www.netresec.com/?page=DigitalBond_S4

[214] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst.*, 2014, pp. 1–8.

[215] Singapore University of Technology and Design (SUTD). *Dataset Characteristics*. Accessed: Jan. 13, 2021. [Online]. Available: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/

[216] D. Myers, S. Suriadi, K. Radke, and E. Foo, "Anomaly detection for industrial control systems using process mining," *Comput. Security*, vol. 78, pp. 103–125, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.cose.2018.06.002

[217] Myers *et al. QUT_S7 Communication Dataset*. Accessed: Dec. 19, 2020. [Online]. Available: https://cloudstor.aarnet.edu.au/plus/index.php/s/9qFfeVmfX7K5IDH

[218] M. F. Abdelaty, R. D. Corin, and D. Siracusa, "DAICS: A deep learning solution for anomaly detection in industrial control systems," *IEEE Trans. Emerg. Topics Comput.*, early access, Apr. 13, 2021, doi: 10.1109/TETC.2021.3073017.

[219] M. Housh and Z. Ohar, "Model-based approach for cyber-physical attack detection in water distribution systems," *Water Res.*, vol. 139, pp. 132–143, Aug. 2018.

[220] Y. Kim and H. K. Kim, "Anomaly detection using clustered deep one-class classification," in *Proc. 15th Asia Joint Conf. Inf. Security (AsiaJCIS)*, Aug. 2020, pp. 151–157. [Online]. Available: https://ieeexplore.ieee.org/document/9194140/

[221] A. Erba *et al. Constrained Concealment Attacks on Reconstruction-Based Anomaly Detectors in Industrial Control Systems*. Accessed: Dec. 7, 2020. [Online]. Available: https://github.com/scy-phy/ICS-Evasion-Attacks

[222] A. Erba *et al.*, "Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems," in *Proc. Annu. Comput. Security Appl. Conf. (ACSAC)*, 2020, pp. 480–495.

[223] M. Kravchik and A. Shabtai, "Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA," *IEEE Trans. Dependable Secure Comput.*, early access, Jan. 8, 2021, doi: 10.1109/TDSC.2021.3050101.

[224] S. Pan, T. Morris, and U. Adhikari, "A specification-based intrusion detection framework for cyber-physical environment in electric power system," *Int. J. Netw. Security*, vol. 17, no. 2, pp. 174–188, 2015.

[225] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 650–662, Jun. 2015.

[226] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.

[227] J. Fürnkranz and G. Widmer, "Incremental reduced error pruning," in *Proc. Mach. Learn.*. 1994, pp. 70–77.

[228] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[229] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, "HAI 1.0: Hil-based augmented ICS security dataset," in *Proc. 13th USENIX Workshop Cyber Security Experimentation Test*, Aug. 2020, pp. 1–5. [Online]. Available: https://www.usenix.org/conference/cset20/presentation/shin

[230] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, 2019, pp. 703–716.

[231] Ö. Yüksel, J. D. Hartog, and S. Etalle, "Reading between the fields: Practical, effective intrusion detection for industrial control systems," in *Proc. ACM Symp. Appl. Comput.*, 2016, pp. 2063–2070.

[232] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2017, pp. 261–272.

[233] K. Demertzis, L. Iliadis, and I. Bougoudis, "Gryphon: A semi-supervised anomaly detection system based on one-class evolving spiking neural network," *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4303–4314, 2020.

[234] A. Mansouri, B. Majidi, and A. Shamisa, "Metaheuristic neural networks for anomaly recognition in industrial sensor networks with packet latency and jitter for smart infrastructures," *Int. J. Comput. Appl.*, vol. 43, no. 3, pp. 257–266, 2018. [Online]. Available: https://doi.org/1206212X.2018.1533613

[235] J. Jägersküpper, "How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms," *Theor. Comput. Sci.*, vol. 361, no. 1, pp. 38–56, 2006.

[236] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[237] Á. L. Perales Gómez, L. Fernández Maimó, A. Huertas Celdrán, F. J. García Clemente, M. Gil Pérez, and G. Martínez Pérez, "SafeMan: A unified framework to manage cybersecurity and safety in manufacturing industry," *Softw. Pract. Exp.*, vol. 51, no. 3, pp. 607–627, 2021.

[238] Y. Li *et al.*, "Cross-domain anomaly detection for power industrial control system," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun.*, 2020, pp. 383–386.

[239] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symp. Comput. Intell. Security Defense Appl.*, 2009, pp. 1–6.

[240] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[241] P. Schneider and K. Böttinger, "High-performance unsupervised anomaly detection for cyber-physical system networks," in *Proc. ACM Conf. Comput. Commun. Security*, 2018, pp. 1–12.

[242] S. D. Anton, L. Ahrens, D. Fraunholz, and H. D. Schotten, "Time is of the essence: Machine learning-based intrusion detection in industrial time series data," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDMW)*, 2018, pp. 1–6.

[243] J. Luswata, P. Zavarsky, B. Swar, and D. Zvabva, "Analysis of SCADA security using penetration testing: A case study on modbus tcp protocol," in *Proc. 29th Biennial Symp. Commun. (BSC)*, 2018, pp. 1–5.

[244] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Tor traffic using time based features," in *Proc. ICISSP*, 2017, pp. 253–262.

[245] D. Bond. *Quickdraw Snort*. Accessed: Oct. 9, 2020. [Online]. Available: https://github.com/digitalbond/Quickdraw-Snort

[246] *NMAP: The Network Mapper—Free Security Scanner*. Accessed: Dec. 11, 2020. [Online]. Available: hhttps://nmap.org/

[247] *SUTD Security Showdown 2017*. Accessed: Apr. 27, 2020. [Online]. Available: https://itrust.sutd.edu.sg/scy-phy-systems-week/2017-2/s317-event/

[248] F. Turrin, A. Erba, N. O. Tippenhauer, and M. Conti, "A statistical analysis framework for ICS process datasets," in *Proc. Joint Workshop CPS IoT Security Privacy*, 2020, pp. 25–30.

[249] C. M. Ahmed, J. Zhou, and A. P. Mathur, "Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in CPS," in *Proc. 34th Annu. Comput. Security Appl. Conf.*, 2018, pp. 566–581.

[250] Q. Lin, S. Adepu, S. Verwer, and A. Mathur, "Tabor: A graphical model-based approach for anomaly detection in industrial control systems," in *Proc. Asia Conf. Comput. Commun. Security*, 2018, pp. 525–536.

[251] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *Proc. NDSS*, 2019, pp. 1–15.

[252] *Security for Industrial Automation and Control Systems. Security Risk Assessment for System Design*, IEC Standard 62443-3-2:2020, 2020.

[253] *Cyber Security—Critical Cyber Asset Identification*, IEC Standard NERC CIP-002-3 through CIP-009-3, 2009.

[254] "ICS security related working groups, standards and initiatives for the report: Good practices for an EU ICS testing," Eur. Union Agency Cybersecurity, Heraklion, Greece, Rep., Dec. 2013, doi: 10.2824/26451.

[255] J. L. Torres, C. A. Catania, and E. Veas, "Active learning approach to label network traffic datasets," *J. Inf. Security Appl.*, vol. 49, Dec. 2019, Art. no. 102388.

[256] M. Zolanvari, M. A. Teixeira, and R. Jain, "Effect of imbalanced datasets on security of industrial IoT using machine learning," in *Proc. IEEE Int. Conf. Intell. Security Informat. (ISI)*, Nov. 2018, pp. 112–117. [Online]. Available: https://ieeexplore.ieee.org/document/8587389/

[257] D. Ramyachitra and P. Manikandan, "Imbalanced dataset classification and solutions: A review," *Int. J. Comput. Bus. Res.*, vol. 5, no. 4, pp. 1–29, 2014.

[258] S. K. Lim, Y. Loo, N. Tran, N. Cheung, G. Roig, and Y. Elovici, "DOPING: Generative data augmentation for unsupervised anomaly detection with gan," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2018, pp. 1122–1127.

[259] G. O. Diaz and V. Ng, "Modeling and prediction of online product review helpfulness: A survey," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, vol. 1, 2018, pp. 698–708.

Mauro Conti (Senior Member, IEEE) received the Ph.D. degree from the Sapienza University of Rome, Italy, in 2009. He is a Full Professor with the University of Padova, Italy. He is also affiliated with TU Delft and University of Washington, Seattle. After his Ph.D., he was a Postdoctoral Researcher with Vrije Universiteit Amsterdam, The Netherlands. In 2011, he joined as an Assistant Professor with the University of Padua, where he became Associate Professor in 2015, and a Full Professor in 2018. He has been a Visiting Researcher with GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He has been awarded with a Marie Curie Fellowship in 2012 by the European Commission, and with a Fellowship by the German DAAD in 2013. His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of security and privacy. In the above area, he published more than 350 papers in topmost international peer-reviewed journals and conferences. He was a Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, and ACNS 2020, and a General Chair for SecureComm 2012, SACMAT 2013, CANS 2021, and ACNS 2022. He is an Area Editor-in-Chief for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and an Associate Editor for several journals, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He is Senior Member of ACM.

Denis Donadel received the master's degree in telecommunication engineering from the University of Padova, Italy, in 2020, where he is currently a Predoctoral Research Fellow. His research interests lie primarily in cyber-physical systems security with a particular focus on critical infrastructures security, electric vehicles security, and Industrial Internet of Things.

Federico Turrin received the master's degree in computer engineering from the University of Padova, Italy, in 2019, where he is currently pursuing the interdisciplinary Ph.D. degree in brain, mind, and computer science. His research interests lie primarily in cyber-physical system security with a particular focus on critical infrastructures security, Industrial Internet of Things, and application of machine learning techniques for anomaly detection.