Computers & Security 125 (2023) 103028

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

# SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues

Manar Alanazi[a,*], Abdun Mahmood[a], Mohammad Jabed Morshed Chowdhury[a]

*Department of Computer Science and Information Technology School of Engineering and Mathematical Science La Trobe University, Bundoora, Melbourne, VIC 3086, Australia*

ARTICLE INFO

*Article history:*
Received 8 January 2022
Revised 16 September 2022
Accepted 22 November 2022
Available online 25 November 2022

*Keywords:*
SCADA vulnerabilities
Cyber-threats
Testbed
Intrusion detection
Taxonomy

ABSTRACT

Supervisory control and data acquisition (SCADA) serves as the backbone of several critical infrastructures, including water supply systems, oil pipelines, transportation and electricity. It accomplishes essential functions, such as monitoring data from pumps, valves and transmitters. Across different generations, SCADA has undergone a significant evolution from a typically isolated environment to a highly interconnected network. Although this conversion has benefits for SCADA, such as enhanced performance efficiency and the cost reduction of heavy equipment, it has made SCADA more vulnerable to various cyber-attacks. Several SCADA security approaches are still provided by IT-based systems that are possibly not efficient enough to deflect the risks and threats originating from SCADA field operations. As a result, it is critically important to analyse cyber risks associated with the industrial SCADA system. The goal of this survey is to explore the security vulnerabilities of SCADA systems and classify the threats accordingly. In this project, we initially reviewed SCADA systems from different scopes, including architecture, vulnerabilities, attacks, intrusion detection techniques (IDS) and testbeds. We proposed taxonomies of vulnerabilities, attacks, IDS and testbeds according to predefined criteria. We concluded the survey by highlighting the research challenges and open issues for future research in the field of SCADA security.



## 1. Introduction

Industrial control systems (ICSs), including supervisory control and data acquisition (SCADA), play a significant role in controlling field devices. They serve as the underpinning technology for critical infrastructures (CIs) and manufacturers. CIs include energy, transportation systems, critical manufacturing and healthcare, as shown in Fig. 1. Cyber-physical systems (CPSs) are integrated with the Internet of Things (IoT) to supplement information-rich operations to conventional CIs (Corallo et al., 2020; Ding et al., 2017). As SCADA systems significantly evolved through four generations, from the monolithic generation to IoTisation, the security level for each generation changed as well. ICSs have several advantages by combining SCADA with the IoT and a cloud environment, such as enhanced cost reduction, flexibility and performance efficiency (Sajid et al., 2016). However, the number of cyber threats against SCADA has risen rapidly due to increased remote access and internet connectivity. In extreme cases, the failure to protect SCADA from such attacks threatens human lives. For example, an adversary can control the water supply system of a city, shut down electricity or induce malfunctions in nuclear reactors. Fig. 2 demonstrates the cyberattacks against SCADA in the past years.

### 1.1. Motivations

The Siberian pipeline explosion in 1982 is believed to be the first cyber incident in the history of SCADA systems (Ismail et al., 2014). A malicious user injected a Trojan horse into the SCADA system to modify the operations of valves and pumps. The malicious code made the gas pressure exceed the acceptable level. Then, in 1994, an attacker gained unauthorised access to the Salt River project through a dial-up modem and was able to steal and modify customer information and the log files of the computer system (Fillatre et al., 2017). In 1999, an attacker broke into Gazprom, the largest gas company in Russia, using a Trojan horse. The attacker gained full control of the central switchboard responsible for monitoring gas flow through the pipelines (Fillatre et al., 2017). Similarly, in 2000, an attacker gained control of 150 wastewater pumping stations using a radio transmitter in Maroochy Shire, Queensland, Australia (Fillatre et al., 2017; Sajid et al., 2016). He

* Corresponding author.
*E-mail address:* ManarAhmedT.ALANAZI@latrobe.edu.au (M. Alanazi).

https://doi.org/10.1016/j.cose.2022.103028

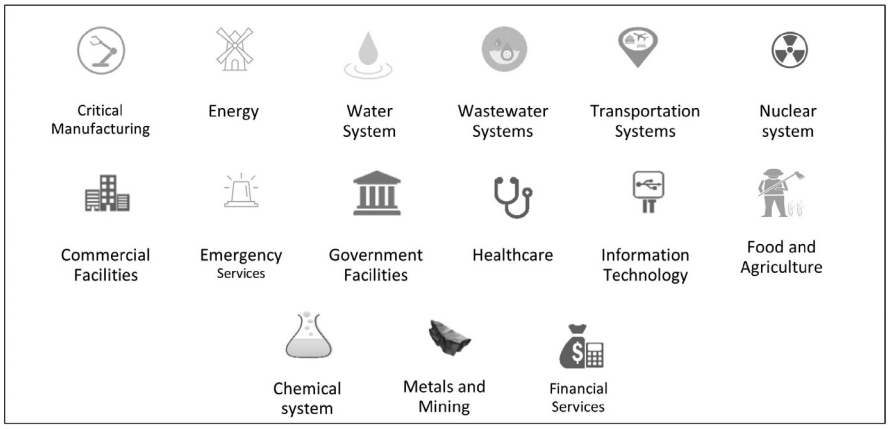


**Fig. 1.** SCADA application areas.

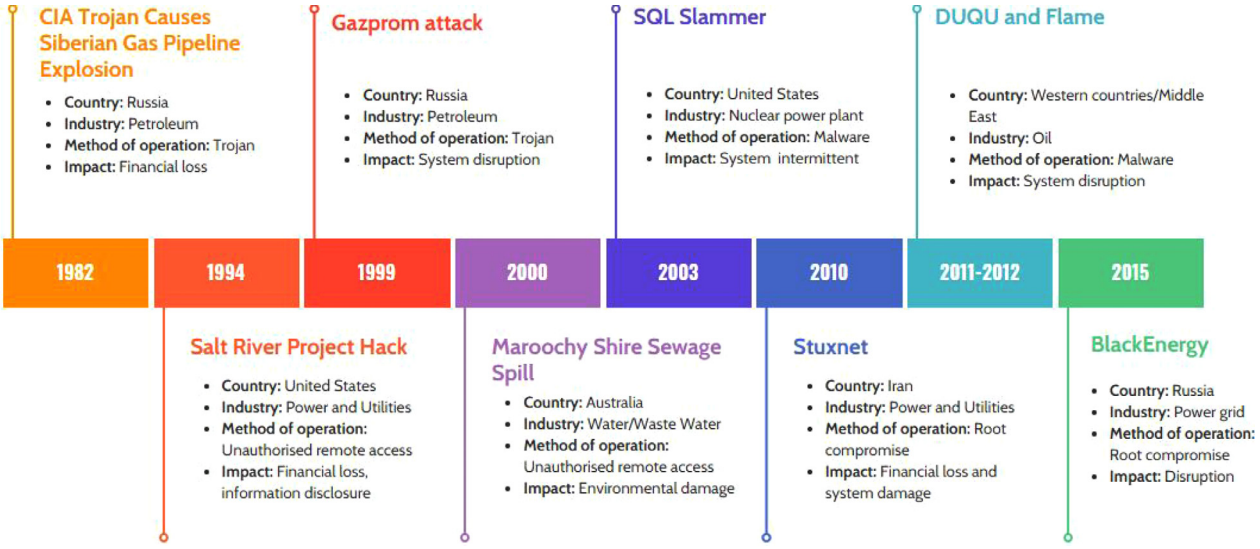


**Fig. 2.** SCADA incidents from 1982 to 2012.

caused a malfunction in the operations of the wastewater system while driving around the area and trying to issue radio commands to the sewage equipment. In 2003, a slammer worm exposed SCADA by exploiting the vulnerability of the MS-SQL database. The worm spread from the enterprise to the SCADA network and disabled a safety monitoring system for around five hours (Miller and Rowe, 2012). In 2010, the Stuxnet worm, which originated from an infected removable drive and hid while propagating, damaged the entire Iranian nuclear system (Falliere et al., 2011). In 2011 and 2012, the malware Duque and Flame appeared. Duque was identical to Stuxnet but had a different purpose. Its goal was to gather information, which could be used by the attacker to conduct future malicious activities (espionage). Similarly, Flame collected technical diagrams, such as for textiles, to conduct stealthy future attacks (Fillatre et al., 2017; Sajid et al., 2016). In 2015, the power grid in Russia was hacked, resulting in a power outage for around 225,000 consumers (Mesbah and Azer, 2019; US-CERT, b). Thus, it is very important to understand and know the total landscape of SCADA vulnerabilities.

### 1.2. Objectives

The survey aims to perform a longitudinal and extensive analysis of the SCADA system with regard to cybersecurity and information warfare. Several research domains in SCADA security are reviewed and critically evaluated to guarantee end-to-end security. The disciplines include SCADA architecture, vulnerabilities, attacks, intrusion detection techniques (IDS) and testbeds. It aims to guide future researchers in the area of ICS security, including SCADA networks. SCADA architecture and communication protocols can help us understand cyber security issues and challenges. Mainly, they answer the question *'What targeted component is the adversary looking for?'* Understanding how SCADA components communicate and connect answers the question *'What are the weaknesses and vulnerabilities in the targeted component the adversary is looking for?'* Understanding the nature of the vulnerabilities in the targeted component answers the question *'How can the adversary use the vulnerable product to launch further attacks?'* Furthermore, understanding the attack chain against the SCADA network can help us develop proper security mechanisms to prevent or at least mitigate future attacks, including IDS. Any proposed security solutions must be trained and validated using SCADA datasets. Due to the lack of datasets for SCADA systems, testbeds are introduced to tackle this issue.

### 1.3. Related works and our contributions

Although several studies have been conducted in the field of SCADA security threats, these studies did not provide a comprehensive analysis of such vulnerabilities and threats. Several pro-

posed taxonomies of SCADA vulnerabilities have either focused on the hardware/software level or network/device level (Corallo et al., 2020; Ding et al., 2017; Ghosh and Sampalli, 2019; Irmak and Erkek, 2018; Papp et al., 2015; Sajid et al., 2016; Xu et al., 2017; Yampolskiy et al., 2013). Corallo et al. (2020) proposed a structural classification of crucial industrial assets in the context of Industry 4.0 and the impact of cyberattacks on business performance. The primary objective of their research was to analyse cybersecurity in terms of the impact on the confidentiality, availability and integrity of data associated with an industrial process through a networked manufacturing machine. However, the study did not highlight SCADA-related vulnerabilities and attacks. A survey by Sajid et al. (2016) focused on the security challenges of IoT-SCADA in a cloud environment, but the survey did not analyse all the security vulnerabilities across the SCADA system functionalities. Ghosh and Sampalli (2019) extended the work conducted by Sajid et al. (2016). Their survey focused on the recent threats against SCADA communication and provided a comparative analysis of SCADA security schemes and standards. Bartman and Carson (2016) performed similar work as Ghosh and Sampalli (2019) but not as comprehensively (Ghosh and Sampalli, 2019). Xu et al. (2017) also presented a taxonomy of cyberattacks on SCADA systems, but the survey only focused on attacks that target SCADA communication protocols. A taxonomy proposed for SCADA by Zhu et al. (2011) categorised attacks at the network, hardware and software levels. While the attacks on software were grouped according to the exploitation of embedded operating systems without privileges, the attack categories in the communication stack were similar to the work done by Ghosh and Sampalli (2019) and Xu et al. (2017). Yampolskiy et al. (2013) proposed a taxonomy that maps cross-domain attacks on SCADA systems. The key feature of this taxonomy is that an influenced element (e.g. an object manipulated by an attack) and the victim element (e.g. an interaction existing in a CPS) are independent of each other. Additionally, each of them can be either in a physical domain or a cyber domain. However, Papp et al. (2015) stated that the proposed taxonomy was generic without attack information. Therefore, Papp et al. (2015) developed the taxonomy proposed by Yampolskiy et al. (2013) but with some modifications in the content and structure. Nevertheless, the precondition dimension defined by Yampolskiy et al. (2013) is more properly mapped to the prerequisite of attack execution. Irmak and Erkek (2018) surveyed some attack vectors that target the SCADA system. The study was quite similar to Zhu et al. (2011), but it did not embody an analysis of SCADA vulnerabilities and attacks as comprehensively as Zhu et al. (2011). Despite the features of those surveys by Ding et al. (2017), Sajid et al. (2016), Ghosh and Sampalli (2019), Xu et al. (2017) and Bartman and Carson (2016), they were still limited to the network level. In other words, those studies did not cover SCADA host-based attacks. Our survey is an extension of the existing taxonomies presented by Yampolskiy et al. (2013) and Zhu et al. (2011), with modifications in terms of content and structure. The contributions of this survey, in comparison with the existing relevant studies, are as follows:

- A security requirement analysis for SCADA and information technology (IT) systems (Section 2.4).
- A comprehensive taxonomy of SCADA vulnerabilities (Section 4).
- A number of taxonomies related to SCADA, namely the types of attacks, targets, deliveries, causes, consequences and impacts of attacks (Section 6).
- The classification and evaluation of the current SCADA IDS (Section 7).
- The classification and evaluation of the current state-of-the-art SCADA-based testbeds (Section 8).
- Existing control and mitigation mechanisms in Section 9.
- A presentation of the current security challenges and open issues for SCADA systems (Section 10).

In addition, we highlight our contributions compared with the exiting literature in Table 1.

### 1.4. Paper organisation

Section 2 describes the background of SCADA in detail, including the architecture and security requirements, and Section 1.3 describes the related works and contributions. Afterwards, Section 3 outlines the review methodology used to conduct this survey. It has two selection process phases. The first part shows the paper selection process, and the second part presents the vulnerability construction process. Additionally, it presents the evaluation methodology for the SCADA IDS and testbeds. Next, Section 4 describes the relationships between SCADA's vulnerabilities, and Section 6 provides a taxonomy of potential attacks against SCADA systems. Sections 7 and 8 also provide a taxonomy of the existing IDS and testbeds. Section 8 evaluates the strengths and weaknesses of the current testbeds, and Section 10 concludes the review with the open issues and security challenges. Section 9 describes the existing controls and mitigation mechanisms to manage the identified risks. Fig. 3 describes the outline of the survey paper.

## 2. Background

It is important to understand SCADA architecture before conducting any security analysis. SCADA has experienced a dramatic change since its convergence with the internet. This section of the paper describes the SCADA life cycle and evaluates the security measures of each SCADA generation. Additionally, it provides a pairwise comparison of ICS and IT requirements.

### 2.1. SCADA components

SCADA utilises a central computer to store information on local or remote devices to control industrial processes and facilities. We can classify the typical SCADA components according to their definitions, as illustrated in Fig. 4.

- **Supervisory control:** It is the primary function of the human-machine interface (HMI). HMI software is an interface that is accountable for the supervision of industrial processes. By contrast, a master terminal unit (MTU) is a central supervisory controller that communicates with lower field devices, such as remote terminal units (RTUs), over the ICS network.
- **Data acquisition:** Data can be acquired from a programmable logic controller (PLC) and RTUs. A PLC is a solid-state device that facilitates decision making by continually controlling and monitoring local industrial physical processes (Mehra, 2012). A PLC utilises sensors to obtain the current state of a process, based on the logic in the PLC, and then sends it to its respective control centre to be graphically displayed by the HMI to the control operator. A PLC performs three processes, known as scanning, while it is in operation. It reads and accepts the inputs from a field device via an input interface, then it executes a control program stored in the memory, and, finally, it writes and updates output devices via an output interface (Senthivel et al., 2017). The RTU and PLC functionalities are overlapped. They both act as physical interfaces between SCADA

**Table 1**
A comparison between the related works and our contributions.

| Criteria | Sub-criteria | | Surveyed papers | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Corallo et al. (2020) | Sajid et al. (2016) | Ghosh and Sampalli (2019) | Bartman and Carson (2016) | Xu et al. (2017) | Zhu et al. (2011) | Papp et al. (2015) | Yampolskiy et al. (2013) | Irmak and Erkek (2018) | Our |
| | **Supervisory control** | HMI | ◐ | ◐ | ◐ | ○ | ◐ | ○ | ◐ | ○ | ● | ● |
| | | MTUs | ◐ | ◐ | ◐ | ◐ | ○ | ○ | ◐ | ○ | ○ | ● |
| | | PLCs | ◐ | ◐ | ◐ | ○ | ◐ | ◐ | ◐ | ◐ | ● | ● |
| | | RTUs | ◐ | ◐ | ● | ○ | ○ | ○ | ◐ | ◐ | ○ | ● |
| | **Data acquisition** | IEDs | ◐ | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ● | ● |
| | **Data exchange** | | ◐ | ● | ● | ◐ | ◐ | ◐ | ● | ● | ● | ● |
| **Architecture** | **Data Storage** | Historian | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ● | ● |
| | | (CWE/NVD) | ○ | ○ | ○ | ○ | ◐ | ◐ | ◐ | ○ | ○ | ● |
| **Vulnerabilities relationship** | **SCADA-specific vulnerabilities** | Other sources | ◐ | ◐ | ◐ | ◐ | ◐ | ○ | ◐ | ○ | ◐ | ● |
| | Local | | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ● |
| | Reconnaissance | | ○ | ◐ | ○ | ○ | ○ | ● | ● | ◐ | | ● |
| **Attack Vector** | Remote/Network | | ○ | ◐ | ● | ◐ | ○ | ● | ● | ● | ● | ● |
| | Preconditions | | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ◐ | ● |
| | Weaponization | | ○ | ○ | ● | ○ | ◐ | ◐ | ○ | ○ | ○ | ● |
| **Attack surfaces** | Delivery | | ○ | ○ | ● | ◐ | ● | ◐ | ○ | ◐ | ◐ | ● |
| | Memory corruption | | ○ | ◐ | ◐ | ○ | ◐ | ○ | ○ | ○ | ○ | ● |
| | DoS/DDoS | | ○ | ● | ● | ● | ○ | ● | ◐ | ● | ● | ● |
| | Information exposure | | ○ | ◐ | ◐ | ○ | ◐ | ● | ○ | ◐ | ● | ● |
| | Red and modify memory/data | | ○ | ◐ | ◐ | ● | ○ | ◐ | ◐ | ● | ● | ● |
| **Consequences** | Ladder logic modification | | ○ | ○ | ○ | ○ | ○ | ○ | ◐ | ○ | ○ | ● |
| | Availability | | ● | ◐ | ● | ● | ○ | ● | ○ | ● | ● | ● |
| | Integrity | | ● | ◐ | ● | ● | ○ | ● | ◐ | ◐ | ● | ● |
| **Attack Impact** | Confidentiality | | ● | ◐ | ● | ● | ○ | ● | ○ | ◐ | ● | ● |
| | Attack detection | | ◐ | ● | ● | ◐ | ○ | ○ | ○ | ● | ○ | ● |
| **Best practices** | Security validation | | ◐ | ◐ | ○ | ◐ | ○ | ○ | ○ | ○ | ○ | ● |
| **Research challenges and open issues** | | | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| **Critical analysis** | | | ● | ○ | ● | ◐ | ◐ | ○ | ○ | ○ | ○ | ● |
| **Time-frame** | | | 2011–2018 | 2011–2016 | 2011–2019 | 2011–2014 | 2011–2017 | 2003–2012 | 2011–2015 | 2003–2012 | 2009–2017 | 2011–2021 |

●Fully covered ◐Partially covered ○Not covered

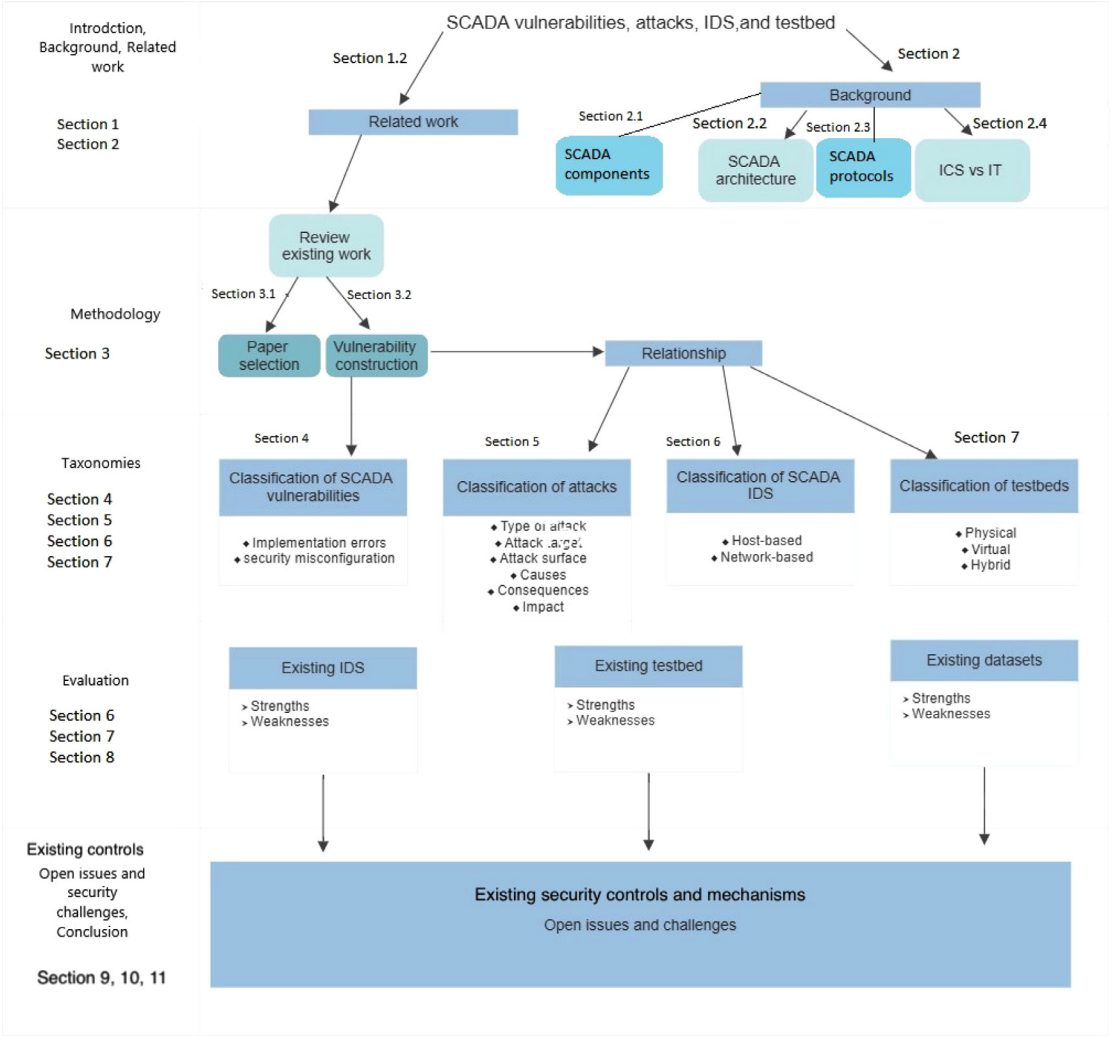


**Fig. 3.** The survey outline.

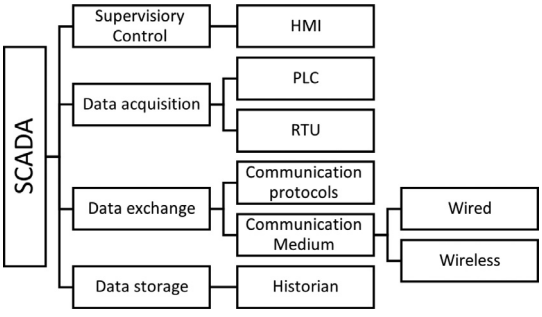


**Fig. 4.** SCADA components.

and the field devices. However, the way they communicate with SCADA is different. RTUs are suitable for wide geographical areas because they use wireless communication. By contrast, PLCs are more tailored to local control.

- **Data storage:** Most SCADA systems use a structured query language (SQL) database to store timestamped data. Historian is a fully integrated SCADA software that collects real-time data from various SCADA devices and stores them in a database, such as SQL.
- **Data exchange:** Communication protocols are used to exchange data between SCADA components. More details about the SCADA communication protocol are provided in Section 2.

### 2.2. SCADA architecture

This section describes the generations of SCADA architecture in detail and provides a summary of each one's strengths and weaknesses.

#### 2.2.1. Monolithic SCADA system

The first generation of SCADA systems, monolithic SCADA, appeared when the terms 'internetworking' and 'interconnectivity' did not yet exist, and it is based on standalone mainframes, which have high computing capabilities (Manoj, 2019; Sen and Dey, 2020). The main components of monolithic SCADA are the master units, including the mainframes and RTUs connected via wide area networks (WANs). One mainframe acts as the primary control system while another mainframe system serves as a standby system to capture any failures of the first one. The primary purpose of WAN implementation is to allow an RTU to interchange data with a master unit (Joshi et al., 2019).

As today's WAN communication protocols were not known to the monolithic SCADA system, RTU vendors developed commu-

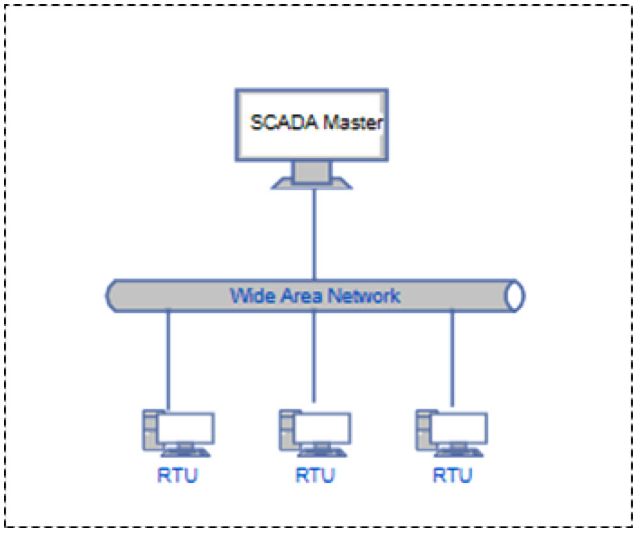


**Fig. 5.** Monolithic SCADA system.

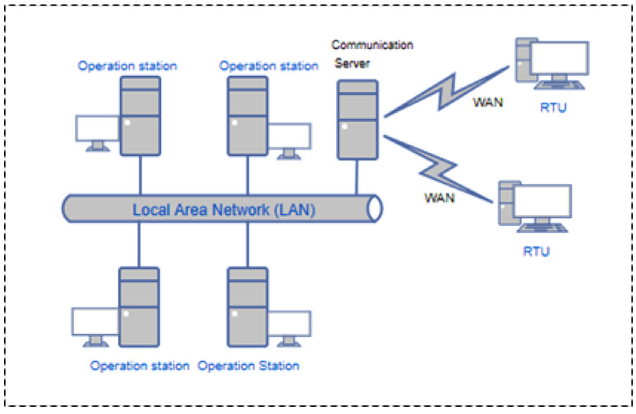


**Fig. 6.** Distributed SCADA system.

nication protocols that were applicable only to proprietary master computers from the same vendor. These communication protocols did not allow any functionalities beyond scanning the data interchange between the RTUs and the master computer (Sen and Dey, 2020). Fig. 5 shows the SCADA monolithic architecture. The connectivity of the RTUs to the master computer is based on the bus level connected via an adapter to the backplate of the central processing units (CPUs) in the master computer (Almalawi et al., 2020). Therefore, it is not feasible to interchange various types of traffic between the RTUs and the master computer. These systems are air-gapped and standalone so that they are not accessible by anyone. However, monolithic SCADA is expensive to maintain, as it requires a standby switching scheme in case of failure. Industrial requirements have forced vendors to address this limitation of SCADA; thus, monolithic SCADA has been converted to a distributed architecture.

*2.2.2. Distributed SCADA system*

The distributed SCADA architecture is based on the existence of internetworking and system miniaturisation. It has made SCADA tinier and less expensive in comparison with the monolithic architecture (Joshi et al., 2019; Sajid et al., 2016). Fig. 6 presents the major components of a distributed SCADA system. It includes the operating stations, mini-computers, RTUs and HMIs.

The configuration of distributed SCADA is based on a central bus associated with a local area network (LAN) to connect the distributed operating stations so that they can interchange data in

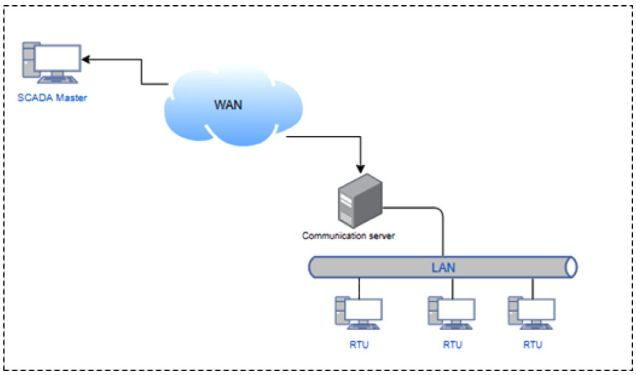


**Fig. 7.** Networked SCADA architecture.

real time (Joshi et al., 2019). By contrast, WANs are implemented to connect RTUs to the communication server using proprietary protocols. What makes a distributed SCADA more reliable is the distributed functionalities in which each operating station has a specific function (Sajid et al., 2016; Shaw, 2006). Some of these distributed operating stations serve as a communication processor with the RTUs, and others serve as the operator interface, providing HMIs for the operator. Additionally, the distributed functionalities can improve redundancy by keeping all the stations online (Sen and Dey, 2020). For instance, in case of an HMI failure, another HMI can operate. However, this system is still limited to proprietary protocols, software and peripheral hardware. Consequently, industrial growth and increased demands by industries and manufacturers have led SCADA to be networked.

*2.2.3. Networked SCADA system*

The networked SCADA architecture, as shown in Fig. 7, is quite similar to the distributed network. However, the main difference is that networked SCADA is oriented to commercial off-the-shelf systems using the internet protocol (IP) instead of proprietary protocols (Sajid et al., 2016).

The configuration of networked architecture is based on connecting the master control station with the RTUs over WAN using the IP, whereas the other SCADA components exchange data over Ethernet (Joshi et al., 2019). The networked SCADA architecture has led SCADA to spread across more than LAN. Additionally, this architecture has enhanced the performance level of SCADA by allowing several servers to run in parallel to handle several tasks (Manoj, 2019).

*2.2.4. IoT SCADA system*

The current SCADA architecture is integrated with the IoT and a cloud environment, as shown in Fig. 8. From human-centric, the internet became thing-centric due to the emergence of the IoT (Manoj, 2019), which brought several advantages to SCADA, such as ease of use, flexibility, availability, cost efficiency, big data processing and scalability (Manoj, 2019; Sajid et al., 2016).

However, IoT integration has exposed SCADA systems to several attacks due to the integration of CPSs with the IoT and a cloud environment (Huong et al., 2021; Sajid et al., 2016). Table 2 summarises the strengths and weaknesses of each SCADA architecture in terms of performance and security, showing that SCADA has been optimised without security in mind. Therefore, SCADA's security decreases significantly due to the connection to the internet. Evidently, monolithic SCADA is the most secure architecture because it is air-gapped from external networks.

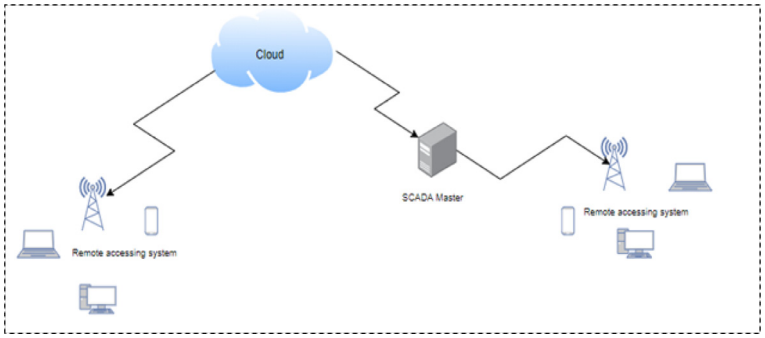


**Fig. 8.** IoT SCADA architecture.

**Table 2**
A comparison between different SCADA systems.

| SCADA generations | Network optimisation | | | | | Security | References |
|---|---|---|---|---|---|---|---|
| | Reliability | Cost efficiency | Scalability | Redundancy | latency | | |
| Monolithic SCADA | Not reliable | Very expensive | Not scalable | Standby switching scheme | High | Air gapped | Sen and Dey (2020) Joshi et al. (2019) |
| Distributed SCADA | More reliable than monolithic SCADA | Less expensive | Scalable | Distributed functionalities | Reduced compared to monolithic SCADA | Security via obscurity | Sen and Dey (2020) Joshi et al. (2019) |
| Networked SCADA | Reliable | Affordable due to off the shelf systems | Very scalable | Parallel servers | Data processing in real time led to reduce latency | Physical security concerns | Sen and Dey (2020) Joshi et al. (2019) |
| IoT SCADA | Exceptionally reliable | Economical and affordable | Very scalable | Cloud environment | Exceptionally low | TLS/SSL | Sen and Dey (2020) |

### 2.3. SCADA communication protocols

Initially, MTUs and RTUs communicated via a wired link, such as a dial-up modem interface (Joshi et al., 2019). However, as wired links are restricted to small-scale networks, industry players moved to more advanced protocols to achieve scalability. In a distributed SCADA system, MTUs and RTUs communicate using a communication server and proprietary protocols. Modbus-RTU, Profibus and Conitel are vendor-specific communication protocols widely adopted for the traditional SCADA system (Joshi et al., 2019). If one component utilises a proprietary protocol, it may not be able to exchange data with other components that are not vendor-specific. This presents a significant challenge in terms of interoperability. Using a standardised and open communication protocol resolves the challenge of interoperability and has several advantages, such as ease of expansion, high flexibility, interoperability and independence.

#### 2.3.1. Modbus

Developed by Gould-Modicon and currently owned by Schneider Electric, Modbus is a communication protocol running on diverse arrays of physical layers (Ghosh and Sampalli, 2019; Liptak, 2018). Commonly used in SCADA to exchange data between MTUs and RTUs, the Modbus protocol has several features, such as being open-source and easy to use. It has two versions:

- **Serial Modbus:** This version is based on the master-slave architecture and supports traditional serial protocols, such as RS 422/RS 485, to allow data transmission between PLCs/RTUs (slaves) and a Modbus device (master) over a serial line (Liptak, 2018; Pliatsios et al., 2020). Before responding to the master, the slave must apply a cyclic redundant checksum to validate the integrity and reliability of the data (Clarke and Reynders, 2004). If the slave receives a normal message, the code in the response message must be identical to the code in the request. However, if an error is detected on the slave side, it modifies the code and sends data to describe the code. The master then checks that code and confirms that the message content is valid (Clarke and Reynders, 2004).
- **Modbus TCP/IP:** This version is based on client-server architecture and supports data transmission between the client and the server over an Ethernet network (Goetz and Shenoi, 2007). It enhances serial Modbus by allowing multiple clients to communicate with multiple servers. Although Modbus does not provide a timestamp, the open platform communication (OPC) server assigns a timestamp according to the local time where the Modbus communication occurs. It does not require a checksum mechanism because the lower layers provide checksum protection (Manoj, 2019). According to Zhu et al. (2011) and Byres et al. (2006), Modbus lacks encryption or other security measures but supports integrity checks.

#### 2.3.2. Profibus

It has the same master-slave feature as Modbus. In a Profibus protocol, all devices follow the proper sequence to connect to a network. Each slave maintains a timer. The slave goes into a state known as a safe state if the master does not talk to a particular slave at the right time. If the master wants to exchange data, it has to go through the start-up sequence again (Clarke and Reynders, 2004).

#### 2.3.3. DNP3

It is an open-source communication protocol. The primary objective of proposing this protocol is to allow flexible and secure communication between different devices. It follows a line and peer-to-peer topology. With several features and advantages, this protocol is commonly used in the power and water sectors, oil and gas, transportation and power generation. It plays a significant role

in the SCADA system and its used devices (i.e. RTUs, intelligent electronic devices and HMIs). In the DNP3 protocol, layer 2 provides error checking, multiplexing and data fragmentation. A transport function is added to layer 4, and layer 7 (application layer protocol) defines the functions and generic data types (Clarke and Reynders, 2004). Data timestamping occurs on a device level (the device is timestamped based on the internal clock of the slave). According to Xu et al. (2017), this protocol does not provide sufficient authentication.

#### 2.3.4. IEC61850

It is an international standard that provides logical communication between a substation and electrical field devices and supports multicast communication. The idea of introducing IEC 61850 is to replace copper wiring with messages called Generic Object Oriented Substation Events (GOOSE). These messages can regularly check whether the receiver is online. In comparison with copper wiring, GOOSE can easily detect broken communication. According to Xu et al. (2017), IEC61850 supports authentication but lacks integrity.

#### 2.3.5. IEC 60870

It is an international standard that allows power plants to control units in geographically distributed locations and supports interoperability and multicast communication.

The interoperability documents provided by the vendors of a remote unit can facilitate a control centre in a communication configuration on that remote terminal unit. Information is initially sent to an RTU without timestamping. However, the RTU initialises a timestamp with an application service data unit. The control centre then creates a sequence of events that occurs in that terminal unit. According to Xu et al. (2017), Maynard et al. (2014), Yang et al. (2013), IEC 60870 lacks authentication control but supports integrity. Table 3 summarises and compares the common SCADA communication protocols in terms of performance and security.

### 2.4. Security requirement analysis for SCADA and IT systems

From a security analyst's perspective, understanding the variance between IT and SCADA security requirements is important because the operative nature of SCADA is different from IT. For example, IT consists of hardware and software and can be protected by a firewall and other security measures without affecting the system's performance. Conversely, firewalls are not recommended by SCADA vendors because a firewall interferes with a remote procedure call, and the operative nature of SCADA does not allow any jitter or delay. Therefore, a security analyst should consider the system's requirements before applying proper security measures. Table 4 describes the difference between SCADA and IT requirements.

#### 2.4.1. Security objectives (CIA/AIC triad)

IT is more info-centric, and its security priorities are confidentiality, integrity and availability (CIA). By contrast, availability is the highest priority in SCADA security, followed by integrity and confidentiality (AIC), respectively (Goetz and Shenoi, 2007; Nazir et al., 2017; Sommestad et al., 2010; Stouffer and Falco, 2006).

#### 2.4.2. Performance and availability

SCADA's availability is the most critical security objective for assuring safety and security. As SCADA controls real-time processes, jitter and delay are not acceptable. In most cases, SCADA uses a maximum latency of $\leq$ one second (Lin et al., 2013). Furthermore, SCADA systems must be operated 24/7, and outages must be scheduled weeks in advance. Thus, redundant systems are required in case an outage occurs. By contrast, IT systems do not require a

**Table 3**
Summary and comparison of SCADA communication protocols.

| Protocol | Attribute | | | | | | | | | | | | | | | | References |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Source | | Timestamp | | Topology | | | | | | Security Weaknesses | | | | Multiple data support | | |
| | Open source | Proprietary | On host machine | On device | Peer 2 Peer | Ring | Star | Bus | Tree | Line | Lack of authentication | Lack of encryption | Lack of integrity | Fixed | Flexible | | |
| Modbus | | | | | | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | Liptak (2018), Plaisios et al. (2020), Clark and Niblet (1989) |
| DNP3 | | | | ✓ | ✓ | | | ✓ | | | ✓ | ✓ | ✓ | | ✓ | | Liptak (2018) |
| Profibus | | | | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | | | Liptak (2018) |
| IEC61850 | ✓ | | ✓ | | | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | | ✓ | | Manoj (2019) |
| IEC 60870 | ✓ | | ✓ | | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | ✓ | ✓ | | Byres et al. (2006), Maynard et al. (2014) |

**Table 4**
The differences between information technology and SCADA cybersecurity requirements.

| Attribute | | IT | SCADA |
|---|---|---|---|
| Security objectives | Confidentiality, integrity and availability (CIA) | | |
| | Availability, integrity, and confidentiality (AIC) | | ✓ |
| Security testing | Standards | | |
| | Simulators and testbeds | | ✓ |
| Performance | Fast performance | | ✓ |
| | Jitter and delay | ✓ | |
| | Real-time processing | ✓ | |
| | Redundancy | | ✓ |
| Communication protocols | Standard | ✓ | |
| | Vendor-specific | ✓ | ✓ |
| Resource constraint | Support security capabilities | ✓ | |

redundant system, as the availability requirement is the least priority in an IT environment. Moreover, the availability requirement depends on the operational nature of the IT system (Goetz and Shenoi, 2007; Sommestad et al., 2010).

*2.4.3. Patch management*

Both IT and SCADA require patch management. On the one hand, IT systems control patch management by pushing patches as soon as they are released. On the other hand, a SCADA system can be left unpatched due to the operational nature of the system Nazir et al. (2017). Furthermore, some SCADA systems use older versions of firmware that are no longer supported by vendors. Hence, the currently available patches may not be applicable (Goetz and Shenoi, 2007).

*2.4.4. Resource constraint*

SCADA systems are resource-constrained and rarely support additional security capabilities due to a lack of computing resources. By contrast, IT systems can support additional security features, such as encryption and password protection (Larkin et al., 2014; Nazir et al., 2017).

*2.4.5. Security testing*

It is expensive to test SCADA systems in real time, as this may lead to a crash, hampering plant operations. By contrast, applying a testing methodology in an IT environment is not as expensive as in a SCADA system. Hence, testbeds and simulators are required to assess the security vulnerabilities of a SCADA system.

*2.4.6. Communication*

SCADA components communicate with each other using vendor-specific or proprietary protocols that are built with no security in mind. However, IT system components communicate with each other using standard protocols that employ security solutions, such as encryption and authentication (Ding et al., 2017; Goetz and Shenoi, 2007; Manoj, 2019; Nazir et al., 2017; Sen and Dey, 2020).

## 3. Review methodology

We followed two approaches to conduct the review. The first approach was related to paper selection according to predefined criteria, and the second approach was related to vulnerability construction.

### 3.1. Paper selection process

We chose all the sources included in this paper from top journals, such as Elsevier, IEEE and ACM. We organised the sources once they met all the criteria. The process is shown in Fig. 9. The selection approach consisted of five stages:

- Searching techniques: We used keywords (e.g. SCADA, HMI, PLC, IDS and testbed) to identify the resources. We applied this method for each scope of our research, including SCADA architecture, attacks, vulnerabilities, IDS and testbeds. For the SCADA architecture, we used keywords to search for the SCADA generations: monolithic SCADA and distributed, networked and IoT-based SCADA systems. A similar procedure was followed for SCADA IDS and testbeds. For SCADA IDS, we selected the papers according to our predefined criteria (i.e. data-centric, either in the host or in the network, and the detection methods, such as signature-based or anomaly-based). For the testbeds, we also assigned the selected papers according to predefined criteria, such as the type of testbed: physical, virtual or hybrid.
- We assessed the selected papers based on the inclusion and exclusion criteria. If the selected article was from a peer-reviewed journal, it proceeded from the first to the second stage. Any research paper written in a language other than English was excluded. Opinion or comment papers were excluded as well.
- We checked the research scope of the selected papers. If they was SCADA-specific, they proceeded to the final stage. Studies that were not directly related to SCADA were excluded.
- In the final stage, a paper was excluded if it was outdated, or the scope of research was not related to SCADA.

### 3.2. Vulnerabilities construction

This section of the survey illustrates the methodology for constructing the SCADA vulnerabilities.

- We referred to the Common Weakness Enumeration (CWE, 2020), Common Vulnerabilities and Exposures (CVE, 2020), the National Vulnerability Database (NVD)(NIST, 2020) and the United States Computer Emergency Readiness Team (US-CERT, a) and mapped them to our taxonomy.
- We obtained a JSON file from the NVD website (https://nvd.nist.gov/vuln/data-feeds) to identify SCADA-related vulnerabilities. The JSON feeds included previous and new vulnerabilities databases in JSON format.
- We then extracted the information from that JSON file using a power query in Microsoft Excel.
- We filtered all the vulnerabilities assigned by the US-CERT. Finally, we sorted the SCADA vulnerabilities using the keywords SCADA, US-CERT, HMI, PLC, RTU, Historian, Siemens, Omron, Rockwell Automation, etc. Fig. 10 presents the vulnerabilities construction process.

The key reasons we chose those databases as the data sources for collecting the SCADA vulnerabilities are as follows:

- Completeness: NVD accommodates CWEs that are synchronised with it. Additionally, NVD contains all the vulnerabilities related to SCADA products. They were assigned by either the vendor or

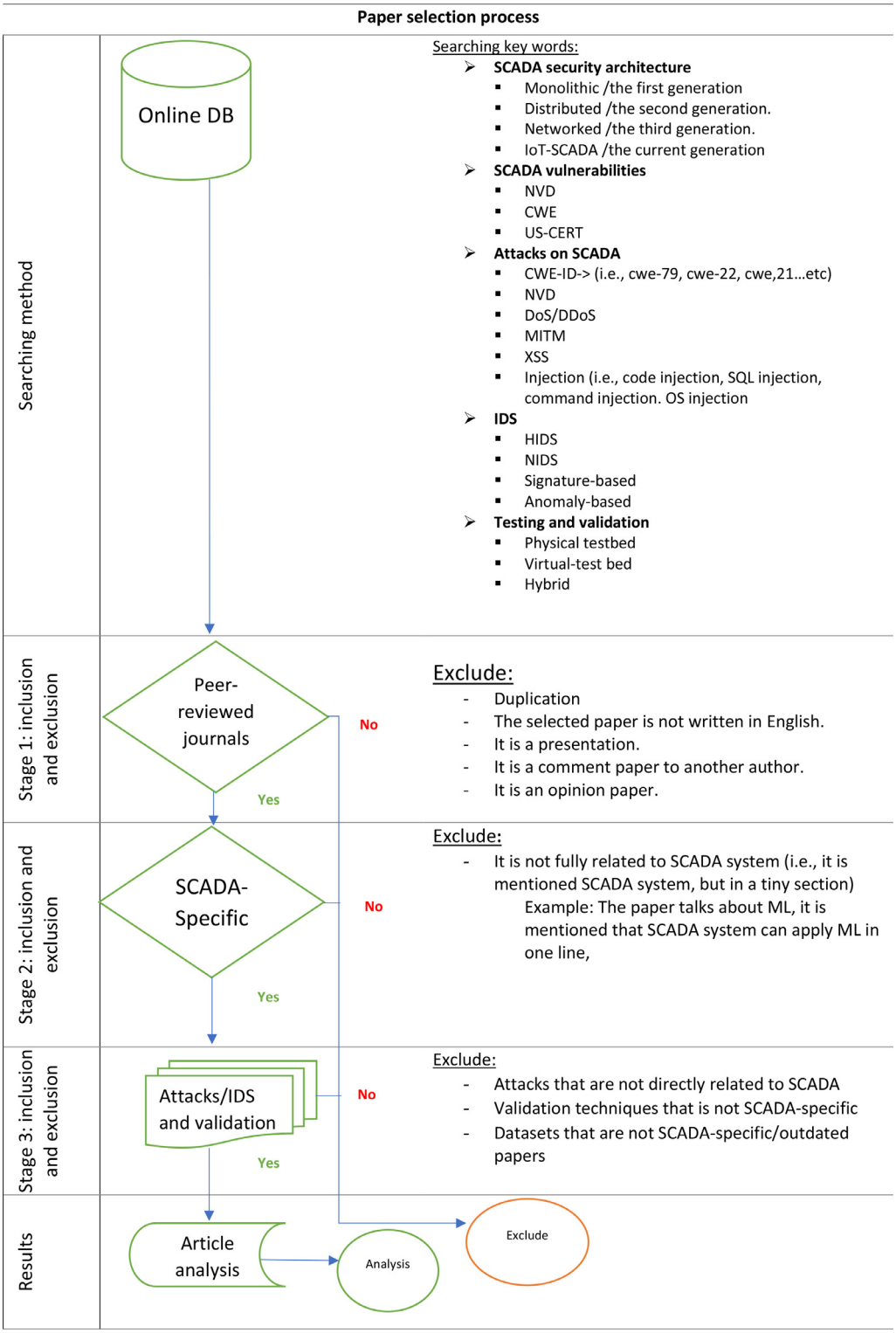


**Fig. 9.** Paper selection process.

the US-CERT. It also accommodates impact metrics, which we can map to our taxonomy.

- Inclusiveness: The US-CERT accommodates inclusive information about SCADA vulnerabilities that are also fully synchronised with CVE, NVD and CWE. It provides information about affected SCADA products and lists the industries and countries that deploy the affected SCADA products.

### *3.3. Attack taxonomy TAACCI methodology*

After we reviewed the SCADA-related vulnerabilities, we classified the attacks into six categories. The TAACCI taxonomy included the type of attack, attack target, attack surface, causes, consequences and impact.

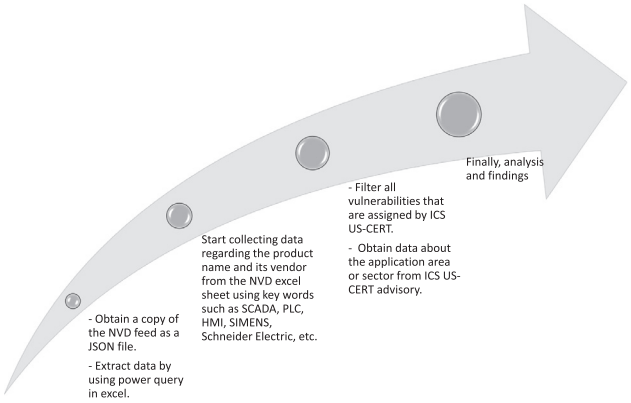


**Fig. 10.** Vulnerabilities selection process.

1. Type of attack: It describes the type of attack a malicious user intends to conduct.
2. Attack target: Attackers can target various SCADA components according to their intentions and objectives. They may target hardware, software or communication protocols and compromise a targeted component to conduct further attacks in the future.
3. Attack surface: It is a collection of vulnerabilities that can be exploited. For a defender, it is important to understand the attack surface because an attacker may cause several future attacks resulting from the previous attacks. The attack surface includes the precondition, attack weapon and operation. Preconditions are requirements for successful attacks (e.g. user interaction and remote access). Attack weapons are the tools used to launch malicious activities, and the attack operation describes the attack flaw.
4. Causes: It describes the reasons an attack occurs. This category is related to SCADA vulnerabilities.
5. Consequence: It describes the results of an attack (e.g. corruption or disruption of SCADA services) and the potential threats that result from the primary one.
6. Impact: It describes the impact of an attack (i.e. on availability, confidentiality or integrity).

We then described the various published attacks on SCADA systems identified in CVE (2020); NIST (2020). Finally, we populated our taxonomy with the attacks on SCADA systems. Each category is demonstrated in Section 6.

### 3.4. Methodology of SCADA IDS evaluation

Intrusion detection involves the following modules: data acquisition, feature extraction, detection, and alert. For the data acquisition module, data are collected from a host or a network for further inspection. The feature extraction module extracts the behavioural features of the collected data. Finally, classification applies a machine-learning algorithm to detect the number of intrusion attempts, and then the alert module reports the event-classified attacks (OSSEC, 0000). Section 7 describes SCADA IDS and evaluates the strength and weaknesses of host-based IDS and network-based IDS based on the following five features:

1. Management: Is it possible for the IDS to manage many hosts in a large environment?
2. Encrypted traffic analysis: Is it possible for the SCADA IDS to inspect encrypted traffic?
3. Zero-day attack detection: Is it possible for the SCADA IDS to detect zero-day attacks?
4. Attack detection in real time: Is it possible for the SCADA IDS to detect real-time attacks?
5. Does the SCADA IDS affect the host's performance?

### 3.5. Methodological evaluation for SCADA testbeds

Any proposed security solution needs a reliable test environment that satisfies its requirements, including repeatability, accuracy and fidelity (Geng et al., 2019). In a broad sense, a test environment includes datasets, testbeds and attack simulation. Several approaches have been used to test and validate security solutions for SCADA systems. This section describes three types of SCADA testbed design approaches: physical, virtual and hybrid (Tripwire, 0000). According to Mallouhi et al. (2011), the primary requirements of a SCADA testbed design are as follows:

1. *Reproducibility:* Is the testbed reproducible and reusable by other researchers?
2. *Scalability:* Does the testbed support a diverse array of devices without any need to redesign the system?
3. *Domain fidelity:* Does the testbed use physical hardware, and is it configured realistically?
4. *Process simulation:* Does the testbed mimic a real-time process?
5. *Multiple ICS protocols:* Does the testbed support more than one ICS protocol?
6. *Cost effectiveness:* Is the testbed affordable for use in SCADA security research?

## 4. SCADA security vulnerabilities

The underlying causes of attacks against IT and SCADA systems result from vulnerabilities. For instance, SQL injection attacks result from improper input sanitisation in both IT and ICS systems. However, a single vulnerability in a SCADA environment can be more far-reaching or critical than in an IT system. Previously, several IT vulnerabilities and exploits did not work in air-gapped ICS environments. Nevertheless, as ICS vendors moved towards commercial off-the-shelf (COTS) network protocols, application architectures and operating systems, these problems started to affect control system networks. Now, many exploits exist for ICS networks' key components, and the vulnerabilities can be exploited via the web. SCADA was built without security considerations; thus, the previous SCADA architectures lacked proper security measures. One vulnerability can flow to many and consequently increase the attack surface so that an attacker can perform further attacks in the future. For example, advanced persistent threats (APTs) deal with protocols in which attackers monitor network activities and steal information for future attacks. APTs deal with protocols in which attackers monitor network activities and steal information for future attacks (Ahmed et al., 2012; Joshi et al., 2019). Hence, understanding the nature of well-known SCADA vulnerabilities and the attack surface is significant for determining possible future attacks. This section illustrates the classification of SCADA vulnerabilities across a system. We referred to (CWE, 2020) as a dictionary of vulnerabilities that expose SCADA during the implementation and misconfiguration of security features. CWE plays a vital role in understanding the patterns of hardware and software vulnerabilities.

### 4.1. Implementation issues

The vulnerabilities in this section are caused due to implementation errors in design and architecture. A programmer does not validate inputs because they believe an attacker cannot modify them. For example, a programmer assumes that the cookies in a web browser cannot be modified. However, any hidden form in a web browser can be altered using a proxy or VPN. Fig. 11 shows the relationships between the vulnerabilities. Each subsection is a cause that leads to another cause. For example, insufficient input validation (cause) leads to a buffer overflow (another cause),

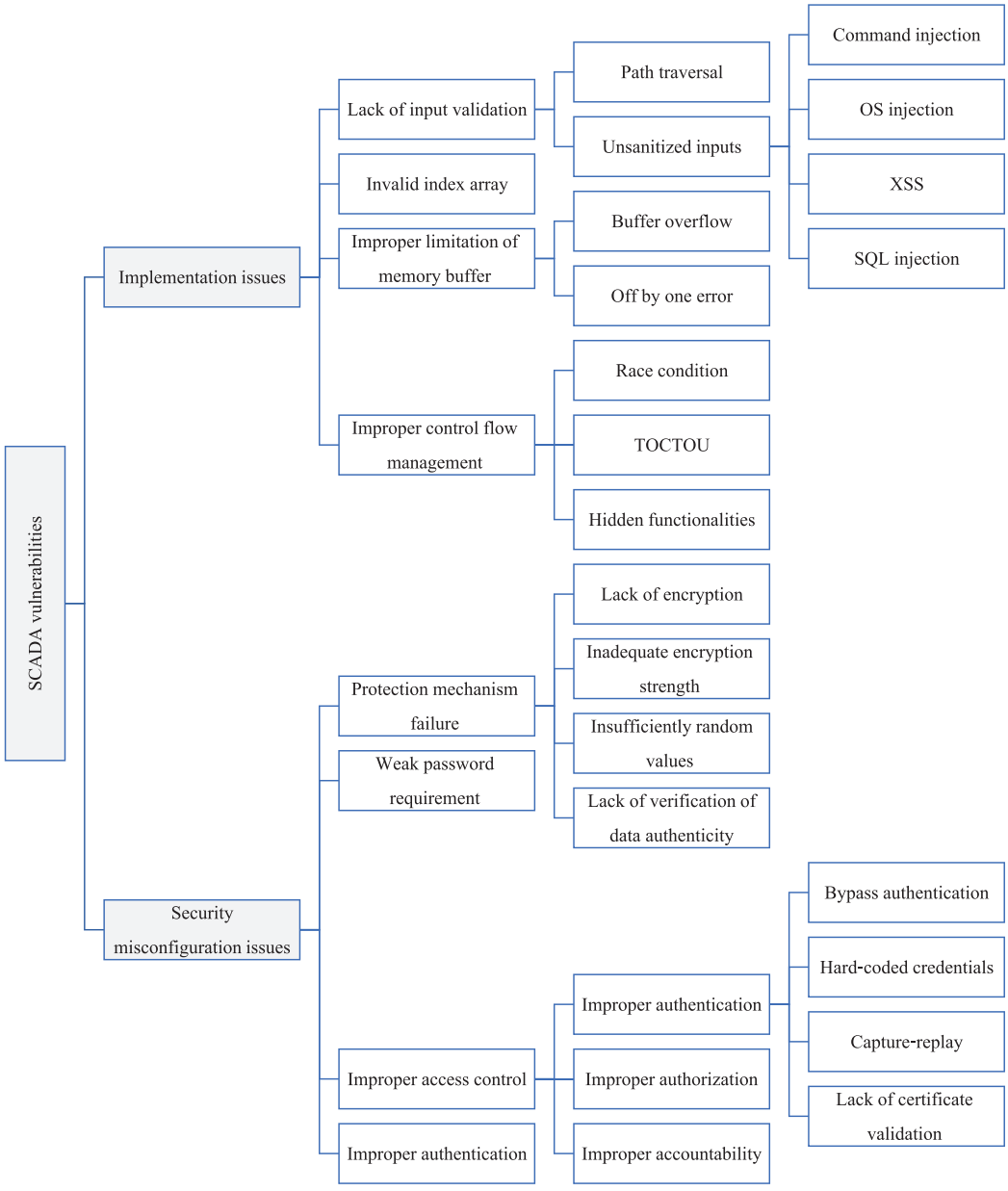


**Fig. 11.** Taxonomy of SCADA vulnerabilities.

which, in turn, leads to a denial of service (DoS) attack (consequence).

*4.1.1. Lack of input validation*

This type of weakness occurs when SCADA software receives inputs from other components but does not validate that those inputs are correct and safe to proceed with (Ding et al., 2017; Sajid et al., 2016; Shahzad et al., 2014). It affects both SCADA hardware and software. This vulnerability can cause other vulnerabilities, such as

- **Path traversal:** It is a web application vulnerability that results from incorrect input validation. A malicious user can use special elements ('dir/../../filename') to trick a web browser. It then returns the pathname to a location that is outside the root directory. Sometimes, the software only checks for one separator (../) at the beginning of the input, but multiple separators, such as (/../../), can bypass that check (CWE-20, 2020; CWE-22, 0000).
- **Lack of input neutralisation:** Any software constructs a data structure or commands from an upstream component, such as a user-controlled data plane. However, the software does not neutralise element inputs (Yadav and Paul, 2019). Those element inputs can be interpreted as control elements. This typically can cause other vulnerabilities, namely
  - **Command injection:** It is a type of special element injection. It occurs when the software receives untrusted inputs from untrusted sources, which are manipulated and implemented as commands by the program. Once a command is executed, the program allows privilege escalation (CWE-22, 0000; CWE-74, 0000).
  - **OS injection:** It is similar to a command injection in that the product's software receives untrusted data from an untrusted source without validation. From a vulnerability perspective, two variants describe the programmers' errors. On the one hand, the application takes OS commands as inputs to be executed. On the other hand, the application receives

inputs from the user as OS commands without input validation. An RTU or PLC employs proprietary operating systems (e.g. Microware OS-9 and VxWorksPliatsios et al. (2020). These products are potentially the most vulnerable to cyberattacks because OSs enhance the attack surface (CWE-77, 0000; Samtani et al., 2016; Seri et al., 2019). LAquis SCADA before 4.1.0.4150 is assigned to this vulnerability. Successful exploitation leads to remote code executions on a SCADA server (US-CERT, 2019b).

- **Cross-site scripting (XSS):** It occurs due to a lack of input neutralisation. XSS occurs when a web application receives untrusted inputs from untrusted parties. The code is injected by an attacker in two ways: reflected or stored. In reflected XSS, an attacker's payload is sent directly to a victim as a malicious link to be excuted. In stored XSS, an attacker's payload is stored in a database and waits for a victim to browse the web page to be executed (CWE-78, 0000; Irfan et al., 2015; Yampolskiy et al., 2013). The Wonderware Intouch HM server is assigned to the XSS vulnerability. Successful exploitation leads to arbitrary code execution or session hijacking (US-CERT, 2014b).
- **SQL injection:** The primary cause of SQL injection is improper input validation. The product constructs SQL queries from upstream components without input sanitation. An attacker can modify the SQL queries directed to a downstream component (CWE-79, 0000). Advantech WebAccess/SCADA is assigned to the SQL injection vulnerability. Successful exploitation leads to information disclosure US-CERT (2018b).

#### 4.1.2. Improper validation of array index

This type of vulnerability occurs due to insufficient input validation. The software does not validate the index references when it receives inputs from the upstream components (CWE, 2020; Irfan et al., 2015).

#### 4.1.3. Untrusted search path

The program looks for critical resources using an untrusted search path that can refer to resources that are not under the direct control of the application. SoMove SCADA software up to 2.6.1 is assigned to this vulnerability. A successful exploit leads to data link library (DLL) hijacking (CVE, 2020; CWE, 2020).

#### 4.1.4. Improper limitation of memory buffer

This type occurs due to the lack of input validation when the software reads from and writes to the memory buffer outside of the intended limits (CWE-426, 0000). It can lead to the following vulnerabilities:

- **Buffer overflow:** It occurs when inputs are not checked properly. If a data array or buffer size is smaller than the input size, it will lead to buffer overflow (Ding et al., 2017; Ghosh and Sampalli, 2019; Irmak and Erkek, 2018; Xu et al., 2017). WebAccess/SCADA software is assigned to this vulnerability (US-CERT, 2019a). Successful exploitation leads to remote code execution and file deletion.
- **Off by one error:** It is a common SCADA software issue in which the program calculates incorrect maximum (N + 1) or minimum (N - 1) values (Dimitrov and Syarova, 2019; Irmak and Erkek, 2018).

#### 4.1.5. Improper control flow management

During execution, the code does not effectively manage its flow, which can unexpectedly modify the execution logic. It can lead to

- **Race conditions:** A race condition occurs when multiple processes and events operate on shared data (CWE-250, 0000; Irmak and Erkek, 2018; Smith, 2014). For example, two processes of shared data read x as 8, given that x = x + 8, which is then assigned back to x. When the first process in the progress reads x as 8, the second process starts reading x as 8. When the first process increments x by 8 to 16, the second process increments x to 16 again. This is incorrect because two processes were executed, and both should have increased x by 8 to 24.
- **Time-of-check time-of-use:** It is a race condition vulnerability in which one process checks the resourse before using it, but the resource may be changed between the check and the use (Bartman and Carson, 2016). Successful exploitation of this vulnerability leads to memory corruption, file directory alteration and data modification.
- **Server-side request forgery:** It is also known as a cross-site port attack. Recently, the mymbCONNECT24 SCADA server was assigned to this vulnerability (NVD, 2020). A successful exploit of this can lead to red data, arbitrary code execution and privilege escalation (Wei et al., 2015; Yeboah-Ofor and Boachie, 2019).
- **Hidden functionality:** The SCADA software component may have a function that does not belong to a specification that is visible to users or administrators. It can exist in the operating system of any computer used for programming PLCs. Successful exploitation of this vulnerability leads to execution logic alteration (Irmak and Erkek, 2018; Smith, 2014).

### 4.2. Security misconfiguration issues

The vulnerabilities that occur due to the implementation of weak security techniques are as follows:

#### 4.2.1. Protection mechanism failure

- **Lack of encryption:** This vulnerability occurs when the SCADA software fails to encrypt data before storage or transmission Ding et al. (2017); Ghosh and Sampalli (2019); Papp et al. (2015). PLC Modicon Premium, Modicon Quantum, Modicon M340, and BMXNOR0200 controllers are assigned to this vulnerability. A successful exploit leads to information exposure.
- **Inadequate encryption strength:** This weakness results from another weakness: reversible one-way hash. When the SCADA software or the SCADA product employs a weak encryption mechanism, it can be subjected to brute force attacks (CWE, 2020; Ding et al., 2017; Ghosh and Sampalli, 2019; Papp et al., 2015; Smith, 2014). IEC 61850 system configurator is assigned to this vulnerability. A successful exploit leads to information exposure.
- **Insufficient random values:** This weakness occurs when the software generates a predictable random value. From a weakness standpoint, there are different variants of predictable random values. The variants are insufficient entropy, predictable from an observable state, a predictable exact value from previous values and predictable value range from previous values (CWE, 2020). Rockwell Automation Wi-Fi Protected Access (WPA and WPA2) that supports IEEE 802.11r is assigned to this vulnerability. This allows an attacker within radio range to decrypt a frame passing between the SCADA components (CVE, 2020).
- **Insufficient verification of data authenticity:** The SCADA software may accept invalid data due to the lack of data authenticity. It results from an origin validation error, which leads to a session fixation vulnerability (CWE, 2020; Ding et al., 2017; Papp et al., 2015). Session fixation occurs when a user is authenticated without validating the existing session ID. It helps an attacker steal a valid session (CWE, 2020; Ding et al., 2017; Papp et al., 2015; Sajid et al., 2016).

#### 4.2.2. Weak password requirement

The software does not force a strong password (Ghosh and Sampalli, 2019; Irmak and Erkek, 2018). A successful exploit leads to privilege escalation.

#### 4.2.3. Improper access control

The software improperly restricts or allows access to critical resources (Ding et al., 2017; Ghosh and Sampalli, 2019; Irmak and Erkek, 2018; Papp et al., 2015; Sajid et al., 2016; US-CERT, 2018c). Access control includes the AAA protection mechanism: authorisation, which ensures that critical resources are accessible by authorised actors; authentication, which identifies the actors; and accountability, which tracks the activities of the actors.

#### 4.2.4. Improper authentication

It occurs when the software does not or improperly validates the identity of a user. Different variants of improper authentication exist:

- **Bypass authentication:** A product requires authentication, but that product has another path that does not require authentication (CWE, 2020; Irmak and Erkek, 2018). Ecava IntegraXor that used to run web-based HMI for the SCADA system is assigned to this vulnerability.
- **Use of hard-coded credentials:** This vulnerability is also known as a backdoor. Weaknesses arise because authentication credentials, such as token and cryptographic keys, cannot be changed (CWE, 2020). Emerson DeltaV Smart Switch is assigned to this vulnerability.
- **Authentication bypass by capture-replay:** Capture-replay occurs when a sniffer can listen to the data transmitted over a network. The attacker then replays it to a server as valid data or with minor changes. A successful exploit can cause privilege escalation (CWE, 2020; Irmak and Erkek, 2018; Yeboah-Ofor and Boachie, 2019). Omron PLC CJ is assigned to this vulnerability.
- **Lack of certificate validation:** Schneider Electric's PowerSCADA Anywhere v1.0 is assigned to this vulnerability. A successful exploit may lead to privilege escalation (CVE, 2020; Ding et al., 2017).
- **Default password configuration:** The SCADA product allows the use of a predefined password; for example, (User:User/Admin:Admin) (CWE, 2020; Ding et al., 2017; Papp et al., 2015; Yeboah-Ofor and Boachie, 2019). If the user does not change the password, many security risks can emerge relating to integrity and confidentiality.
- **Improper restriction of invalid authentication attempts:** The SCADA system does not prevent invalid authentication attempts, making it more prone to brute force attacks (CWE, 2020; Irmak and Erkek, 2018).
- **Improper authorisation:** The SCADA system does not check properly where the critical resources are only accessible by an authorised actor. A successful exploit of this vulnerability can lead to reading and modifying data, files or directories (CWE, 2020; Ding et al., 2017; Ghosh and Sampalli, 2019; Irmak and Erkek, 2018; Papp et al., 2015; Sajid et al., 2016; Xu et al., 2017).
- **Unsalted hash value:** The SCADA system employs one-way hash protection, such as passwords. The hash value is not reversible, but the system does not salt the hash value as part of the inputs (CWE, 2020; Irmak and Erkek, 2018). This vulnerability can lead to rainbow table attacks.

## 5. IT versus ICS attacks

Cyberattacks against IT and ICS systems can be conducted locally, remotely or adjacently. Furthermore, adversary attack research and exploits kits are well known, publicly available and scalable to general IT networks. Adversaries prefer to remotely exploit IT vulnerabilities that do not require authentication and vulnerabilities that affect the confidentiality, integrity and availability of ICS components (Roumani and Nwankpa, 2020). Adversaries also generally need more time and skill to significantly impact ICS systems. The introduction of COTS to cut the costs of heavy equipment, proprietary communication protocols and operating systems increases network optimisation and cost efficiency. However, cyber threats against critical infrastructures have increased due to the integration of COTS software into the ICS environment. Exploits and attack methodologies used against IT networks can be employed against ICS systems. Consequently, they significantly ease the complexity of performing cyberattacks against the industrial control systems. As ICS databases are increasingly integrated into organizations' business or financial networks, which are connected to the public internet, critical infrastructure faces a new level of threats. While cyber incidents in traditional IT environments can lead to digital data corruption, sensitive information breaches, data destruction and business application system downtime, cyberattacks in ICS environments can cause direct or indirect physical damage to engineering assets, introduce environmental impacts and lead to human injury or death. Previously, cyberattacks against ICS systems were sophisticated and required high complexity to succeed, but recently, they have become far less sophisticated due to the involvement of IT insiders in the corporate network. Ransomware attacks can be launched in an ICS system by infecting a workstation via AUTORUN files on USB drives without the need to have any knowledge about the ICS process. By contrast, an adversary with good computing knowledge can exploit zero-day vulnerabilities using remote administration tool (RAT) malware to launch ransomware attacks against IT systems. Moreover, attack tactics against ICS are different from IT systems. It has been evident that the industrial process directly connects to IT (i.e. the enterprise network) since the introduction of the Purdue Enterprise Reference Architecture. Therefore, traditional IT cyber kill chains are being used in ICS systems. However, it is not practical for insider threats. In other words, some adversaries or attack campaigns target ICS/SCADA only.

### 5.1. Attack execution tools for ICS

A specific attack tool must be developed for attacks against a SCADA system. Havex is used to gather sensitive data about ICS systems and allows adversaries to deploy additional payloads and functionalities. By mapping Havex to Attack Matrix for Enterprise and Attack Matrix for ICS designed by MITRE (0000), it can be observed that adversaries use the software for different tactics in each environment, including IT and ICS systems.

**PLC-Blaster worm:** Several S7-PLCs have an open TCP port 102, which is a likely target for many adversaries. This malware transfers from the TCP to the PLC and attempts to infect other S7-PLCs in the same network. It employs an endless loop to trigger and violate PLC's life cycle, causing DoS. More importantly, it allows an adversary to manipulate the output of a PLC using any value of POU POKE with the process image (Spenneberg et al., 2016).

**TRITON:** It is an attack framework engineered to manipulate the industrial safety system at a critical infrastructure facility. It uses the UDP port 1502 to communicate with other devices over the TriStation protocol. It can also detect the key (TSkeystate) and program (TSprogstate) states referenced in TriStation. Moreover, it performs arbitrary code execution by injecting a payload (imain.bin) into the firmware to gain supervisor privileges. It is configured to masquerade as trilog.exe, allowing the adversary to analyse the system logic in the Tricon software. It can also perform arbitrary code execution on the firmware of the safety controller. Therefore,

it allows an adversary to modify a device's operations to an unsafe or hazardous state (Johnson et al., 2017).

**Havex:** It is a remote Trojan horse (RAT) used in cyber warfare to gather sensitive information about industrial control systems and critical infrastructure facilities. Using OPC protocol, it communicates with a command-and-control server. It can be executed in three ways: Trojanized software installer, spearphishing or watering hole attacks. Unlike Stuxnet, it does not require a zero-day exploit to spread over an ICS network. Havex can enumerate OPC tags that can reveal the nature of an industrial process (Rrushi et al., 2015).

**Industroyer:** It is a sophisticated multicomponent malware used to disrupt the industrial processes of ICS systems, including power grid sectors. Industroyer's OPC protocol can send stVal requests to read the operational variable in the control state. It abuses ICS communication protocols without any vulnerability exploits. It provides attackers with the capabilities of downloading and executing modules to expand the functionalities and exfiltrate the files of an infected machine. It can replace an image path with a registry value to ensure persistence. More importantly, it has a secondary backdoor as a backup mechanism in case the primary backdoor is detected or disabled. It also employs a report scanner that an attacker can implement instead of a publicly available tool, such as NMAP. It can cause loss of control, visibility and destruction (Lee et al., 2017).

**Stuxnet:** It is a digital weapon of mass destruction used in the context of an ICS system. It was discovered lurking in the data bank of a power plant traffic and control system. It has arrays of capabilities to turn up the pressure in nuclear reactors or switch off oil pipelines. Stuxnet utilises several complex tactics, including a sophisticated Windows rootkit, multiple zero-day vulnerabilities and network infection routines. It can modify the control system and reprogram the critical operational functions of a PLC in such a way that a legitimate command is requested. It is also configured to masquerade as s7otbxdx.dll wherein a DLL is responsible for PLC communication. It can propagate remotely by executing malicious SQL commands, including xpcmdshell (Langner, 2011).

## 6. Attack taxonomy TAACCI

An attack taxonomy can help security analysts identify unknown attacks. It can also help system designers avoid flawed design features. In this section, we classify attacks into five categories, as shown in Fig. 12 and described in Section 6. We describe various published attacks on SCADA systems identified in CVE (2020); NIST (2020). Then, we populate our taxonomy with attacks on the SCADA system, as in Table 5.

### 6.1. Classifications based on types of attacks

This subcategory involves common security attacks against SCADA. It also provides a brief definition of each type of attack and then maps these attacks to the target entity (attack target) in Section 6.2, Table 5.

- **DoS** floods a target entity with more traffic. By contrast, distributed denial of service (DDoS) is a type of DoS wherein multiple compromised computers simultaneously attack a target entity.
- **Memory corruption attacks** occur when the memory location is modified due to programming errors.
- **Privilege escalation attacks** occur when a threat actor obtains unauthorised access to a user account with administrative privileges to increase permissions.
- **Privilege elevation attacks** occur when a threat actor obtains direct unauthorised access to a SCADA system with privileges.
- **Arbitrary and remote code execution** is an attack resulting from related attacks and SCADA vulnerabilities, such as buffer overflow.
- **Reconnaissance** allows an attacker to gather information about a SCADA network's topology and data values, device functions or sensitive information stored on automation controllers.
- **Resets function code attack** occurs when unprotected SCADA communication protocols lack the proper authentication and authorisation. An adversary may change the original state of a SCADA device by resetting the function code of that device to be in an inconsistent state, causing an outage of service.
- **SQL injection attack** is a code injection attack that exposes data-driven applications.

### 6.2. Classifications of attacks based on attack targets

This section maps each type of primary attack to the SCADA component in Section 2.1. For example, DoS can be exposed to a supervisory control component, such as HMI, as well as a data acquisition component, PLC. We assigned a code as a reference number to each attack to expose specific SCADA components. The reference number was the first letter of each component. For example, we used H1 to refer to DoS exposed as HMI and P1 for a ladder logic modification that exposes the PLC. We applied this referencing method to all attacks that exposed SCADA components.

#### 6.2.1. Supervisory control HMI

The responsibility of an HMI is to monitor and control the operational processes between operators and machines by translating signals from humans to signals of the machinery. It can be a built-in screen on a machine linked to supervisory computers or tablets to provide intuition to the performance of industrial processes. HMIs are prone to the following cyberattacks:

- **DoS H1:** Recently, Eaton HMiSoft VU3 (versions 3.00.23 and prior) was assigned to buffer overflow, stack overflow and out-of-bounds read vulnerabilities (CVE, 2020; NIST, 2020; US-CERT, a). The successful exploitation of these vulnerabilities leads to DoS attacks. For example, an attacker can create crafted input files that trigger DoS when loaded by affected products. According to a US-CERT report (NIST, 2020), no countermeasure, except the replacement of the products, exists for these security issues, Table 5.
- **Memory corruption H2:** In 2020, Advantech WebAccess HMI, versions 1.2.3.9 and prior, was assigned to multiple heap overflow (CVE, 2020; NIST, 2020; US-CERT, a). Once successfully exploited, it may cause arbitrary code execution. An attacker can exploit heap overflow by overwriting a critical function in the memory and pointing it to their code. The user interaction 'open the crafted project file' leads to successful exploitation, after which the attacker can write to or read from the memory, Table 5.
- **Privilege elevation H3:**The open automation software, OPC Systems.NET HMI application, is assigned to the uncontrolled search path element vulnerability (CVE, 2020; NIST, 2020; US-CERT, a). An attacker performs DLL hijacking, described in Fig. 13, through search order hijacking. To do this, an attacker needs to rename a malicious DLL to the same name as a legitimate missing DLL, then place it in the search path before the location of a legitimate DLL. It guarantees the execution of the malicious DLL when it is called by the victim program, Table 5.
- **Arbitrary code execution H4:** In 2019, all HMI versions manufactured by SIMATIC were assigned to the XSS vulnerability. An attacker initially steals cookies to bypass authentication. Once

**Table 5**
Attacks related to SCADA systems.

| | | | Supervisory control | | | | | Data acquisition | | | | | | | | Data exchange | | | | | Data storage | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | H1 | H2 | H3 | H4 | H5 | P1 | P2 | P3 | P4 | P5 | R1 | R2 | R3 | E1 | E2 | E3 | E4 | E5 | S1 | S2 | S3 |
| **Attack Surface** | **Preconditions** | Remote access | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | | Local access | | | ✓ | | | | | | | | | | | | | | | | | | |
| | | Reconnaissance | | | | | ✓ | | | | | | | | | | | ✓ | ✓ | ✓ | | | |
| | **Cyber weapon** | Crafted inputs | | | | | | | | | | | | | | | | | | | ✓ | ✓ | ✓ |
| | | Arbitrary code | | ✓ | | | | | | | | | | | | | | | | | | | |
| | | Crafted DLL | | | ✓ | | | | | | | | | | | | | | | | | | |
| | | Malicious Script | | | | ✓ | | | | | | | | ✓ | | | | | | | | | |
| | | Capture reply | | | | | | | | | | | | | | ✓ | | ✓ | ✓ | | | | |
| | | Crafted project files | | | | | | | | ✓ | | | | | | | | | | | | | |
| | | Special element | | | | | | | | | | | ✓ | | | | | | | | | | |
| | | CSRF | | | | | | | | | | ✓ | | | | | | | | | | | |
| | | Crafted packet | ✓ | | | | | ✓ | | | ✓ | | | | | | | | | | | | |
| | | Compromised computer | | | | | | | | | | | | | | | ✓ | | | | | | |
| | **Attack delivery** | Buffer overflow | ✓ | | | | | | | | ✓ | | | | | | | | | | | | |
| | | Heap overflow | | ✓ | | | | | | | | | | | | | | | | | | | |
| | | DLL hijacking | | | ✓ | | | | | | | | | | | | | | | | | | |
| | | Arbitrary code execution | | | | ✓ | | | | ✓ | | | | | | | | | | | | | |
| | | MITM | | | | | | ✓ | | | | | | | | ✓ | | | | | | | |
| | | DoS\DDoS | | | | | | | | | ✓ | | | | | | ✓ | | | | | ✓ | |
| | | Reset function code to 1 | | | | | | | | | | | | | | | | | | ✓ | | | |
| | | Administrative privileges | | | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | ✓ | | |
| | | Cookies hijacking | | | | | | | | | | | | ✓ | | | | | | | | | |
| | | SQL injection | | | | | | | | | | | | | | | | | | | | ✓ | ✓ |
| **Causes** | | Unrestricted buffer | ✓ | | | ✓ | | | | | | | | | | | | | | | | | |
| | | Lack of input validation | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ | ✓ | | | | | | | | | |
| | | Uncontrolled Search path | | | ✓ | | | | | | | | | | | | | | | | | ✓ | |
| | | Use of hard coded credentials | | | | | ✓ | | | | | | | | | | | | | | | | |
| | | Improper access control | | | | | | ✓ | | | | | | | | ✓ | | ✓ | ✓ | ✓ | | | |
| | | Insufficient data authenticity | | | | | | | ✓ | | | ✓ | | | | | | | | | | | |
| | | Default credential configuration | | | | | | | | | | | ✓ | | ✓ | | | | | | ✓ | | |
| | | Uncontrolled resource consumption | | | | | | | | | | | | | | | ✓ | | | | | ✓ | |
| | | Lack of encryption | | | | | | | | | | | | | | ✓ | | | | | | | |
| **Consequences** | | Corruption | | ✓ | | | | | | ✓ | | | | | | | | ✓ | | ✓ | | | |
| | | Privilege elevation | | | ✓ | | | | | | | ✓ | | | | | | | | | | | |
| | | Disruption | ✓ | | | | ✓ | | | | ✓ | | ✓ | | ✓ | | ✓ | | | ✓ | ✓ | ✓ | |
| | | Disclosure | | | | | ✓ | | ✓ | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| | | Modification | | ✓ | | | | | | | | | | | ✓ | | | ✓ | ✓ | | | | ✓ |
| | | Device reconfiguration | | | | ✓ | | ✓ | | | | | | | | | | | | | | | |
| **Impact** | | Loss of availability and safety | ✓ | ✓ | ✓ | | | ✓ | | ✓ | ✓ | | ✓ | | | | ✓ | | | ✓ | | ✓ | |
| | | Confidentiality | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ |
| | | Integrity | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ | | ✓ |
| **References** | | | [85] [123] | [85] [123] | [85] [123] | [50] | [50] | [85] [123] | [21] | [20] | [132] | [52] | [127] | [53] | [53] | [128] | [51] | [28],\ [124] | [101] | [85] | [126] | [20] | [85] |

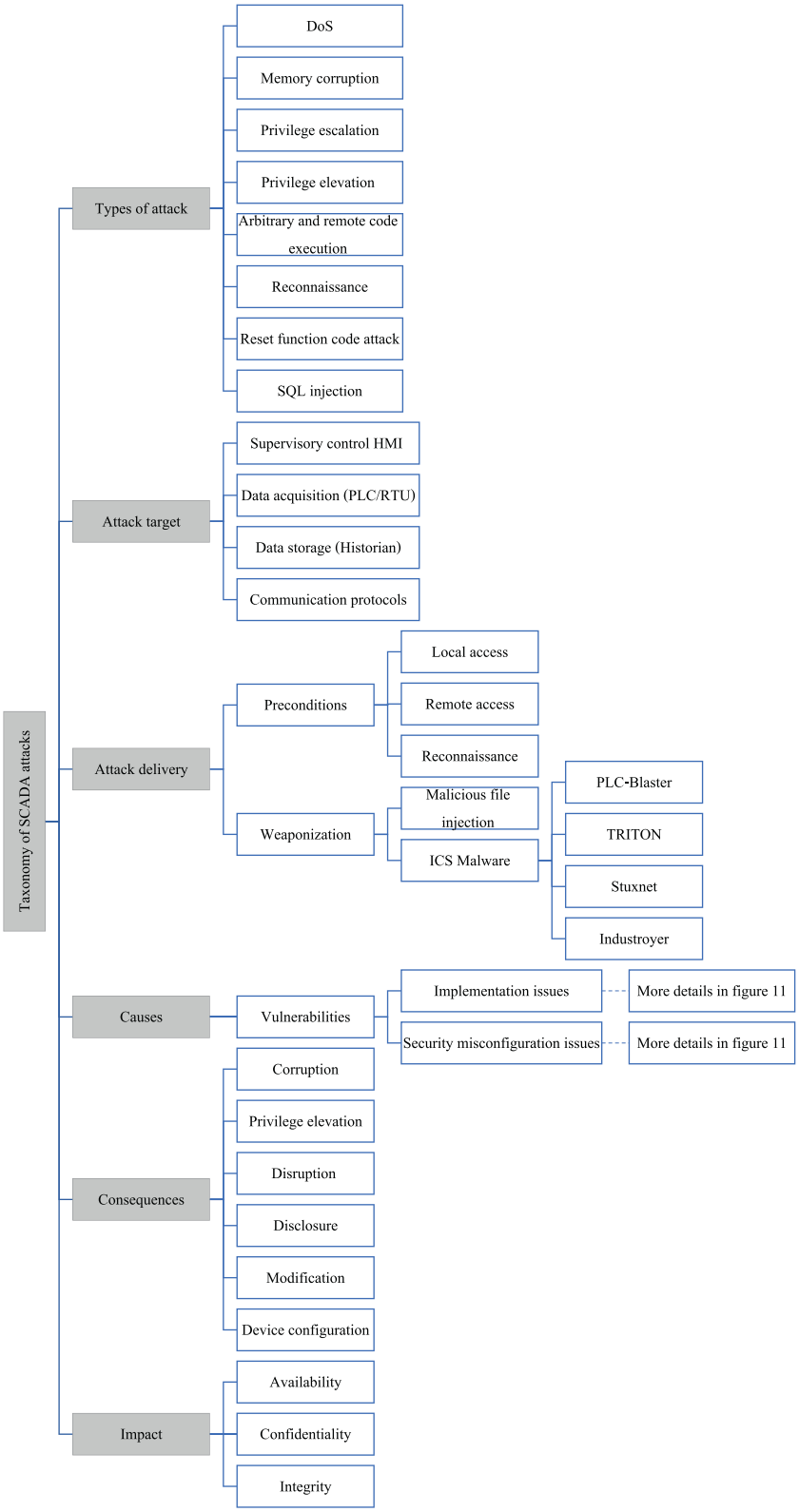


**Fig. 12.** Taxonomy of attacks on SCADA.

the attacker gains a cookie, they can execute a malicious code on the end-point system to conduct further attacks. The exploitation can be successful if the attacker modifies the device configuration via the simple network management protocol (SNMP) (ICS-US-CERT, 2018a), Table 5.

- **Read and modify data application H5:** SIMATIC HMI Comfort Panels before version 15.1 are assigned to the hard-coded credential vulnerability. If an attacker successfully exploits a hard-coded vulnerability, it can allow him to modify and read all the variables over the SNMP (ICS-US-CERT, 2018a) (see Table 5).

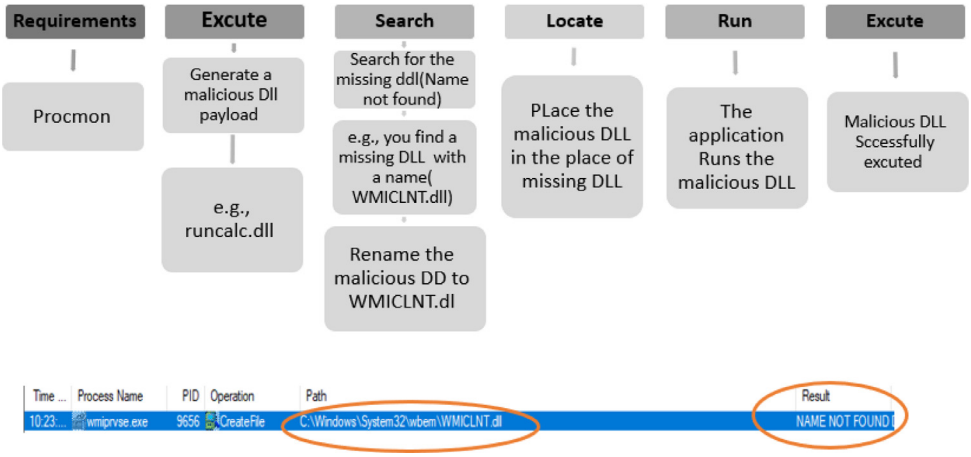


**Fig. 13.** The procedure of DLL hijacking.

*6.2.2. Data acquisition (PLC/RTU)*

Automation begins with either an RTU or a PLC. Both technologies employ a tiny brain CPU to process inputs and outputs from sensors and actuators. These devices include flow transmitters, intrusion alarms and motor starters. This section describes the potential attacks that target data acquisition devices, Table 5.

- **Ladder logic modification P1:** Rockwell Automation MicroLogix 1400 PLCs are assigned to the improper access control vulnerability (CVE, 2020; NIST, 2020), US-CERT (a). An attacker needs to send an unauthenticated packet with a read-write operation to exploit this vulnerability, and the key switches must be in the REMOTE mode. This can be achieved by performing man-in-the-middle (MITM). The index offset of the read-write operation is 0xFFFFFFFF, which is considered as NaN (not a number) in the float data type. If the float data type is set to w/r functions, the fault is triggered. Successful exploitation leads to information disclosure and logic modification, Table 5).
- **Information disclosure P2:** The SIMATIC S7-300 CPU family is vulnerable to information disclosure due to improper credential management. An attacker can access the port 120/TCP remotely, and exploit this weakness through Profibus and obtain credentials from the PLC if level-2 protection is applied to the affected products (CVE, 2020; CWE, 2020), see Table 5.
- **Memory corruption P3:** A PLC editor V1, V3 and 3U manufactured by Weacon is assigned to improper input validation. An attacker can corrupt the memory by crafting a project file to execute their malicious code under the current process (Ding et al., 2017). Similarly, the EZ PLC Editor, versions 1.8.41 and earlier, allows an attacker to execute code under the privilege of the EZ PLC Editor (CVE, 2020), see Table 5.
- **DoS P4:** The COMMGR PLC Simulator from Delta Electronics, versions 1.08 and earlier, are assigned to stack overflow, as it employs a fixed length of stack buffer. An attacker can send a crafted packet that exceeds the stack size through a particular network port, causing the stack to be overwritten and DoS (CVE, 2020; US-CERT.CISA, 2019), see Table 5.
- **Privilege escalation through cross-site request forgery (CSRF)P5:** S7 PLC devices manufactured by Siemens with firmware before version 1.5.0 are assigned to the CSRF vulnerability. A remote attacker can bypass authentication and force a victim to plant unauthorised commands on the PLC web server (ICS-US-CERT, 2019), Table 5.
- **Privilege elevation via path traversal R1:** Path traversals occur because of improper input validation. An attacker can craft a path request using a special element, such as (../../). Consequently, it can give an attacker insight into the file structure of an affected RTU and access to critical files or directories (US-CERT, 2018c), Table 5.
- **Privilege escalation through XSS injection R2:** According to ICSA (2020), data are not properly sanitised over a web socket. This allows an adversary to inject an XSS payload into a web application. As a result, the attacker can successfully steal the session cookie of a legitimate user to gain unauthorised access to their account, Table 5.
- **Packet modification R3:** Like PLCs, RTUs are also vulnerable to DoS attacks. As an RTU utilises default credential configuration to connect, unprivileged users can modify a system configuration or take complete control over an RTU. Additionally, the affected RTU creates more than one connection to I/O without closing them properly. It can lead to resource exhaustion, which, in turn, causes DoS on the targeted RTU ICSA (2020), Table 5.

*6.2.3. Data exchange components*

The number of cyberattacks against data during their transmission has increased due to various security weaknesses in the SCADA communication protocol. This section describes the threats against the communication protocol:

- **Data interception through MITM E1:** The firmware variant DNP3 TCP for the EN100 Ethernet module is assigned to the improper authentication and inadequate encryption vulnerability. It allows an MITM party to access the device port 80/TCP, capture traffic and perform administrative operations (US-CERT, 2018a), Table 5.
- **DOS/DDOS E2:** DDoS can expose DNP3 due to improper resource control. This occurs when several compromised systems attempt to attack a system to exhaust its bandwidth or resources. The Elipse SCADA 2.29 DNP3 driver builds 141 and earlier are assigned to this vulnerability in which an attacker sends a packet with formatting errors, causing resource exhaustion (DoS) (ICS-US-CERT, 2018b), Table 5.
- **Reconnaissance E3:** Reconnaissance occurs due to weak encryption within DNP3 messages. It allows an attacker to gain information about network topology and data values, device functions or sensitive information stored on automation controllers Pliatsios et al. (2020). Once the attacker successfully detects the information about the SCADA process, the reconnaissance module initialises its codes to masquerade and conduct further attacks against the target component (CWE-78, 0000; US-CERT, 2014a), Table 5.
- **Privilege escalation via capture reply E4:** An adversary can capture the packet passing between MTU and RTU/PLC devices

and tamper with its contents (minor changes). This occurs due to improper authentication. All versions of the PLC CJ/CS manufactured by Omron are assigned to the capture reply vulnerability. Successful exploits can cause a delay or repeat in the operational processes of industrial valves (Rodofile et al., 2016), Table 5.

- **Reset function via reply attack, spoofing and sniffing E5:** Unprotected SCADA communication protocols may be subjected to reset function attacks. An adversary can send a reset function to a device that initiates a restart, causing an outage of services for a while. Telecare F25 Series Radio Controls have been subjected to reset function attacks (NIST, 2020; Pliatsios et al., 2020), Table 5.

#### 6.2.4. Data storage (historian)

Historian is a database that stores the real-time data of a SCADA network. Several security concerns target Historian:

- **Information disclosure S1:** Wonderware Historian lacks proper credential management. It applies default passwords as a security practice for authentication, providing unauthorised access to the Historian database (US-CERT, 2017), Table 5.
- **DOS attacks S2:** Schneider Electric Wonderware Historian is assigned to the improper XML parsing vulnerability. Successful exploits can cause DoS (CVE, 2020), Table 5.
- **Memory corruption through SQL injection attacks S3:** Insufficient input neutralisation causes SQL injection. SCADA Historian comprises a web interface either for users or administrators. A malicious entity can leverage those websites to execute a malicious query string that can damage the history of the entire SCADA system. The AVEVA eDNA Enterprise Data Historian is assigned to this vulnerability (NIST, 2020). To create a crafted SOAP web request, an attacker needs to inject a query string in the CHaD.asmx web service (CVE, 2020; NIST, 2020). A successful exploit can cause data modification and information disclosure, Table 5.

### 6.3. Classifications based on attack surfaces

This section describes the steps of a successful attack. They are classified into three subcategories. The first is the precondition, which is the attack vector required to exploit SCADA vulnerabilities, Table 5.

#### 6.3.1. Precondition

- **User interaction:** A malicious action can be executed if its preconditions are true. In some cases, SCADA may have an attack surface, but user interaction is required to successfully exploit the SCADA vulnerabilities. Additionally, attackers need a path to access their target components (CVE, 2020; NIST, 2020; US-CERT, a).
- **Local access:** It is a path wherein an attacker can exploit the vulnerability. For example, privilege escalation and USB attacks require local access to be exploited successfully (Daryabar et al., 2012; US-CERT, a). In Table 5, OPC systems.net HMI application is vulnerable to DLL hijacking. An adversary must have local access in order to rename a malicious DLL. Similarly, LAquis SCADA 4.3.1.71 is assigned to the improper input validation vulnerability. An adversary can craft a project file and consequently access the file systems or directories.
- **Remote access:** It is a path wherein an attacker does not need network access to the vulnerable component because it is located in the network stack (Daryabar et al., 2012; NIST, 2020). With SIMATIC HMI (H4), as shown in Table 5, an attacker only needs to steal cookies to bypass authentication and execute a malicious code in the end-point system. Similarly, Ecava IntegraXor, used to run web-based HMI, is assigned to the CRLF injection vulnerability. An adversary can inject the arbitrary code into a SCADA HMI web application by conducting HTTP response splitting attacks. More examples of remote attacks are in Table 5.
- **Reconnaissance:** This is important for an attacker before conducting any malicious activities. The attacker may need to gather information about the SCADA network ports, the number of machines and so on. It can be achieved by using sniffing tools. One of the most popular sniffing tools is Wireshark. It captures packets from a local network and stores them for offline analysis (Hall and Ngalamou, 2019; Hilal and Nangim, 2017; Lee et al., 2014; Sayegh et al., 2013). Another way to gather information is network scanning using various tools, such as NMAP (Jicha et al., 2016; Rosa et al., 2019a). It collects information about a SCADA network, such as the number of available hosts and the services provided by the hosts.

#### 6.3.2. Weaponisation

The second subcategory of attack surface comprises attack weapons, Table 5. It describes the tools used by attackers to conduct their malicious activities:

- **Malware and virus:** An attacker copies malware to an infectable USB drive and then inserts it locally into the target component (Miller and Rowe, 2012). The attacker can use the link life to propagate copies of the worm and a rootkit to hide all the malicious processes. A remote access trojan is malware that comes with backdoors to control the target SCADA component with administrative privileges (Guillén et al., 2019). Stuxnet is an example of malware that exposes nuclear systems by overwriting some setpoints in a SCADA system. An attacker can also use a virus, such as a Trojan horse, to infect targeted machines and conduct malicious activities Miller and Rowe (2012). For example, worms such as Slammer, Stuxnet and Blaster can propagate to a new targeted SCADA component by the use of SQL UDP port 1443, TCP port 135 and TCP 145, respectively. Moreover, Duqu has been used for spying on industrial project documents. Shamoon was intended to sabotage the oil industry in Saudi Arabia. Finally, BlackEnergy is a Trojan that is used to conduct DDoS attacks against SCADA systems. According to Kaspersky (2017b), this Trojan is active in energy/SCADA companies worldwide (Kaspersky, 2017a).
- **Malicious file injection:** It allows an attacker to locally or remotely inject malicious commands into the target component of a SCADA system. It can be achieved by crafted inputs, arbitrary code execution, crafted DLL, crafted project files, malicious scripts or special elements. In Table 5, MicroLogix 1400 PLCs (P1) are assigned to insufficient input validation in which an adversary sends a crafted packet with reading and writing operations to exploit the vulnerability (NIST, 2020). Similarly, the mymbCONNECT24 SCADA server is assigned to cross-site request forgery (CSRF) wherein an attack can cause cross-site attacks to execute arbitrary codes in end-point systems. Additionally, an attacker can corrupt the memory of a vulnerable PLC by crafting a project file under the current process. Moreover, an adversary can obtain privileged access to a user's account in a SCADA web application by injecting a malicious script into the web page of the SCADA application. Furthermore, an attacker can access the critical files or directories of a vulnerable RTU by crafting the path request with a special element (../.../). Finally, HMiSoft VU3 is assigned to the buffer overflow vulnerability in which an attacker can create crafted input files that trigger DoS when loaded by the affected SCADA HMI.

### 6.4. Classifications based on causes of attacks

Cyberattacks do not happen without a cause. They normally result from security weaknesses, such as vulnerabilities or usability issues. Other causes can be the impacts by which an attacker can perform further attacks based on the current one. For example, an attacker exploits a buffer overflow vulnerability (the current attack) and then performs DoS based on the impact of the buffer overflow (Ghosh and Sampalli, 2019; Xu et al., 2017). The improper limitation of a memory buffer and improper input validation can cause a buffer overflow in the vulnerable SCADA component. Additionally, an uncontrolled search path can cause arbitrary code execution. Insufficient data authenticity, default credentials, the use of hard-coded credentials, improper access control and a lack of encryption can cause information disclosure. In Table 5, reconnaissance occurs due to weak encryption within DNP3 messages. DNP3 is vulnerable to DDoS due to improper resource control. Similarly, DNP3 is vulnerable to MITM attacks due to weak encryption. Therefore, a successful exploit leads to various consequences, as stated in the following section.

### 6.5. Classifications based on consequences of attacks

The potential consequences mapped to our taxonomy are memory corruption against the SCADA software, privilege elevation wherein a lower privilege user gains administrative privileges, a disruption of the SCADA service, sensitive information disclosure to unauthorised actors, PLC ladder logic modification and, finally, device reconfiguration. By consequences, we mean attacks resulting from one or more attacks. For example, for DDoS to be successful, an attacker needs a series type of attack. First, the attacker needs to compromise the computers in the host through virus infection. Second, the attacker performs DDoS on a single victim machine. One consequence of performing DDoS or DoS against a SCADA system is memory corruption. For example, the COMMGR PLC has a fixed length of the memory buffer. An attacker can send a crafted packet that exceeds the buffer size. Consequently, the attacker can perform DoS by causing the stack of the buffer to be overwritten CWE (2020). Additionally, improper access control and weak encryption can lead to information disclosure and data modification. For example, the Wonderware Historian database lacks proper encryption and credential management. It utilises default password configuration. Hence, a successful exploit leads to information disclosure to an unauthorised actor. Improper access control vulnerabilities are assigned to MicroLogix 1400 PLCs, and an attacker can send an unauthenticated packet to exploit this vulnerability. Consequently, a successful exploit can lead to both information disclosure and ladder logic modification (CVE, 2020; CWE, 2020). Finally, an adversary with high privileges can capture the packet transmitting between MTU and RTU/PLC devices and tamper with its contents, causing information exposure and data modification.

### 6.6. Classifications based on impacts of attacks

Each cyberattack violates security policies and leads to the following:

- **Impact on availability:** An attacker intercepts the available service of a SCADA system. For example, multiple heap overflow is assigned to Advantech WebAc-cess HMI wherein an attacker can cause arbitrary code execution by overwriting a critical function in the memory and pointing to their code. HMiSoft VU3 was also assigned to the buffer overflow vulnerability. A successful exploit can cause DoS, which has a high impact on the availability of SCADA services.
- **Impact on confidentiality:** An attacker has unauthorised access to sensitive data. For example, SIMATIC HMI was assigned to the hard-coded credential vulnerability in which an attacker can read and modify all the variables over the SNMP. Additionally, the firmware variant DNP3 TCP was assigned to the improper authentication and lack of encryption vulnerability. A successful exploit has a high impact on confidentiality.
- **Impact on integrity:** An attacker alters and modifies data. Improper access control vulnerability was assigned to RTUs and PLCs. A successful exploit has a high impact on integrity. We highlighted more examples of attacks against SCADA that have high impacts on availability, confidentiality and integrity in Table 5.

## 7. Comparison of SCADA IDS

### 7.1. Host-based intrusion detection

Host-based intrusion detection (HIDS) is an IDS running on a host for monitoring and analysing malicious logging activities. HIDS has a wide scope of use. It can perform file system integrity management and monitor the log files generated by an endpoint, creating a historical record of activities and allowing the system manager to search for an anomaly that may have occurred. It can decrypt events in packets to detect attacks against a host in the network. Hence, it is effective for detecting zero-day and hidden attacks. However, HIDS may reduce the performance level of a SCADA host because IDS consumes computing resources.

- **File system integrity management:** HMI is built with a standard operating system. It is not possible to apply patches and updates because of the operational nature of SCADA. Instead, file integrity management tools can help detect any modifications to the logs and registries of the operating system. Robert E (2010) claimed that it was possible to immediately detect Stuxnet without waiting for antivirus vendors to identify it if file integrity management was implemented at that time. Ding et al. (2017) also claimed that file system integrity management can verify harmful blacklist files and whitelist files. Tripwire (0000), OSSEC (0000) and Radoglou-Grammatikis et al. (2019) use the whitelists of file systems. They scan file systems to check file integrity and perform registry monitoring and log file analysis.
- **Memory dump analysis via function codes:** Volatile and non-volatile memory analysis can help identify hidden malware and other intrusions (Ding et al., 2017). On the one hand, many studies have shown that it is challenging to conduct memory analysis in real time due to the operational nature of a SCADA system (Larkin et al., 2014). On the other hand, memory extraction or dumping can solve the challenge of real-time memory analysis. In the context of the SCADA system, memory extraction can be done either via function codes or the Joint Test Action Group (JTAG). For example, Awad et al., 2019 attempted to obtain the entire memory of the Modicon PLC through the function codes of communication protocols such as read/write. The authors attempted to acquire the memory via JTAG, but the checksum verification was not successful because the JTAG pins were disabled by the manufacturer. Furthermore, Awad et al., 2019 conducted a dump analysis of two PLCs from different vendors. PLC A consisted of a removable compact flash card that stored the entire file system. By contrast, PLC B only consisted of firmware that could dump DRAM to a flashcard if the power was unexpectedly lost by using a backup battery. The conclusion of the experiment indicated that it was possible to acquire DRAM from PLC B, whereas it was not possible to acquire data from PLC A. Similarly, it was possible to acquire NVRAM from PLC A, yet PLC B did not support it. The experiment also indicated that it is not difficult to extract DRAM con-

tent and acquire data from PLCs while they are in operating mode due to the differences in design and architecture. A gap exists, as there are limited studies on code injection identification, and none of the previous works tackled code injection or undetected attacks on volatile memory. Marco et al. (2020) also examined three different PLCs to investigate the differences in acquirable data between models from the same vendor. The authors followed the same approach as Larkin et al. (2014), Awad et al., 2019, Ahmed et al. (2017), in which third-party communication libraries were used as data acquisition methods. However, Marco et al. (2020) aimed to identify particular components of data and their related artefacts. The authors highlighted the artefacts according to high-level types, including variable content, application codes, metadata and device logs. Therefore, it is believed that the PLCs are rich in data, which may have evidential values. However, the authors stated that not all the data artefacts were acquirable. Additionally, the format of the data differed when similar artefacts were acquired. Although the PLC vendor provided the ability to acquire the application codes from these PLCs, the vendor software had a certain number of vulnerabilities. From a forensic standpoint, we cannot rely on software vendors to acquire data. Again, the authors did not consider the decentralised nature of the current SCADA system. This approach may be applicable in some cases in which an attack occurs against a local PLC.

- **Memory dump via the JTAG:** The JTAG allows manufacturers to test and debug the physical connections between the pins on a chip in a printed circuit board (PCB) at an integrated circuit level. The JTAG has been adapted as a data acquisition method for forensic analysis, as it provides the ability to extract the whole physical image of a device's memory if normal forensic tools fail to acquire the device. The first memory imaging via the JTAG was conducted by Awad et al., 2019. Breeuwsma (2006) attempted to evaluate memory extraction via the JTAG on the M221 Modicon PLC manufactured by Schneider Electric. The authors disassembled the device to find the TAP port on the PCB and connected it to the JTAGulator device to find the JTAG pins and access the raw data stored in that device. The authors stated that they found the JTAG pins, but they could not extract the memory, as the JTAG pins were disabled by the manufacturer. This approach can be efficient, as the PLC consists of RAM and EEPROM, which store the firmware so that the RAM has substantial information about the PLC. Thus, by using the JTAG, we can dump the RAM and acquire a full image of this rich information. Moreover, the JTAG is a viable acquisition method for compatible devices. However, this approach has several limitations. First of all, discovering the JTAG on an unknown device that does not have a TAP port can be challenging (Awad et al., 2019). Additionally, it can be difficult to obtain the JTAGs when the TAP ports are disabled by the device manufacturer, even with the presence of a TAP port. More importantly, it is challenging to remotely acquire firmware images over a network.

### 7.2. Network-based intrusion detection

Traffic inspection and protocol analysis are examples of network intrusion detection techniques. Traditional IDS are examined by security analysts who evaluate alerts and track decisions Yasakethu and Jiang (2013), Rosa et al. (2019b), Snort (0000), Yang et al. (2013), Suricata and Zeek - also known previously as Bro - can be used to detect intrusions in a SCADA network (Radoglou-Grammatikis et al., 2019; Rakas et al., 2020; Wong et al., 2017). These tools can perform traffic inspection and protocol analysis but with some limitations. Wireshark provides support for many common SCADA protocols, including GOOSE, DNP3, IEC 60870 and Fieldbus/Modbus Rakas et al. (2020). For industrial analysis purposes, NetDecoder Wong et al. (2017) is a popular traffic analysis tool, as its primary objective is to troubleshoot communication issues in ICS (Senthivel et al., 2017). NetCoder supports some ethernet protocols, such as EtherNet/IP, DNP3, CSP/PCCC, IEC 60870, Modbus/TCP, PROFINET and CC-Link IE. In network-based intrusion detection (NIDS), data can be acquired from function codes in ICS communication protocols (Liptak, 2018; Tripwire, 0000). NIDS is more frequently employed in SCADA systems because of the limitations of SCADA components. As Tripwire performs constant monitoring, significant resources are required from the host (Tripwire, 0000). Therefore, Tripwire is not applicable to SCADA field devices. Table 6 shows the difference between HIDS and NIDS. Moreover, NIDS is limited to encrypted traffic.

### 7.3. Classifications of feature extraction

Feature extraction must be achieved for learning models. Most studies in this field can be classified into three categories:

#### 7.3.1. Model-based algorithms

Artificial intelligence techniques help a system learn from the current environment by evaluating what it learns in real time (Cheung et al., 2007; Hahn and Govindarasu, 2012; Kalech, 2019a). It is also configured as an example of a signature-based system.

#### 7.3.2. Expert system-based algorithms

This intelligent system emulates decision-making properties. It makes decisions based on experience, heuristics and non-formalised knowledge (Hahn and Govindarasu, 2012; Mayadev et al., 2014).

#### 7.3.3. Machine learning-based algorithms

Anomalies can be learnt from historical data and new data classified based on the learnt model (Cui et al., 2019; Kalech, 2019a). Hence, machine learning can be considered a type of anomaly detection system (Mayadev et al., 2014; RK, 2013; SUABOOT et al., 2020; Yasakethu and Jiang, 2013). Kalech (2019b) stated that features can be extracted with the use of pattern recognition based on the function code, time factor or both. A support vector machine (SVM) is a supervised machine learning algorithm that can distinguish between two objects. The primary objective of applying SVM is to identify a hyperplane that can maximise margins and reduce the risk of overfitting. A soft-margin SVM is used to maximise the margin and reduce the risk of slack (Hasan et al., 2014; Vapnik, 1999). Using the method of Lagrange multipliers, we can obtain a dual formulation expressed in terms of variables (Schölkopf et al., 2002). It is easy to implement and proper for large datasets. The key nearest neighbour (KNN) is a practical machine learning algorithm that classifies problems. The decision depends on the number of neighbours, where $K = 5$ is the nearest neighbour. The prediction is based on the distance calculation between the test data and the inputs using the Euclidean distance formula (Mafra et al., 2010). The advantage of KNN is that it does not require the prior assumption of data. However, the main drawback of KNN is the complexity of searching for the nearest neighbour for each sample. Therefore, it is not suitable for large datasets. Linear regression (LR) is a practical machine learning algorithm employed to predict values based on the value of another variable. Dependent variables refer to the measurement of the influence of one variable on another variable. An independent variable is called a predictor and refers to predicting one variable based on another variable. A regression model can be determined based on learning the values of B0 and B1 in the following equation. $\epsilon$ represents the errors in the model.

**Table 6**
Host-based IDS versus network-ba.

| Feature | Description | Network-based IDS | Host-based IDS |
|---|---|---|---|
| **Management** | Is it possible for IDS to manage many hosts in large environment? | ● | ○ |
| **Encrypted traffic analysis** | Is it possible for SCADA IDS to inspect encrypted traffic? | ◐ | ● |
| **Successful attack detection** | Is it possible for SCADA IDS to detect successful attacks? | ○ | ● |
| **Remote attack detection** | Is it possible for SCADA IDS to detect remote attacks? | ● | ○ |
| **Impact on host performance** | Does SCADA IDS affect the host ™s performance? | ○ | ● |
| | **Example** | Snort Suricata Zeek | OSSEC\Tripwire |
| | ○Yes ●No ◐Partly | | |

### 7.4. Classifications of detection analysis methodologies via ML algorithms

There are two broad analysis categories of IDS that can be tailored to SCADA and ICS systems. The first category is signature based (Cheung et al., 2007; Hahn and Govindarasu, 2012; Kalech, 2019a), Han et al. (2014); SUABOOT et al. (2020), and the second is anomaly-based detection (Barbosa, 2014; Hahn and Govindarasu, 2012; Kalech, 2019a; Ren et al., 2018; Tripwire, 0000).

#### 7.4.1. Signature-based IDS

It is typically used to identify known threats. It operates using a programmed indicator of compromise and a list of known threats (e.g. misuse patterns). It can achieve high accuracy, but this technique cannot detect future attacks due to the absent attack signatures of various known threats.

#### 7.4.2. Anomaly-based IDS

The system compares network traffic regularly. It then raises alerts if extremely abnormal behaviour appears. The distinctive normal model is trained and learns through statistical and mathematical techniques. Nevertheless, the challenge of modelling correct general behaviour is an increased false alarm rate (Dussel et al., 2010). This approach is efficient in detecting zero-day attacks. Unsupervised learning algorithms can be used to detect anomalous patterns (Ahmed et al., 0000; Ghodratnama et al., 2020):

Nearest-neighbour based approach: A local outlier factor (LOF) detects an anomaly based on the local density of data points (Ahmed et al., 0000; Breunig et al., 2000). The connectivity-based outlier factor (COF) Tang et al. (2007) is an evolution of the LOF approach Ahmed et al. (0000). It detects anomalies based on the chaining distance (low density) of a pattern. Papadimitriou et al. (2007) calculated the outlier score from the local correlation integral. The LoOP score represents the outlier score based on the probability of the set distance (Ahmed et al., 0000).

### 7.5. Existing attack datasets

Five datasets related to power, gas and water systems have been released (Morris et al., 2011b). They included network data, log data and field device data. The Morris-1 dataset included 37 event scenarios in power systems, switches, routers and normal and abnormal events in field devices. However, no communication protocol was provided in Morris-1. Therefore, the Morris-2, Morris-3 and Morris-4 datasets included the Modbus communication protocols between the HMI and the control devices in the gas pipeline testbed (Morris and Gao, 2014). Morris-5 involved the data collected from the realistic energy management system over a month, which was considered the longest time compared with other datasets. UNSW-NB15 was released in 2015 by the University of New South Wales in Canberra, Australia, for network intrusion detection (Moustafa and Slay, 2015). The dataset consisted of hybrid attack activities related to network traffic, including reconnaissance, worm, exploit, fuzzers and DoS. However, many duplicates were found in the training set. For intrusion detection purposes, a network traffic dataset was released by Lemay and Fernandez (2016). It consisted of Modbus TCP communication protocols and channel commands in the SCADA system. Security analysts use the KDD99 dataset for testing NIDS (Revathi and Malathi, 2013). It is used for competition with data mining tools. NSL-KDD is an improved version of KDD99 that removes duplicate records (Revathi and Malathi, 2013). The water supply system dataset consisted of log files of multiple features, such as temperature, water, inflow and outflow (Arnold, 1994).

**Table 7**
Attack distribution in the UNSW-NB15.

| Attack category | Number of samples | Distribution of Attacks in % |
|---|---|---|
| Normal | 37,000 | 44.939999 |
| Generic | 18,871 | 22.920614 |
| Exploits | 11,132 | 13.520867 |
| Fuzzers | 6062 | 7.362872 |
| DoS | 4089 | 4.966477 |
| Reconnaissance | 3496 | 4.246223 |
| Analysis | 677 | 0.822281 |
| Backdoor | 583 | 0.708109 |
| Shellcode | 378 | 0.459117 |
| Worms | 44 | 0.053442 |

#### 7.5.1. Experiment setup and result

Our experiment used training and testing subsets of UNSW-NB15 to train and test the three machine learning classifiers, including LR, random forest, SVM and KNN. All experiments were performed using Python 3 with the Keras 2.4.3 framework installed for deep learning utilities on an 8-GB MacBook Pro-(13-inch, M1, 2020). Table 7 represents the attack distribution in the UNSW-NB15 dataset.

A confusion matrix was utilised to evaluate the performance of KNN and SVM using the accuracy measure and false alarm rate. The confusion matrix consisted of the following:

- True Positive (TP) was the number of times the model predicted positive values as positives.
- True Negative (TN) was the number of times the model predicted negative values as negatives.
- False Positive (FP) was the number of times the model predicted negative values as positives.
- False Negative (FN) was the number of times the model predicted positive values as negatives.

Accuracy was the overall prediction that was correct.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Recall, also called sensitivity, indicated the probability of detecting true positives.

$$Recall = TP/(TP + FN)$$

**Table 8**
Performance comparison of four ML algorithms on UNSW-NB15.

| Dataset | Model | % Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|---|
| UNSW-NB15 | LR | 00.74 | 00.73 | 00.73 | 00.72 |
| | KNN | 00.84 | 00.84 | 00.84 | 00.84 |
| | Random Forest | 00.86 | 00.86 | 00.86 | 00.83 |
| | SVM | 00.85 | 00.85 | 00.81 | 00.83 |

Precision was the positive predictive values, which indicated the overall positive values and how often they were correct.

$$Precision = TP/(TP + FP)$$

F1score was the harmonic mean of recall and precision.

$$F1 = 2\times(Precision \times Recall)/(Precision + Recall)$$

The UNSW-NB15 dataset achieved the highest validation accuracy. Employing the random forest classifier in the UNSW-NB15 dataset demonstrated reasonable performance and was comparable to the results obtained in the cases of the other classifiers, Table 8.

## 8. Critical evaluation of SCADA testbeds

A SCADA testbed is an isolated environment typically used to validate security techniques in a safe manner. It can be classified into three categories: physical testbed, virtual testbed and hybrid.

### 8.1. Physical testbed

It consists of real hardware and software for conducting a line of security validation experiments. Several physically simulated testbeds have been proposed to validate a new security solution (Geng et al., 2019). The National SCADA Testbed (NSTB) is a multi-location security-testing environment with scalable hardware and software. It consists of seven substations and around 3000 monitoring sites (Kuipers, 2008). However, the NSTB is not free for academics and researchers. Due to limited costs or funding, it is challenging for scientific researchers to develop a testbed with high fidelity and scalability. SWaT, proposed by Mathur and Tippenhauer (2016), is a small-scale testbed used for security and research training on industrial control systems. However, the authors pointed out that SWaT is limited when detecting some attacks in experiments. Mississippi State University built an ICS testbed, which was a small-scale vulnerability discovery (Geng et al., 2019; Goetz and Shenoi, 2007; Morris et al., 2011a). Ahmed et al. (2016) proposed a small-scale SCADA testbed for security and forensic analysis at the University of New Orleans. It consists of three models of industrial processes, including a gas pipeline, wastewater system and power distribution system. The testbed utilises real-world SCADA devices, such as PLCs, transformers and HMIs. It also supports multiple communication protocols, such as Modbus TCP/IP, EtherNet/IP and Profinet. Although the testbed is a close representation of a real SCADA system, it is very expensive. It also does not support fieldbus I/O to connect a PLC with sensors and actuators. Fieldbus is critical, as an adversary can search for the lowest level of a SCADA system to conduct DoS and MITM attacks. Idaho State University built a small-scale testbed, PowerCyber, for testing power grid communications Ashok et al. (2020). It can be used for vulnerability analysis or to simulate metasploit attacks, wireless attacks and sniffing attacks. It supports multiple communication protocols, including IEC 61850, DNP3, Modbus, C37.118 and OPC UA.

### 8.2. Virtual testbed

Virtualisation technology is efficient for developing an affordable security testbed with high fidelity and flexibility. Queiroz et al. (2011) proposed SCADASim as a security testing framework, which consists of a real hardware and software environment to simulate physical processes and control networks. SCADASim is built on top of OMNET++. It supports the integration of real-world devices with industrial communication protocols, such as Modbus TCP/IP and DNP3. It can be easily scaled. It simulates spoofing, eavesdropping, MITM and DoS attacks. A SCADA-specific security testbed was proposed by Yang et al. (2014) to evaluate security vulnerabilities in SCADA systems. The testbed consists of the main SCADA components, including HMI, Historian, field devices and IEC 60870-5-103 communication protocols. A malicious host is implemented to simulate cyberattacks, such as MITM, DoS and ARP spoofing. However, it does not support fidelity. GridLAB-D is an open-source power distribution system simulation and analysis tool. This tool employs advanced modelling techniques that provide useful information to designers and operators of distribution systems (Chassin et al., 2008). It simulates the power flow from a generation station to the end user. It can run on multiprocessor machines, but it cannot support fidelity. ASTORIA is an attack simulation toolset for smart grid infrastructures (Wermann et al., 2016). It permits the simulation of attacks, such as malicious command injection and DoS, and the evaluation of attack impacts on power grid systems. It has a set of built-in attack profiles to help security researchers enhance the security level of smart grid systems. However, the toolset of attack profiles should be extended in future research, and additional simulation scenarios should be considered. TASSCS is a virtual testbed used to test the security of ICS systems. It utilises the Opnet tool to reproduce control network activities (Yadav and Paul, 2021). It also uses the PowerWorldi simulation system to simulate operations behaviour (Mallouhi et al., 2011). The TASSCS approach is a combination of simulators, hardware-in-the-loop and emulators. It simulates ARP spoofing, MITM, unauthorised access and DoS attacks. This approach only supports communication over the Modbus and DNP3 protocols, and the software components are not open source (Koutsandria et al., 2015). Pacific Northwest National Laboratory proposed a testbed called PRIME for smart grid systems (Becejac et al., 2020). It can be used to perform several cyberattack scenarios and impact analyses of power grid systems. The testbed is implemented to model a typical SCADA system. It supports high fidelity and multiple communication protocols. As the testbed implements communication over real SCADA networks, it allows live cyber defence exercises. However, it does not support flexibility. RINSE is an example of a hybrid testbed (Ding et al., 2017; Morris et al., 2011a). It is a large-scale testbed used to assess and evaluate smart grid network attacks.

### 8.3. Hybrid testbed

In this approach, the SCADA system can simulate, virtualise and emulate physical devices. The primary focus of a hybrid testbed is to provide a testbed for cybersecurity purposes. SCADA-SST is a virtual testbed propoy Ghaleb et al. (2016b) at King Fahd University of Petroleum and Minerals. It consists of three nodes: virtual node, external node and interface node. The virtual node involves PLC, RTU and HMI, which are simulated with OMNet++. The external node includes physical SCADA devices located outside the simulated environment. The interface node allows communication between the virtual node and the external node. Although the SCADA-SST testbed supports high fidelity and reliable experiments, low flexibility and high costs limit the scale of the testbed. The HITL testbed was proposed by Keliris et al. (2016) to simulate Tennessee Eastman chemical processes. It consists of physical PLC and OPC servers over TCP/IP protocols. It can simulate ARP spoofing and false data injection attacks. Queiroz et al. (2009) proposed a SCADA testbed based on the combination of network simula-

**Table 9**
Comparison between SCADA testbeds (= Yes, X= No, NA= not specified).

| Type of testbed | Examples | Characteristics | | | | | Application area | | | Testing capability | | | | | | | | | | Overall shortcoming | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Reproducibility | Domain fidelity | Scalability | Support multiple ICS protocols | Cost-effectiveness | Vulnerability analysis | Security validation | Impact analysis | Reconnaissance | (Cloud-based) brute force attack | Command injection | MITM | ARP Spoofing | DoS | Wireless Attacks | Sniffing Attacks | Vulnerability discovery | Scalability issues | Maintenance | High cost | Reproducibility issues |
| Physical | National-SCADA testbed [74] | X | High | Large | ✓ | X | ✓ | X | X | NA | NA | NA | NA | NA | NA | NA | NA | X | X | NA | ✓ | ✓ |
| | Swat [86] | ✓ | High | Small | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | X | X | X | X | X | X | X | ✓ | NA | X | X |
| | Ahmed et al. [3] | ✓ | High | Small | ✓ | X | ✓ | ✓ | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | X | ✓ | X | ✓ | X |
| | PowerCyber [9] | ✓ | NA | Small | ✓ | ✓ | X | ✓ | ✓ | X | X | X | X | X | X | ✓ | ✓ | ✓ | ✓ | NA | NA | ✓ |
| | Mississippi State University testbed [48] | X | High | Small | X | ✓ | ✓ | ✓ | X | X | X | ✓ | X | X | ✓ | X | X | ✓ | ✓ | ✓ | X | ✓ |
| Virtual | SCADASim [13] | ✓ | NA | Small | ✓ | ✓ | ✓ | X | X | X | X | X | ✓ | ✓ | ✓ | X | ✓ | X | NA | NA | X | X |
| | Yang et al. [167] | ✓ | Moderate | NA | ✓ | ✓ | ✓ | ✓ | ✓ | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | X | NA | X |
| | GridLAB-D [18] | ✓ | Moderate | Small | ✓ | ✓ | X | ✓ | X | NA | NA | NA | NA | NA | NA | NA | NA | NA | ✓ | NA | X | X |
| | Astoria [161] | X | Moderate | Small | ✓ | ✓ | ✓ | ✓ | X | X | X | ✓ | X | X | ✓ | X | X | X | ✓ | X | X | ✓ |
| | PRIME [13] | ✓ | Low | small | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | X | X |
| | TASSCS [165] | X | Moderate | Large | X | X | X | ✓ | X | X | X | X | ✓ | ✓ | ✓ | X | X | X | X | NA | NA | NA |
| Hybrid | SCADA-SST [38], [100] | ✓ | High | Small | ✓ | X | X | X | ✓ | X | X | X | X | X | ✓ | X | X | X | ✓ | X | ✓ | X |
| | HITL Tennessee Eastman [72] | NA | High | Small | ✓ | X | ✓ | X | ✓ | X | X | ✓ | X | ✓ | X | X | X | X | X | X | ✓ | NA |
| | Queiroz et al [112] | ✓ | High | Small | X | X | ✓ | ✓ | ✓ | X | X | X | X | X | ✓ | X | X | X | ✓ | X | ✓ | X |
| | VPST [14] | ✓ | High | Small | ✓ | ✓ | X | X | ✓ | X | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ | X | ✓ | X |

tion and real device connectivity. It allows sensors and actuators to be attached to the simulator to investigate the impacts of DDoS attacks against field devices in a simulated network. The testbed consists of OMNET++, Lego Mindstorms NXT, a discrete event simulation tool, Modbus TCP/IP protocol and a programmable microcontroller. Despite enabling a SCADA network in this co-simulated environment, it requires hardware-in-the-loop for high fidelity. Bergman et al. (2009) proposed a virtual power system testbed (VPST), which is a combination of virtual and physical SCADA components. VPST supports multiple ICS protocols and enables several cybersecurity attacks to study the impacts of such attacks against a simulated environment. Table 9 summarises the key differences between the three types of SCADA testbeds.

### 8.4. The scope of SCADA testbeds

SCADA testbeds need to have capabilities that offer realistic SCADA environments. The scope of SCADA testbeds describes the research applications of SCADA testbeds. For example, a researcher may use a SCADA testbed for vulnerability analysis or attack impact analysis. According to Poudel et al. (2017a), SCADA testbeds can follow several scopes and directions:

- Vulnerability analysis: It is challenging to conduct a vulnerability assessment on a real SCADA system. Therefore, security researchers need SCADA testbeds to find weaknesses in industry-standard software configurations and communication protocols. For example, SCADASim, Mississippi State University's testbed and the NSTB can be used for vulnerability analysis Ghodratnama et al. (2020), Queiroz et al. (2011), Morris et al. (2011b), Poudel et al. (2017b).
- Security control validation: SCADA security examiners can use SCADA testbeds to evaluate the adaptability of different control mechanisms that propose to eliminate or detect cyberattacks against SCADA systems. For example, SWaT, Mississippi State University's testbed, PowerCyber and TASSCS can be used for security validation (Ashok et al., 2020; Ghodratnama et al., 2020; Mallouhi et al., 2011; Morris et al., 2011b).
- Impact analysis: Security analysts can use SCADA testbeds to analyse the impacts of cyberattacks against a SCADA system by quantifying the degree of damage of such attacks. For example, SWaT, PowerCyber and SCADA-SST can be used for the impact analysis of attacks against a SCADA system (Mathur and Tippenhauer, 2016), Ashok et al. (2020), Ghaleb et al. (2016a).

## 9. Existing specific cybersecurity controls and mitigation mechanisms

Overall, the nuances of industrial operations and fundamental differences in how similar technologies are used and deployed between IT and ICS networks mean that IT solutions cannot be de-

ployed over ICS. Furthermore, defensive mechanisms deployed in IT conflict with ICS system requirements. One notable difference is that the demand for high availability in ICS systems complicates the security of such systems. As a result, traditional IT cybersecurity controls cannot be applied to operational technology (OT) systems and ICS due to the operational nature of such systems. For example, penetration testing and vulnerability scanning can trigger significant disruptions of SCADA services. The following cybersecurity controls can help enhance the security of a SCADA system:

### 9.1. Assets management inventory

Asset inventory can help discover unauthorised devices connected to a SCADA network. Device configuration, software and firmware are essential to document an asset inventory with the device locations in the network. The use of active asset inventory and ping response approaches, such as TCP SYN or ACK, are challenging in the context of a SCADA system. However, passive asset inventory can be applicable to a control system because it avoids generating additional traffic as it travels along the network without interrupting critical processes. Passive asset discovery can be accomplished using non-intrusive methods, including MAC-ARP tables, DNS or ICS-specific tools (Eden et al., 2015). Real-time asset inventory saves time, increases accuracy and detects unauthorised devices connected to a network (Mir and Ramachandran, 2019).

**Industrial Defender Automation Systems Manager (ASM):** ASM collects information on software and hardware versions across a SCADA network. It can scan IP and non-IP devices. If an unknown driver is attached to the network, the ARP watch generates an alert by comparing IP and MAC addresses with the presented IP and MAC addresses in the ASM system. Dragos Security CyberLean is used for passive asset inventory in IT and ICS systems. It utilises sensors around a network to perform passive monitoring and capture data network traffic for offline analysis. PAS Cyber Integrity offers an accurate inventory of device configurations for both OT and IT systems.

### 9.2. Vulnerability assessment and management

Although asset inventory can help an organisation identify its attack surface, vulnerability assessment can help determine several entries that attackers may use. On the one hand, in several ICS organisations, the field devices are vulnerable to malware attacks due to weaknesses in the structural design. It is crucial to conduct risk assessment on a routine basis (Ranathunga et al., 2016). On the other hand, automated scanning tools are used to manage and patch vulnerabilities in IT networks. System upgrades can be installed according to scan reports. However, this approach is not practical for assessing SCADA vulnerabilities because it halts the system's primary functions and services. Passive vulnerability assessment can help manage organisational risks.

### 9.3. Using safe memory languages

Rust is a memory-safe language (Matsakis and Klock, 2014). It is used to develop reliable and efficient systems. It guarantees memory safety, isolation and concurrency. It also supports direct stack allocation by allowing fine-grained control of memory representation. If this language is used to design future IoT-SCADA operating environments, a buffer overflow will no longer be an issue.

### 9.4. Integrity checks

Integrity checks in field devices can prevent fuzzy attempts to crash a SCADA server or network by DoS and DDoS attacks. Yang et al. (2022) proposed a trusted encryption verification model to guarantee secure communication between a SCADA server and a TCP-RTU converter. It is based on token authentication and TLS. The author stated that the proposed mechanisms are efficient and compatible with real field devices.

### 9.5. Input validation

SCADA designers should consider all the possible entries where attackers can input data. Input validation frameworks, such as OWASP Enterprise Security API (ESAPI) and Struts, can help validate inputs by employing a whitelist approach. A whitelist should account for all data types, the amounts of data and the structure of the data integrated into the SCADA application or software. Adopting input validation techniques that validate user inputs against predefined rules, including range, length, divide by zero and format check, can mitigate SQL, XSS and command injection attacks. Administrators should also use parametrised or stored statements to process SQL queries. These statements are parsed by the database server separately from any parameters.

### 9.6. Output encoding

Output encoding involves the direct transformation of a user's inputs into a safe form in which the inputs cannot be interpreted as code in an HTML browser. Adopting HTML, URL, JavaScript and CSS encoding can prevent or mitigate XSS attacks. URL encoding should only be applied to the parameter values in a URL.

### 9.7. Privilege access management

Privilege management refers to managing privileged users to access critical assets in a control system. To prevent privilege escalation, administrators should consider the following:

- System installations should not run in privileged mode.
- There should be rule-based access control for SCADA field devices.
- The principle of least privilege (PoLP) should be considered because it prevents malicious behaviour. The PoLP ensures that programmers and users only have the necessary privileges to perform their tasks.
- The most often used authentication, authorisation and accounting (AAA) server is the Remote Authentication Dial-In User Service (RADIUS) (RFC 2865 and 2866) using IEEE 802.1x with the extensible authentication protocol. It plays a significant role in user authentication at all levels in a network. A salted password provides higher security in terms of user authentication.
- VBN gateways should be implemented to manage remote access to a SCADA network.
- Physical tokens should be considered for accessing physical areas.

### 9.8. Credential management

It plays a vital role in improving cybersecurity in an industrial SCADA system. SCADA system administrators should consider the following:

- Avoid using default password configurations to prevent information disclosure.
- All user passwords should be salted to reduce rainbow table attacks. Password salting involves adding data, up to 32-bit characters or more, to a password and then hashing it.

### 9.9. Intrusion detection, prevention and prediction systems

An IDS can prevent malware and viruses from propagating over a SCADA network. Integrated IDSs, including host based and network based, are the most efficient at detecting and controlling attacks against SCADA networks and devices. More details about host-based and network-based IDSs can be found in Section 6. In addition, a distributed IDS is recommended to be implemented in a distrusted SCADA architecture. It deals with heavy network traffic and log management because it employs a dedicated storage node. Therefore, it provides high monitoring functionality to field devices. Mohan et al. (2020) proposed an efficient algorithm that generates rule stets based on SCADA traffic patterns and integrated with a distributed intrusion detection framework. The proposed framework allows continuous monitoring of DNP3 traffic at the substation network and the detection of abnormal behaviour.

## 10. Challenges and open issues

A traditional SCADA system follows a centralised and static architecture. A SCADA centralised architecture is limited to interoperability Mikhail et al. (2017). Thus, an IoT- and cloud-based architecture has been proposed to overcome these weaknesses. However, the revolutionised SCADA system is prone to cyberattacks due to the integration of SCADA with cloud environments (Ye and Heidemann, 2006). Additionally, information leakage and privacy should be considered before moving to the cloud. Moreover, MITM can expose a communication link that relies on cloud communication because the cloud opens backdoors to ICSs and CIs (Ding et al., 2017; US-CERT, a). Manoj (2019) proposed a virtual SCADA architecture. However, the proposed framework focuses solely on cloud migration. Indeed, a SCADA system should be private, scalable, dynamic and fault-tolerant (Mikhail et al., 2017; Rivera and K. Tosh, 2019; Yadav and Paul, 2021).

**Datasets need further development:** Although NVD and CVE are good vulnerability databases, they fail to focus on the SCADA system. Security analysts use the KDD99 dataset for testing NIDS (Tavallaee et al., 2009). Additionally, it is used for the data mining tools competition (SUABOOT et al., 2020). However, KDD99 has some drawbacks. First, the dataset is extensive, which leads to an increase in the computational cost of an IDS. Therefore, only 10% of the dataset is used. Furthermore, KDD99 has several duplicate records and redundant data. As a result, the learning process is difficult and harmful for the system (Al-Jarrah et al., 2014). NSL-KDD is an improved version of KDD99 that removed the duplicate records (Meena and Choudhary, 2017). DARPA has a drawback in that it was recorded to an isolated network not connected to the internet (McHUGH, 2000). Hence, it is not synchronised with the current SCADA architecture. The water supply system dataset consists of log files of multiple features, such as temperature, water, inflow and outflow (SUABOOT et al., 2020; Tamy et al., 2019; Zohrevand et al., 2016). However, this dataset is outdated because it has not been updated since 2014. The water supply system dataset consists of log files of several features, including the level of water, inflow, temperature and outflow (SUABOOT et al., 2020; Zohrevand et al., 2016). Accordingly, a specific SCADA dataset should be considered. More importantly, the dataset should include all SCADA application areas.

**Intrusion prevention and prediction are not completely studied:** Although there are several studies on ID, there are few studies on attack prevention and prediction. More research effort is required to develop an IPS tool with prediction capability, one that can detect and predict future attacks that target a SCADA system. Tamy et al. (2019) stated that machine learning algorithms can predict attacks on SCADA (i.e. SVM, random forest, regression and Naive Bayes). The authors noted that the rainforest algorithm performs best. However, this approach is only feasible for reconnaissance attacks. Much effort is required to test feasibility and performance in a different environment.

**Distributed IDS for a SCADA system:** SUABOOT et al. (2020) suggested that DIDSs enable IDS for large-scale networks. The aggregation and correlation between different data sources can help in detecting the distribution of exploits or hidden malware. Additionally, the author pointed out that multiple learning models are required to improve the efficiency of DIDS (Clark and Niblet, 1989; Cruz et al., 2016; Shosha et al., 2011).

**Test and validation:** Testbed development is a complex and costly process because it needs an extensive amount of funding (Davis and Magrath, 2013; Imran et al., 2010). As seen from the test and validation literature, scalability is a big issue, and testbeds should have a high degree of fidelity. Researchers should focus on scalable simulated testbeds that support SCADA applications (water, gas, power, energy, etc.) with a high degree of fidelity (Geng et al., 2019; Morris et al., 2011a; SUABOOT et al., 2020).

**Research consideration for IoT-SCADA in a cloud environment:** From a safety perspective, IoTisation and cloud-based SCADA may lead to several issues, such as sharing data and information over the web, increasing bandwidth overload and latency. Any delay in decision making can cause losses in production. Hence, high bandwidth with low latency can improve system optimisation. From a security perspective, IoTisation makes SCADA vulnerable to various attacks, as the IoT has issues. More security recommendations can be found in (Ding et al., 2017).

**Forensic analysis, incident response and decision making:** Cyber forensics has been introduced to the SCADA research field. However, no conclusion has been reached yet due to several limitations and challenges of live forensic identification. For instance, the operative feature of a SCADA system requires it to be active 24/7. Most forensic tools used in IT environments do not support ICS. Therefore, vendors should add forensic capabilities to their products to help forensic investigators obtain a dump of memory and conduct further analysis. Additionally, more studies are required to find an effective method to verify volatile data during the process of collecting forensic evidence. Automation, data science and big data analysis can help solve the problems of live forensic investigations Eden et al. (2015). Applying analytic techniques to collected ICS data can extract valuable artefacts that can improve decision-making processes Rehman et al. (2019).

## 11. Conclusion

Several cyber threats can expose SCADA systems due to the integration of SCADA with the IoT and cloud environments. This survey analysed SCADA security from various aspects, including architecture, vulnerabilities, attacks, SCADA IDS and testbeds. Additionally, the survey outlined several open challenges related to SCADA threats, defences and validations to help future researchers. The survey indicated that several scopes need improvement in the fields of SCADA and ICS. For IDS and risk assessment, a specific SCADA dataset needs to be developed. For validation, SCADA testbeds can be improved by supporting high fidelity and scalability.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that may have appeared to influence the work reported in this paper.

## Data Availability

No data was used for the research described in the article.

## Acknowledgments

This research was supported by the Saudi Arabian Cultural Mission in Australia.

## References

Ahmed, I., Obermeier, S., Naedele, M., Richard, G.G., 2012. Scada systems: challenges for forensic investigation. IEEE 45, 44–51. doi:10.1109/MC.2012.325.
Ahmed, I., Obermeier, S., Sudhakaran, S., Roussev, V., 2017. Programmable logic controller forensics 15 (6).
Ahmed, I., Roussev, V., Johnson, W., Senthivel, S., Sudhakaran, S., 2016. A scada system testbed for cybersecurity and forensic research and pedagogy. In: Proceedings of the 2nd Annual Industrial Control System Security Workshop. Association for Computing Machinery, New York, NY, USA, p. 1–9. doi:10.1145/3018981.3018984.
Ahmed, M., Anwar, A., Shah, Z., Mahmood, A. N., Maher, M. J.,. An investigation of performance analysis of anomaly detection techniques for big data in scada systems. Ind. Netw. Intell. Syst. 2 (3).
Al-Jarrah, O.Y., Siddiqui, A., Elsalamouny, M., Yoo, P.D., Muhaidat, S., Kim, K., 2014. Machine-learning-based feature selection techniques for large scale network intrusion detection. In: Proceedings of the International Conference on Distributed Computing Systems Workshops (ICDCSW) doi:10.1109/ICDCSW.2014.14.
Almalawi, A., Tari, Z., Fahad, A., Yi, X., 2020. SCADA Security: Machine Learning Concepts for Intrusion Detection and Prevention. John Wiley & Sons.
Arnold, J., 1994. Swat-soil and water assessment tool.
Ashok, A., Krishnaswamy, S., Govindarasu, M., 2020. Powercyber: a remotely accessible testbed for cyber physical security of the smart grid. Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT). 10.1109/ISGT.2016.7781277
Awad, R., Lopez Jr, J., Rogers, M., 2019. Volatile memory extraction-based approach for level 0–1 cps forensics. Proceedings of the IEEE International Symposium on Technologies for Homeland Security (HST). doi:10.1109/HST47167.2019.9032919.
Barbosa, R. R. R., 2014. Anomaly detection in scada systems: a network based approach.
Bartman, T., Carson, K., 2016. Securing communications for scada and critical industrial systems. Proceedings of the IEEE 69th Annual Conference for Protective Relay Engineers (CPRE), 2–3. doi:10.1109/CPRE.2016.7914914.
Becejac, T., Eppinger, C., Ashok, A., Agrawal, U., O'Brien, J., 2020. Prime: a real–time cyber-physical systems testbed: from wide-area monitoring, protection, and control prototyping to operator training and beyond. IET Cyber-Phys. Syst. Theory Appl. 5 (2), 186–195.
Bergman, D.C., Jin, D.K., Nicol, D.M., Yardley, T., 2009. The virtual power system testbed and inter-testbed integration. CSET.
Breeuwsma, M., 2006. Forensic imaging of embedded systems using jtag (boundary-scan). Digital Invest. 3 (1).
Breunig, M.M.B., Kriegel, H.-P., T. Ng, R., Sander, J., 2000. Lof: identifying density-based local outliers. ACM.
Byres, E.J., Hoffman, D., Kube, N., 2006. On shaky ground–a study of security vulnerabilities in control protocols. In: Proceedings of the 5th American Nuclear Society International Meeting on Nuclear Plant Instrumentation, Controls, and HMI Technology.
Chassin, D.P., Schneider, K., Gerkensmeyer, C., 2008. Gridlab-d: An open-source power systems modeling and simulation environment. In: 2008 IEEE/PES Transmission and Distribution Conference and Exposition, pp. 1–5. doi:10.1109/TDC.2008.4517260.
Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Skinner, K., Valdes, A., 2007. Using model-based intrusion detection for scada networks.
Clark, P., Niblet, T., 1989. The cn2 induction algorithm. machine learning, 261–283.
Clarke, G., Reynders, D., 2004. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Newnes.
Corallo, A., Lazoi, M., Lezzi, M., 2020. Cybersecurity in the context of industry 4.0: a structured classification of critical assets and business impacts. Comput. Ind. 114, 103165.
Cruz, T., Rosa, L., Proença, J., Maglaras, L., Aubigny, M., Lev, L., Jiang, J., Simões, P., 2016. A cybersecurity detection framework for supervisory control and data acquisition systems. IEEE doi:10.1109/TII.2016.2599841.
Cui, M., Wang, J., Yue, M., 2019. Machine learning-based anomaly detection for load forecasting under cyberattacks. IEEE 10 (5).
CVE, 2020. Common vulnerabilities exposure. Accessed: https://cve.mitre.org/.
CWE, 2020. Common weakness enumeration. Accessed: https://cwe.mitre.org/about/board.html.
CWE-20,. Cwe-20: Improper input validation, 2020Accessed: 12 October https://cwe.mitre.org/data/definitions/20.html.
CWE-22,. Cwe-22: Improper limitation of a path name to a restricted directory ('path traversal'). Accessed: 19 October 2020 https://cwe.mitre.org/data/definitions/22.html.
CWE-250, Cwe-250: Execution with unnecessary privileges. Accessed: 25 October 2020 https://cwe.mitre.org/data/definitions/250.html.
CWE-426,. Cwe-426: Untrusted search path. Accessed: 22 October 2020 https://cwe.mitre.org/data/definitions/426.html.
CWE-74,. Cwe-74: improper neutralization of special elements in output used by a downstream component ('injection'). Accessed: 16 October 2020 https://cwe.mitre.org/data/definitions/74.html.
CWE-77,. Improper neutralization of special elements used in a command ('command injection'). Accessed: 19 October 2020 https://cwe.mitre.org/data/definitions/77.html.
CWE-78,. Cwe-78: Improper neutralization of special elements used in an os command ('os command injection'). Accessed: 16 September 2020https://cwe.mitre.org/data/definitions/78.html.
CWE-79,. Cwe-79: Improper neutralization of input during web page generation ('cross-site scripting'). Accessed: 17 October 2020https://cwe.mitre.org/data/definitions/79.html.
Daryabar, F., Dehghantanha, A., Udzir, N.I., bin Shamsuddin, S., et al., 2012. Towards secure model for scada systems. In: Proceedings of the Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec). IEEE, pp. 60–64. doi:10.1109/CyberSec.2012.6246111.
Davis, J., Magrath, S., 2013. A Survey of Cyber Ranges and Testbeds. Cyber and Electronic Warfare Division.
Dimitrov, W., Syarova, S., 2019. Analysis of the functionalities of a shared ICS security operations center. IEEE 2. doi:10.1109/BdKCSE48644.2019.9010607.
Ding, J., Atif, Y., Andler, S.F., Lindstrom, B., Jeusfeld, M., 2017. Cps-based threat modeling for critical infrastructure protection. ACM 45.
Dussel, P., Gehl, C., Laskov1, P., Bußer, J.-U., Jens-Uwe, C., Kastner, J., 2010. Cyber-critical infrastructure protection using real-time anomaly detection. In: Proceedings of the International Workshop on Critical Information Infrastructures Security.
Eden, P., Blyth, A., Burnap, P., Jones, K., Stoddart, K., 2015. A Forensic Taxonomy of Scada Systems and Approach to Incident Response. BCS Learning & Development Ltd.
Falliere, N., Murchu, L. O., Chien, E., 2011. W32. Stuxnet dossier. White paper, Symantec Corp., Security Response 5 (6), 29.
Fillatre, L., Nikiforov, I., Willett, P., et al., 2017. Security of scada systems against cyber–physical attacks. IEEE Aerosp. Electron. Syst. Mag. 32 (5), 28–45.
Geng, Y., Wang, Y., Liu, W.L., Wei, Q., Liu, K., Wu, H., 2019. A survey of industrial control system testbeds. In: Proceedings of the IOP Conference Series: Materials Science and Engineering.
Ghaleb, A., Zhioua, S., Almulhem, A., 2016a. Scada-sst: a scada security testbed, 1–6. doi:10.1109/WCICSS.2016.7882610.
Ghaleb, A., Zhioua, S., Almulhim, A., 2016b. Scada-sst: a scada security testbed. IEEE. doi:10.1109/WCICSS.2016.7882610.
Ghodratnama, S., Zakershahrak, M., Sobhanmanesh, F., 2020. An intelligent summarization approach for identifying hidden anomalies.
Ghosh, S., Sampalli, S., 2019. A survey of security in scada networks:current issues and future challenges. IEEE Access 7. doi:10.1109/ACCESS.2019.2926441.
Goetz, E., Shenoi, S., 2007. Critical Infrastructure Protection. Springer, p. 194.
Guillén, J.H., del Rey, A.M., Casado-Vara, R., 2019. Security countermeasures of a sciras model for advanced malware propagation. IEEE Access 7, 135472–135478. doi:10.1109/ACCESS.2019.2942809.
Hahn, A., Govindarasu, M., 2012. Model-based intrustion detection for the smart grid(minds). ACM doi:10.1145/2459976.2460007.
Hall, K.B., Ngalamou, L., 2019. Securing wireless scada systems in rural american power grids. In: Proceedings of the IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, pp. 0257–0262. doi:10.1109/UEMCON47517.2019.8993004.
Han, S., Xie, M., Chen, H.-H., Ling, Y., 2014. Intrusion detection in cyber-physical systems: techniques and challenges. IEEE Syst. J. doi:10.1109/JSYST.2013.2257594.
Hasan, M.A.M., Nasser, M., Pal, B., Ahmad, S., 2014. Support vector machine and random forest modeling for intrusion detection system (ids). J. Intell. Learn. Syst. Appl. 2014.
Hilal, H., Nangim, A., 2017. Network security analysis scada system automation on industrial process. In: Proceedings of the International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP). IEEE, pp. 1–6. doi:10.1109/BCWSP.2017.8272569.
Huong, T.T., Bac, T.P., Long, D.M., Luong, T.D., Dan, N.M., Thang, B.D., Tran, K.P., et al., 2021. Detecting cyberattacks using anomaly detection in industrial control systems: a federated learning approach. Comput. Ind. 132, 103509.
ICS-US-CERT, 2018a. Open automation software opc systems net dll hijacking vulnerability. Accessed: 23 September 2020https://us-cert.cisa.gov/ics/advisories/ICSA-15-344-02.
ICS-US-CERT, 2018b. Siemens siprotec 4, siprotec compact, digsi 4, and en100 ethernet module (update d). Accessed: 23 August 2020https://us-cert.cisa.gov/ics/advisories/ICSA-18-067-01.
ICS-US-CERT, 2019. Omron plc cj and cs series. Accessed: 29 August 2020https://us-cert.cisa.gov/ics/advisories/icsa-19-346-02.
ICSA, U.-C., 2020. Sae it-systems fw-50 remote telemetry unit (rtu). Accessed: 04 February 2020https://us-cert.cisa.gov/ics/advisories/ICSA2012602.
Imran, M., Said, A.M., Hasbullah, H., 2010. A survey of simulators, emulators and testbeds for wireless sensor networks. In: Proceedings of the International Symposium on Information Technology, ITSim.
Irfan, Siddavatam, I.A., Kazi, F., 2015. Security assessment framework for cyber physical systems: a case-study of dnp3 protocol. IEEE 3. doi:10.1109/IBSS.2015.7456631.
Irmak, E., Erkek, I., 2018. An overview of cyber-attack vectors on scada. IEEE doi:10.1109/ISDFS.2018.8355379.
Ismail, S., Sitnikova, E., Slay, J., 2014. Towards developing scada systems security measures for critical infrastructures against cyber-terrorist attacks. In: Proceedings of the IFIP International Information Security Conference. Springer, pp. 242–249.

Jicha, A., Patton, M., Chen, H., 2016. Scada honeypots: an in-depth analysis of conpot. In: Proceedings of the IEEE conference on intelligence and security informatics (ISI). IEEE, pp. 196–198. doi:10.1109/ISI.2016.7745468.
Johnson, B., Caban, D., Krotofil, M., Scali, D., Brubaker, N., Glyer, C., 2017. Attackers deploy new ICS attack framework "triton" and cause operational disruption to critical infrastructure. Threat Res. Blog 14.
Joshi, V., Adhikari, M.S., Patel, R., Singh, R., Gehlot, A., 2019. Industrial Automation: Learn the Current and Leading-Edge Research on SCADA Security. BPB Publications.
Kalech, M., 2019. Cyber-attack detection in scada systems using temporal pattern recognition techniques. Comput. Secur. 84. doi:10.1016/j.cose.2019.03.007.
Kalech, M., 2019. Cyber-attack detection in scada systems using temporal pattern recognition techniques. Comput. Secur. 84, 225–238. doi:10.1016/j.cose.2019.03.007.
Kaspersky, 2017a. From shamoon to stonedrill: Wipers attacking saudi organizations and beyondhttps://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07180722/Report_Shamoon_StoneDrill_final.pdf.
Kaspersky, I., 2017b. Threat landscape for industrial automation systemshttps://www.kaspersky.com/resource-center/threats/blackenergy.
Keliris, A., Salehghaffari, H., Cairl, B., Krishnamurthy, P., Maniatakos, M., Khorrami, F., 2016. Machine learning-based defense against process-aware attacks on industrial control systems. In: Proceedings of the IEEE International Test Conference (ITC). IEEE, pp. 1–10.
Koutsandria, G., Gentz, R., Jamei, M., 2015. A real-time testbed environment for cyber-physical security on the power grid. ACM doi:10.1145/2808705.2808707.
Kuipers, D., 2008. Common cyber security vulnerabilities observed in control system assessments by the inl NSTB program. Idaho National Lab.(INL), Idaho Falls, ID (United States), Tech. Rep.
Langner, R., 2011. Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. 9 (3), 49–51.
Larkin, R.D., Lopez Jr, J., W. Butts, J., Grimaila, M.R., 2014. Evaluation of security solutions in the scada environment. ACM 45, 40–42. doi:10.1145/2591056.2591060.
Lee, D., Kim, H., Kim, K., Yoo, P.D., 2014. Simulated attack on dnp3 protocol in scada system. In: Proceedings of the 31th Symposium on Cryptography and Information Security, Kagoshima, Japan, pp. 21–24.
Lee, R., Slowik, J., Miller, B., Cherepanov, A., Lipovsky, R., 2017. Industroyer/Crashoverride: Zero Things Cool About a Threat Group Targeting the Power Grid. Black Hat USA.
Lemay, A., Fernandez, J.M., 2016. Providing {SCADA} network data sets for intrusion detection research. In: Proceedings of the 9th Workshop on Cyber Security Experimentation and Test (CSET 16).
Lin, H., Slagell, A., Kalbarczyk, Z., Sauer, P.W., Iyer, R.K., 2013. Semantic security analysis of scada networks to detect malicious control commands in power grids. In: Proceedings of the first ACM Workshop on Smart Energy Grid Security, pp. 29–34.
Liptak, B.G., 2018. Modbus. CRC Press, p. 784.
Mafra, P.M., Moll, V., da Silva Fraga, J., Santin, A.O., 2010. Octopus-iids: an anomaly based intelligent intrusion detection system. In: Proceedings of the IEEE symposium on Computers and Communications. IEEE, pp. 405–410.
Mallouhi, M., Al-Nashif, Y., Cox, D., Chadag, T., Hariri, S., 2011. A testbed for analyzing security of scada control systems tasscs. In: Proceedings of the IEEE ISGT doi:10.1109/ISGT.2011.5759169.
Manoj, K., 2019. Industrial Automation with SCADA: Concepts, Communications and Security. Notion Press.
Marco, C., Stavrou, I., Dimmock, S., Jhonson, C., 2020. Introducing a forensics data type taxonomy of acquirable artefacts from programmable logic controllers. IEEE doi:10.1109/CyberSecurity49315.2020.9138879.
Mathur, A.P., Tippenhauer, N.O., 2016. Swat: A water treatment testbed for research and training on ics security. In: Proceedings of the International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater). IEEE, pp. 31–36. doi:10.1109/CySWater.2016.7469060.
Matsakis, N.D., Klock, F.S., 2014. The rust language. ACM SIGAda Ada Lett. 34 (3), 103–104.
Mayadev, Ushakumari, S.S., Vinodchandra, S.S., 2014. Scada-based operator support system for power plant equipment fault forecasting. Inst. Eng. doi:10.1007/s40031-014-0117-9.
Maynard, P., McLaughlin, K., Haberler, B., 2014. Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks. In: Proceedings of the 2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014) 2, pp. 30–42.
McHUGH, J., 2000. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM doi:10.1145/382912.382923.
Meena, G., Choudhary, R.R., 2017. A review paper on ids classification using kdd 99 and nsl kdd dataset in weka. IEEE doi:10.1109/COMPTELIX.2017.8004032.
Mehra, R., 2012. PLCs & SCADA: Theory and Practice. Laxmi Publications.
Mesbah, M., Azer, M., 2019. Cyber threats and policies for industrial control systems. In: Proceedings of the International Conference on Smart Applications, Communications and Networking (SmartNets). IEEE, pp. 1–6.
Mikhail, A., Kamil, I.A., Mahajan, H., 2017. Increasing scada system availability by fault tolerance techniques. In: Proceedings of the IEEE International Conference on Computing, Communication, Control and Automation (ICCUBEA) doi:10.1109/ICCUBEA.2017.8463911.
Miller, B., Rowe, D., 2012. A survey scada of and critical infrastructure incidents. In: Proceedings of the 1st Annual conference on Research in information technology, pp. 51–56. doi:10.1145/2380790.2380805.
Mir, A.W., Ramachandran, R.K., 2019. Security gaps assessment of smart grid based scada systems. Inf. Comput. Secur..
MITRE,. Attamp;ck®for industrial control systems. Acessed: 22 May 2022. https://collaborate.mitre.org/attackics/index.php/Main_Page.
Mohan, S.N., Ravikumar, G., Govindarasu, M., 2020. Distributed intrusion detection system using semantic-based rules for scada in smart grid. In: Proceedings of the IEEE/PES Transmission and Distribution Conference and Exposition (T&D). IEEE, pp. 1–5.
Morris, T., Gao, W., 2014. Industrial control system traffic data sets for intrusion detection research. In: Proceedings of the International Conference on Critical Infrastructure Protection. Springer, pp. 65–78.
Morris, T., Vaughn, R., Dandass, Y., 2011a. A testbed for scada control system cybersecurity research and pedagogy.
Morris, T., Vaughn, R., Dandass, Y.S., 2011. A testbed for scada control system cybersecurity research and pedagogy. In: Proceedings of the 7th Annual Workshop on Cyber Security and Information Intelligence Research, p. 1.
Moustafa, N., Slay, J., 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: Proceedings of the Military Communications and Information Systems Conference (MilCIS). IEEE, pp. 1–6.
Nazir, S., Patel, S., Patel, D., 2017. Assessing and augmenting scada cyber security: a survey of techniques. Comput. Secur. 70, 438. doi:10.1016/j.cose.2017.06.010.
NIST, 2020. National vulnerability database. Accessed: https://nvd.nist.gov/vuln/data-feeds.
NVD, 2020. National vulnerability databasehttps://nvd.nist.gov/vuln/detail/CVE-2020-35558.
OSSEC,. Host intrusion detection for everyone„ 2021. Accessed: 19 July 2021 https://www.ossec.net.
Papadimitriou, S., Kitagawa, H., Gibbons, P., Faloutsos, C., 2007. LOCI: fast outlier detection using the local correlation integral.
Papp, D., Ma, Z., Buttyan, L., 2015. Embedded systems security: threats,vulnerabilities, and attack taxonomy. In: Proceedings of the IEEE 13th Annual Conference on Privacy, Security and Trust (PST) doi:10.1109/PST.2015.7232966.
Pliatsios, D., Sarigiannidis, P., Lagkas, T., Sarigiannidis, A.G., 2020. A survey on scada systems: secure protocols, incidents, threats and tactics. IEEE Commun. Surv. Tutor. 22 (3), 1942–1976.
Poudel, S., Ni, Z., Malla, N., 2017. Real-time cyber physical system testbed for power system security and control. Int. J. Electr. Power Energy Syst. 90, 124–133.
Poudel, S., Ni, Z., Malla, N., 2017. Real-time cyber physical system testbed for power system security and control. Int. J. Electr. Power Energy Syst. 90, 124–133. doi:10.1016/j.ijepes.2017.01.016.
Queiroz, C., Mahmood, A., Hu, J., Tari, Z., Yu, X., 2009. Building a scada security testbed. In: Proceedings of the Third International Conference on Network and System Security. IEEE, pp. 357–364.
Queiroz, C., Mahmood, A., Tari, Z., 2011. Scadasim' A framework for building scada simulations. IEEE Trans. Smart Grid 2 (4), 589–597. doi:10.1109/TSG.2011.2162432.
Radoglou-Grammatikis, P., Sarigiannidis, P., Giannoulakis, I., Kafetzakis, E., Panaousis, E., 2019. Attacking iec-60870-5-104 scada systems. In: Proceedings of the IEEE World Congress on Services (SERVICES), Vol. 2642. IEEE, pp. 41–46.
Rakas, S. V. B., Stojanović, M. D., Marković-Petrović, J. D., 2020. A review of research work on network-based scada intrusion detection systems. doi:10.1109/ACCESS.2020.2994961.
Ranathunga, D., Roughan, M., Nguyen, H., Kernick, P., Falkner, N., 2016. Case studies of scada firewall configurations and the implications for best practices. IEEE Trans. Netw. Serv. Manag. 13 (4), 871–884.
Rehman, M.H.u., Yaqoob, I., Salah, K., Imran, M., Jayaraman, P.P., Perera, C., 2019. The role of big data analytics in industrial internet of things. Future Gener. Comput. Syst. 99, 247–259. doi:10.1016/j.future.2019.04.020.
Ren, W., Yardley, T., Nahrstedt, K., 2018. Edmand: edge-based multi-level anomaly detection for scada networks. doi:10.1109/SmartGridComm.2018.8587533.
Revathi, S., Malathi, A., 2013. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int. J. Eng. Res. Technol. (IJERT) 2 (12), 1848–1853.
Rivera, A.O.G., K. Tosh, D., 2019. Towards security and privacy of scada systems through decentralized architecture. IEEE doi:10.1109/CSCI49370.2019.00230.
RK, S., 2013. Security and protection of scada: a bigdata algorithmic approach. ACM doi:10.1145/2523514.2523595.
Robert E, J., 2010. Survey of scada security challenges and potential attack vectors. IEEE.
Rodofile, N.R., Radke, K., Foo, E., 2016. Dnp3 network scanning and reconnaissance for critical. ACM 1–10. doi:10.1145/2843043.2843350.
Rosa, L., Freitas, M., Mazo, S., Monteiro, E., Cruz, T., Simões, P., 2019. A comprehensive security analysis of a scada protocol: from osint to mitigation. IEEE Access 7, 42156–42168. doi:10.1109/ACCESS.2019.2906926.
Rosa, L., Freitas, M., Mazo, S., Monteiro, E., Cruz, T., Simões, P., 2019. A comprehensive security analysis of a scada protocol: from osint to mitigation. IEEE Access 7, 42156–42168. doi:10.1109/ACCESS.2019.2906926.
Roumani, Y., Nwankpa, J., 2020. Examining exploitability risk of vulnerabilities: a hazard model. Commun. Assoc. Inf. Syst. 46 (1), 18.
Rrushi, J., Farhangi, H., Howey, C., Carmichael, K., Dabell, J., 2015. A quantitative evaluation of the target selection of havex ics malware plugin. In: Proceedings of the Industrial Control System Security (ICSS) Workshop.

Sajid, A., Abbas, H., Saleem, K., 2016. Cloud-assisted IOT-based scada systems security: a review of the state of the art and future challenges. IEEE 4. doi:10.1109/ACCESS.2016.2549047.
Samtani, S., Yu, S., Zhu, H., Patton, M., Chen, H., 2016. Identifying scada vulnerabilities using passive and active vulnerability assessment techniques. In: Proceedings of the IEEE Conference on Intelligence and Security Informatics (ISI). IEEE, pp. 25–30. doi:10.1109/ISI.2016.7745438.
Sayegh, N., Chehab, A., Elhajj, I.H., Kayssi, A., 2013. Internal security attacks on scada systems. In: Proceedings of the Third International Conference on Communications and Information Technology (ICCIT). IEEE, pp. 22–27. doi:10.1109/ICCITechnology.2013.6579516.
Schölkopf, B., Smola, A.J., Bach, F., et al., 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT press.
Sen, S.K., Dey, C., 2020. Industrial Automation Technologies. CRC Press.
Senthivel, S., Ahmed, I., Roussev, V., 2017. Scada network forensics of the pccc protocol. Digital Invest. doi:10.1016/j.diin.2017.06.012.
Seri, B., Vishnepolsky, G., Zusman, D., 2019. Critical vulnerabilities to remotely compromise vxworks, the most popular rtos.
Shahzad, A., Musa, S., Aborujilah, A., Ifran, M., 2014. Industrial control systems (ICSS) vulnerabilities analysis and scada security enhancement using testbed encryption. ACM doi:10.1145/2557977.2558061.
Shaw, W.T., 2006. Cybersecurity for SCADA Systems. PennWell Books, p. 17.
Shosha, A.F., Gladyshev, P., Wu, S.-S., Liu, C.-C., 2011. Detecting cyber intrusions in scada networks using multi-agent collaboration. In: Proceedings of the IEEE 16th International Conference on Intelligent System Applications to Power Systems doi:10.1109/ISAP.2011.6082170.
Smith, S., 2014. A proposal for a taxonomy for vulnerabilities in supervisory control and data acquisition (scada) systems. ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD.
Snort,. Snort intrusion detection tool. https://www.snort.org/.
Sommestad, T., Ericsson, G.N., Nordlander, J., 2010. Scada System Cyber Security -A Comparison of Standards. IEEE, p. 2.
Spenneberg, R., Brüggemann, M., Schwartke, H., 2016. Plc-Blaster: A Worm Living Solely in the Plc, 16. Black Hat Asia, pp. 1–16.
Stouffer, K., Falco, J., 2006. Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security. National Institute of Standards and Technology.
SUABOOT, J., FAHAD, A., TARI, Z., GRUNDY, J., MAHMOOD, A.N., ALMALAWI, A., Y. ZOMAYA, A., DRIRA, K., 2020. A taxonomy of supervised learning for idss in scada environments. ACM Comput. Surv. 53. doi:10.1145/3379499.
Tamy, S., Belhadaoui, H., Rabbah, M.A., 2019. An evaluation of machine learning algorithms to detect attacks in scada network. IEEE doi:10.1109/CMT.2019.8931327.
Tang, J., Chen, Z. C., Fu, A. W., 2007. Capabilities of outlier detection schemes in large datasets, framework and methodologies. doi:10.1007/s10115-005-0233-6.
Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD cup 99 dataset. IEEE 53–58. doi:10.1109/CISDA.2009.5356528.
Tripwire,. Tripwire home page2021Accessed: 21 July 2021https://www.tripwire.com/.
US-CERT, a. Cyber security and information security agency2020. Accessed: 17 October 2020 https://us-cert.cisa.gov/.
US-CERT, b. Russia cyber threat overview and advisories, 2021. 2021-08-21 https://www.cisa.gov/uscert/russia.
US-CERT, 2014a. Elipse scada dnp3 denial of service. Accessed: 17 September 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-14-303-02.
US-CERT, 2014b. Schneider electric wonderware vulnerabilities. Accessed: 08 December 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-14-238-0.
US-CERT, 2017. Schneider electric wonderware historian. Accessed: 27 September 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-17-024-01.
US-CERT, 2018a. Martem telem-gw6/gwm (update b). Accessed: 10 October 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-18-142-01.
US-CERT, 2018b. Schneider electric somove software and dtm software components. Accessed: 30 December 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-18-065-02.
US-CERT, 2018c. Siemens simatic s7-1500 cpu firmware vulnerabilities. Accessed: 11 October 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-14-073-01.
US-CERT, 2019a. Advantech webaccess/scada. Accessed: 16 April 2021 https://us-cert.cisa.gov/ics/advisories/icsa-19-178-05.
US-CERT, 2019b. Lcds - leão consultoria e desenvolvimento de sistemas ltda me laquis scada. Accessed: 01 December 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-19-015-0.
US-CERT.CISA, 2019. Siemens simatic panels and wincc (tia portal). Accessed: 04 November 2020 https://us-cert.cisa.gov/ics/advisories/ICSA-19-134-09.
Vapnik, V., 1999. The Nature of Statistical Learning Theory. Springer science & Business Media.
Wei, J., Yan, L.K., Muhammad, A.H., 2015. Mose: Live Migration Based on-the-Fly Software. ACM.
Wermann, A.G., Bortolozzo, M.C., Germano da Silva, E., Schaeffer-Filho, A., Paschoal Gaspary, L., Barcellos, M., 2016. Astoria: a framework for attack simulation and evaluation in smart grids. In: Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, pp. 273–280. doi:10.1109/NOMS.2016.7502822.
Wong, K., Dillabaugh, C., Seddigh, N., Nandy, B., 2017. Enhancing suricata intrusion detection system for cyber security in scada networks. doi:10.1109/CCECE.2017.7946818.
Xu, Y., Yang, Y., Li, T., Ju, J., Wang1, Q., 2017. Review on Cyber Vulnerabilities of Communication Protocols in Industrial Control Systems.
Yadav, G., Paul, K., 2019. Assessment of Scada System Vulnerabilities. IEEE.
Yadav, G., Paul, K., 2021. Architecture and security of scada systems: a review. Int. J. Crit. Infrastruct. Prot. 100433. doi:10.1016/j.ijcip.2021.100433.
Yampolskiy, M., Horvath, P., Koutsoukos, X., Xue, Y., Sztipanovits, J., 2013. Taxonomy for description of cross-domain attacks on CPS. ACM doi:10.1145/2461446.2461465.
Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Pranggono, B., Wang, H., 2013. Intrusion detection system for IEC 60870-5-104 based scada networks. In: Proceedings of the IEEE power & energy society general meeting. IEEE, pp. 1–5.
Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Im, E.G., Pranggono, B., Wang, H.F., 2014. Multiattribute scada-specific intrusion detection system for power networks. IEEE Trans. Power Deliv. 29 (3), 1092–1102. doi:10.1109/TPWRD.2014.2300099.
Yang, Y.-S., Lee, S.-H., Chen, W.-C., Yang, C.-S., Huang, Y.-M., Hou, T.-W., 2022. Securing scada energy management system under DDOS attacks using token verification approach. Appl. Sci. 12 (1), 530.
Yasakethu, S., Jiang, J., 2013. Intrusion detection via machine learning for scada system protection.
Ye, W., Heidemann, J., 2006. Enabling Interoperability and Extensibility of Future 'Scada' Systems. Networked Embedded Control for Cyber Physical Systems.
Yeboah-Ofor, A., Boachie, C., 2019. Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning, p. 67.
Zhu, B., Joseph, A., Sastry, S., 2011. A taxonomy of cyber attacks on scada systems. Proceedings of the IEEE International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing. doi:10.1109/iThings/CPSCom.2011.34.
Zohrevand, Z., Glasser, U., Shahir, H.Y., Tayebi, M.A., Costanzo, R., 2016. Hidden Markov Based Anomaly Detection for Water Supply Systems. IEEE.

**Manar ALANAZI** is currently a PhD candidate in the school of mathematical science and engineering at La Trobe University, Australia. She received the mater degree in internetworking and cybersecurity from Macquarie university, Sydney, Australia, and the bachelor's degree in computer science from Aljouf university, Saudi Arabia. Manar has worked as a teaching assistant at Aljouf university, Saudi Arabia. She lectured CS student in IT subjects both theoretical and practical. Her research is about cyber security in the context of SCADA system.

**Dr. Abdun Mahmood** received his PhD from the University of Melbourne, Australia, in 2008 the MSc (Research) degree in computer science and the B.Sc. degree in applied physics and electronics from the University of Dhaka, Bangladesh, in 1999 and 1997, respectively. Dr. Mahmood had an academic career in University since 2000, working at University of Dhaka, RMIT University, UNSW Canberra and currently in La Trobe University as an Associate Professor (Reader). Dr. Mahmood leads a group of researchers focusing on Machine Learning and Cybersecurity including Anomaly Detection in Smart Grid, SCADA security, Memory Forensics, and False Data Injection. Dr. Mahmood has been successful to attract over a $1M+ in grant funding as a CI, including two ARC Linkage Projects.

**Mohammad Jabed Morshed Chowdhury** is currently working as Lecturer in Cyber Security Program at La Trobe University, Melbourne, Australia. He has earned his PhD from Swinburne University of Technology, Melbourne, Australia. He has earned his double Masters in Information Security and Mobile Computing from Norwegian University of Science and Technology, Norway and University of Tartu, Estonia under the European Union's Erasmus Mundus Scholarship Program. He has published his research in top journal and conferences. He is currently working with Security, Privacy and Trust. He has published research work related to Data Sharing, Privacy, and Blockchain. He also worked as PC member in different top tier conferences.