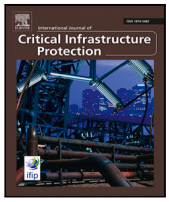
international journal of critical infrastructure protection 42 (2023) 100615

Contents lists available at ScienceDirect

International Journal of Critical Infrastructure Protection

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



# Software-Defined Networking approaches for intrusion response in Industrial Control Systems: A survey

Xabier Etxezarreta *, Iñaki Garitano, Mikel Iturbe, Urko Zurutuza

*Electronics and Computing Department, Mondragon University, Goiru 2, 20500 Arrasate-Mondragón, Spain*

ARTICLE INFO

*Keywords:*
Software-Defined Networking
Industrial Control Systems
Cyber-physical systems
Critical Infrastructure
Intrusion response

ABSTRACT

Industrial Control Systems (ICSs) are a key technology for life-sustainability, social development and economic progress used in a wide range of industrial solutions, including Critical Infrastructures (CIs), becoming the primary target for multiple security attacks. With the increase of personalized and sophisticated attacks, the need for new tailored ICS cybersecurity mechanisms has increased exponentially, complying with specific ICS requirements that Information Technology (IT) security systems fail to meet. In this survey, a comprehensive study of ICS intrusion response is conducted, focusing on the use of Software-Defined Networking (SDN) for the development of intrusion response strategies in ICS. With its centralized control plane, increased programmability and global view of the entire network, SDN enables the development of intrusion response solutions that provide a coordinated response to mitigate attacks. Through the survey, an analysis of ICS security requirements and the applicability of SDN is conducted, identifying the advantages and disadvantages compared to traditional networking and protocols. Furthermore, a taxonomy on intrusion response strategies is presented, where different proposals are discussed and categorized according to intrusion response strategy and deployment environment characteristics. Finally, future research directions and challenges are identified.

## 1. Introduction

Industrial Control System (ICS) is a general term that covers various types of control systems and associated instruments used to operate and/or automate industrial processes (NIST SP 800-82 [1]). ICSs are heterogeneous systems that include actuators, sensors, control and networking components [2]. They support manufacturing and control at different scales, such as, nuclear plants, electrical grids, hydroelectric dams and so on. ICSs encompass several types of control systems, including Supervisory Control and Data Acquisition (SCADA) systems and other control systems such as Programmable Logic Controllers (PLCs) or Remote Terminal Units (RTUs), often found in the industrial sectors and Critical Infrastructures (CIs).

Industry 4.0 or fourth industrial revolution are widely used concepts to refer to the digitization that ICSs are undergoing through the integration of electronics and computing systems [3]. ICSs are composed of two main zones. On the one side, the Operational Technology (OT) network contains the hardware and software used to monitor and control industrial processes. On the other side, the Information Technology (IT) network used for information processing by using workstations, server and databases. Originally, OT and IT were independent concepts in the industrial domain, but the convergence of these two technologies has given networking capabilities to ICSs, enabling cost savings and flexibility in the remote management and monitoring of industrial processes with the use of cyber-components [4]. The modern trend is the complete adoption of IT and the massive interconnection of systems, resulting in the emergence of Cyber Physical Systems (CPSs) [5].

In the actual state of the society, where a failure in a CI such as an electrical grid or a nuclear plant could affect the health, physical integrity, safety and the well-being of the population, it is extremely necessary to develop highly reliable strategies to avoid an unimaginable catastrophe [6]. Interconnection is an increasing event in CIs, and due to their interdependence, a security breach in a certain process can compromise the whole industrial network to which it is connected [7]. Isolation has been a key factor in ensuring the security of industrial operations, but the adoption of IT systems and increasingly interconnected ICSs, exposes the originally isolated ICSs to corporate networks, including the Internet [8]. This change resulted in traditional isolation and security through obscurity techniques no longer being effective in ICS environments. Less isolation implies a greater need to secure these systems against attacks. This has pushed the research and development of security measures that could mitigate threats in such a sensitive scenario, and which is still active today [9].

To prevent such catastrophic scenarios, it is extremely recommended applying secure-by-design approaches (NIST SP 800–160),

* Corresponding author.
*E-mail address:* xetxezarreta@mondragon.edu (X. Etxezarreta).

https://doi.org/10.1016/j.ijcip.2023.100615
Received 3 November 2022; Received in revised form 14 March 2023; Accepted 13 June 2023
Available online 19 June 2023

**Table 1**
Comparison of related survey articles.

| References | SDN | ICS | Response | Review taxonomy |
|---|---|---|---|---|
| Rakas et al. [14] | No | Yes | No | Signature-based, statistical-based, knowledge-based, Machine Learning (ML)-based (neural networks and clustering/outlier detection), specification-based and hybrid intrusion detection. |
| Chica et al. [15] | Yes | No | Yes | Threat detection, NFV/cloud-based security, attack remediation (denial of service, side-channel, rogue device infiltration, malformed packets), identityaccess/management, network status monitoring, security assessment and forensics. |
| Mazhar et al. [16] | Yes | No | Yes | Anomaly/Entropy-based detection, ML-based detection, Manufacturing Usage Description (MUD)-based detection, ML and MUD-based prevention. |
| Hande et al. [17] | Yes | No | No | ML-based detection, Deep Learning-based detection and entropy-based detection. |
| Yurekten et al. [18] | Yes | No | Yes | Defense against scanning, spoofing, Denial of Service (DoS), sniffer, malware, social engineering and web application attacks. |
| Yungaicela et al. [19] | Yes | No | Yes | Reactive defense (statistics-based detection, ML/DL-based detection, reinforcement learning-based mitigation, adversarial learning) and proactive (Moving Target Defense, Network Function Virtualization, deception, network slicing, blockchain) defense. |
| Our survey | Yes | Yes | Yes | Dynamic traffic filtering, network survivability, Moving Target Defense and honeypot-based intrusion response. |

and where it is not possible, to implement mitigation and prevention techniques. In general, ICSs are configured and designed with static pre-defined policy rules, in order to meet high performance and resiliency requirements in critical operations. In most cases, this is achieved by manually implementing management functions and rules in proprietary Command Line Interfaces (CLI) provided by different industrial vendors devices. Since most of the existing industrial network infrastructures are application-specific and statically deployed, they are unable to support different types of industrial applications with diverse requirements. This calls for a network infrastructure that enables dynamic configuration and interoperability among different industrial applications [10], giving to Software-Defined Networking (SDN) the opportunity to be the key technology in building ICSs. SDN opens up a new horizon of possibilities to ease the management of communication networks based on the concept of separating the control plane (e.g., failure management, topology discovery) and the data plane (e.g., packet routing, forwarding), making the network dynamic and programmable [11]. Nowadays, the integration of SDN into industrial environments is in an early stage of development and requires an extensive testing and validation work [12]. However, SDN have been successfully adopted in IT (e.g., datacenters) and telecommunication (e.g., Wide-Area Networks, 5G mobile networks) environments, and is expected to be useful for the development of intelligent and reliable security solutions in ICSs [13], particularly in the field of intrusion detection and response.

This work aims to analyze the possibilities of leveraging SDN for ICS intrusion response, by systematically reviewing existing literature on related fields. As shown in Table 1, the paper complements the aspects already covered in existing surveys [14–19] by considering and combining aspects not included before in a single survey, such as the use of SDN for developing intrusion response strategies for ICSs. The core contributions of this paper are the following:

- A comprehensive literature review on the applicability of SDN to develop intrusion response strategies in ICS.
- A novel taxonomy to classify SDN-based intrusion response strategies in ICS.
- A discussion of open problems and future trends regarding ICS, SDN and intrusion response.

The rest of the paper is structured as follows: Section 2 provides an introduction to ICSs by analyzing their characteristics, security needs, the main cyberattacks and intrusion response techniques. Section 3 analyzes SDN, the differences with traditional networking, how SDN is used for securing ICS and the main security issues related to the architecture. Section 4 provides a literature review of SDN-based intrusion response in ICS by making a categorization based on intrusion response strategies. Section 5 discusses and compares the articles listed in the previous section. Section 6 identifies and proposes research challenges and future lines. Finally, the paper is concluded in Section 7.

### 1.1. Literature review methodology

For the development of the survey paper in SDN-based intrusion response in ICS, a literature review methodology described underneath has been followed for relevant contributions search. The different steps of the methodology, the search strategy, the keywords used, and the inclusion criteria are described below.

Publications were retrieved through a computerized search of the Compendex and Inspec databases via Engineering Village, IEEE Xplore and ScienceDirect in order to find relevant contributions published in English in the related research area. In addition, internationally recognized technical documents and standards such as RFC, IEC or IEEE documents have been used to obtain specification information for different technologies and protocols.

The survey paper was conducted in an iterative manner. First, a global view of the current state in ICS and SDN security was sought. The search query used for this step was: “SDN” AND “ICS” AND “security” OR “architectures” OR “resilience”. Controlled terms were used to exclude all publications outside the research domain.

Once the key concepts in the current state of ICS and SDN security were understood, the search terms were modified to focus on SDN-based ICS intrusion response and to identify different technologies and approaches to achieve this goal. The search terms in this step included “SDN” AND “ICS” AND “intrusion response” OR “attack mitigation” OR “resilience” OR “survivability”.

After identifying the main approaches in SDN-based ICS intrusion response, a more specific search was carried out in order to find contributions in each identified intrusion response approach or strategy. The following query was used: “SDN” AND “ICS” AND “network survivability” OR “network reconfiguration” OR “MTD” OR “honeypot” OR “traffic filtering” OR “drop” OR “block”. Controlled vocabulary terms were used to exclude publications related to non-relevant research areas. Duplicated papers were also discarded. The bibliographies of all relevant articles and review papers were also searched and taken into account. Selected relevant papers were analyzed in-depth and included in this survey paper.

## 2. Industrial control systems

The nature of ICSs makes traditional IT security systems fail to meet industrial systems security requirements. Although many security mechanisms have been designed for IT systems, the use of these security

**Table 2**
Differences between traditional IT systems and ICSs.

| Criteria | IT systems | ICS |
|---|---|---|
| Applicable domain | Corporate and home environments | Industry |
| Primary function | Data processing and transfer | Physical equipment control |
| Traffic behavior | Unpredictable | Predictable |
| Transmission nature | Aperiodic large packets | Periodic and aperiodic small packets |
| Determinism | Low | High |
| Temporal consistency | Not required | Required |
| Environment | Clean environment | Hostile environment |
| Failure severity | Low | High |
| Primary security requirement | Confidentiality | Availability |
| Updates | Continuous updates | Limited updates capabilities |
| Components lifetime | 3–5 years | 10–15 years or longer |

solutions in ICSs may not be suitable. Table 2 aims to highlight the main differences between the IT and ICS systems characteristics. According to NIST SP 800-82 [1], ICSs differ from traditional IT systems in the following aspects:

- ICSs are intended to control and monitor physical devices and processes. Any problem or incident in the operation of the ICS can cause consequences in the physical world.
- Availability is a priority in ICSs. While IT systems focus on data confidentiality and integrity, ICSs concerns include fault tolerance and human safety. An unexpected system outage is not acceptable.
- ICSs are often time-critical and must be within an acceptable delay. In contrast, IT systems can withstand a certain level of delay or jitter.
- The component lifetime in ICS is very long compared to IT components. ICS components lifetime can be 10–15 years, in some cases even longer.
- Applying security patches are postponed in ICSs due to availability and reliability requirements. These updates have to be extensively tested before being implemented. In contrast, IT systems are updated regularly.
- ICS devices are designed for industrial processes control and in many cases do not have the capability to integrate security mechanisms.

This translates into the occurrence of several number of events related to security breaches [20]. Example of such events include the well-known Stuxnet [21], a malware which was able of modifying the operation of a nuclear plant in Iran, resulting in the failure of centrifuges of a uranium enrichment plant.

### 2.1. Cyberattacks in ICSs

Insufficient security systems in legacy ICSs and an increase in the connectivity of these systems, securing an ICS environment becomes a difficult task. Besides this, cyberattacks have evolved to levels of sophistication, diversity, personalization and intelligence never seen before [22]. As an example, Advanced Persistent Threats (APTs), that remain active for long periods of time without being detected, represent a threat to ICSs [23,24]. Understanding different cyberattack scenarios is very important for developing and deploying ICS intrusion response solutions to meet security, safety, resilience and adaptability needs. The most common attack model is to gain access to the manufacturing zone network in order to compromise the ICS. An attacker can gain control through social engineering or by exploiting vulnerabilities in legacy ICS devices connected to the internet. Most vulnerabilities in ICS occur due to lack of authentication mechanisms, encryption and integrity checks [25], allowing attackers to launch attacks from unauthorized hosts or modifying the content of network packets. Cyberattacks in ICSs can be classified into the following groups [26]:

**Denial of Service (DoS)** is an attack on a computer system or network that causes a service or resource to be inaccessible to legitimate users. It usually results in the loss of network connectivity by excessive bandwidth consumption or overloading the computational resources of the attacked system. Examples of such attacks include SYN flood, UDP flood, ICMP flood, Ping of Death or low-rate DoS attacks.

**Reconnaissance attack** is a process or set of attacks that aims to collect information about devices on a network. Information such as network topology, IP addresses, device names, open ports, etc. can be obtained. Reconnaissance attacks can be performed passively or actively. Examples of attacks include information gathering attacks such as passive reconnaissance (e.g., eavesdropping, passive fingerprinting, sniffing) and active reconnaissance (e.g., nmap, active fingerprinting).

**Man-in-the-Middle (MitM) attack** allows to an attacker sit in the middle of a communication. The attacker is able to intercept communication packets in order to read, modify, inject commands or interrupt communication flows.

**Injection attack** refers to the use of techniques through which an attacker, impersonates a different entity by injecting falsified data in a communication. For example, an attacker could inject false sensor reading or control commands from a compromised device. Examples of attacks include IP spoofing, MAC flooding, ARP spoofing, data injection, command injection or packet alteration attacks.

**Replay attack** consists of retransmitting a packet previously seen on the network. For example, an attacker could retransmit lower or higher temperature readings to prevent safety systems from alerting ICS operators. Such attacks are often difficult to detect.

**Physical Process attacks** aim to alter the industrial process by performing attacks that physically affects ICS components. Examples of such attacks include link destruction, stealth attacks by introducing small perturbations into the system process and direct damage attacks.

### 2.2. Cybersecurity demands in ICSs

Industrial networks are built upon heterogeneous and diverse network devices and protocols that support traffic filtering or access control protocols, as well as physical redundancy in order to enable network survivability. Several aspects related to cybersecurity requirements in ICS are detailed below.

#### 2.2.1. Redundancy and network survivability

A resilient system has to be flexible enough to withstand changes or modifications and return to a stable mode of operation after an incident occurs. In order to achieve high availability and guarantee the delivery of all packets in an industrial communication, network redundancy becomes the most accepted technique. Network redundancy consists of maintaining alternative communication paths in case the primary one fails, reconfiguring the network by selecting an alternative path and avoiding an interruption.

There are several protocols that enable redundancy control and which differ from each other in the supported topologies and in the recovery time [27]. Rapid Spanning Tree Protocol (RSTP, IEEE 802.1w) supports different types of topologies, but the failover time depends on the location of the failure and different vendors individual implementation and does not guarantee a deterministic failover time.

Another well-known protocol in the industrial domain is the Media Redundancy Protocol (MRP, IEC 62439-2). It provides a deterministic failover behavior and the recovery time is usually lower than RSTP, but it is only supported in ring topologies. In addition to these two mentioned protocols, it is worth to mention the Parallel Redundancy Protocol (PRP) and the High Availability Seamless Redundancy (HSR), both defined in IEC 62439-3. They provide a zero recovery time in case a failure occurs.

#### 2.2.2. Timeliness and predictability

Industrial control network traffic is by nature repetitive and predictable, since most of the traffic is generated by automated processes [28,29]. Timeliness requirements in ICSs depend entirely on each individual application and the latency, timing, synchronization and predictability of network traffic may vary for each use case. As an example, closed-loop industrial control applications often require a deterministic communication, with a low-delay and an acceptable jitter value. In addition, time synchronization is critical for ICS security as is useful for log correlation, quality of service and authentication/authorization mechanisms (NIST SP 1500-201 [30]). Network Time Synchronization (NTP, RFC 5905) and Precision Time Protocol (PTP, IEEE 1588) can provide time synchronization among industrial devices. In order to prevent malicious modifications of network time, it is recommended the use of the secure versions of NTP (RFC 8915) and PTP (enabling 'time, length and value' integrity checking as defined in IEEE 1588).

#### 2.2.3. Network monitoring and management

Network visibility and traffic management are fundamental concepts to monitor unexpected behaviors and avoid undesired interruptions in an industrial network. In ICS networks, monitoring capabilities are typically implemented through the use of Switch Port Analyzer (SPAN), also called port mirroring, a technique that copies incoming Ethernet frames on a switch and forwards them through an outgoing port (NISTIR-8219 [31]). There are other alternatives that offer more advanced network monitoring capabilities [32]. Protocols such as the encrypted and authenticated Simple Network Management Protocol version 3 (SNMPv3, RFC 3410) can be used to gather real-time basic network usage data such as bandwidth as well as device readings such as CPU load or memory usage. For a more comprehensive and specialized network traffic reporting solution, NetFlow can be used as a packet aggregator into flows, for further processing and analysis (e.g., incident detection, network performance or quality of service verification). However, NetFlow is a widely used Cisco-only proprietary network protocol and in response, Internet Protocol Flow Information Export (IPFIX, RFC 7011) was created as a common, open and universal standard of network flow information export. Another open technology worth mentioning is sFlow (RFC 3176), which enables exporting network packets in a mandatory sampling mode, but does not contain a packet-to-flow aggregator itself.

In addition to monitoring protocols mentioned above, network management can be achieved via NETCONF (RFC 6241) protocol, which is an advance over vendor-specific proprietary configuration platforms and tools. NETCONF enables to manipulate and install network devices configurations on top of a Remote Procedure Call (RPC) layer [33].

#### 2.2.4. Incident and fault detection

Compared to IT systems, ICS network traffic is deterministic by nature and this can be leveraged to support network monitoring for unexpected behavior or fault detection. Understanding the normal state and operation of an ICS network is often a prerequisite for many Intrusion Detection Systems (IDSs) [14]. Signature-based IDSs monitors all packets traversing the network and compares them against a ruleset of attack signatures of known malicious threats. The main advantage of signature-based IDSs is the high accuracy, fast pattern matching and low false positive ratio. The main disadvantage is the lack of ability of zero-day attacks detection. If the signature of an attack is not registered, the attack will not be detected. Some industrial equipment vendors have already incorporated signature-based IDSs for a many industrial protocols such as DNP3 or Modbus TCP.[1] Another approach to detect incidents are the anomaly-based IDSs, where thresholds are defined to determine whether a monitored behavior is licit or illicit. The main advantage of an anomaly-based IDS is the ability to detect zero-day or previously unseen attacks. Concerning to detection rate, anomaly-based detection techniques result in more false positives than signature-based ones and the efficiency depends on the precision in defining thresholds. For an in depth analysis of the performance of different Machine Learning (ML) approaches (supervised, unsupervised, semi-supervised and reinforcement learning) in ICS, refer to the work by Umer et al. [34].

#### 2.2.5. Incident prevention and mitigation

Network architectures play a fundamental role in preventing security issues in ICSs. Network separation between IT and OT systems, as detailed in industry-recognized architectures such as the Purdue model or the IEC 62443 standard, is difficult to achieve with new emerging trends such as Industrial Internet of Things (IIoT) or cloud computing [35]. As an example, sensor data can be retrieved in the lowest level of the ICS and directly sent to the cloud for further processing (e.g., predictive maintenance), not complying with segmentation rules defined in previously mentioned reference architectures. Apart from authentication, authorization and integrity checking mechanisms, ICS security relies on enforcing physical and logical access restrictions to cyber-components [36]. On the one side, network segmentation can be achieved by physically using different switches or logically implementing Virtual Local Area Network (VLAN, IEEE 802.1Q) configurations. On the other side, network isolation can be implemented by using Access Control Lists (ACLs), firewalls or Deep Packet Inspection (DPI) devices. Network isolation devices can be configured to enforce policies permitting only allowed flows or packets and blocking everything else.

Network segmentation and isolation does not protect against all ICS threats and manually handling network incident alerts may not be the most efficient way. Intrusion Prevention Systems (IPSs) that automatically deploy countermeasures to mitigate attacks can cause a disruption in the network and an extensive testing and validation work is required before the deployment into mission-critical networks.

### 2.3. Shortcomings of traditional ICS protocols and tools

In most cases, industrial control networks are deployed and managed using pre-configured static policies to meet high availability and reliability needs. Often, to satisfy previously mentioned ICS security demands, low-level manual configurations are required by using vendor-specific configuration and management tools. Moreover, with the evolution of industrial networks into dynamic and increasingly interconnected heterogeneous devices [37,38], those security requirements are difficult to achieve without centralized network management tools. Because of this, SDN is becoming a promising technology for the operation and development of tailored industrial security tools [12,39], providing a global view over the entire network, increasing network programmability and allowing a dynamic and centralized network management.

## 3. Software-defined networking

Software-Defined Networking (SDN) refers to a new approach for network programmability, that is, the capacity to initialize, control, change, and manage network behavior dynamically via open interfaces (RFC 7426). SDN enables the central and intelligent management and

[1] https://github.com/digitalbond/Quickdraw-Snort.

**Table 3**
Traditional networking vs. Software-Defined Networking.

| Criteria | Traditional networking | Software-Defined Networking |
|---|---|---|
| Control and data planes | Coupled | Decoupled |
| Control plane | Decentralized | Centralized |
| Network supervision | Limited view | Global view |
| Network management | Changes implemented separately at each device | Easier with the help of the controller |
| Network programmability | Low | High |
| Extensibility | Static and inflexible | High |
| Maintenance cost | Higher | Lower |

control of individual hardware components with the help of software [40,41]. For this purpose, the control plane is detached from the network devices and centralized in an external entity. The data plane, on the other hand, remains in the network devices and its function is reduced to packet forwarding. Within the operation of SDNs, an ecosystem of platforms, protocols and applications can be found. Currently, the Open Networking Foundation (ONF) is in charge of promoting SDN and its adoption, as well as standardizing and managing the OpenFlow [42,43] standard. As represented in Fig. 1, the SDN architecture is divided into three different planes [44].

**Data plane.** Implements forwarding decisions made by the control plane. It is composed of traffic forwarding and processing devices (e.g., switches, routers). These devices use specific protocols that allow communication with controllers to establish flow rules and share traffic statistics. The most widely used protocol in SDN is OpenFlow, although there are some others such as NETCONF (RFC 6241).

**Control plane.** It is used to grant logic to the data plane. In the control plane, SDN controllers can be found. The ecosystem of a controller can be divided into three areas:

*Northbound interface:* It is used for the interaction between external user-developed applications and the controller. Programmers can create, modify or delete flow entries and retrieve network statistics.

*Controller core:* Also known as the SDN controller, it is responsible for the interaction between the southbound and northbound interfaces. Applications can be developed that offer functionalities.

*Southbound interface:* Composes the lowest layer of the stack of an SDN controller and is used to interact with the forwarding devices in the data plane. Controllers typically support multiple protocols for the southbound interface, including the OpenFlow protocol.

**Application plane.** Consists of end-user business applications that use SDN services through the northbound interface. Allows services and applications to simplify and automate the tasks of configuring, provisioning and managing new services in the network.

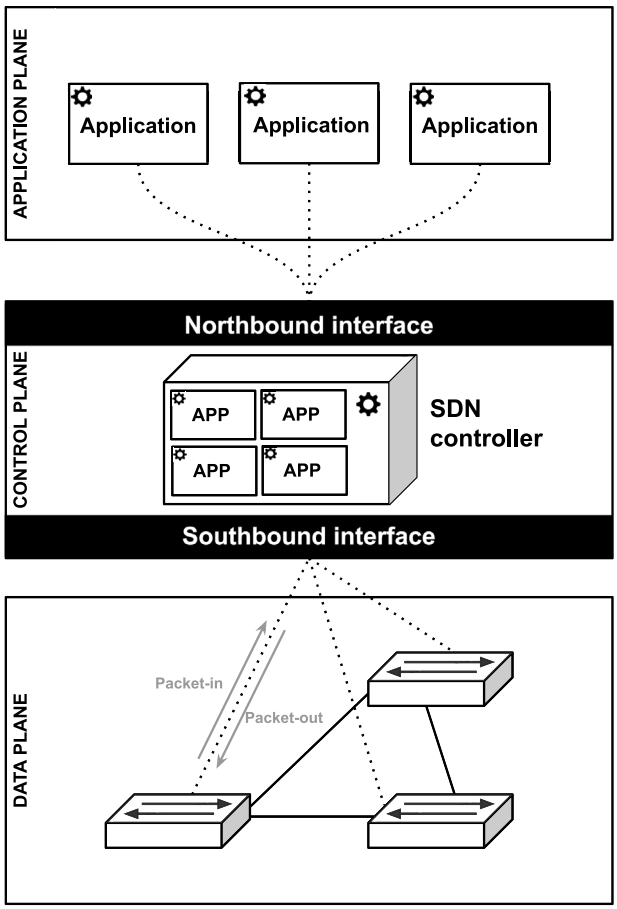



**Fig. 1.** SDN architecture.

### 3.1. SDN for securing ICSs

Traditional network architectures are not optimized to meet current and future networking requirements of ICSs that demand more flexibility, heterogeneity and interoperability in dynamic environments [12]. In traditional networking, to make any kind of change (topology, rules, protocols, etc.), the network administrator must manually configure each of the devices, increasing complexity and possible errors in the network operation. Because of this, network architectures are evolving towards dynamic and programmable topologies such as SDN. The main differences between traditional networking and SDN are highlighted in Table 3. From a ICS network security, reliability and interoperability point of view, SDN technology brings benefits mainly due to the following characteristics:

- **Network-wide visibility** allows users and applications to collect traffic information or monitor network status in real-time. This information can be used to develop applications to detect and respond to attacks, as well as to define security policies or to ensure high availability of the industrial process avoiding interruptions.
- **Network programmability** allows developing and integrating applications that interact with the network. A centralized control plane and well-defined communication interfaces, make the network operation more dynamic and scalable.
- **Dynamic flow control** allows to decide the behavior of the network traffic in a centralized way. Among many other things, it is possible to modify the traffic route, decide whether to drop or forward packets, etc.

Next, the potential applications of SDN in ICS are discussed, with a focus on network monitoring, security, timeliness, reliability, interoperability and management. A summary of the main topics discussed in this section is provided in Table 4.

#### 3.1.1. Network monitoring and security

From a network monitoring perspective, the global view over the network offered by SDN provides detailed understanding of what is happening in the network at any given moment. The controller, among many other things, is capable of monitoring and handling

**Table 4**
Summary of how SDN covers different ICS needs.

| Monitoring and security |
|---|
| - Network traffic and status monitoring through a network-wide visibility.<br>- Network-wide intrusion detection using gathered traffic and network status data (real-time or offline mode).<br>- Intrusion response due to a dynamic, real-time and centralized decision-making.<br>- Network-wide intrusion prevention with security policy enforcement (e.g., access control, network segmentation, software-defined firewalls). |
| **Timeliness and reliability** |
| - In advance backup flow rules definition in flow tables to avoid table-misses and delay introduction.<br>- Network interruption identification by querying the data plane or handling port status changes notifications from switches.<br>- Topology-agnostic, fast and deterministic recovery mechanisms to avoid unexpected interruptions and fulfilling high availability requirements (e.g., OpenFlow Fast-Failover groups, control plane-based network reconfiguration).<br>- Dynamic flow control to meet real-time and quality of service related requirements (e.g., load balancing, avoid overloaded devices or links, priority traffic routing). |
| **Interoperability and management** |
| - Abstraction layer in the data plane homogenizing low-level and vendor-specific network configuration interfaces and tools.<br>- Ease the interoperability between different ICS ecosystems by using open protocols to manage forwarding tables.<br>- Improves manageability and allows faster innovation cycles.<br>- Support for multiple industrial network traffic types, topologies and protocols. |

switches ports and links status changes, handling table-misses (packets that have not matched any flow rule), querying flow rules statistics (e.g., packet/byte count, flow rule duration) or sending network packets to the controller by encapsulating them into *Packet-In* packets before or during a forwarding decision is made at the data plane [45]. Intrusion detection can be developed leveraging monitoring strategies mentioned above. As an example, network packets can be sent to the controller for attack signature matching, anomaly detection or deep packet inspection before the forwarding is made. For a non-real-time intrusion detection, packets can be mirrored to the controller for further packet-by-packet analysis or packets-to-flows aggregation.

Compared to previously analyzed traditional network isolation or segmentation techniques, SDN can provide static and dynamic traffic filtering rules enforcement to prevent and/or mitigate attacks [46]. Whitelisting has been advocated by industry as an effective method for securing industrial networks (NIST SP 800–82) and SDN can be used to easily install, update and remove static filtering policies allowing only authorized or whitelisted communications. This way, limiting communications to only authorized ones, ICS attack surface is reduced. In addition to static filtering, leveraging different monitoring strategies mentioned above for a constant network checking, a centralized intrusion detection and flow management, the network can be dynamically reconfigured to mitigate attacks in SDN. An automated intrusion response could cause adverse consequences in the operation of ICS and an extensive testing is often required before the deployment of these systems into production networks.

#### 3.1.2. Network timeliness and reliability

From a network resilience point of view, an SDN controller can offer failure response by installing pre-defined forwarding rules that serve as backup in case a network reconfiguration is needed. For this, network monitoring modules can be developed on top of the controller to gather network status data (e.g., ports or links status) in order to detect failures and reconfigure the network by using predefined backup forwarding rules or by installing new ones [47]. Instead of the controller querying the data plane to obtain information about the state of the network, network switches can be configured to notify ports and links status changes to the controller, allowing to the controller react and avoid end-to-end connectivity failures. For a faster recovery time, a feature available since OpenFlow version 1.1 named OpenFlow Fast-Failover groups [48], can be used to detect and overcome port failures. The motivation of Fast-Failover groups is that the reconfiguration process occurs in the data plane by predefining a list of port monitoring and actions buckets, preventing the controller from querying the data plane or handling status changes. Apart from self-healing, OpenFlow groups can be leveraged for different purposes such as load-balancing by using the OpenFlow Select group or to improve the performance avoiding forwarding rules duplication for a set of similar flows by using the OpenFlow Indirect group.

Compared to traditional redundancy control protocols, SDN can provide topology-agnostic network recovery mechanisms, as opposed to MRP which only supports ring topologies. Moreover, SDN can also implement deterministic and faster recovery times than RSTP by leveraging network-wide awareness and in advance flow rules definition.

#### 3.1.3. Network interoperability and management

SDN overcomes low-level vendor-specific network configuration interfaces by introducing an abstraction layer for the data plane, enabling network operation via open interfaces and the support for multiple industrial topologies and protocols. The use of open protocols to manage forwarding tables and a centralized control plane eases the interoperability between different ICS ecosystems, improving manageability and allowing faster innovation cycles compared to traditional legacy industrial networks. However, a major challenge is the interoperability of legacy ICS devices within the SDN ecosystem. ICS components lifetime is usually longer compared to IT systems and the transition from traditional industrial networking to SDN may not always be possible due to availability requirements and lack of support for required protocols. Authors in [49] analyze different hybrid SDN approaches that combine legacy forwarding devices and programmable SDN switches in networks that a fully SDN deploy is not possible.

### 3.2. Security issues in SDN

Like any other network technology, the protocols, devices and applications participating in an SDN network can be the target of an intentional security attack or an accidental incident due to misconfiguration or misuse [50]. Although SDN can offer several security related advantages over traditional networking, potential SDN vulnerabilities must be addressed to meet ICS security requirements. In fact, the ONF presented a manuscript [51] that analyzes different SDN related security issues and proposes guidelines for designing secure SDN solutions capable of withstanding threats. All layers and communication interfaces of the SDN architecture represented in Fig. 1 are susceptible to specific security attacks or vulnerabilities introduced due to configuration or operation errors. As the different planes of the SDN architecture are interconnected through communication interfaces (northbound and southbound interfaces), an attack launched in one of the planes can affect other parts of the architecture. The following is an overview of the main vulnerabilities that can be found in the different planes of the SDN architecture [15]:

**Application plane vulnerabilities.** Vulnerable or buggy applications, lack of applications authentication and authorization, malicious applications deployment.

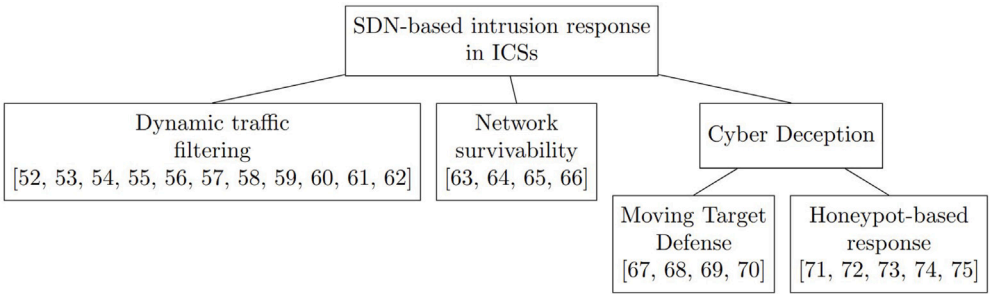



**Fig. 2.** Taxonomy of SDN-based intrusion response strategies in ICSs used the survey.

**Control plane vulnerabilities.** Single point of failure due to control plane centralization, vulnerable SDN controllers, vulnerable or buggy applications, SDN controller overload, malicious control packet injection, lack of network devices authentication/authorization.

**Northbound and Southbound interfaces vulnerabilities.** Most of the vulnerabilities occur due to the use of communications that are vulnerable to attacks such as the use of unencrypted communications, vulnerable protocols or errors in the configuration.

**Data plane vulnerabilities.** Vulnerable or legacy ICS protocols, vulnerable network devices, devices overload, malicious traffic injection, lack of network devices authentication and authorization.

## 4. SDN-based intrusion response in ICSs

Dynamic network behavior modification, continuous monitoring and centralized decision-making capabilities are recurrent resources on many SDN security related research papers. The SDN architecture provides tools to develop new security mechanisms in a more effective way, enhancing programmability and knowledge gathering capabilities compared to traditional networking. In the field of ICS security, the applicability of SDN has focused mainly on the development of different intrusion detection and response solutions. Focusing on intrusion response, this section provides a review of the state-of-the-art in the field of SDN-based intrusion response in ICSs. Fig. 2 defines the taxonomy followed in this review, whereby publications are classified based on the intrusion response strategy followed. The intrusion response strategies can be categorized into the following three groups:

1. *Dynamic traffic filtering:* Compared to traditional networking, SDN eases the dynamic definition of security policies by using traffic statistics or network status information. These approaches implement different detection techniques (e.g. deep packet inspection, signature pattern matching, anomaly detection, Machine Learning) and leverage SDN to dynamically drop packets or block network flows to avoid reaching the destination device.
2. *Network survivability:* Consists of maintaining network operation when one or more network components fail or suffer an attack. Leveraging the global view of the network that SDN provides, network status monitoring modules are developed to detect a failure and reconfigure the network to avoid an interruption.
3. *Cyber deception:* The main goal of cyber deception techniques is to confuse attackers by providing false information on network resources, increasing their uncertainty in the decision-making and allowing knowledge gathering by the defender. Cyber deception approaches in SDN-based ICSs can be classified into two groups:

    (a) *Moving Target Defense:* The network configuration is randomized over time (e.g. IP addresses, network paths), reducing the attack surface and providing false information to the attacker.

    (b) *Honeypot-based response:* Fake industrial processes and devices are deployed so that attacks strike these systems instead of production ones. Keeping the attack active on fake systems allows intelligence to be gathered without affecting production systems.

Following this taxonomy, the section has been divided into three subsections, each one corresponding to a different intrusion response strategy. Table 6 summarizes, compares and classifies all SDN-based intrusion response approaches for ICS analyzed in this section. It includes:

- **Intrusion response** section by including:
  - **Cyber-threats** against the proposed solutions are tested during the experimental phase.
  - The adopted **Response approach** based on different response strategies defined in Table 5.
- The SDN **Controller** used in the solution.
- The industrial **Protocol** considered in the solution.
- **Deployment** section by including:
  - A tick if **NFV** is used in the solution.
  - Equipment, tools and technologies that are used for **Testbed** deployment.
  - Intrusion **Detection module**s deployment location based on the SDN architecture planes.
  - Intrusion **Response module**s deployment location based on the SDN architecture planes.

### 4.1. Dynamic traffic filtering

To prevent malicious network traffic from reaching its destination, traffic filtering becomes a very effective technique. Sainz et al. [52] present a solution for detecting and mitigating payload alterations. In the experiment, a bottle filling plant is simulated in which the SCADA server and the PLC communicate through the Manufacturing Message Specification (MMS) protocol. No forwarding rules are installed on the switches, forcing all packets to go through the SDN controller. A machine is placed between the PLC and the SCADA server. The request packages from the SCADA server are intercepted by this machine and the payload altered. The entire payload signature matching process is performed on the SDN controller and the malicious packets are dropped to prevent from reaching the PLC. The main drawback of the proposed solution is that the signature matching process is only made with control packets flowing from the SCADA server to the PLC. Sensor readings flowing from the PLC to the SCADA server are not considered in the attack detection process. Moreover, processing all packets in the SDN controller can lead to controller overload problems. In addition to packet alteration mitigation mechanism, ICMP traffic rate is limited in order to mitigate ICMP flood attacks. Physical industrial equipment is used in conjunction with Emulab [76] for testbed deployment.

**Table 5**
Categorization of different intrusion response strategies of the literature.

| Intrusion response category | Response approach | Short identifier | Description |
|---|---|---|---|
| Dynamic traffic filtering | Drop packet | DROP-PKT | Packets are dropped according to rules or matches. |
| | Block flow | BLOCK-FLOW | Flows are blocked according to rules or flow matches. |
| Network survivability | Path reconfiguration | PATH-RECONF | Change flow path in order to avoid an interruption. |
| | Limit traffic rate | RATE-LIMIT | Limit traffic rate to avoid network resources overload. |
| | Packet sanitation | SANITIZE-PKT | Replacing network packets affected by an attack. |
| Cyber deception | IP randomization | IP-RAND | Proactively or reactively randomize IP addresses. |
| | Path randomization | PATH-RAND | Proactively or reactively randomize flow paths. |
| | Send to honeypot | SEND-HONEY | Send traffic to a honeypot or honeynet. |

**Table 6**
Classification of SDN approaches for intrusion response in ICS.

| | | [52] Sainz et al. | [53,54] Piedrahita et al. | [55] Brugman et al. | [56] Ndonda et al. | [57] Tsuchiya et al. | [58] Melis et al. | [59] Rivera et al. | [60] Radoglou et al. | [61,62] Holik et al. | [63] Genge et al. | [64] Sándor et al. | [65] Jin et al. | [66] Aydeger et al. | [67] Almusaher et al. | [68] Silva et al. | [69] Ndonda et al. | [70] Chavez et al. | [71] Antonioli et al. | [72] Petroulakis et al. | [73] Salazar et al. | [74] Bernieri et al. | [75] Du et al. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Dynamic filtering | | | | | | | | | Survivability | | | | MTD | | | | Honeypot | | | | |
| Intrusion response | **Cyber-threats** | | | | | | | | | | | | | | | | | | | | | | |
| | Denial of Service | ■ | □ | ■ | ■ | □ | ■ | □ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | ■ |
| | Reconnaissance | □ | □ | ■ | ■ | ■ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | □ | □ |
| | Man-in-the-Middle | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ |
| | Injection | ■ | ■ | ■ | ■ | □ | ■ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ |
| | Replay | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ |
| | Physical | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Not specified | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | ■ | □ | ■ | ■ | □ | ■ | ■ | □ | □ |
| | **Response approach** | | | | | | | | | | | | | | | | | | | | | | |
| | DROP-PKT | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | BLOCK-FLOW | □ | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | PATH-RECONF | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | RATE-LIMIT | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | SANITIZE-PKT | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | IP-RAND | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | □ | □ | □ |
| | PATH-RAND | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | □ | □ | □ | □ | □ |
| | SEND-HONEY | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ |
| Controller | OpenDaylight | ■ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | ■ | □ | □ | □ |
| | ONOS | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ |
| | RYU | □ | □ | □ | □ | □ | □ | ■ | ■ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ |
| | POX | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | ■ | ■ | □ | □ | □ | □ | □ | □ |
| | Other | □ | □ | □ | □ | ■ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Not specified | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | □ | □ | ■ | ■ |
| Protocol | MMS | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Modbus | □ | □ | ■ | ■ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | □ | □ | □ | □ | ■ | □ |
| | EtherNet/IP | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | ■ | □ | □ |
| | OPC UA | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | DNP3 | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Not specified | □ | □ | □ | □ | □ | □ | ■ | □ | ■ | ■ | ■ | ■ | □ | ■ | □ | □ | ■ | □ | ■ | □ | □ | ■ |
| Deployment | **NFV** | □ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ |
| | **Testbed** | | | | | | | | | | | | | | | | | | | | | | |
| | Physical equipment | ■ | □ | □ | □ | □ | □ | ■ | □ | ■ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Mininet | □ | □ | □ | ■ | □ | ■ | ■ | □ | ■ | ■ | ■ | □ | ■ | ■ | ■ | ■ | □ | □ | □ | ■ | ■ | □ |
| | MiniCPS | □ | ■ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | ■ | □ | □ |
| | Virtual machines | □ | ■ | ■ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | □ | ■ | ■ | □ | □ |
| | Other | ■ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Not specified | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ |
| | **Detection module** | | | | | | | | | | | | | | | | | | | | | | |
| | Application plane | □ | ■ | ■ | ■ | □ | □ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | □ | □ |
| | Control plane | ■ | □ | ■ | □ | ■ | □ | □ | □ | ■ | ■ | □ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Data plane | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | No detection | □ | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ |
| | **Response module** | | | | | | | | | | | | | | | | | | | | | | |
| | Application plane | □ | □ | □ | □ | □ | □ | □ | ■ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| | Control plane | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | ■ | ■ | ■ | ■ |
| | Data plane | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ■ | ■ | ■ | ■ | ■ |

Piedrahita et al. [53] present an SDN based incident response approach in ICS. The authors describe a scenario where the SDN controller is used together with an IDS deployed in a NFV infrastructure to mitigate attacks launched from a compromised water level sensor. The IDS receive a copy of the measurements made by the sensor, identifying differences between actual and estimated values by using a mathematical model. If the difference is greater than a given threshold, the IDS notifies the SDN controller, deploying response actions. The SDN controller discards anomalous measurements by exchanging them for estimated measurements, ensuring the continuous operation of the real industrial process. The results show that the system is capable of detecting and mitigating attacks early, ensuring the safe operation of the process. This work extends MiniCPS emulation software [77] in order to include a SDN/NFV incident response mechanism. Based on the previously mentioned work, same authors [54] presented a variant where the IDS is deployed in a cloud-based infrastructure. In addition to mitigating attacks launched from compromised sensors, attacks launched from compromised PLCs and SCADA servers are also considered. This mechanism is also based on the analysis of sensor and control packets, replacing them with expected values if an attack is detected by the IDS.

Brugman et al. [55] present a cloud-based IDPS to detect and mitigate threats in ICS by using SDN. The architecture is composed of three security modules. First, the SDN controller acts as a firewall, filtering traffic based on IP address and protocol. Second, a signature-based IDPS deployed on a cloud server using NFV. Finally, a deep packet inspector also deployed in the cloud with NFV. When a packet arrives to a switch, it is forwarded to the cloud for analysis. If an intrusion is detected, the traffic is dropped. Otherwise, the traffic is forwarded to its destination. The architecture is tested in a virtual energy management system.

The need to not negatively affect the real-time performance of the ICSs makes it necessary to dynamically apply and update filters directly on the forwarding devices. Ndonda et al. [56] present an SDN-based two level IDPS for ICS networks. The first level is implemented using P4 [78] programming language on top of network switches and consists of an allowlist-based filter for the Modbus protocol. If a packet is not matched against any allowlist entry, it is forwarded to the second level of the IDPS. The second level is a security engine running on a dedicated host that performs Deep Packet Inspection (DPI) with the packet analyzer tool Bro [79]. If an intrusion is detected at the second level, the allowlist of the first level is updated, so that next time the intrusion will be detected in the first one.

Tsuchiya et al. [57] propose an SDN-based firewall designed for securing ICSs. The traffic filtering approach is divided into three modules: transparent firewall, temporal filtering and spatial filtering modules. Firstly, transparent firewall consists on forwarding or blocking flows in the data plane based on IP and Ethernet header fields values by installing rules on the forwarding devices. Secondly, temporal filtering consists of using the Trema [80] SDN controller to keep the access rules updated on the switches, dropping packets that do not match those rules. Finally, spatial filtering consists of rewriting the access rules based on the OPC UA application level authorization. First, OPC UA access control server requests OPC UA applications about their trust list and network interfaces. After that, an access control list is generated and the SDN controller updates the forwarding rules of the switches.

Several articles focus on making security policies definition easier for network administrators. Melis et al. [58] base their proposal in the usage of four security modules developed on top of the SDN controller that help network administrators implementing security policies in industrial networks for mitigating attacks. The first two modules called "Live Capture" and "Packet inspector", allow the network administrator to decide to accept or reject a packet flow. The third module called "Reachability" is used to verify the security actions taken by the network administrator. This is performed by using the NetPlumber policy checking tool [81]. Finally, the "To WayPoint" module consists of verifying the communication between two nodes.

Rivera et al. [59] propose a security mechanism for defending robotics systems. The architecture is composed of two main modules. First, an anomaly detection engine deployed outside the SDN controller that performs a pattern matching process to detect network attack flows and logs. Second, the policy engine, an abstraction layer deployed in the SDN controller translates domain specific policies to SDN understandable language. This module allows users to define rules based on source and destination fields. By default, allow, drop, log, and copy actions are implemented. The performance of the architecture is tested in a physical and in an emulated environment using Mininet [82] and the Gazebo [83] robot simulator.

Some articles implement Machine Learning (ML) models to predict attacks or anomalous behavior in the network traffic data in order to respond to attacks. In the approach presented by Radoglou et al. [60] an intrusion detection and mitigation for a DNP3 SCADA system is presented. The proposed system is divided into two blocks, one module for the attack detection and another module for the attack mitigation, both developed using the northbound interface of the SDN controller. The IDPS is composed of two ML models. First, a supervised ML model for detecting whether a DNP3 flow is related to a specific DNP3 attack. Second, an unsupervised ML model, trained only with attack-free flow data, that consists of a deep neural network with an encoder/decoder architecture called auto-encoder. Finally, the response module notifies the SDN controller to drop a flow in case an attack is detected.

Similarly, Holik et al. [61] present an ICS protection system based on network traffic filtering with two feed-forward neural networks, the first one for layer 3 protocols and the second one for layer 4. The neural networks, deployed in the SDN controller, analyze the incoming flow characteristics and assigns a flag to each network flow indicating the response to apply. The system is tested with ICMP DoS, TCP DoS and UDP DoS attacks, showing that the use of neural networks for traffic filtering in ICS is a feasible and effective solution. In [62], the same authors provide a more detailed explanation of the development and internals of the neural networks.

### 4.2. Network survivability

To ensure the survivability of ICSs and avoid unexpected outages, network reconfiguration becomes an effective way to guarantee availability and end-to-end communications in industrial networks. Genge et al. [63] present an SDN-based network optimization solution for ICS that reconfigures the network topology when a link failure occurs. As shown in Fig. 3, network status is constantly monitored (e.g., switch port status) in order to detect changes in the network. In response to these changes, a new flow distribution is calculated according to the results given by an optimization problem. The new network configuration is implemented with the SDN controller by installing static forwarding rules on the switches. The solution is tested in single and multi-domain SDN networks using the FloodLight [84] SDN controller.

Sándor et al. [64] design an attack detection algorithm and an optimal intervention strategy that meet the communication and security needs of industrial applications. The detection algorithm performs an anomaly detection process. With continuous flow monitoring and anomaly detection data, a localization algorithm is used for identifying the affected segments of the network. For isolating the critical segments, a new network configuration is calculated and later notified to the SDN controller that installs new forwarding rules on the switches. The architecture is tested in a physical robotic control system and in an emulated environment, demonstrating that the solution can be used in ICSs, enabling survivability.

Network reconfiguration techniques are also widely applied in smart grids to avoid undesired interruptions. Jin et al. [65] propose the utilization of the SDN controller for dynamic self-healing in smart grids.

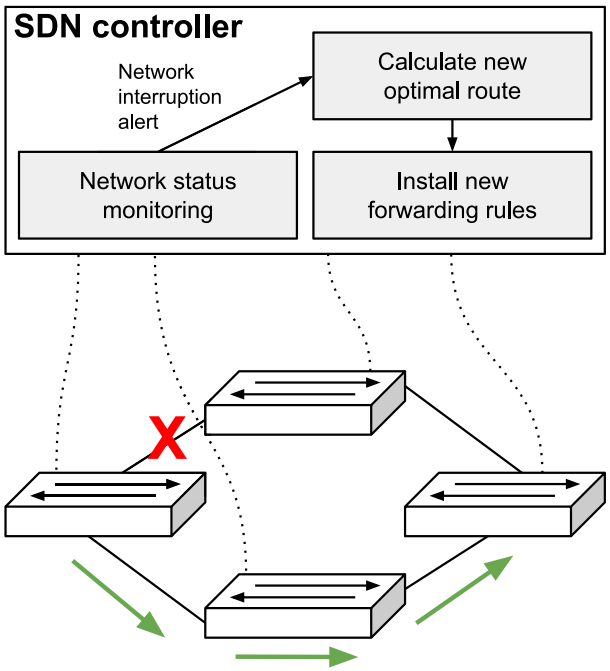



**Fig. 3.** Example of SDN-based network reconfiguration.

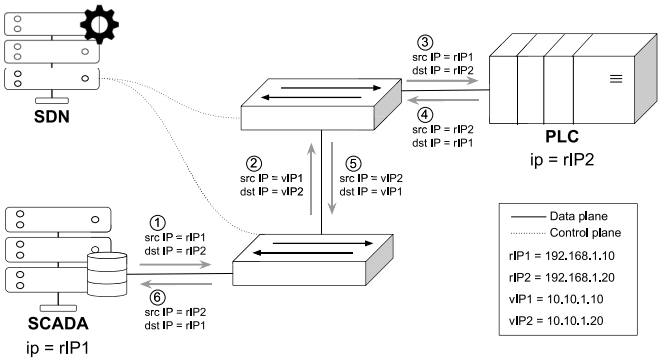



**Fig. 4.** SDN-based IP-Hopping in SCADA-PLC communication. The real IPs (rIP) remain unchanged on the devices, whereas the virtual IPs (vIP) are randomly assigned to each device and translated into rIP before reaching their final destination.

The goal of the proposed architecture is to ensure critical communications when switches or links are compromised. For the test scenario, the SDN controller is used to create forwarding rules for communication between the control center and the energy storage device. A simulated link destruction attack is performed between the control center and the energy storage device. Each time the network topology is modified, an update interruption arrives to the SDN controller. Realizing the compromise of the link, a new shortest path is calculated with Dijkstra's [85] algorithm. DSSnet [86] is used to emulate the network and evaluate the solution.

Similarly, Aydeger et al. [66] present a solution for avoiding service interruption in smart grids when a wired link fails. In order to solve the problem, wired and wireless connections are combined. When a wired connection fails, an event is triggered in the SDN controller. This event performs a change from the wired connection to the wireless one, ensuring the correct operation of the network. The test scenario is deployed with Mininet in combination with the ns-3 [87] network simulator.

### 4.3. Cyber deception

Cyber deception techniques are designed to mislead, disrupt, induce uncertainty and gather information about an attack [88]. In the field of SDN-based ICS, cyber deception techniques can be classified into two groups: (1) Moving Target Defense and (2) Honeypot-based response.

#### 4.3.1. Moving target defense

The static nature of ICS networks gives an advantage to attackers, allowing to explore vulnerabilities before performing the attack [89]. Due to this problem, an approach named Moving Target Defense (MTD) has emerged as a solution for static systems. Although there is no standard definition of MTD, it can be defined as a constantly changing system that moves or reduces the attack surface making it difficult for attackers to exploit. As detailed in MTD-related surveys [90–92], SDN has become a powerful framework to develop novel MTD techniques. Research has focused on developing techniques that attempt to make a network unpredictable by randomizing some industrial network attributes or configurations over time. There are three techniques: (1) IP randomization, (2) Path randomization and (3) Hybrid approaches.

**IP randomization:** Also known as IP-hopping, is a defense strategy that consists of modifying the IP addresses of network packets randomly. This makes it difficult for an attacker to see and identify active network devices, distorting the results of the reconnaissance phase of an attack. Almusaher et al. [67] aim to demonstrate the feasibility of applying MTD to ICS. For this purpose, IP randomization technique is implemented in an emulated industrial environment. This method assigns each device a new random virtual IP every certain fixed time intervals, without modifying hosts configuration and being transparent to them. The SDN controller is the responsible for installing forwarding rules on the network switches in order to perform IP translations. Fig. 4 shows an implementation example of the IP randomization framework in a simple SDN-based industrial network. In terms of performance, the work concludes by demonstrating the feasibility of applying IP randomization in ICS.

**Path randomization:** Also known as route mutation or multipath routing, was first proposed in the 1970s and has been successfully applied in different types of networks [93]. This security mechanism consists of varying and changing the flow routes, making the attack surface more difficult to predict. The article published by Silva et al. [68] propose a defense strategy based on path randomization to improve ICS security. To prevent all traffic from going down the same path, their strategy disperses traffic over multiple paths to defend against unauthorized traffic interceptions. This traffic dispersal is done by programming OpenFlow switches and installing static and dynamic forwarding rules. The dynamic rules are based on the hard-timeouts offered by the OpenFlow protocol, allowing the expiration of the rules after a certain period of time. With this as a basis, traffic is sent from a particular path until the dynamic rule expires. A new dynamic rule is then assigned so that the traffic continues along a different path. The major drawback of this solution is that when a rule expires, before continuing forwarding traffic, the switch has to notify the SDN controller to install a new flow rule, generating latency spikes.

Following and extending the approach presented by Silva et al. [68], Ndonda et al. [69] propose to use the priority field of the flow rules to solve the latency spikes. When installing flow rules for communication between A and B, two rules *r1* and *r2* are installed. The second rule *r2* has a hard-timeout twice that of rule *r1* and a lower priority. Thus, rule *r1* will be the one that the packets match. When rule *r1* expires, rule *r2* becomes the main one with the highest priority. On the expiration of rule *r1*, a flow-removed message is sent to the SDN controller, causing the controller to install a third rule *r3*. With this method, there are always two rules installed on the switches, with one of them prevailing. This method reduces latency because traffic does not have to wait for the controller to install new forwarding rules on the switch.

**Hybrid** approaches combine various network attribute randomization techniques. Chavez et al. [70] implements a host level Port-hopping and SDN-based IP-hopping and route-mutation for CIs. On the

one hand, the IP randomization is implemented in the switches, being transparent to hosts. When the traffic arrives to a switch, the source and destination pair are validated and the installed forwarding rules apply a randomization of IP addresses. The duration of these forwarding rules can be fixed or variable. On the other hand, a route randomization is performed in a configurable period of time. When two endpoints communicate, a random path is selected by the SDN controller.

#### 4.3.2. Honeypot-based response

A honeypot, better known as a "trap system" or "decoy", is located in a network or computer system with the objective to face possible attacks. There are systems that can simulate the real behavior of a system, making attackers believe that they have entered a real environment, and that it is easy to take control. Honeypots are more commonly found in IT environments, but with the increasing number of attacks in ICS, the incorporation of honeypots in these environments is becoming more common in order to detect and mitigate possible attacks [94,95]. The use of SDN technology in conjunction with honeypots has resulted in more sophisticated security solutions. These solutions can be classified into two groups:

**SDN within the honeypot** for its dynamic management and configuration. The first attempt of using SDN and honeypots in ICS is the one presented by Antonioli et al. [71]. In their proposal, a high interaction water treatment plant honeypot is proposed where an SDN controller is deployed and used inside the honeypot. The SDN controller allows an easy management of the honeypot, extending functionalities, deploying new services or modifying traffic flows dynamically inside the honeypot.

**SDN outside the honeypot** for attack traffic redirection and mitigation. Petroulakis et al. [72] propose a service chaining function architecture for SDN/NFV enabled industrial network. The author deploys multiple intrusion detection services such as firewalls and IDSs using NFV. The SDN controller defines several function chains that combine one or more deployed services, so that they can be used for different traffic types. If an intrusion is detected in any of the chains, the malicious traffic is forwarded to a honeynet. The testbed is deployed using Proxmox [96] virtualization environment. Another similar proposal is the one presented by Salazar et al. [73], where the malicious traffic is forwarded to a fake industrial network identical to the original one. In this approach, all the traffic is mirrored to the IDS to avoid introducing delays in the industrial network. The SDN controller receives notifications from the IDS in case an attack is detected and forwards the malicious traffic to the honeypot by installing new forwarding rules on the switches.

Bernieri et al. [74] present an ICS honeypot called MimePot. The honeypot is composed of two modules, one to simulate the interaction of PLCs with physical processes and the other to simulate a SCADA workstation. A data integrity attack is simulated where the SDN controller redirects the malicious traffic to the honeypot. Du et al. [75] focus on mitigating DDoS attacks in a SDN-enabled industrial scenario. A new attack type is presented, where the attacker identifies the honeypot in the network and disables it. To overcome this, a game-theoretic pseudo-honeypot approach is developed to model interactions between attackers and defender and offer optimal strategies. The testbed show that the proposed framework defends against DDoS attacks.

## 5. Discussion

We have conducted a study on different strategies to respond to intrusions in ICSs using SDN. These strategies can be used to mitigate (reactively) or prevent (proactively) malicious actions. This section provides a review of SDN-based intrusion response techniques summarized in Table 6. The motivation and evolution of these techniques are discussed with a special focus on the application in ICSs. In Table 7 shortcomings for each analyzed SDN-based intrusion response strategy for ICSs are identified.

### 5.1. Intrusion response trends

The main objective of an industrial intrusion response system is to mitigate the adverse effects that an attack may cause on an ICS. The initial SDN-based intrusion response approaches use network flow statistics or deep packet inspection techniques to detect possible attack attempts or malicious activities in the network. When an intrusion is detected, response measures are deployed reactively by blocking network flows or dropping packets according to rules or matches. From the intrusion detection point of view, a wide variety of detection methods are used, signature-based IDSs being the most common among the solutions analyzed in the state-of-the-art. The major drawback of these approaches is the risk of disrupting the network operation when network traffic is blocked or dropped. Delegating the security of an ICS to an IDS/IPS that makes decisions autonomously may cause interruptions or an undesired behavior in the system. Availability is one of the most important features in ICSs and dropping or blocking network traffic due to a wrong decision of an IDPS can lead to unexpected outages. Because of this, authors in [53,54] are the only ones that propose a solution to overcome this problem, replacing dropped packets by others with estimate values.

The use of encrypted industrial protocols or VPNs can hinder routing or forwarding decisions in SDN by using network packet payload information. As defined in its protocol specification [48], OpenFlow only provides network traffic engineering capabilities for layers 2, 3 and 4 (L2-L4). There are two different approaches that support application layer-based traffic forwarding or routing: (1) to extend the OpenFlow protocol to support different application layer protocols (as proposed in [97]) or (2) to forward or mirror network packets to the SDN controller or an external host/entity with application layer processing capabilities, as done in all approaches cited in the survey that use DPI for attack detection and response. With encrypted communication protocols or VPNs, access to the payload of the packets would be severely limited, and in most cases the network administrator would only have the packet header information to make forwarding or routing decisions. As a result, the use of DPI techniques or payload information for intrusion detection and response in ICS would not be feasible. In the literature, different traffic analysis techniques have been proposed for detecting attacks in encrypted communications, including in SDN [98].

As availability is a priority in ICS, new attack response systems were developed in order to enable survivability by reconfiguring the network when needed. Leveraging the network-wide visibility that SDN offers, these approaches continuously monitor the industrial network status to locate critical points where an outage may occur (e.g., overloaded network devices, link or port status changes). With this information, the network is reconfigured to avoid interruptions and to prevent the attack from affecting the operation of the industrial network. Assigning alternative routes to network traffic is the most used network reconfiguration technique in SDN-based ICSs, which consists of ensuring that network traffic does not flow through paths affected by an attack. For this, network interruption events are handled by the SDN controller, reconfiguring the network to ensure survivability. Network monitoring for fault detection is always the first step in designing and implementing network reconfiguration techniques to avoid interruptions in industrial communications. For example, in a site-to-site VPN connection, if a VPN endpoint is compromised, the SDN controller must be notified that the VPN endpoint has been compromised, so that it can react to changes in the network. With this information, SDN allows us to dynamically change the behavior of network flows. Network traffic could be routed through alternative paths preventing the traffic from flowing through the compromised VPN endpoint, thus minimizing the effects of the attack. As the detection occurs in the data plane and the reconfiguration in the control plane, additional delay is introduced due to information exchange between the data and control planes. A network reconfiguration solution implemented in the data plane, such as OpenFlow Fast-Failover groups where fault detection

**Table 7**
Summary of objectives and shortcomings of different SDN-based intrusion response strategies for ICSs.

| Intrusion response strategy | Objectives | Shortcomings |
|---|---|---|
| Dynamic traffic filtering | Prevent attack traffic reaching its destination by blocking flows or dropping packets. | The effectiveness depends on the accuracy of the IDS. May cause disruptions in the network. |
| Network survivability | Ensure network operation avoiding interruptions. Enable network survivability and ensure end-to-end communications. | Real-time systems performance may be affected. Control plane-based reconfiguration introduces additional delay in the recovery time. |
| Moving Target Defense | Move or reduce the attack surface by constantly changing the network configuration. Increase the cost of launching attacks. | Long randomization intervals reduces the effectiveness. Short randomization intervals increase network overhead. Difficult to set an optimal randomization interval time. Security through obscurity. |
| Honeypot-based response | Redirect attack traffic for knowledge gathering and avoiding attacks to target production systems. | Can be fingerprinted. Complexity is added to the network design. A vulnerable honeypot can be used by the attacker to launch attacks against other systems. |

and reconfiguration occurs in the data plane (without consulting the controller), a quicker convergence time is guaranteed.

The next great qualitative leap in the development of SDN-based intrusion response strategies for ICS begins by proposing proactive solutions for mitigating attacks. A proactive response consists of dynamically and continuously changing the attack surface, regardless of whether the network is under attack or not. In the state-of-the-art, MTD solutions are the only ones that follow a proactive response approach by modifying the network configuration at certain defined time intervals. Without relying on the effectiveness of IDSs or network monitoring modules for decision-making, the probability of causing a network outage in ICS is minimized. A major challenge that time-based MTD solutions face is the definition of the optimal time interval at which the network configuration will be modified or randomized. On the one hand, if the time interval is too long, an attacker may have enough time to scan and compromise the system, resulting in a security breach. On the other hand, if the interval is too short, the MTD is triggered even when the system is not under attack, wasting defense resources and degrading network performance. Furthermore, strategies that base their security on preventing an attacker from discovering possible attack vectors by hiding the configuration, services or devices available on the network can be considered as a *security through obscurity* solution. NIST 800–160 Volume 2 recommends the use of *security through obscurity* techniques as a complementary security layer for secure by design and resilient systems.

In addition to MTD strategies, honeypot-based cyber deception provides a framework for reactive attack mitigation. State-of-the-art solutions leverage SDN to dynamically detect and redirect attack traffic into honeypots, maintaining the attack active and enabling threat-intelligence gathering. A disadvantage of honeypots is the fingerprinting, an attacker could identify the identity of the honeypot by analyzing the behavior of the system. Furthermore, with the deployment of honeypots, complexity is added to the network design, increasing the attack surface. By keeping the attack active instead of blocking it, there is always the risk of an attacker compromising the honeypot and launching attacks against other systems.

Although there is a wide variety of techniques to respond to intrusions, a complete solution for securing SDN-based ICSs involves combining different security approaches. Reactive and proactive intrusion responses complement each other and together can create a more comprehensive security solution.

### 5.2. Intrusion response testbeds

To validate whether the results of the proposed solutions are satisfactory or not, each analyzed solution deploy and adapts a test environment according to the need of the problem that is being solved. The wide variety of network types and sizes poses a problem when comparing different state-of-the-art intrusion response approaches. Most of the analyzed articles deploy simple and small-scale networks, consisting of a few network devices and a single SDN controller responsible for managing the entire network. Although these types of networks are very practical for evaluation purposes, the applicability and suitability of these solutions in real and different scale ICS networks remain to be validated. Even though the variety of testbeds is large, we can classify them into three groups: (1) Emulated, (2) Virtualized and (3) Physical testbeds. These environments may also include some simulated industrial processes or components.

**Emulated testbeds.** Mininet is the most popular emulation tool used to prototype SDN networks, allowing in a quick and easy way to emulate switches, links and hosts in a single machine. MiniCPS is also a widely used tool for evaluating solutions designed for the industrial domain, extending the functionality of Mininet to add real-time simulation of CPSs, physical processes and control devices. These emulation tools allow to quickly deploy networks of different scales, providing easy replicability and a framework for integrating user-developed SDN applications that interact with the network.

**Virtualized testbeds.** Regarding virtualized environments, virtual machines are used to prototype the evaluation networks. This technique allows making deployments in a single machine or to divide the network among different computers. Similar to emulated testbeds, the use of virtual machines allow deploying networks of different scales.

**Real/Physical testbeds.** The use of real ICS equipment is a step forward compared to emulated and virtualized testbeds. The main advantage is that the behavior of the network is closer to a real industrial environment, providing results that are more representative and closer to a real system. The main drawback of physical testbeds is the low replicability they offer due to the uniqueness of the equipment and specific use cases of the state-of-the-art proposals. In addition, due to the cost of having many industrial devices to deploy large-scale networks, small networks with a limited number of devices are the most common in physical testbeds. Only four [52,59,61,64] analyzed articles use physical equipment for evaluation purposes.

### 5.3. Intrusion response architectures

As represented in Table 6, most of the intrusion response solutions heavily relay on the SDN controller or the application plane to implement intrusion detection and response mechanisms. Solutions that focus on implementing the detection system in the application plane, capture data from the data plane and transmit detection results to the SDN controller which installs mitigation flow rules in the data plane through the southbound interface. Other solutions, leveraging global view, centralized control and computational resources of the SDN controller, deploy detection and mitigation algorithms in the control plane. These architectural designs can limit the scalability and performance of different intrusion response strategies. Most of the state-of-the-art solutions use the data plane only to implement forwarding or filtering decisions made in the control plane, delegating all responsibility for intrusion detection and mitigation to external applications deployed in the application plane or to a centralized SDN controller. This could lead to undesired delays in network traffic, overhead and possible single points of failure in the network. Proposals [56,64] are the only ones that leverage the data plane to implement detection and mitigation modules, providing a lower impact on communication latencies.

Single point of failure due to control plane centralization is a well-known SDN vulnerability in architectures where redundant or distributed SDN controllers are not available. On the one side, single controller architectures refer to a single SDN controller responsible for controlling all forwarding devices in the network. On the other side, multi-controller architectures divide the network into domains each controlled by an SDN controller. During the analysis of state-of-the-art, we realized that the proposal presented by Genge et al. [63] is the only one that considers single and multi-controller scenarios for intrusion response.

Regarding the SDN controllers, the variety of available controllers is diverse and there is no preference for any of them. ODL and ONOS are best suited to the needs of industrial networks, offering features such as control plane scalability, high performance and high availability for critical operation networks. However, for research purposes, Ryu [99] and POX [100] controllers remain a popular option, offering an easy and highly programmable frameworks, but their design is not intended for industrial use.

## 6. Future research directions

This section identifies open research issues and suggests future work directions based on the insight gathered from Sections 4 and 5. The section is divided into three categories, each mapping with a future research line: (1) towards proactive and adaptive intrusion response (2) more scalable and reliable intrusion response architectures and (3) evaluation methodologies and datasets.

### 6.1. Towards proactive and adaptive intrusion response

Throughout this study, we have observed that the defense of an SDN-based industrial network can be approached from a reactive or proactive perspective, with reactive response being the most widespread technique in this research area. Next, we propose the development of new proactive responses to intrusions as one of the challenges for future research. It also emphasizes the need for the development of adaptive intrusion response techniques in ICS.

**Proactive intrusion response.** Reactive intrusion response approaches, currently the most common response method, use intrusion detection systems in order to decide when to apply response actions so that the attack can be mitigated. In industrial networks where an outage is not acceptable and high availability is required, deploying response actions on false-positive alerts due to a wrong decision of the IDS could lead to outages or unexpected behavior. With the evolution of technology and the development of increasingly sophisticated attacks, reactive intrusion response may not be sufficient to ensure reliability and security of critical operations. To address these problems, it is necessary to develop preventive solutions rather than techniques that base their strategy on detecting and responding to attacks. Because of this, the field of proactive intrusion response is becoming a promising research field in the industrial domain and requires an extensive research in the development of proactive techniques capable of responding current and future ICS threats. In the field of proactive intrusion response, MTD solutions, in conjunction with SDN paradigm, have become leading techniques in adopting a proactive approach. Although most proactive SDN-based MTD solutions are limited to performance testing and do not offer dedicated testing to mitigate specific industrial attacks, in the IT domain, multiple proactive MTD solutions have been proposed to mitigate different attacks. As an example, header randomization solutions [101,102] that hide header identifiers (e.g. IP address, TTL value, MAC address, TCP/UDP port number, etc.) by replacing them with random values.

**Adaptive intrusion response.** Adaptive intrusion response is a technique that consists of adapting the response actions based on the attack patterns, attacker behavior or the security status of the network. The main objective of adaptive responses is to dynamically adapt and deploy the optimal security measures for dealing with attacks. For this, the defense solution requires advanced detection and learning capabilities by the defender. The ability to learn about the attackers behavior and the security status of the network becomes vital for the defenders decision-making and the development of adaptive responses. The design of intrusion response mechanisms that consider attacker capabilities to design optimal intervention strategies for mitigating threats represents a challenging research line in the industrial domain and studies are needed in this field. Looking to the IT domain, adaptive intrusion response techniques have been proposed that represent promising for the development of adaptive intrusion response in ICS. As an example, by assuming that attackers and defenders behave rationally, game theory can be used to model interactions between attackers and defenders to adaptively mitigate DDoS attacks [103,104].

### 6.2. More scalable and reliable intrusion response architectures

For developing intrusion response solutions in SDN industrial infrastructures, reliability and scalability are essential goals for network management software and devices. For a more reliable and scalable intrusion response solutions, several architectural designs are proposed below that could help to achieve reliability and scalability goals in the industrial domain.

**Stateful data plane.** Delegating all processing to a centralized entity such as the SDN controller can lead to bottleneck problems, overhead or delay in the network and increase the risk of DoS attacks. For this, stateful data plane is proposed to offload the logics from the control to the data plane, allowing network switches to implement some logic to reduce network and controller overhead [105]. This can be achieved by well-known protocol independent packet processor programming language P4 [78], which enables packet processing without any match fields limitation (as opposed to OpenFlow that contains a limited number of match fields) and a full integration with SDN control protocols such as OpenFlow. There are other alternative programming languages to P4 such as the ones presented in [106,107]. Taking into account available computational resources, these programming languages can be leveraged to develop some threat detection and mitigation modules on top of forwarding devices.

**Data and control planes collaboration.** Some intrusion response strategies may require high computational resources that forwarding devices are unable to provide, making a fully stateful data plane implementation not possible. As an example, implementing ML algorithms or MTD solutions that require a network-wide coordination on top of forwarding devices is not a realistic approach. Usually, advanced algorithms with high computational requirements such as ML algorithms, MTD solutions or optimization algorithms are deployed in the controller whereas simpler procedures such as attack signatures pattern matching or traffic filtering can be implemented in the data plane. Instead of sending all flows to the SDN controller, authors in [108] perform a lightweight anomaly detection for suspicious flows detection. If an anomalous flow is detected, the forwarding device sends to the controller for a more comprehensive deep learning-based attack detection. After that, the controller installs forwarding rules in the data plane for mitigating detected attack. As shown in this approach, the collaboration between data and control planes can reduce controller overhead and delay introduced in the network.

**Distributed control plane deployment.** A single SDN controller managing the entire network can lead to bottlenecks in control plane-intensive intrusion response solutions. Multi-controller architectures are gaining ground by enabling efficiency, scalability and high availability compared to centralized control plane deployments [109]. Developing intrusion response strategies on top of multi-controller architectures will need new modules to coordinate controllers actions without introducing any problem in the operation of the network and minimizing the risk of outage. There are two types of multi-controller architectural designs [110]; (1) a fully distributed flat design (2) a

hierarchical design. In flat multi-controller architectures the network is divided into domains, each managed by an SDN controller. Controllers communicate each other to get the global view of the network and coordinate management operations, requiring advanced and extra control mechanisms to guarantee a consistent control. To overcome this problem, hierarchical multi-controller designs use two types of controllers layers: domain controllers that manage its own network domain and a root controller to manage domain controllers in order to get a global view of the network. The hierarchical architecture facilitates the management of distributed controllers through the root controller but can still introduce bottlenecks or single points of failure in the network. Authors in [111,112] propose DDoS attack mitigation strategies in non-industrial multi-controller SDN networks. A multi-controller SDN deployment can provide a more scalable architecture, minimize the risks of a single point of failure and reduce overhead by spreading computation across different controllers.

### 6.3. Evaluation methodologies and datasets

The comparison of different intrusion response techniques becomes a difficult task due to the wide variety of existing evaluation procedures in the literature. While conducting this study, we realized that there is not a standardized methodology for evaluating intrusion response approaches. Each research work uses its own evaluation procedure, metrics and network architectures that are not standardized in a contrasted methodology. A common methodology would facilitate the assessment process and help making comparisons of different intrusion response approaches. Regarding evaluation metrics, a wide variety and diverse metrics are used to measure the effectiveness and performance of the presented solution. As an example, some MTD solutions evaluate the performance of the network by using Round Trip Time (RTT) metric whereas others use the jitter or packet loss value. Similarly, solutions that use ML models for intrusion detection, accuracy, precision or F1-score are widely used to validate the detection effectiveness. These differences make it difficult to compare the different solutions in the literature.

Focusing on the deployment of evaluation architectures, networks of all types and sizes can be found in the literature. As an example, publications using tools such as Mininet or MiniCPS, deploy different sizes and varying numbers of devices architectures. Similarly, solutions using virtual machines or real industrial equipment make replicability and comparability difficult due to the diverse number of devices used tailored for each specific use case. The standard is to use small scale networks with a limited number of devices that facilitate the evaluation process.

Finally, having varied, diverse and high quality data is a very important resource for developing detection algorithms, implementing attack response approaches and evaluating the applicability of different response techniques in industrial environments. In [113,114], non-industrial SDN datasets are presented for developing and evaluating detection algorithms, both containing legitimate and attack traffic. The lack of industry-specific SDN datasets demonstrates the need to collect realistic traffic in industrial SDN environments that contains both legitimate traffic and multiple types of attack traffic. This is an important step for developing and evaluating intrusion detection and response algorithms in the industrial SDN domain.

## 7. Conclusion

Software-Defined Networking (SDN) is an emerging technology that provides numerous benefits for dynamically managing networks. The adoption of this paradigm for intrusion detection and response purposes represents a step forward compared to traditional networking. Although the effectiveness of SDN in responding to attacks has already been demonstrated, it is necessary to dedicate research effort in the development of new SDN-based intrusion response solutions for the ICS domain.

This work reviews SDN-based intrusion response solutions for ICS, classifying them based on different response strategies. A discussion of different approaches covering topics such as intrusion response trends, testbeds and architectural characteristics is also conducted. Finally, open research areas are discussed, and the most promising future research directions suggested.

The study highlights the need of SDN-based intrusion response solutions for ICSs. First, proactive and adaptive intrusion response strategies represents a promising research area for the development of ICS defense solutions. Second, the need for scalable and reliable SDN architectures for intrusion response in ICS is identified. Finally, the lack of methodologies for the development of intrusion response solutions hinders the evaluation process and the comparison between different intrusion response techniques.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Xabier Etxezarreta reports financial support was provided by Basque Government Grupos. Inaki Garitano reports financial support was provided by Basque Government Grupos. Mikel Iturbe reports financial support was provided by Basque Government Grupos. Urko Zurutuza reports financial support was provided by Basque Government Grupos. Inaki Garitano reports financial support was provided by Gipuzkoako Diputazioa Gaitzerdi. Mikel Iturbe reports financial support was provided by Gipuzkoako Diputazioa Gaitzerdi. Urko Zurutuza reports financial support was provided by Gipuzkoako Diputazioa Gaitzerdi.

## Data availability

No data was used for the research described in the article

## Acknowledgments

This work has been developed by the Intelligent Systems for Industrial Systems group, supported by the Department of Education, Language Policy, and Culture of the Basque Government (IT1676-22). This project has received support from the Department of Economic Development, Sustainability, and Environment of the Basque Government, within the ELKARTEK 2023 call, under the BEACON project (registration number 2023RTE00242510). This work has been partially funded by the GAITZERDI Project of the Gipuzkoa Science, Technology, and Innovation Network (2022-CIEN-000065-01).

## References

[1] K. Stouffer, J. Falco, K. Scarfone, Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800, 2015, p. 16.
[2] R. Mitchell, I.-R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACM Comput. Surv. 46 (4) (2014) http://dx.doi.org/10.1145/2542049.
[3] N. Jazdi, Cyber physical systems in the context of Industry 4.0, in: 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, 2014, pp. 1–4, http://dx.doi.org/10.1109/AQTR.2014.6857843.
[4] A. Sari, A. Lekidis, I. Butun, Industrial networks and IIoT: Now and future trends, in: I. Butun (Ed.), Industrial IoT : Challenges, Design Principles, Applications, and Security, Springer International Publishing, Cham, 2020, pp. 3–55, http://dx.doi.org/10.1007/978-3-030-42500-5_1.
[5] D. Ding, Q.-L. Han, X. Ge, J. Wang, Secure state estimation and control of cyber-physical systems: A survey, IEEE Trans. Syst. Man Cybern.: Syst. 51 (1) (2021) 176–190, http://dx.doi.org/10.1109/TSMC.2020.3041121.
[6] C. Pursiainen, Critical infrastructure resilience: A Nordic model in the making? Int. J. Disaster Risk Reduct. 27 (2018) 632–641, http://dx.doi.org/10.1016/j.ijdrr.2017.08.006, URL: https://www.sciencedirect.com/science/article/pii/S2212420917301644.

[7] J. Harašta, Legally critical: Defining critical infrastructure in an interconnected world, Int. J. Crit. Infrastruct. Prot. 21 (2018) 47–56, http://dx.doi.org/10.1016/j.ijcip.2018.05.007, URL: https://www.sciencedirect.com/science/article/pii/S1874548216300841.

[8] C. Zhou, B. Hu, Y. Shi, Y.-C. Tian, X. Li, Y. Zhao, A unified architectural approach for cyberattack-resilient industrial control systems, Proc. IEEE 109 (4) (2021) 517–541, http://dx.doi.org/10.1109/JPROC.2020.3034595.

[9] R. Osei-Kyei, V. Tam, M. Ma, F. Mashiri, Critical review of the threats affecting the building of critical infrastructure resilience, Int. J. Disaster Risk Reduct. 60 (2021) 102316, http://dx.doi.org/10.1016/j.ijdrr.2021.102316, URL: https://www.sciencedirect.com/science/article/pii/S221242092100282X.

[10] H. Xu, W. Yu, D. Griffith, N. Golmie, A survey on industrial internet of things: A cyber-physical systems perspective, IEEE Access 6 (2018) 78238–78259, http://dx.doi.org/10.1109/ACCESS.2018.2884906.

[11] M. Alsaeedi, M.M. Mohamad, A.A. Al-Roubaiey, Toward adaptive and scalable OpenFlow-SDN flow control: A survey, IEEE Access 7 (2019) 107346–107379, http://dx.doi.org/10.1109/ACCESS.2019.2932422.

[12] E. Molina, E. Jacob, Software-defined networking in cyber-physical systems: A survey, Comput. Electr. Eng. 66 (2018) 407–419, http://dx.doi.org/10.1016/j.compeleceng.2017.05.013, URL: https://www.sciencedirect.com/science/article/pii/S0045790617313368.

[13] M. Sainz, M. Iturbe, I. Garitano, U. Zurutuza, Software defined networking opportunities for intelligent security enhancement of industrial control systems, in: H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, E. Corchado (Eds.), International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding, Springer International Publishing, Cham, 2018, pp. 577–586.

[14] S.V.B. Rakas, M.D. Stojanović, J.D. Marković-Petrović, A review of research work on network-based SCADA intrusion detection systems, IEEE Access 8 (2020) 93083–93108, http://dx.doi.org/10.1109/ACCESS.2020.2994961.

[15] J.C. Correa Chica, J.C. Imbachi, J.F. Botero Vega, Security in SDN: A comprehensive survey, J. Netw. Comput. Appl. 159 (2020) 102595, http://dx.doi.org/10.1016/j.jnca.2020.102595, URL: https://www.sciencedirect.com/science/article/pii/S1084804520300692.

[16] N. Mazhar, R. Salleh, M.A. Hossain, M. Zeeshan, SDN based intrusion detection and prevention systems using manufacturer usage description: A survey, Int. J. Adv. Comput. Sci. Appl. 11 (12) (2020) http://dx.doi.org/10.14569/IJACSA.2020.0111283.

[17] Y. Hande, A. Muddana, A survey on intrusion detection system for software defined networks (SDN), in: Research Anthology on Artificial Intelligence Applications in Security, IGI Global, 2021, pp. 467–489, http://dx.doi.org/10.4018/978-1-7998-7705-9.ch023.

[18] O. Yurekten, M. Demirci, SDN-based cyber defense: A survey, Future Gener. Comput. Syst. 115 (2021) 126–149, http://dx.doi.org/10.1016/j.future.2020.09.006, URL: https://www.sciencedirect.com/science/article/pii/S0167739X20303277.

[19] N.M. Yungaicela-Naula, C. Vargas-Rosales, J.A. Pérez-Díaz, M. Zareei, Towards security automation in Software Defined Networks, Comput. Commun. 183 (2022) 64–82, http://dx.doi.org/10.1016/j.comcom.2021.11.014, URL: https://www.sciencedirect.com/science/article/pii/S0140366421004436.

[20] T. Alladi, V. Chamola, S. Zeadally, Industrial Control Systems: Cyberattack trends and countermeasures, Comput. Commun. 155 (2020) 1–8, http://dx.doi.org/10.1016/j.comcom.2020.03.007, URL: https://www.sciencedirect.com/science/article/pii/S0140366419319991.

[21] S. Karnouskos, Stuxnet worm impact on industrial cyber-physical system security, in: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, 2011, pp. 4490–4494, http://dx.doi.org/10.1109/IECON.2011.6120048.

[22] T. Miller, A. Staves, S. Maesschalck, M. Sturdee, B. Green, Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems, Int. J. Crit. Infrastruct. Prot. 35 (2021) 100464, http://dx.doi.org/10.1016/j.ijcip.2021.100464, URL: https://www.sciencedirect.com/science/article/pii/S1874548221000524.

[23] J. Lopez, C. Alcaraz, J. Rodriguez, R. Roman, J.E. Rubio, Protecting industry 4.0 against advanced persistent threats, Euro CIIP Newslett. 11 (2017) 27–29.

[24] J.E. Rubio, C. Alcaraz, R. Roman, J. Lopez, Current cyber-defense trends in industrial control systems, Comput. Secur. 87 (2019) 101561, http://dx.doi.org/10.1016/j.cose.2019.06.015, URL: https://www.sciencedirect.com/science/article/pii/S0167404819301245.

[25] Á.L.P. Gómez, L.F. Maimó, A.H. Celdran, F.J.G. Clemente, C.C. Sarmiento, C.J.D.C. Masa, R.M. Nistal, On the generation of anomaly detection datasets in industrial control systems, IEEE Access 7 (2019) 177460–177473, http://dx.doi.org/10.1109/ACCESS.2019.2958284.

[26] M. Conti, D. Donadel, F. Turrin, A survey on industrial control system testbeds and datasets for security research, IEEE Commun. Surv. Tutor. 23 (4) (2021) 2248–2294, http://dx.doi.org/10.1109/COMST.2021.3094360.

[27] K.S. Kiangala, Z. Wang, An effective communication prototype for time-critical IIoT manufacturing factories using zero-loss redundancy protocols, time-sensitive networking, and edge-computing in an industry 4.0 environment, Processes 9 (11) (2021) http://dx.doi.org/10.3390/pr9112084, URL: https://www.mdpi.com/2227-9717/9/11/2084.

[28] M. Cheminod, L. Durante, A. Valenzano, Review of security issues in industrial networks, IEEE Trans. Ind. Inform. 9 (1) (2013) 277–293, http://dx.doi.org/10.1109/TII.2012.2198666.

[29] R.R.R. Barbosa, R. Sadre, A. Pras, Flow whitelisting in SCADA networks, Int. J. Crit. Infrastruct. Prot. 6 (3) (2013) 150–158, http://dx.doi.org/10.1016/j.ijcip.2013.08.003, URL: https://www.sciencedirect.com/science/article/pii/S1874548213000437.

[30] E. Griffor, C. Greer, D. Wollman, M. Burns, Framework for Cyber-Physical Systems: Volume 1, Overview, 2017, http://dx.doi.org/10.6028/NIST.SP.1500-201.

[31] M. Powell, J. McCarthy, C. Tang, K. Stouffer, T. Zimmerman, W. Barker, T. Ogunyale, D. Wynne, Securing Manufacturing Industrial Control Systems: Behavioral Anomaly Detection, 2020, http://dx.doi.org/10.6028/NIST.IR.8219.

[32] P. Matoušek, O. Ryšavý, M. Grégr, V. Havlena, Flow based monitoring of ICS communication in the smart grid, J. Inf. Secur. Appl. 54 (2020) 102535, http://dx.doi.org/10.1016/j.jisa.2020.102535, URL: https://www.sciencedirect.com/science/article/pii/S2214212619311329.

[33] J. Schönwälder, M. Björklund, P. Shafer, Network configuration management using NETCONF and YANG, IEEE Commun. Mag. 48 (9) (2010) 166–173, http://dx.doi.org/10.1109/MCOM.2010.5560601.

[34] M.A. Umer, K.N. Junejo, M.T. Jilani, A.P. Mathur, Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations, Int. J. Crit. Infrastruct. Prot. 38 (2022) 100516, http://dx.doi.org/10.1016/j.ijcip.2022.100516, URL: https://www.sciencedirect.com/science/article/pii/S1874548222000087.

[35] C. Alcaraz, Secure interconnection of IT-OT networks in industry 4.0, in: D. Gritzalis, M. Theocharidou, G. Stergiopoulos (Eds.), Critical Infrastructure Security and Resilience: Theories, Methods, Tools and Technologies, Springer International Publishing, Cham, 2019, pp. 201–217, http://dx.doi.org/10.1007/978-3-030-00024-0_11.

[36] R. Arief, N. Khakzad, W. Pieters, Mitigating cyberattack related domino effects in process plants via ICS segmentation, J. Inf. Secur. Appl. 51 (2020) 102450, http://dx.doi.org/10.1016/j.jisa.2020.102450, URL: https://www.sciencedirect.com/science/article/pii/S2214212619308993.

[37] Y. Bai, Industrial Internet of things over tactile Internet in the context of intelligent manufacturing, Cluster Comput. 21 (1) (2018) 869–877.

[38] P.K. Malik, R. Sharma, R. Singh, A. Gehlot, S.C. Satapathy, W.S. Alnumay, D. Pelusi, U. Ghosh, J. Nayak, Industrial internet of things and its applications in industry 4.0: State of the art, Comput. Commun. 166 (2021) 125–139, http://dx.doi.org/10.1016/j.comcom.2020.11.016, URL: https://www.sciencedirect.com/science/article/pii/S0140366420319964.

[39] C. Urrea, D. Benítez, Software-defined networking solutions, architecture and controllers for the industrial internet of things: A review, Sensors 21 (19) (2021) http://dx.doi.org/10.3390/s21196585, URL: https://www.mdpi.com/1424-8220/21/19/6585.

[40] N. Feamster, J. Rexford, E. Zegura, The road to SDN: An intellectual history of programmable networks, SIGCOMM Comput. Commun. Rev. 44 (2) (2014) 87–98, http://dx.doi.org/10.1145/2602204.2602219.

[41] S. Sezer, S. Scott-Hayward, P.K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, N. Rao, Are we ready for SDN? Implementation challenges for software-defined networks, IEEE Commun. Mag. 51 (7) (2013) 36–43, http://dx.doi.org/10.1109/MCOM.2013.6553676.

[42] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: Enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74, http://dx.doi.org/10.1145/1355734.1355746.

[43] A. Lara, A. Kolasani, B. Ramamurthy, Network innovation using OpenFlow: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2014) 493–512, http://dx.doi.org/10.1109/SURV.2013.081313.00105.

[44] ONF, SDN architecture, 2014, URL: https://opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf.

[45] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, C.-S. Yang, Network monitoring in software-defined networking: A review, IEEE Syst. J. 12 (4) (2018) 3958–3969, http://dx.doi.org/10.1109/JSYST.2018.2798060.

[46] H. Hu, W. Han, S. Kyung, J. Wang, G.-J. Ahn, Z. Zhao, H. Li, Towards a reliable firewall for software-defined networks, Comput. Secur. 87 (2019) 101597, http://dx.doi.org/10.1016/j.cose.2019.101597, URL: https://www.sciencedirect.com/science/article/pii/S016740481930152X.

[47] G.N. Kumar, K. Katsalis, P. Papadimitriou, P. Pop, G. Carle, Failure handling for time-sensitive networks using SDN and source routing, in: 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), 2021, pp. 226–234, http://dx.doi.org/10.1109/NetSoft51509.2021.9492666.

[48] O.N. Foundation, OpenFlow switch specification, version 1.5.1, 2015, p. 283, URL: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

[49] R. Amin, M. Reisslein, N. Shah, Hybrid SDN networks: A survey of existing approaches, IEEE Commun. Surv. Tutor. 20 (4) (2018) 3259–3306, http://dx.doi.org/10.1109/COMST.2018.2837161.

[50] M.B. Jiménez, D. Fernández, J.E. Rivadeneira, L. Bellido, A. Cárdenas, A survey of the main security issues and solutions for the SDN architecture, IEEE Access 9 (2021) 122016–122038, http://dx.doi.org/10.1109/ACCESS.2021.3109564.

[51] Principles and practices for securing software-defined networks, 2015, URL: https://opennetworking.org/wp-content/uploads/2014/10/Principles_and_Practices_for_Securing_Software-Defined_Networks_applied_to_OFv1.3.4_V1.0.pdf.

[52] M. Sainz, I. Garitano, M. Iturbe, U. Zurutuza, Deep packet inspection for intelligent intrusion detection in software-defined industrial networks: A proof of concept, Log. J. IGPL 28 (4) (2020) 461–472, http://dx.doi.org/10.1093/jigpal/jzz060, arXiv:https://academic.oup.com/jigpal/article-pdf/28/4/461/33554856/jzz060.pdf.

[53] A.F.M. Piedrahita, V. Gaur, J. Giraldo, A.A. Cardenas, S.J. Rueda, Leveraging software-defined networking for incident response in industrial control systems, IEEE Softw. 35 (1) (2018) 44–50, http://dx.doi.org/10.1109/MS.2017.4541054.

[54] A.F.M. Piedrahita, V. Gaur, J. Giraldo, A.A. Cardenas, S.J. Rueda, Virtual incident response functions in control systems, Comput. Netw. 135 (2018) 147–159, http://dx.doi.org/10.1016/j.comnet.2018.01.040, URL: https://www.sciencedirect.com/science/article/pii/S1389128618300434.

[55] J. Brugman, M. Khan, S. Kasera, M. Parvania, Cloud based intrusion detection and prevention system for industrial control systems using software defined networking, in: 2019 Resilience Week, Vol. 1, RWS, 2019, pp. 98–104, http://dx.doi.org/10.1109/RWS47064.2019.8971825.

[56] G.K. Ndonda, R. Sadre, A two-level intrusion detection system for industrial control system networks using P4, in: 5th International Symposium for ICS & SCADA Cyber Security Research 2018 5, 2018, pp. 31–40, http://dx.doi.org/10.14236/ewic/ICS2018.4.

[57] A. Tsuchiya, F. Fraile, I. Koshijima, A. Órtiz, R. Poler, Software defined networking firewall for industry 4.0 manufacturing systems, J. Ind. Eng. Manag. (JIEM) 11 (2) (2018) 318–333, http://dx.doi.org/10.3926/jiem.2534, URL: http://hdl.handle.net/10419/188867.

[58] A. Melis, D. Berardi, C. Contoli, F. Callegati, F. Esposito, M. Prandini, A policy checker approach for secure industrial SDN, in: 2018 2nd Cyber Security in Networking Conference (CSNet), 2018, pp. 1–7, http://dx.doi.org/10.1109/CSNET.2018.8602927.

[59] S. Rivera, S. Lagraa, C. Nita-Rotaru, S. Becker, R. State, ROS-defender: SDN-based security policy enforcement for robotic applications, in: 2019 IEEE Security and Privacy Workshops, SPW, 2019, pp. 114–119, http://dx.doi.org/10.1109/SPW.2019.00030.

[60] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, P.-A. Karypidis, A. Sarigiannidis, DIDEROT: An intrusion detection and prevention system for DNP3-based SCADA systems, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES '20, Association for Computing Machinery, New York, NY, USA, 2020, http://dx.doi.org/10.1145/3407023.3409314.

[61] F. Holik, P. Dolezel, Industrial network protection by SDN-based IPS with AI, in: P. Sitek, M. Pietranik, M. Krótkiewicz, C. Srinilta (Eds.), Intelligent Information and Database Systems, Springer Singapore, Singapore, 2020, pp. 192–203.

[62] F. Holik, P. Dolezel, J. Merta, D. Stursa, Development of artificial intelligence based module to industrial network protection system, in: K. Arai, S. Kapoor, R. Bhatia (Eds.), Intelligent Systems and Applications, Springer International Publishing, Cham, 2021, pp. 229–240.

[63] B. Genge, P. Haller, A hierarchical control plane for software-defined networks-based industrial control systems, in: 2016 IFIP Networking Conference (IFIP Networking) and Workshops, 2016, pp. 73–81, http://dx.doi.org/10.1109/IFIPNetworking.2016.7497208.

[64] H. Sándor, B. Genge, Z. Szántó, L. Márton, P. Haller, Cyber attack detection and mitigation: Software defined survivable industrial control systems, Int. J. Crit. Infrastruct. Prot. 25 (2019) 152–168, http://dx.doi.org/10.1016/j.ijcip.2019.04.002, URL: https://www.sciencedirect.com/science/article/pii/S187454821930006X.

[65] D. Jin, Z. Li, C. Hannon, C. Chen, J. Wang, M. Shahidehpour, C.W. Lee, Toward a cyber resilient and secure microgrid using software-defined networking, IEEE Trans. Smart Grid 8 (5) (2017) 2494–2504, http://dx.doi.org/10.1109/TSG.2017.2703911.

[66] A. Aydeger, K. Akkaya, A.S. Uluagac, SDN-based resilience for smart grid communications, in: 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), 2015, pp. 31–33, http://dx.doi.org/10.1109/NFV-SDN.2015.7387401.

[67] H. Almusaher, G. Alam, How feasible moving target defense is within ICS environment, in: International Petroleum Technology Conference, in: IPTC International Petroleum Technology Conference, vol. Day 3 Wed, January 15, 2020, 2020, http://dx.doi.org/10.2523/IPTC-19649-MS, arXiv:https://onepetro.org/IPTCONF/proceedings-pdf/20IPTC/3-20IPTC/D031S062R003/1188059/iptc-19649-ms.pdf. D031S062R003.

[68] E. Germano da Silva, L.A. Dias Knob, J.A. Wickboldt, L.P. Gaspary, L.Z. Granville, A. Schaeffer-Filho, Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM, 2015, pp. 165–173, http://dx.doi.org/10.1109/INM.2015.7140289.

[69] G.K. Ndonda, R. Sadre, A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems, in: 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017, pp. 1–7, http://dx.doi.org/10.1109/NFV-SDN.2017.8169840.

[70] A.R. Chavez, W.M. Stout, S. Peisert, Techniques for the dynamic randomization of network attributes, in: 2015 International Carnahan Conference on Security Technology, ICCST, 2015, pp. 1–6, http://dx.doi.org/10.1109/CCST.2015.7389661.

[71] D. Antonioli, A. Agrawal, N.O. Tippenhauer, Towards high-interaction virtual ICS honeypots-in-a-box, in: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, in: CPS-SPC '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 13–22, http://dx.doi.org/10.1145/2994487.2994493.

[72] N.E. Petroulakis, K. Fysarakis, I. Askoxylakis, G. Spanoudakis, Reactive security for SDN/NFV-enabled industrial networks leveraging service function chaining, Trans. Emerg. Telecommun. Technol. 29 (7) (2018) e3269, http://dx.doi.org/10.1002/ett.3269, e3269 ett.3269.

[73] L.E. Salazar, A.A. Cardenas, Enhancing the resiliency of cyber-physical systems with software-defined networks, in: Proceedings of the ACM Workshop on Cyber-Physical Systems Security and Privacy, in: CPS-SPC'19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 15–26, http://dx.doi.org/10.1145/3338499.3357356.

[74] G. Bernieri, M. Conti, F. Pascucci, MimePot: a model-based honeypot for industrial control networks, in: 2019 IEEE International Conference on Systems, Man and Cybernetics, SMC, 2019, pp. 433–438, http://dx.doi.org/10.1109/SMC.2019.8913891.

[75] M. Du, K. Wang, An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things, IEEE Trans. Ind. Inform. 16 (1) (2020) 648–657, http://dx.doi.org/10.1109/TII.2019.2917912.

[76] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, A. Joglekar, An integrated experimental environment for distributed systems and networks, SIGOPS Oper. Syst. Rev. 36 (SI) (2003) 255–270, http://dx.doi.org/10.1145/844128.844152.

[77] D. Antonioli, N.O. Tippenhauer, MiniCPS: A toolkit for security research on CPS networks, in: Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/Or PrivaCy, in: CPS-SPC '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 91–100, http://dx.doi.org/10.1145/2808705.2808715.

[78] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker, P4: Programming protocol-independent packet processors, SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 87–95, http://dx.doi.org/10.1145/2656877.2656890.

[79] R. Sommer, Bro: An open source network intrusion detection system, in: Security, E-Learning, E-Services, 17. DFN-Arbeitstagung Über Kommunikationsnetze, 2003.

[80] Trema: Full-Stack OpenFlow Framework in Ruby. URL: https://github.com/trema/trema.

[81] P. Kazemian, M. Chang, H. Zeng, G. Varghese, N. McKeown, S. Whyte, Real time network policy checking using header space analysis, in: 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), USENIX Association, Lombard, IL, 2013, pp. 99–111, URL: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/kazemian.

[82] B. Lantz, B. Heller, N. McKeown, A network in a laptop: Rapid prototyping for software-defined networks, in: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, in: Hotnets-IX, Association for Computing Machinery, New York, NY, USA, 2010, http://dx.doi.org/10.1145/1868447.1868466.

[83] Gazebo: Open source robotics simulator. URL: https://gazebosim.org/.

[84] Floodlight SDN OpenFlow Controller. URL: https://github.com/floodlight/floodlight.

[85] E.W. Dijkstra, et al., A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.

[86] C. Hannon, J. Yan, D. Jin, DSSnet: A smart grid modeling platform combining electrical power distribution system simulation and software defined networking emulation, in: Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, in: SIGSIM-PADS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 131–142, http://dx.doi.org/10.1145/2901378.2901383.

[87] G.F. Riley, T.R. Henderson, The ns-3 network simulator, in: K. Wehrle, M. Güneş, J. Gross (Eds.), Modeling and Tools for Network Simulation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 15–34, http://dx.doi.org/10.1007/978-3-642-12331-3_2.

[88] V.E. Urias, M. William Stout, B.V. Leeuwen, On the feasibility of generating deception environments for industrial control systems, in: 2018 IEEE International Symposium on Technologies for Homeland Security, HST, 2018, pp. 1–6, http://dx.doi.org/10.1109/THS.2018.8574141.

[89] R. Zhuang, S.A. DeLoach, X. Ou, Towards a theory of moving target defense, in: Proceedings of the First ACM Workshop on Moving Target Defense, MTD '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 31–40, http://dx.doi.org/10.1145/2663474.2663479.

[90] P. Kampanakis, H. Perros, T. Beyene, SDN-based solutions for Moving Target Defense network protection, in: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, 2014, pp. 1–6, http://dx.doi.org/10.1109/WoWMoM.2014.6918979.

[91] J.-H. Cho, D.P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T.J. Moore, D.S. Kim, H. Lim, F.F. Nelson, Toward proactive, adaptive defense: A survey on moving target defense, IEEE Commun. Surv. Tutor. 22 (1) (2020) 709–745, http://dx.doi.org/10.1109/COMST.2019.2963791.

[92] J. Zheng, A.S. Namin, A survey on the moving target defense strategies: An architectural perspective, J. Comput. Sci. Tech. 34 (1) (2019) 207–233.

[93] N.F. Maxemchuk, Dispersity routing, in: Proceedings of ICC, Vol. 75, Citeseer, 1975, 41–10.

[94] N. Dutta, N. Jadav, N. Dutiya, D. Joshi, Using honeypots for ICS threats evaluation, in: E. Pricop, J. Fattahi, N. Dutta, M. Ibrahim (Eds.), Recent Developments on Industrial Control Systems Resilience, Springer International Publishing, Cham, 2020, pp. 175–196, http://dx.doi.org/10.1007/978-3-030-31328-9_9.

[95] S. Maesschalck, V. Giotsas, B. Green, N. Race, Don't get stung, cover your ICS in honey: How do honeypots fit within industrial control system security, Comput. Secur. 114 (2022) 102598, http://dx.doi.org/10.1016/j.cose.2021.102598, URL: https://www.sciencedirect.com/science/article/pii/S0167404821004211.

[96] Proxmox Virtual Environment. URL: https://www.proxmox.com/.

[97] R. Udechukwu, R. Dutta, Extending openflow for service insertion and payload inspection, in: 2014 IEEE 22nd International Conference on Network Protocols, 2014, pp. 589–595, http://dx.doi.org/10.1109/ICNP.2014.94.

[98] E. Papadogiannaki, S. Ioannidis, A survey on encrypted network traffic analysis applications, techniques, and countermeasures, ACM Comput. Surv. 54 (6) (2021) http://dx.doi.org/10.1145/3457904.

[99] Ryu SDN Framework. URL: https://ryu-sdn.org/.

[100] S. Kaur, J. Singh, N.S. Ghumman, Network programmability using POX controller, in: Proc. Int. Conf. Commun., Comput. Syst., Vol. 138, ICCCS, 2014, pp. 134–138.

[101] R. Skowyra, K. Bauer, V. Dedhia, H. Okhravi, Have no PHEAR: Networks without identifiers, in: Proceedings of the 2016 ACM Workshop on Moving Target Defense, MTD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 3–14, http://dx.doi.org/10.1145/2995272.2995276.

[102] Y. Wang, Q. Chen, J. Yi, J. Guo, U-TRI: Unlinkability through random identifier for SDN network, in: Proceedings of the 2017 Workshop on Moving Target Defense, MTD '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 3–15, http://dx.doi.org/10.1145/3140549.3140554.

[103] A. Aydeger, M.H. Manshaei, M.A. Rahman, K. Akkaya, Strategic defense against stealthy link flooding attacks: a signaling game approach, IEEE Trans. Netw. Sci. Eng. 8 (1) (2021) 751–764.

[104] Y. Zhou, G. Cheng, S. Yu, An SDN-enabled proactive defense framework for DDoS mitigation in IoT networks, IEEE Trans. Inf. Forensics Secur. 16 (2021) 5366–5380.

[105] X. Zhang, L. Cui, K. Wei, F.P. Tso, Y. Ji, W. Jia, A survey on stateful data plane in software defined networks, Comput. Netw. 184 (2021) 107597, http://dx.doi.org/10.1016/j.comnet.2020.107597, URL: https://www.sciencedirect.com/science/article/pii/S1389128620312305.

[106] A. Sivaraman, A. Cheung, M. Budiu, C. Kim, M. Alizadeh, H. Balakrishnan, G. Varghese, N. McKeown, S. Licking, Packet transactions: High-level programming for line-rate switches, in: Proceedings of the 2016 ACM SIGCOMM Conference, SIGCOMM '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 15–28, http://dx.doi.org/10.1145/2934872.2934900.

[107] A. Tulumello, G. Belocchi, M. Bonola, S. Pontarelli, G. Bianchi, Pushing services to the edge using a stateful programmable dataplane, in: 2019 European Conference on Networks and Communications (EuCNC), 2019, pp. 389–393, http://dx.doi.org/10.1109/EuCNC.2019.8802031.

[108] P. Krishnan, S. Duttagupta, K. Achuthan, VARMAN: Multi-plane security framework for software defined networks, Comput. Commun. 148 (2019) 215–239, http://dx.doi.org/10.1016/j.comcom.2019.09.014, URL: https://www.sciencedirect.com/science/article/pii/S0140366419308217.

[109] O. Blial, M. Ben Mamoun, R. Benaini, An overview on SDN architectures with multiple controllers, J. Comput. Netw. Commun. 2016 (2016) 9396525, http://dx.doi.org/10.1155/2016/9396525.

[110] T. Hu, Z. Guo, P. Yi, T. Baker, J. Lan, Multi-controller based software-defined networking: A survey, IEEE Access 6 (2018) 15980–15996, http://dx.doi.org/10.1109/ACCESS.2018.2814738.

[111] B. Rashidi, C. Fung, CoFence: A collaborative DDoS defence using network function virtualization, in: 2016 12th International Conference on Network and Service Management, CNSM, 2016, pp. 160–166, http://dx.doi.org/10.1109/CNSM.2016.7818412.

[112] S. Hameed, H. Ahmed Khan, SDN based collaborative scheme for mitigation of ddos attacks, Future Internet 10 (3) (2018) http://dx.doi.org/10.3390/fi10030023, URL: https://www.mdpi.com/1999-5903/10/3/23.

[113] M.S. Elsayed, N.-A. Le-Khac, A.D. Jurcut, InSDN: A novel SDN intrusion dataset, IEEE Access 8 (2020) 165263–165284.

[114] A.K. Sarica, P. Angin, A novel SDN dataset for intrusion detection in IoT networks, in: 2020 16th International Conference on Network and Service Management, CNSM, IEEE, 2020, pp. 1–5.