Securing Industrial Control Systems: A Systematic Review of Software-Defined Networking (SDN) for Mitigating Attacks on Modbus/TCP and HTTP Protocols

MUPATIWA ZULU*, LUKUMBA PHIRI
Department of Electrical and Electronics Engineering
University of Zambia (UNZA)
Great East Road Campus: P.O. Box 32379, Lusaka
ZAMBIA

**Corresponding Author*

Abstract: - Many communication protocols used in Industrial Control Systems, such as Modbus/TCP and web-based services, were developed many years ago and were not originally designed with strong security features. As a result, Industrial Control Systems became vulnerable to cyberattacks.
This paper reviews existing studies that explore how Software-Defined Networking (SDN) can improve security in Industrial Control Systems. The review focuses on how researchers simulated attacks and designed defensive strategies designed for protocols like Modbus/TCP and HTTP. It also examine Software-Defined Networking-based security frameworks, compares performance results, and identifies research trends and gaps. The findings indicate that integrating Software-Defined Networking with ICS can enhance network security in the environment. SDN and ICS Systems are used together to enhance network security by improving monitoring and protection against attacks. However, several challenges remain, including delays caused by SDN mechanisms and compatibility issues with existing industrial systems. To address the issues raised, a few solutions have been researched on the protection of several protocols altogether, and many studies do not evaluate their approaches under real industrial conditions.

Keywords: - Cybersecurity, HTTP, Industrial Control System (ICS) security, Intrusion Detection, Literature Review, Modbus/TCP, Software-Defined Networking (SDN).

Received: March 22, 2026. Revised: April 16, 2026. Accepted: June 11, 2026. Published: September 14, 2026.

1 Introduction

Industrial Control Systems (ICS) play a critical role in modern industries. They include technologies such as Supervisory Control and Data Acquisition (SCADA) systems and Distributed Control Systems (DCS), which provide the framework used to monitor and control industrial processes. These systems are widely applied in sectors like power generation, water treatment, and manufacturing. By enabling continuous monitoring and automated control, ICS helps ensure that industrial operations run efficiently, safely, and reliably [1]. SCADA systems focus on overseeing and managing widely distributed assets through centralized servers and Human Machine Interfaces (HMIs), whereas DCS structures prioritize distributed, closely linked control for immediate process automation in industrial facilities. These architectures define the broader ICS ecosystem, ensuring operational continuity and safety. Traditionally, ICS were deployed in isolated environments using proprietary communication channels; however, increasing integration with corporate networks and internet-based services has led to widespread adoption of standard IP-based protocols such as Modbus/TCP and web-based HTTP interfaces [2],[3]. These protocols were not originally designed with cybersecurity as a primary consideration and therefore lack native mechanisms for authentication, encryption, and integrity protection, rendering ICS environments vulnerable to reconnaissance, unauthorized command injection, replay attacks, and denial-of-service incidents when exposed to malicious networks [4], [5]. As the frequency and sophistication of cyberattacks targeting industrial

environments continue to rise, conventional perimeter-based security mechanisms have proven insufficient to address the dynamic and protocol-specific threat landscape of ICS [6], [7]. In this context, Software-Defined Networking (SDN) has become an approach to improve ICS security. SDN separates the control part from the data part, allowing for programmable network management. This enables fine-grained control of traffic, a complete view of the network, and quick response to network attacks on vulnerable protocols like Modbus/TCP and HTTP [8]. For example, security frameworks that use SDN are being looked at to make modern ICS setups stronger and better in dealing with cyber threats.

The search for this study was done in academic databases. These included IEEE Xplore, Scopus, Web of Science, Google Scholar, ACM Digital Library and ScienceDirect. These databases were chosen because they contain extensive research on networking, cybersecurity, and industrial automation. This is important for SDN and ICS setups. The primary search period spanned publications from 2017 through to 2025, as they contained updated literature information on SDN. The primary keywords used were "Software-Defined Networking" and "Industrial Control System," "OpenFlow-based intrusion detection," and "SDN intrusion response industrial networks." These terms ensured coverage of both protocol-specific vulnerabilities and broader SDN security frameworks.

Google Scholar was the main search engine used in this study to identify important literature. While it is widely accessible, it is also known to introduce certain indexing and ranking biases. To mitigate this limitation, a backward and forward citation analysis was conducted. This method helped identify papers that might not have appeared in the initial search. To ensure quality, only papers that discussed using SDN to enhance ICS security were included in this paper. The primary focus were papers on detecting and mitigating attacks for the Modbus/TCP and HTTP protocols. If a paper was not about ICS, did not discuss these protocols, or did not explain its methods clearly, it was excluded from this study.

This study extends prior qualitative surveys by introducing formal models and quantitative analysis to evaluate SDN-based ICS security mechanisms. In particular, mathematical representations of network flows, detection functions, optimization objectives, and performance metrics are integrated throughout the paper to meet the expectations of engineering and systems-oriented research. Ultimately, this study aims to build a knowledge base to support the design of security systems. These systems are for ICS and are based on SDN. They need to withstand cyber threats.

Fig. 0.1 (Appendix A) shows how the research was conducted. The study began by defining the research objectives. Relevant keywords and databases were then selected to guide the search. The chosen papers were carefully read, and the main ideas, including new findings from different researchers, were noted. The results were then examined to see what is already known and what problems still need to be resolved.

This process contributed to the development of a security framework for ICS using SDN. The remainder of the paper is organized as follows:

a) 2 Section 2 reviews existing literature and related work on the topic
b) Section 3 discusses how SDN can enhance the security of ICS
c) Section 4 examines the application of SDN in defending ICS environments
d) Section 5 highlights the research challenges and gaps in SDN-based security for ICS
e) Section 6 concludes the paper and provides recommendations for future work on SDN-based security frameworks for ICS.

2 Literature Review: SDN-Based Security for ICS Communication Protocols

ICS use special communication protocols to support real-time monitoring and control of physical processes. However, most of these protocols were originally designed without strong security features, which makes them vulnerable in today's highly interconnected network environments.

Recent research suggests that SDN could offer practical security improvements in ICS environments. One of the main reasons is that SDN

separates the control plane from the data plane, enabling centralized management of network policies. Because of this centralized control and programmability, security rules can be updated or enforced dynamically when threats are detected. In this section, the discussion focuses on previous work that applied SDN to secure ICS networks, particularly those that used the Modbus/TCP and HTTP protocols.

2.1 Security Challenges in ICS Communication Protocols

Modbus/TCP is a widely used industrial communication protocol that is well appreciated because of its simplicity and low communication overhead. However, it does not include key security features such as authentication (authenticating user identities), encryption (protecting data from being captured), or data validity checks (detecting tampering), which makes ICS networks exposed to attacks like replay (reusing captured data to repeat actions), command injection (inserting unauthorized instructions), and unauthorized register manipulation (changing device states without consent). Attackers can exploit Modbus function codes (specific codes used to communicate commands) to delete Programmable Logic Controller (PLC) states or interfere with operations. Equally, HTTP, which stands for Hypertext Transfer Protocol exist in SCADA dashboards (graphic displays for monitoring), web-based HMIs (operator interfaces), and industrial gateways that brings web-specific vulnerabilities such as cross-site scripting (injecting scripts), SQL injection (inserting malicious database commands), and denial-of-service (overloading systems). The combination of Mobus/TCP and HTTP traffic in combined IT/OT networks expands the total attack surface, making it difficult to detect and prevent cyber threats. Fig. 0.2 (Appendix A) shows some of the ways that Modbus/TCP and HTTP can be attacked in an ICS environment. This is really important because Modbus/TCP and HTTP are used a lot in these systems and they require to be protected.

2.2 Conventional Security Approaches for ICS Protocols

Conventional ICS security measures consist of firewalls, demilitarized zones (DMZs), Virtual Local Area Networks (LANs), and intrusion detection systems (IDS). Although these methods offer fundamental protection, they tend to be static and centered on perimeter security, rendering them inadequate against insider threats and attacks that are aware of protocols [9]. Signature-based IDS solutions such as Snort perform well for known attack patterns but fail to detect zero-day exploits or subtle deviations in industrial traffic behavior [10]. Anomaly-based IDS systems improve detection of unknown attacks but often generate high false positive rates due to the deterministic yet sensitive nature of industrial traffic [11]. Moreover, these mechanisms typically lack centralized manageability, resulting in delayed or fragmented responses to coordinated attacks.

2.3 SDN as an Enabler for ICS Security

SDN provides a flexible framework that separates the control plane from the data plane, allowing for centralized traffic oversight and specific flow regulation. Multiple research findings indicate that SDN greatly enhances network visibility and allows for immediate implementation of security policies within ICS settings [12], [13]. Utilizing OpenFlow-capable switches, SDN controllers can analyze Modbus/TCP function codes, track HTTP request movements, and actively block or redirect harmful traffic. The centralized aspect of SDN facilitates worldwide threat detection, proving beneficial for identifying distributed or multi-stage attacks. Fig. 0.3 (Appendix A) shows what the security architecture for these networks looks like in the industrial environment.

Fig. 0.3 (Appendix A) presents the SDN-based security architecture proposed for ICCS (Integrated Command and Control System) networks. The architecture is structured into layered components to clearly separate data forwarding, control, and security intelligence. Basically, ICS field devices involving PLCs, RTUs, sensors, actuators, and HMIs, communicate operational data through industrial and web protocols like Modbus/TCP and HTTP. These messages pass through SDN-capable switches that make up the data plane and manage the forwarding of packets based on flow rules.

The control plane is harnessed by a centralized SDN controller that holds a total view of the network. The controller continuously gathers flow data from switches and implements security policies that are knowledgeable of the protocols. Through this communication infrastructure, ICS field devices are able to send and receive messages that support monitoring and control functions. This information passes through SDN-capable switches that make up the data plane and handle the forwarding of packets based on flow rules.

The control plane is applied by a centralized SDN controller that holds a complete view of the network. The controller constantly gathers flow data from switches and enforces security policies that are knowledgeable of the protocols. The combined security features, such as intrusion detection and anomaly detection, can identify traffic patterns including harmful or typical behaviors. Once identified, the controller quickly enforces mitigation measures to restrict, redirect, or regulate the volume of malicious traffic while guaranteeing that critical control communication is prioritized.

In the uppermost layer, management and security applications connect to the controller through northbound APIs to deliver enhanced functionalities such as automated response management, logging, and alert creation. **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** above illustrates how SDN facilitates centralized administration, flexible policy implementation, and responsive security in ICS settings without modifying existing industrial equipment, matching the goals and framework of the suggested research.

2.4 SDN-Based Intrusion Detection and Mitigation Techniques

Recent research has suggested integrating intrusion detection systems directly into SDN controller so that security threats can be detected and addressed more quickly within the network. SDN/IDS frameworks based on machine learning (ML) investigate flow statistics to identify anomalies in Modbus/TCP traffic, attaining detection accuracies over 95% in simulated settings [14], [15]. Deep learning methods, such as convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) models, enhance detection capabilities but also bring added computational demands and delays [16]. Flow-rule-based mitigation techniques actively separate compromised devices or prevent malicious packets, reducing attack speed. However, reactive flow installations may cause temporary delays during attack bursts, potentially affecting time-critical industrial operations [17]. Fig. 0.4 (Appendix A) illustrates conventional ICS security mechanisms and their limitations.

2.5 Security Framework and Security Information and Event Management (SIEM)

International standards such as NIST SP 800-82 and IEC 62443 offer organized recommendations for industrial cybersecurity, advocating for a multi-layered security approach, access control measures, and effective incident response strategies. SIEM solutions collect logs and alerts from various devices to allow for centralized monitoring and analysis. While SIEM platforms can improve situational awareness, integrating them with legacy ICS infrastructure introduces considerable complexity and cost. Engineers must address challenges in sensor placement, data normalization, and rule tuning, which can introduce latency and limit real-time response capabilities for resource-constrained operators.

2.6 Analysis of SDN-Based ICS Security Frameworks

Several studies have discovered the use of SDN to develop security in ICS networks, particularly for protocols such as Modbus/TCP and HTTP. Using SDN to filter network traffic and enhance the security of Modbus/TCP communications was suggested in [18]. Their rule-based detection method offers low latency and is relatively easy to deploy in industrial environments. However, the approach may not effectively detect more complicated cyber-attacks. In this context, both SDN and Modbus/TCP protocol play a crucial role in enabling traffic monitoring and threat mitigation. Similarly, a machine learning-based SDN intrusion detection system designed to identify malicious activities in Modbus/TCP traffic was proposed in [19]. The method uses anomaly detection techniques and demonstrates high detection accuracy. Nevertheless, the performance of the approach depends on the availability of high-quality training data, which is often limited in

real industrial environments. Therefore, the quality of data plays a detailed role in ensuring the success of SDN-based security mechanisms for Modbus/TCP networks.

In another study, a deep learning-based SDN security system proposed to strengthen both Modbus/TCP and HTTP protocols was developed in [20]. The structure uses neural network models to sense irregularities in network traffic, resulting in high detection accuracy. However, the methodology requires considerable computational resources, which may model challenges for utilization in industrial environments with limited processing capacity. In this framework, SDN, Modbus/TCP, and HTTP serve as key components that enable network monitoring and anomaly detection. A reactive mitigation strategy for SDN environments was proposed to address security threats in Modbus/TCP networks. The approach dynamically inserts security rules into the network controller, enabling a rapid response to detect attacks. However, the process of dynamically adding rules may introduce delays that could affect time-sensitive industrial operations. In this strategy, both SDN and Modbus/TCP protocol play critical roles in enabling effective threat detection and mitigation. Additionally, a multi-protocol SDN security framework proposed to protect both Modbus/TCP and HTTP communications was described in [21]. The framework combines machine learning techniques with SDN flow rules to improve detection capability and provide broader protocol reporting. Although the approach demonstrates promising results, its performance in large-scale industrial deployments remains uncertain. In this framework, SDN, Modbus/TCP, and HTTP serve as essential components for enabling network monitoring and threat detection.

2.7 Limitations and Research Gaps

Despite promising results, most SDN-based ICS security solutions are validated primarily in simulated environments such as GNS3 or Mininet, limiting their real-world applicability [22]. A number of studies focus merely on Modbus/TCP, forgetting HTTP-based industrial services and their relationship with control traffic. The performance indicators, like latency, jitter, and throughput, are really important for ICS operations. Without these indicators, there can be problems that can lead to underreporting. It is for this reason that there is a need for security systems to work with different protocols to adapt to different situations. These systems also need to be able to work within the limits of what's possible in industry. Fig. 0.5 (Appendix A) shows the areas where more research is needed based on what was found when looking at what people are saying about using SDN to secure ICS systems.

As outlined in Fig. 0.5 (Appendix A), the primary research deficiencies in SDN-centered security, emphasizing resilience problems, inadequate multi-protocol defense structures, and difficulties in merging SDN solutions with traditional systems and industrial security criteria.

3 The Role of SDN in ICS Security: Formal Models and Analysis

SDN offers a programmable and centralized control plane for ICS, allowing dynamic flow supervision, intrusion detection, and the application of mitigation policies. To carefully evaluate the effectiveness of SDN in improving ICS security, it is necessary to develop formal network models, decision functions, control policies, and appropriate performance metrics. This section presents a quantitative framework for modelling SDN-based ICS networks, formulating intrusion detection, defining mitigation strategies, and assessing performance using standardized metrics.

3.1 SDN-Based ICS Network Model

An SDN-enabled ICS network can be represented using a directed graph model:

$$G = (V, E) \quad (1)$$

In this model, $V = V_d \; U \; V_s \; U \; V_c$ represents the set of nodes in the network. In this model, V_d represents ICS devices like Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Human-Machine Interfaces (HMIs). The set V_s represents SDN switches, while V_c denotes SDN controllers. The set E represents the communication links that connect these nodes within the network.

Each network flow f_i can be formally described as:

$$f_i = (src_i, dst_i, p_i, b_i, t_i) \quad (2)$$

Where src_i and dst_i represent the source and destination nodes of the flow, respectively. The parameters p_i denotes the communication used,

such as Modbus/TCP or HTTP. The variable b_i represents the required bandwidth for the flow, while t_i defines the timing constraints associated with the corresponding control loop.

This abstraction allows both traffic patterns and protocol-specific characteristics to be modelled effectively, which is essential for conducting realistic performance and security analysis in ICS networks.

The set of flow rules can be represented as:

$$R=\{r_j=(match_j, action_j, priority_j)\} \quad (3)$$

These rules are installed dynamically by the SDN controller through the OpenFlow protocol. Each rule defines how packets that match specific header fields ($match_j$) should be processed. The parameter ($action_j$) specifies the action to be applied to the matched packets, while ($priority_j$) determines the priority level of the rule when multiple rules are applicable.

This rule structure enables fine-grained traffic control within the network, allowing protocol-aware forwarding, effective traffic isolation, and real-time mitigation of potential threats.

3.2 Intrusion Detection as a Decision Function

Intrusion detection in SDN-enabled ICS can be formalized as a decision function. A detection function can be defined as:

$$D : \mathbb{R}^n \rightarrow \{0, 1\} \quad (4)$$

Where a feature vector x = [x_1, x_2, …., x_n] is extracted from traffic statistics such as packet rate, byte count, Modbus function codes, and HTTP methods. The function *D(X) = 1* indicates that the traffic is classified as malicious. This formalism allows the application of both supervised and unsupervised ML models for anomaly detection or attack classification [23], [24].

For supervised ML classifiers, the model is trained by minimizing a loss function.

$$min_\theta \sum_{\{i=1\}}^{N} L\left(D_{\theta(x_i)}, y_i\right) \quad (5)$$

Where y_i is the ground-truth label of flow i, and θ represents the model parameters.

In unsupervised settings, anomaly scores $A(x)$ are computed for each observation and compared against a threshold T. This enables the detection of previously unseen attacks or variations from normal ICS behavior [25], [26]. The inclusion of ML-based IDS within the SDN controller introduces a quantifiable computation cost, which is incorporated into controller load and latency analysis (see Section 4).

3.3 Mitigation and Control Policy Formulation

Upon detecting malicious flows, mitigation is formulated as a control policy:

$$\pi : f_i \rightarrow a_k \in \{block, rate-limit, reroute\} \quad (6)$$

This mapping assigns an action a_k to each network flow f_i, where the possible actions include blocking the flow, applying a rate limit, or rerouting it. The policy must respect real-time operational limitations:

$$\Delta t_{ctrl} + \Delta t_{sw} \leq T\_\max \quad (7)$$

Here, Δt_{ctrl} represents the processing delay at the SDN controller, Δt_{sw} is the enforcement delay at the switch, and $T\_\max$ represents the maximum allowable latency to ensure safe operation of the ICS network. This formulation ensures that security interventions do not compromise control loop stability or violate industrial timing constraints [27].

Mitigation strategies such as selective packet blocking, rate limiting, or rerouting traffic are implemented via flow-rule updates in the SDN switches. The optimal action selected by the policy depends on several factors, including the severity of the detected attack, the criticality of the network flow, and the available resources of the SDN controller. These factors create an inherent trade-off between security enforcement, network latency, and system resilience in ICS environments [28].

3.4 Performance Metrics and Analytical Definitions

To allow consistent evaluation across different studies, several quantitative performance metrics are defined:

4 Detection Rate (DR):

$$DR = \frac{TP}{TP+FN} \quad (8)$$

This measurement shows the ratio of malicious activities that are correctly identified by the detection system.

5 False Positive Rate (FPR):

$$FPR = \frac{FP}{FP+TN} \quad (9)$$

This measure shows the proportion of traffic that is wrongly labelled as bad.

Precision:

$$P = \frac{TP}{TP+FP} \quad (10)$$

Precision estimates how many flows identified as attacks are essentially malicious.

6 End-to-End Latency:

$$L = L_{data} + L_{control} \quad (11)$$

This metric represents the overall delay faced in the system, combining the network data-plane delay, the control-plane processing and implementation delay.

The following abbreviations, TP, FP, TN, and FN, in these equations, mean:

a) TP represents true positives
b) FP represents false positives
c) TN represents true negatives
d) FN represents false negatives.

The parameter L_{data} represents the delay within the network data plane, while $L_{control}$ corresponds to the processing and enforcement latency in the control plane.

Together, these metrics provide a structured framework for evaluating SDN-based security solutions across different ICS deployments and experimental environments [29], [30].

SDN has been widely recognized in the literature as a promising paradigm for strengthening security in ICS, particularly in response to the growing limitations of traditional, static defense mechanisms. Conventional ICS networks rely on fixed configurations, distributed control logic, and perimeter-based security. These mechanisms are often ineffective against cyberattacks that are designed to bypass traditional security protocols. This issue has become more significant considering the interconnection of Information Technology (IT) and Operational Technology (OT), which exposes ICS to greater security risks. [31].

This problem is being addressed through SDN, which helps to separate the network's control functions from the data forwarding functions. This separation makes the network stronger, more flexible, and capable of applying security policies more effectively. As a result, SDN offers a promising approach for handling the developing security threats faced by ICS and helps improve their protection against cyberattacks.

3.5 Comparison of SDN-based Intrusion Detection Approaches

A number of studies have looked at the use of SDN to enhance intrusion detection in ICS systems. These methods naturally examine the communication protocols used in ICS, such as Modbus/TCP and HTTP, to detect malicious traffic patterns. They often use machine learning methods to identify unusual behavior in network traffic. To assess the performance of these machine learning models, the use of metrics like Detection Rate (DR) and False Positive Rate (FPR) is applied, where Detection Rate shows how effectively the system can identify real attacks, while False Positive Rate shows how normal traffic is mistakenly classified as malicious. For ICS, it is important to achieve a high detection rate while keeping false alarms as low as possible to ensure both security and stable system operation.

Previous research in [32] proposed an SDN-based intrusion detection framework that focuses on the Modbus/TCP protocol. A Random Forest classifier was used to analyze network traffic features, including packet rate, Modbus function codes, and byte count. The evaluation produced a detection rate of 95.2% with a false positive rate of 3.1%. These results suggest that ensemble learning methods can efficiently detect abnormal patterns in ICS network communication.

Similarly, the use of deep learning for detecting attacks in HTTP-based communications within ICS environments was explored in [33]. This model uses a Long Short-Term Memory (LSTM) neural network to study features such as HTTP request methods. Payload size and flow duration. The proposed system achieved a detection rate of 93.5% and a false positive rate of 2.8%, highlighting the ability of recurrent neural networks to capture temporal patterns in network traffic.

Another approach was presented in [34], which developed an unsupervised anomaly detection system based on an Autoencoder model. Their framework analyses both Modbus/TCP and HTTP traffic by extracting flow statistics and timing constraints associated with industrial control operations. The study reported a detection rate of 91.7% with a false positive rate of 4.2%, demonstrating that unsupervised learning

techniques can identify abnormal behaviors even without labeled training data.

In addition, the research proposed an SDN-based intrusion detection mechanism using a Support Vector Machine (SVM) classifier. The simulation examines Modbus/TCP traffic and uses features such as packet rate and PLC command types to identify potential threats. The results generated showed that the system was able to detect 90.8% of threats with a false positive rate of 5.0%, which indicated that traditional machine learning processes can still be effective for detecting interference in industrial communication protocols.

A way to detect intrusions in HTTP-based ICS communications using the K-Nearest Neighbors (KNN) algorithm was studied in in [35]. This simulation examines features such as HTTP request methods and the occurrence of requests to identify irregular behavior. According to the researcher, the detection rate of 88.9% with a false positive rate of 6.3% of the system indicates that it can successfully detect threats while maintaining a relatively simple computational model [36].

These studies show that using machine learning to detect intrusion and combining it with SDN controllers can make ICS networks a lot more secure. The writer indicated that different algorithms have their own strengths and weaknesses in terms of detection accuracy, computational complexity, and flexibility to specific protocols [37]. Choosing applicable features and machine learning models is key when designing security solutions for industrial settings that use SDN.

While the approaches mentioned above mainly focus on detecting intrusions, other research has also looked at using SDN to mitigate threats [38]. These mitigation strategies typically involve SDN control actions such as traffic blocking, rate limiting, flow rerouting, and network segmentation in order to reduce the impact of detection attacks [39], [40].

3.6 SDN-Based Mitigation and Control Strategies in ICS

Some studies have looked at ways to use SDN to make ICS stronger against cyber-attacks [41]. These methods looks on various mitigation methodologies, such as blocking malicious activities, placing controllers in different locations, and using various redundancy mechanisms. These design choices can influence factors such as network response time and the overall dependability of the system [42].

A mitigation approach that blocks suspicious traffic flows was proposed in [43]. In this approach, it was noted that the SDN controller detects traffic that appears malicious and then prevents that traffic from being transmitted over the network by using a centralized controller to make all decisions, which allows it to react quickly. The system responds to these threats in approximately 15 milliseconds, demonstrating quick mitigation capability. Nevertheless, this method also has a limitation because the design relies on a single controller, which creates a single point of failure. If the controller fails or is compromised, the entire system may stop functioning properly, reducing the overall resilience of the network [44].

In contrast, a rate-limiting mechanism was implemented through a distributed controller architecture with a redundant controller instance. In this approach, the system does not completely block suspicious traffic but instead limits the rate at which malicious flows are transmitted. This method improves the reliability of the system and helps it better handle potential failures. However, because the system must coordinate with multiple components, the response time is slightly increased to about 18 milliseconds. Despite this additional delay, the design focuses on improving the system's reliability and fault tolerance, and the response time remains relatively fast [45].

Another approach was proposed, which implemented a hybrid mitigation framework combining centralized and local control mechanisms. In this approach, the system has included a number of mechanisms to maintain security by switching traffic and separating parts of the network when suspicious activity is detected. The architecture uses multiple controllers, so if one controller fails, the system can continue operating. The maximum response time is about 20 milliseconds, which provides a good balance between fast response and system security.

An access control approach was proposed in which an SDN controller manages and regulates who is allowed to access the system. The controller can dynamically update access permissions to prevent unauthorized users from entering the network. This approach is very fast, requiring only about 12

milliseconds to apply changes. The design depends on a central controller, which can lead to the failure of the system if that controller becomes unavailable.

A method that uses multiple communication paths together with multiple controllers was proposed. This design improves system reliability because the network can switch to alternative paths if a failure occurs. Although the response time is slightly longer, around 22 milliseconds, the approach increases network availability and resilience, particularly when controllers fail or when the network is disrupted [46].

Generally, these security approaches demonstrate important trade-offs of systems that rely on a single centralized controller to respond faster but may also suffer from a single point of failure. Systems that use a lot of controllers and redundant paths provide better dependability and fault tolerance, although they may introduce slightly higher response times. These kinds of designs are really important for keeping the ICS safe and working properly. On the other hand, using a system that is spread out or a mix of different systems can help make the ICS stronger and more reliable, but it can also slow things down and make them work harder. The effectiveness of these approaches is typically evaluated using quantitative metrics such as

end-to-end latency, controller processing delay, and network overhead, as formally defined in Section 3.4.

4 SDN-Based Defense Mechanism: A Quantitative Perspective

This section provides a quantitative analysis of SDN-based defense mechanisms for ICS, building directly on the formal models introduced in Section 3 and Equations (1)-(11). Rather than presenting SDN security techniques in a purely descriptive manner, the discussion emphasizes analytical performance metrics, latency considerations, and optimization trade-offs that are critical in industrial environments [47].

Using the SDN-enabled ICS network model defined in Equations (1)-(3), security enforcement is analyzed in terms of control-plane and data-plane interactions. The attack and detection models in Equations (4)-(7) allow intrusion detection accuracy, false alarm rates, and reaction time to be expressed quantitatively. In particular, the analytical latency model in Equations (8)-(10) decomposes end-to-end response delay into sensing, controller processing, and mitigation components, which is essential for assessing the feasibility of SDN-based defenses under real-time industrial constraints [48].

Machine-learning-based IDS integrated with the SDN controller introduces additional computational overhead. As captured in Equation (11), the inference cost of ML models directly affects controller load and processing delay. This establishes an explicit trade-off between detection accuracy and control-plane responsiveness, especially in large-scale or multi-protocol ICS deployments [49], [50]. Consequently, lightweight or distributed ML inference strategies are often required to maintain bounded response times.

Several SDN defense mechanisms, including flow rule enforcement, traffic isolation, and dynamic access control, can be expressed within this analytical framework. Their effectiveness depends not only on detection capability but also on controller placement and redundancy strategies. Optimization tasks, like deciding where to place controllers, can be expressed as:

$$\min_{C} \max_{v} \in V_d \; d(v, C)$$

Subject to availability and redundancy constraints (12)

This formulation illustrates the inherent trade-off between reducing control-plane latency and maintaining resilience through redundancy. Placing controllers in a centralized manner can lower average delays, but it also risks creating single point of failure. On the other hand, deploying distributed or redundant controllers enhances fault tolerance, though it introduces additional coordination overhead [51], [52].

Overall, this quantitative perspective demonstrates that SDN-based defense mechanisms for ICS must be evaluated holistically, considering detection performance, latency guarantees, controller load, and resilience requirements simultaneously. Such an analytical approach is essential for designing deployable and dependable SDN-enabled security architectures in industrial control networks [53], [54].

To support this analytical discussion, the next section leverages the models and optimization

formulation in Equations (1)-(12) to evaluate representative SDN-based security strategies under realistic performance metrics. In particular, detection accuracy, end-to-end latency, controller overhead, and network impact are assessed to illustrate the practical implications of the trade-offs identified in this section.

Fig. 0.6 (Appendix A) illustrates the end-to-end SDN-based framework that underpins the security frameworks reviewed in this paper.

In the context of securing ICS networks and vulnerable protocols such as Modbus/TCP and HTTP, the pipeline begins with continuous flow monitoring, where SDN switches collect traffic statistics and packet features without disrupting real-time operations. These data are analyzed in the detection stage using protocol-aware intrusion detection and machine-learning techniques to identify anomalies and attack patterns specific to industrial traffic.

Once a threat is identified, the decision-making stage within the SDN controller evaluates risk severity and generates appropriate mitigation policies. The enforcement of these policies at this stage is applied by installing flow rules that limit traffic rates or isolate suspicious flows. This approach allows attacks to be stopped quickly while ensuring that legitimate communications continue without delay [55].

In the final dynamic adaptation stage, techniques such as moving target defense and automatic traffic control are applied. This stage is supported by a continuous feedback loop that monitors traffic and identifies opportunities to improve both detection and response strategies.

Overall, the figure encapsulates how SDN enables centralized, adaptive, and protocol-aware defense mechanisms, aligning with the article's objective of evaluating SDN as a scalable and responsive security paradigm for modern ICS environments.

A critical systems consideration for SDN defenses is controller placement and distribution. Centralized controllers simplify policy management but increase latency and create potential single points of failure. Contemporary studies therefore emphasize distributed or hierarchical controller architectures (multi-controller, edge-assisted SDN) and optimized controller placement to minimize worst-case latency while maintaining detection coverage and resilience. These works show that careful partitioning of the control plane, placing the controller near latency-sensitive ICS segments, or delegating lightweight detection to edge nodes can substantially reduce reaction times and improve scalability, but they require more complex synchronization and consistency management.

Operational validation is another recurring issue where most SDN-based defense proposals are validated in Mininet, GNS3, or small emulated testbeds. Only a minority have been tested on realistic industrial testbeds or field deployments. This gap is important because emulators often underrepresent jitter, PLC cycle timing constraints, and mixed legacy-proprietary equipment behavior, all of which affect the real latency/throughput impact of SDN interventions [56]. Consequently, claims of "negligible latency" or "near real-time mitigation" must be treated cautiously unless validated on production-representative testbeds.

Finally, protocol specificity matters as many SDN defenses target Modbus/TCP because it is simple to parse and a common ICS vector with fewer works provides holistic, multi-protocol defenses that simultaneously account for HTTP (web HMIs), DNP3, OPC-UA, and application-layer interactions. Multi-protocol frameworks tend to have higher parsing costs and higher controller processing requirements, and therefore, face greater latency concerns, yet they are essential for practical ICS protection because attackers commonly combine web-based and control-protocol vectors [57].

5 Challenges and Open Research Problems

The application of SDN in ICS security has demonstrated important benefits, yet multiple technical and operational challenges continue to delay practical implementation. The following is the discussion of the major challenges identified in the literature, expressing concrete open research problems and directions that follow from them. Meanwhile, citations are provided to recent studies that analyze or illustrate each issue.

One of the technical challenges is controller reliability and a single point of failure. Centralized controllers simplify policy management and global visibility but generate an attractive high-value target where successful attacks or failures at the controller can paralyze application and detection

capabilities. Research must therefore develop resilient control-plane architectures that combine fault tolerance, controller replication, and secure consensus while protecting low latency. Work on distributed/hierarchical controllers and controller placement strategies shows promise, but optimal placement under ICS latency and availability constraints remains an open problem. Formal methods for verifying controller redundancy protocols and secure state synchronization are also required.

Closely related are latency and real-time constraints. Industrial processes often require bounded, deterministic communication delays; however, SDN operations such as packet-in processing, controller-based machine learning inference, and reactive flow-rule installations may introduce additional latency that can breach these strict timing constraints. Although procedures such as proactive flow-rule provisioning and edge-based processing can reduce controller round-trip delays, they naturally introduce a trade-off between rapid response and temporal predictability. Therefore, future research should focus on describing worst-case latency under realistic ICS operating conditions, developing hybrid edge-controller architectures that ensure guaranteed response bounds, and designing scheduling and Quality of Service mechanisms that prioritize time-critical control traffic during security enforcement actions.

Scalability and controller load present another practical barrier. ML/DL models integrated into controllers for anomaly detection provide high accuracy but increase computational and memory demands. Large industrial infrastructures with many flows will stress naïve central analytics. Research must explore (1) lightweight algorithmic alternatives suitable for on-device or edge inference, (2) hierarchical distribution of analytics, and (3) elastic offloading mechanisms that maintain detection fidelity while limiting controller overload. Strategies that combine model compression, incremental learning, and approximate inference are promising directions.

A pervasive problem is legacy device interoperability and constrained endpoints. Many PLCs, RTUs, and embedded HMIs cannot be upgraded to support modern security protocols or SDN southbound interfaces. This requires SDN solutions that operate transparently through inspecting and enforcing policies without modifying field devices or the design of secure gateways that preserve timing semantics. Research questions include how to perform deep packet or protocol-aware inspection safely on encrypted tunnels, and how to introduce security proxies or gateways that do not break control loops [58].

From a research methodology perspective, there is an urgent need for standardized datasets, benchmarks, and realistic testbeds. Much of the claimed performance in literature derives from Mininet/GNS3 simulations or synthetic traces that do not capture the jitter, cycle timing, or multi-vendor heterogeneity of production ICS. The community must converge on representative traffic mixes, labeled multi-protocol datasets (Modbus/TCP, HTTP), and shared testbeds to enable fair comparison and reproducible results. Creating such benchmarks raises practical and ethical problems that require careful study. Machine learning components introduce a set of security and trustworthiness problems. ML models are susceptible to adversarial evasion, data poisoning, concept drift, and lack of interpretability. In ICS contexts, false negatives have direct physical consequences. Open research problems, therefore, include adversarial-robust training for protocol-aware models, continual learning methods that adapt to operational changes without problems, and explainable ML approaches that provide human operators with actionable justifications for automated mitigations [59]. Methods for certifying ML model behavior under bounded perturbations would be particularly valuable. Automated manageability and safe actuation remain a delicate challenge. Hybrid systems that automatically translate detection outputs into flow-rule changes must include mechanisms for safety verification and rollback to prevent automated defenses from causing outages [60]. Research is needed on policy-verification tools, human-in-the-loop intervention thresholds, and formal guarantees that enforcement actions preserve critical control invariants. Techniques from control theory and formal methods should be adapted for SDN actuation pipelines.

Multi-protocol and cross-layer attacks are underexplored, and attackers usually combine web-based exploits (HTTP/XSS/SQL injection) with control-protocol manipulations to achieve stealthy, multistage compromises. Most SDN defenses target a single protocol, resulting in

designing unified detection and mitigation strategies that reason across application, transport, and network layers is an open research area. This requires efficient multi-protocol parsers, correlation engines, and semantics-aware models that understand ICS process context [61].
Ultimately, challenges related to regulation, privacy, and organization are significant [62]. Implementing SDN security in industrial facilities might clash with operational policies, regulatory requirements, or operator activities [63], [64], [65]. Research should, therefore, encompass socio-technical analyses to assess deployability, user confidence, incident response procedures, and the legal ramifications of automated network management in safety-sensitive systems.

6 Conclusion and Future Research Directions

6.1 Conclusion

This paper focuses on security in ICS and examines how SDN can be applied to enhance protection in these environments. The study consolidates existing research through formal network and attack models, analytical performance metrics, and optimization-based perspectives, thereby moving beyond purely descriptive surveys.
Recent surveys from IEEE and ACM show that security architectures based on SDN significantly enhance ICS protection, especially when integrated with intrusion detection systems leveraging machine learning and strategies for latency-aware placement of controllers. This study reviews previous research on how SDN can be used as a defense mechanism in industrial environments. It examines several performance factors discussed in earlier studies, including detection accuracy, latency, scalability, and network overhead. This study emphasizes both the benefits and the limitations associated with using SDN-based security solutions in these environments.
The results show that SDN offers a number of advantages for improving network security, and it gives better visibility of network happenings. It allows centralized control of the network and facilitates security policies to be updated more easily when threats occur. These capabilities make SDN a promising method for improving security in ICS. However, the analysis also emphasizes that the successful use of SDN in industrial settings depends on several important considerations. These involve the applicable settings of the SDN controller, effective prioritization of network traffic, and the use of security mechanisms that are capable of understanding industrial communication protocols.
From this review, researchers and industry practitioners are able to develop and evaluate security frameworks for ICS environments that include SDN tools. By bringing together results from previous work, it can help to explain how SDN can successfully strengthen the security and reliability of industrial networks.

6.2 Future Research Directions

Although the literature indicates progress has been made, numerous unresolved research challenges persist. Future research needs to create complete systematic frameworks that simultaneously embrace various industrial protocols and different traffic relationships, mirroring actual ICS implementations. To evaluate possible delays and the consistency of security mechanisms in industrial operations, it is important to ensure that these systems comply with true requirements for timeliness and safety. The assessments conducted should not rely only on average performance metrics but must consider worst-case scenarios and operational reliability.
In addition, future studies should evaluate proposed solutions on industrial testbeds and large-scale experimental platforms that carefully reproduce real-world environments. Such platforms would make it possible to identify differences between simulation results and actual system behavior. The conditions, such as attack scenarios, normal functioning traffic, and potential system failures, should be considered in these environmental settings in order to provide a more accurate assessment of security mechanisms.
Another important future direction includes the use of distributed and edge-maintained SDN designs to address scalability and latency problems that are often linked with centralized controllers. Through the combination of certain detection and enforcement functions to control units while maintaining the global coordination through higher-level controllers, it is possible to get faster

response times and improved resilience without losing centralized network visibility.
Additionally, there is a need to investigate machine learning methodologies that are strong, interpretable, and suitable for combination within SDN control systems. The research should also look at the ways of reducing computational overhead, improving resilience to adversarial interference, and improving simulation interpretability. These methodologies are important as they make sure that system operators understand and trust automated decisions in key infrastructure environments.
In order to improve the SDN-based security approaches for ICS, a combination of automation and policy validation mechanisms. This includes formal verification, real-time policy checking, and operator assisted control system to make sure that automated mitigation actions do not unintentionally disrupt industrial processes or compromise operational safety.

6.3 Final Remarks

SDN represents a powerful yet complex foundation for enhancing ICS security. Its successful deployment requires careful system modelling, quantitative performance evaluation, and close alignment with industrial operational constraints. By highlighting systematic rigor and engineering relevance, this paper aims to contribute to the development of more reliable, secure, and deployable SDN-based security solutions for industrial control networks.

References:

[1] ISA, "ISA/IEC 62443 Series of Standards." Accessed: Dec. 18, 2025. [Online]. Available: https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards.

[2] Modbus Organization, "Modbus Specifications." [Online]. Available: https://www.modbus.org/modbus-specifications. Accessed: Dec. 18, 2025.

[3] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing," Jun. 2014. Available: https://doi.org/10.17487/RFC7230.

[4] R. J. Rodríguez, S. Marrone, I. Marcos, and G. Porzio, "MOSTO: A toolkit to facilitate security auditing of ICS devices using Modbus/TCP," *Comput. Secur.*, vol. 132, p. 103373, Sep. 2023, https://doi.org/10.1016/j.cose.2023.103373.

[5] Modbus Organization, "MODBUS/TCP Security Protocol Specification," Jul. 2021. [Online]. Available: https://assets.noviams.com/novi-file-uploads/modbus/pdfs-and-documents/MB-TCP-Security-v36_2021-07-30.pdf. Accessed: Dec. 18, 2025.

[6] X. Etxezarreta, I. Garitano, M. Iturbe, and U. Zurutuza, "Software-Defined Networking approaches for intrusion response in Industrial Control Systems: A survey," *Int. J. Crit. Infrastruct. Protect*, vol. 42, p. 100615, Sep. 2023, https://doi.org/10.1016/j.ijcip.2023.100615.

[7] X. Le, J. Li, Y. Zhao, and Z. Fan, "A Security-Enhanced Scheme for ModBus TCP Protocol Based on Lightweight Cryptographic Algorithm," *Electronics*, vol. 14, no. 18, Art. no. 3674, Sep. 2025, https://doi.org/10.3390/electronics14183674.

[8] A. Nadhum and A. Al-Saadi, "Exploring the Landscape of Software Defined Networking: A Comprehensive Survey of Applications, Challenges, and Future Directions," *AIP Conference Proceedings*, vol. 3169, Feb. 10, 2025, Art. no. 030018, https://doi.org/10.1063/5.0254293.

[9] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for Industrial Communication Systems," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005, https://doi.org/10.1109/JPROC.2005.849714.

[10] J. C. Correa Chica, J. C. Imbachi, and J. F. Botero Vega, "Security in SDN: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 159, p. 102595, Jun. 2020, https://doi.org/10.1016/j.jnca.2020.102595.

[11] M. Abdallah, S. Bagchi, S. D. Bopardikar, K. Chan, X. Gao, M. Kantarcioglu, C. Li, P. Liu, Q. Zhu, "Game Theory in Distributed Systems Security: Foundations, Challenges,

and Future Directions," arXiv preprint arXiv:2309.01281, 2024, https://arxiv.org/abs/2309.01281. Accessed: Dec. 18, 2025.

[12] M. R. Ahmed, S. Islam, S. Shatabda, A. K. M. M. Islam, and M. T. I. Robin, "Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques: A Comprehensive Survey," TechRxiv preprint, Dec. 10, 2021, https://doi.org/10.36227/techrxiv.17153213.v1. Accessed: Dec. 18, 2025

[13] K. M. R. Mokoena, R. L. Moila, and P. M. Velempini, "Improving Network Management with Software Defined Networking using OpenFlow Protocol," in *2023* International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, Aug. 2023, pp. 1–5, https://doi.org/10.1109/icABCD59051.2023.10220519.

[14] J. Kim, J. H. Yun, and H. C. Kim, "Anomaly Detection for Industrial Control Systems Using Sequence-to-Sequence Neural Networks," in *Computer Security*, Lecture Notes (LNCS) in Computer Science, Springer, Cham, Feb. 22, 2020, pp. 3–18, https://doi.org/10.1007/978-3-030-42048-2_1.

[15] Z. Chen, J. Liu, Y. Shen, M. Simsek, B. Kantarci, H. T. Mouftah and P. Djukic, "Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, May 2023, https://doi.org/10.1145/3530812.

[16] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial Internet of Things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, Oct. 2023, https://doi.org/10.1016/j.aej.2023.09.023.

[17] N. E. Petroulakis, K. Fysarakis, I. Askoxylakis, and G. Spanoudakis, "Reactive security for SDN/NFV-enabled industrial networks leveraging service function chaining," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 7, Art. no. e3269, Jul. 2018, https://doi.org/10.1002/ett.3269.

[18] S. A. More and A. V Kachavimath, "SDN Intrusion Detection using Meta-Heuristic Optimization and K-Nearest Neighbors Classifier," *Procedia Comput. Sci.*, vol. 260, pp. 1137–1144, 2025, https://doi.org/10.1016/j.procs.2025.03.299.

[19] J. Suchorab, S. Plamowski, and M. Ławryńczuk, "Anomaly detection system for Modbus data based on an open source tool," *Comput. Secur.*, vol. 157, p. 104572, Oct. 2025, https://doi.org/10.1016/j.cose.2025.104572.

[20] H. Kheddar, Y. Himeur, and A. I. Awad, "Deep transfer learning for intrusion detection in industrial control networks: A comprehensive review," *Journal of Network and Computer Applications*, vol. 220, p. 103760, Nov. 2023, https://doi.org/10.1016/j.jnca.2023.103760.

[21] B. R. Ray, M. U. Chowdhury, and J. H. Abawajy, "A Multi-protocol Security Framework to Support Internet of Things," in *Security in Smart Cities*: *Models, Applications, and Challenges*, Springer, Cham, Jun. 14, 2017, pp. 257–270, https://doi.org/10.1007/978-3-319-59608-2_14.

[22] X. Qin, R. Doss, F. Jiang, X. Qin, and B. Long, "Securing ICS networks: SDN-based Automated Traffic Control and MTD Defensive Framework against DDoS attacks," *Comput. Commun.*, vol. 241, p. 108252, Sep. 2025, https://doi.org/10.1016/j.comcom.2025.108252.

[23] P. Ohri and S. G. Neogi, "Software-Defined Networking Security Challenges and Solutions: A Comprehensive Survey," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 383–400, Jul. 2022, https://doi.org/10.12785/ijcds/120131.

[24] P. Mahadevappa, R. K. Murugesan, R. Al-amri, R. Thabit, A. H. Al-Ghushami, and G. Alkawsi, "A secure edge computing model using machine learning and IDS to detect and isolate intruders," *MethodsX*, vol. 12, p. 102597, Jun. 2024, https://doi.org/10.1016/j.mex.2024.102597.

[25] Claroty, "The 2026 Cybersecurity Guide to Industrial Control Systems," Apr. 16, 2025. [Online]. Available: https://claroty.com/blog/cybersecurity-dictionary-industrial-control-systems-ics-security. Accessed: Jan. 08, 2026.

[26] S. Usman, I. Winarno, and A. Sudarsono, "Implementation of SDN-based IDS to protect Virtualization Server against HTTP DoS attacks," in *2020 International Electronics Symposium (IES)*, Surabaya, Indonesia, Sep. 2020, pp. 195–198, https://doi.org/10.1109/IES50839.2020.9231699.

[27] F. Zobary, "Optimizing SDN Controller to Switch Latency for Controller Placement Problem," *Informatica*, vol. 48, no. 8, pp. 165–176, Jun. 2024, https://doi.org/10.31449/inf.v48i8.5846.

[28] D. Um, H.-S. Park, H. Ryu, and K.-J. Park, "Predictive Forwarding Rule Caching for Latency Reduction in Dynamic SDN," *Sensors*, vol. 25, no. 1, Art. no. 155, Dec. 2024, https://doi.org/10.3390/s25010155.

[29] A. Almalawi, S. Hassan, A. Fahad, A. Iqbal, and A. I. Khan, "Hybrid Cybersecurity for Asymmetric Threats: Intrusion Detection and SCADA System Protection Innovations," *Symmetry*, vol. 17, no. 4, Art. no. 616, Apr. 2025, https://doi.org/10.3390/sym17040616.

[30] H. N. Nguyen, M. Girdhar, Y.-H. Kim, and J. Hong, "Machine-Learning-Based Anomaly Detection for GOOSE in Digital Substations," *Energies*, vol. 17, no. 15, Art. no. 3745, Jul. 2024, https://doi.org/10.3390/en17153745.

[31] A. Hidouri, N. Hajlaoui, H. Touati, M. Hadded, and P. Muhlethaler, "A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking," *Computers*, vol. 11, no. 12, Art. no. 186, Dec. 2022, https://doi.org/10.3390/computers11120186.

[32] V. G. da Silva Ruffo, D. M. B. Lent, M. Komarchesqui, V. F. Schiavon, M. V. O. de Assis, L. F. Carvalho, and M.L Proenca Jr, "Anomaly and intrusion detection using deep learning for software-defined networks: A Survey," Expert Systems with Applications, vol. 256, Art. no. 124982, Dec. 2024, https://doi.org/10.1016/j.eswa.2024.124982.

[33] K. Ahmed, J. O. Blech, M. A. Gregory, and H. W. Schmidt, "Software Defined Networks in Industrial Automation," *Journal of Sensor and Actuator Networks*, vol. 7, no. 3, Art. no. 33, Aug. 2018, https://doi.org/10.3390/jsan7030033.

[34] S. K. Dey and Md. M. Rahman, "Flow Based Anomaly Detection in Software Defined Networking: A Deep Learning Approach With Feature Selection Method," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, Dhaka, Bangladesh, Sep. 2018, pp. 630–635, https://doi.org/10.1109/CEEICT.2018.8628069.

[35] R. Wainwright, M. Bagheri, A. Salama, and R. Saatchi, "Software-Defined Networking Security Detection Strategies and Their Limitations with a Focus on Distributed Denial-of-Service for small to medium-sized enterprises," *Applied Sciences*, vol. 15, no. 23, Art. no. 12389, Nov. 2025, https://doi.org/10.3390/app152312389.

[36] S. Moghanian, F. B. Saravi, G. Javidi, and E. O. Sheybani, "GOAMLP: Network Intrusion Detection with Multilayer Perceptron and Grasshopper Optimization Algorithm," *IEEE Access*, vol. 8, pp. 215202–215213, Nov. 2020, https://doi.org/10.1109/ACCESS.2020.3040740.

[37] G. M. Rao and D. Ramesh, "A Hybrid and Improved Isolation Forest Algorithm for Anomaly Detection," *in Soft Computing and Signal Processing*, Singapore, Springer, Oct. 2020, pp. 589–598, https://doi.org/10.1007/978-981-15-7234-0_55.

[38] S. R. Mishra, B. Shanmugam, K. C. Yeo, and S. Thennadil, "SDN-Enabled IoT Security Frameworks: A Review of Existing Challenges," *Technologies*, vol. 13, no. 3, Art. no. 121, Mar. 2025, https://doi.org/10.3390/technologies13030121.

[39] G. R. M. Raman, C. M. Ahmed, and A. Mathur, "Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation," *Cybersecurity*, vol. 4, no. 1, Art. no. 27, Dec. 2021, https://doi.org/10.1186/s42400-021-00095-5.

[40] B. Spoorthi and D. Harekal, "SCADA Security in Various Industrial Sectors," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, May 2018, pp. 339–346, https://doi.org/10.1109/RTEICT42901.2018.9012395.

[41] M. A. Al-Shareeda, A. Abdullah Alsadhan, H. H. Qasim, and S. Manickam, "Software defined networking for internet of things: review, techniques, challenges, and future directions," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 638–647, Feb. 2024, https://doi.org/10.11591/eei.v13i1.6386.

[42] J. Arevalo-Herrera, J. Camargo Mendoza, J. I. Martínez Torre, T. Zona-Ortiz, and J. M. Ramirez, "Assessing SDN Controller Vulnerabilities: A Survey on Attack Typologies, Detection Mechanisms, Controller Selection, and Dataset Application in Machine Learning," *Wirel. Pers. Commun.*, vol. 140, no. 1–2, pp. 739–775, Jan. 2025, https://doi.org/10.1007/s11277-025-11748-w.

[43] A. Houkan, A. K. Sahoo, S. P. Gochhayat, and P. K. Sahoo, "Artificial intelligence approach to intrusion detection in industrial control systems with real world dataset generation and model evaluation," *Discover Artificial Intelligence*, vol. 5, Art. no. 307, Nov. 2025, https://doi.org/10.1007/s44163-025-00507-2.

[44] X. Ni, S. Jiang, K. Yu, C. An, Y. Zhang, and H. Huang, "Smart Grid Intrusion Detection System Based on Incremental Learning," *Electronics*, vol. 14, no. 19, Art. no. 3820, Sep. 2025, https://doi.org/10.3390/electronics14193820.

[45] A. H. Janabi, T. Kanakis, and M. Johnson, "Survey: Intrusion Detection System in Software-Defined Networking," *IEEE Access*, vol. 12, pp. 164097–164120, 2024, https://doi.org/10.1109/ACCESS.2024.3493384.

[46] N. M. Yungaicela-Naula, C. Vargas-Rosales, J. A. Pérez-Díaz, and M. Zareei, "Towards security automation in Software Defined Networks," *Comput. Commun.*, vol. 183, pp. 64–82, Feb. 2022, https://doi.org/10.1016/j.comcom.2021.11.014.

[47] G. Logeswari, S. Bose, and T. Anitha, "An Intrusion Detection System for SDN Using Machine Learning," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 867–880, 2023, https://doi.org/10.32604/iasc.2023.026769.

[48] E. Sogut, A. Tekerek, and O. A. Erdem, "Machine Learning-Based DDoS Attack Detection on SDN-Based SCADA Systems," *Gazi Journal of Engineering Sciences*, vol. 9, no. 3, pp. 596–611, Dec. 2023, https://doi.org/10.30855/gmbd.0705090.

[49] L. Rajesh and P. Satyanarayana, "Detecting Flooding Attacks in Communication Protocol of Industrial Control Systems," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 396–401, 2020, https://doi.org/10.14569/IJACSA.2020.0110149.

[50] A. Abuarqoub, "A Review of the Control Plane Scalability Approaches in Software Defined Networking," *Future Internet*, vol. 12, no. 3, Art. no. 49, Mar. 2020, https://doi.org/10.3390/fi12030049.

[51] M. J. Islam, A. Rahman, S. Kabir, M. R. Karim, U. K. Acharjee, M. K. Nasir, S. S. Band, M. Sookhak, and S. Wu, "Blockchain-SDN-Based Energy-Aware and Distributed Secure Architecture for IoT in Smart Cities," *IEEE Internet Things Journal*, vol. 9, no. 5, pp. 3850–3864, Mar. 2022, https://doi.org/10.1109/JIOT.2021.3100797.

[52] K. Ahmed, N. S. Nafi, J. O. Blech, M. A. Gregory, and H. Schmidt, "Software defined industry automation networks," in *27th International Telecommunication*

Networks and Applications Conference (ITNAC), Melbourne, Australia, Nov. 2017, pp. 1–3, https://doi.org/10.1109/ITNAC41115.2017.

[53] A. T. J. Akem, M. Gucciardo, and M. Fiore, "Ultra-Low Latency User-Plane Cyberattack Detection in SDN-based Smart Grids," in *The 15th ACM International Conference on Future and Sustainable Energy Systems*, New York, NY, USA: *ACM*, 2024, pp. 676–682, https://doi.org/10.1145/3632775.3661995.

[54] G. D. Singh, V. Tripathi, A. Dumka, R. S. Rathore, M. Bajaj, J. Escorcia-Gutierrez, N. O. Aljehane, V. Blazek, and L. Prokop, "A novel framework for capacitated SDN controller placement: Balancing latency and reliability with PSO algorithm," *Alexandria Engineering Journal*, vol. 87, pp. 77–92, Jan. 2024, https://doi.org/10.1016/j.aej.2023.12.018.

[55] A. Abderrahmane, H. Drid, and A. Behaz, "A Survey of Controller Placement Problem in SDN-IoT Network," *International Journal of Networked and Distributed Computing*, vol. 12, no. 2, pp. 170–184, Jul. 2024, https://doi.org/10.1007/s44227-024-00035-y.

[56] T. Magara and Y. Zhou, "Internet of Things (IoT) of Smart Homes: Privacy and Security," *Journal of Electrical and Computer Engineering*, vol. 2024, pp. 1–17, Apr. 2024, https://doi.org/10.1155/2024/7716956.

[57] M. S. Ataa, E. E. Sanad, and R. A. El-Khoribi, "Intrusion detection in software defined network using deep learning approaches," *Scientific Reports*, vol. 14, Art. no. 29159, 2024, https://doi.org/10.1038/s41598-024-79001-1.

[58] J. M. Storm, S. H. Houmb, P. Kaliyar, L. Erdodi, and J. M. Hagen, "Testing Commercial Intrusion Detection Systems for Industrial Control Systems in a Substation Hardware in the Loop Testlab," *Electronics*, vol. 13, no. 1, Art. no. 60, 2023, https://doi.org/10.3390/electronics13010060.

[59] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018, https://doi.org/10.1016/j.patcog.2018.07.023.

[60] D. Bhamare, M. Zolanvari, A. Erbad, R. Jain, K. Khan, and N. Meskin, "Cybersecurity for industrial control systems: A survey," *Comput. Secur.*, vol. 89, Art. no. 101677, Feb. 2020, https://doi.org/10.1016/j.cose.2019.101677.

[61] B. B. Bulle, A. O. Santin, E. K. Viegas, and R. R. dos Santos, "A Host-based Intrusion Detection Model Based on OS Diversity for SCADA," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Oct. 2020, pp. 691–696, https://doi.org/10.1109/IECON43393.2020.9255062.

[62] M. R. Asghar, Q. Hu, and S. Zeadally, "Cybersecurity in industrial control systems: Issues, technologies, and Challenges." *Computer Networks*, vol. 165, Art. no. 106946, Dec. 2019, https://doi.org/10.1016/j.comnet.2019.106946.

[63] A. Dehlaghi-Ghadim, A. Balador, M. H. Moghadam, H. Hansson, and M. Conti, "ICSSIM — A framework for building industrial control systems security testbeds," *Comput. Ind.*, vol. 148, p. 103906, Jun. 2023, https://doi.org/10.1016/j.compind.2023.103906.

[64] NIST, "Guide to Operational Technology (OT) Security, SP 800-82 Rev. 3," Sep. 2023. Accessed: Dec. 18, 2025. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/82/r3/final.

[65] E. A. Lee and S. A. Seshia, "Introduction to Embedded Systems - A Cyber-Physical Systems Approach (Errata; V.2.2; Errata_all; V1.08; V2.0)," pp. 1–519, 2017, Accessed: Dec. 18, 2025. [Online]. Available: https://ptolemy.berkeley.edu/books/leeseshia/releases/LeeSeshia_DigitalV2_2.pdf.

[66] A. Derhab, M. Guerroumi, A. Gumaei, L. Maglaras, M. A. Ferrag, M. Mukherjee, and F. A. Khan, "Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial IoT Security," *Sensors*, vol. 19, no. 14, Jul. 2019, Art. no. 3119, https://doi.org/10.3390/s19143119.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Mupatiwa Zulu conducted the literature review, designed and implemented the SDN-based simulation framework, performed the experiments, analyzed the results, and prepared the manuscript.

Dr. Lukumba Phiri provided academic supervision, research guidance, technical review of the study, and reviewed the final manuscript.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This research received no external funding. The study was conducted purely for academic purposes as part of the requirements for the author's Master's degree program.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

APPENDIX

Appendix A: Supporting SDN-based ICS Security Frameworks

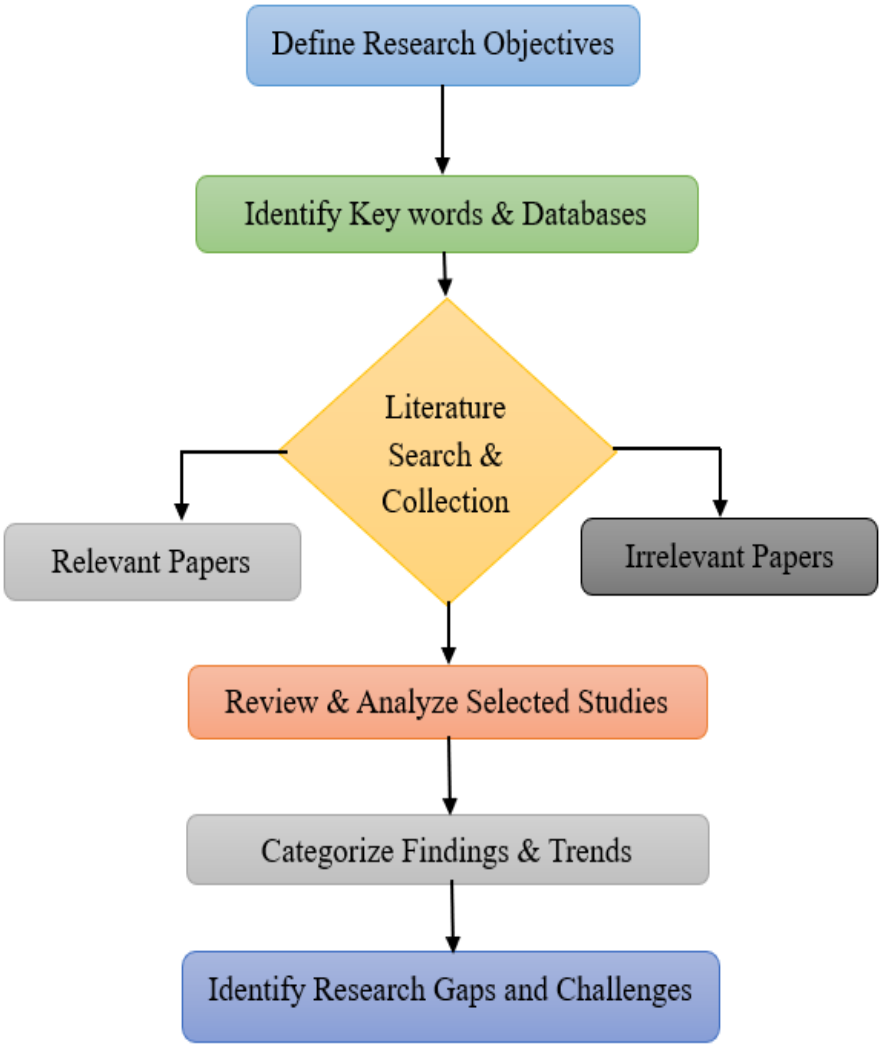

Fig. 0.1. SDN-based ICS security research Flowchart
Source: Created by the authors

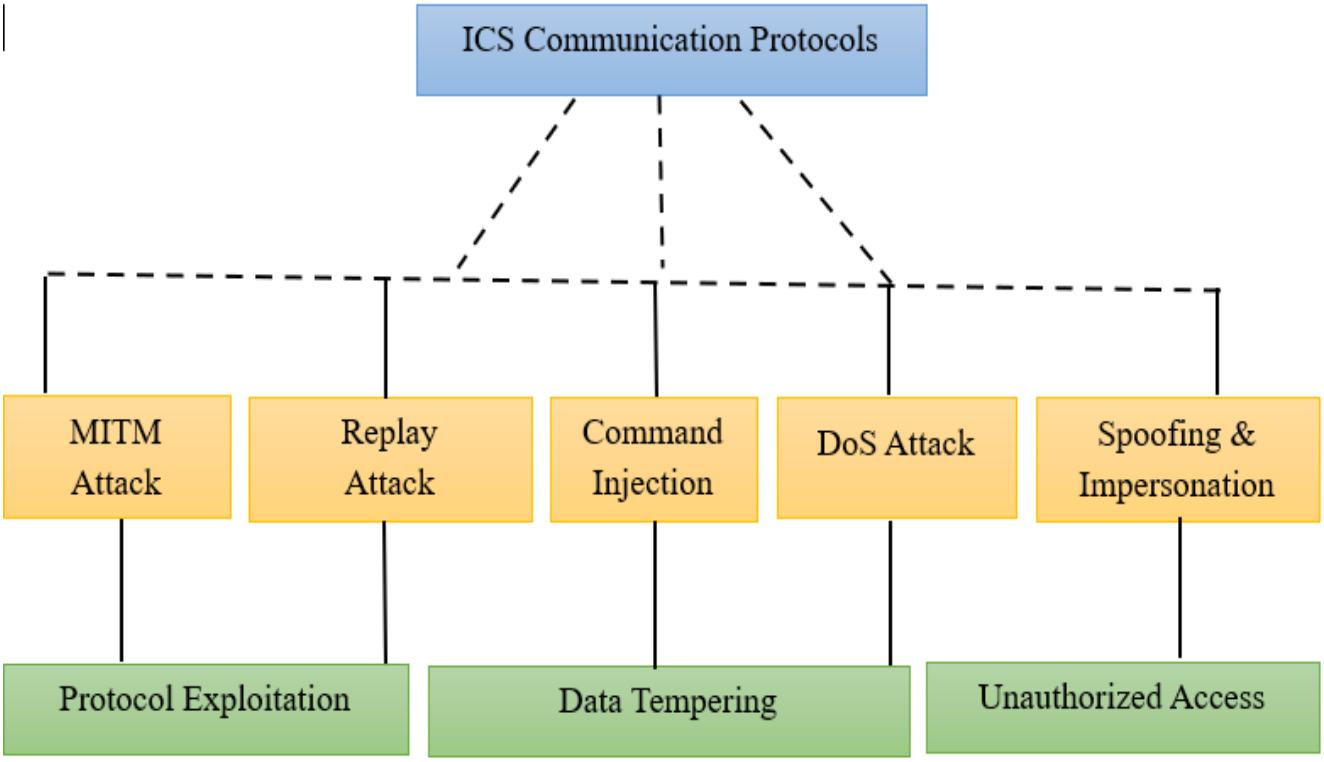

Fig. 0.2. Frequent Attack Methods Aiming at ICS Communication Protocols
Source: Created by the authors

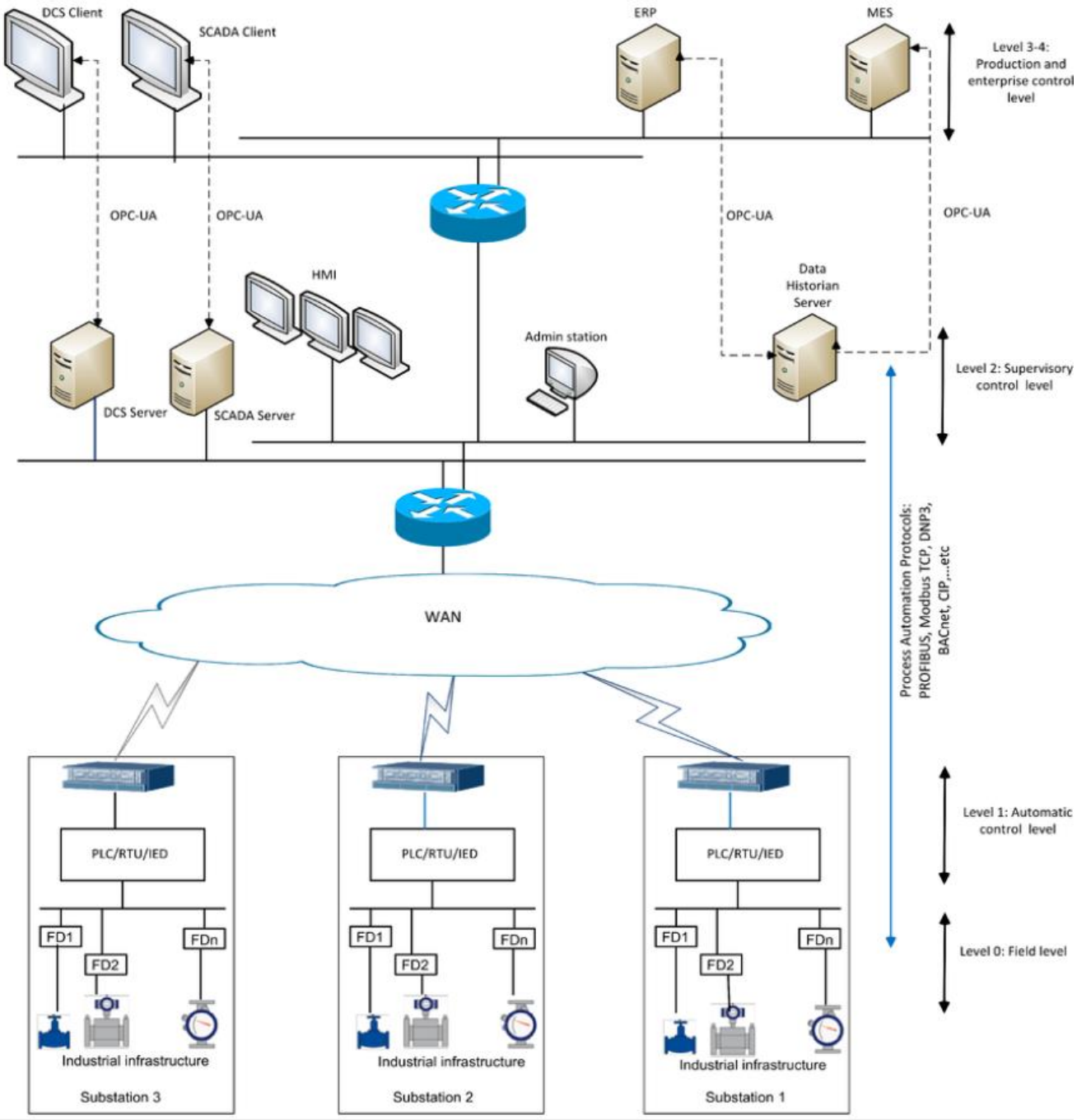

Fig. 0.3. Security architecture for SDN-based ICS Networks
Source: [66]

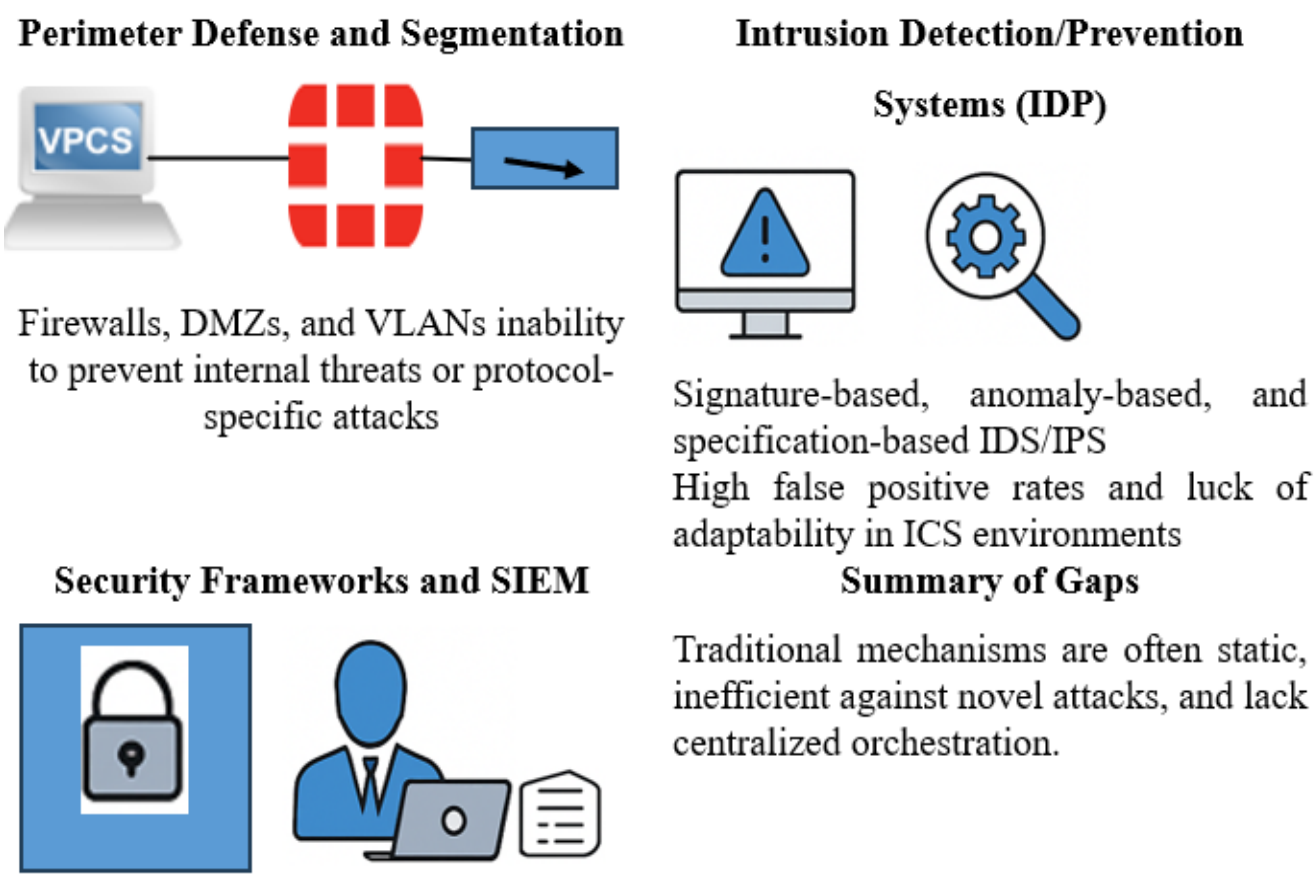

Fig. 0.4. Conventional ICS security mechanisms and limitations.
Source: Created by the authors

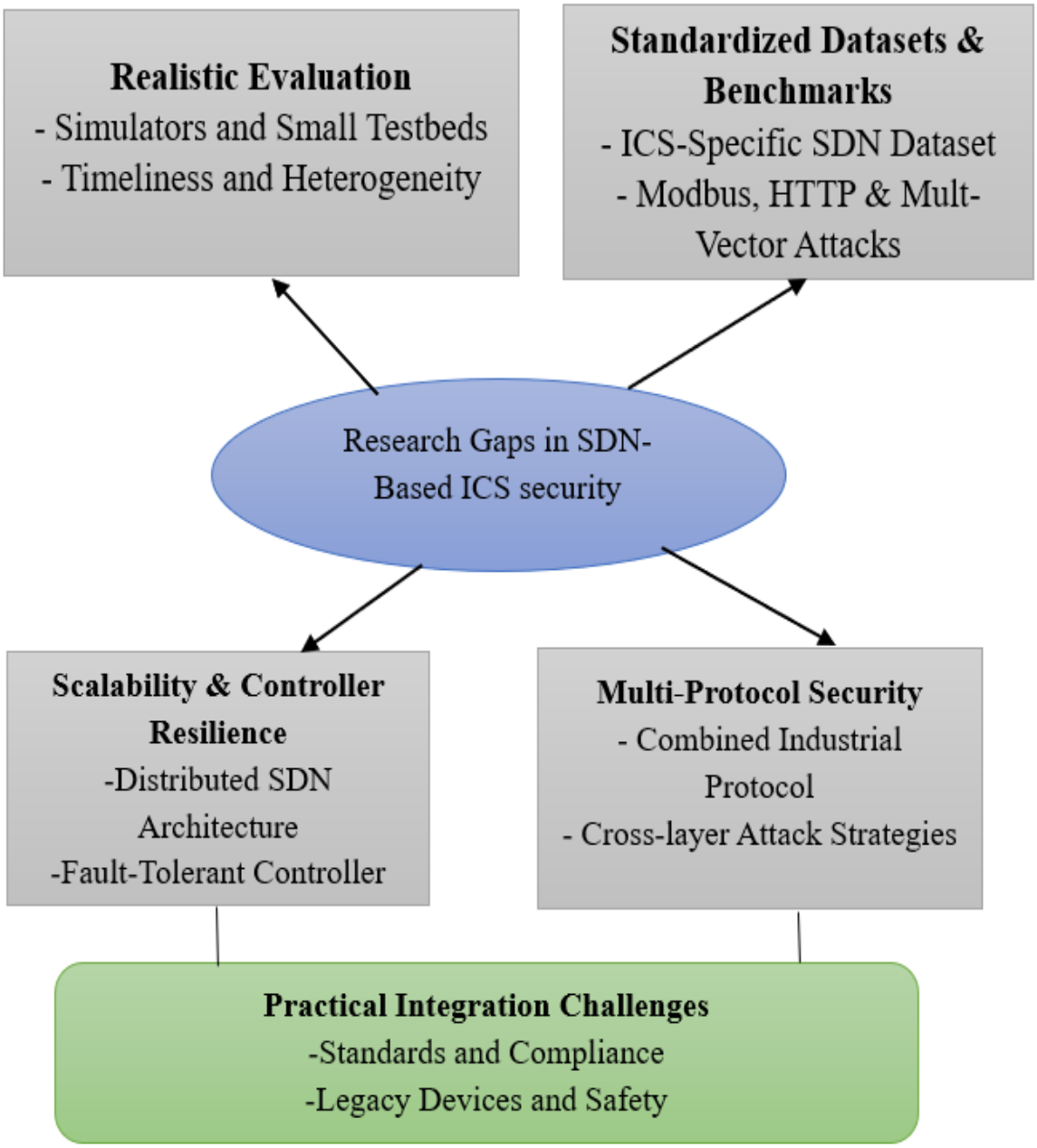

Fig. 0.5. Identified Research Gaps in SDN-Based ICS Security
Source: Created by the authors

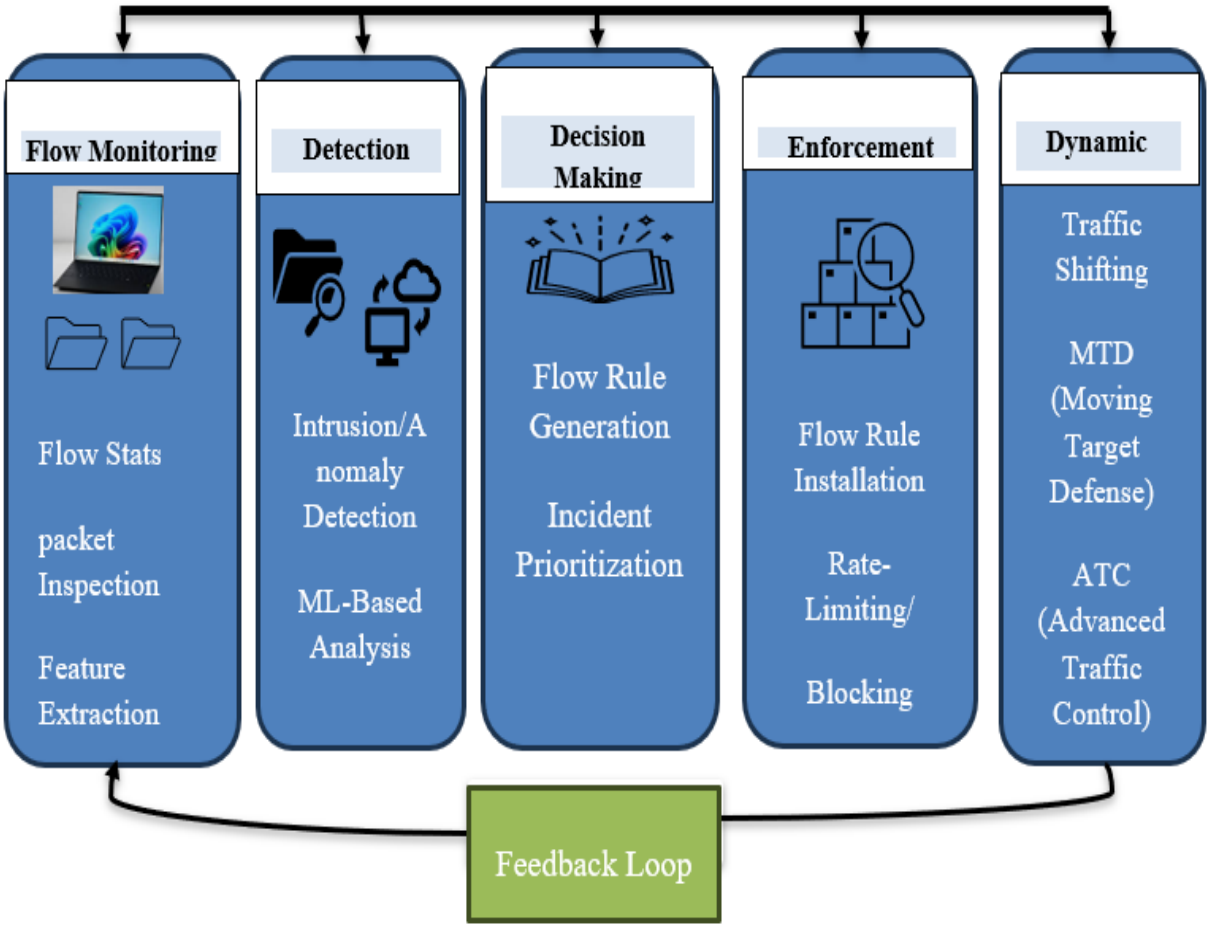

Fig. 0.6. SDN Defense Pipeline
Source: Created by the authors