Using Decision Support to Fortify Industrial Control System against Cyberattacks

Alireza Dehlaghi-Ghadim
and Niclas Ericsson
Research Institutes of Sweden,
Mälardalen University
{alireza.dehlaghi.ghadim,
niclas.ericsson}@ri.se,mdu.se

Lars-Göran Magnusson
and Mats Eriksson
Arctos Labs Scandinavia AB
{lars-goran.magnusson,
mats.eriksson}@arctoslabs.com

Mahshid Helali Moghadam,
Ali Balador and Hans Hansson
Mälardalen University
{mahshid.helali
ali.balador, hans.hansson}@mdu.se

***Abstract*—This paper presents a cybersecurity solution designed to fortify Industrial Control Systems (ICS) against cyberattacks. The proposed solution integrates a Network-based Intrusion Detection System (NIDS) with a Decision Support System (DSS), leveraging machine learning to detect anomalies in network data and employing a filtering mechanism to reduce false alarms. The NIDS protects a simulated ICS testbed, detecting anomalies and forwarding them to the DSS for further analysis and selection of mitigation strategies. We outline the system architecture and showcase promising outcomes from a prototype implementation. Our proof of concept evaluation demonstrates high accuracy in detecting attack scenarios. Challenges such as detection delays between attacks and potential mitigations highlight areas for future improvement. This research contributes to bridging the gap between ML-based IDS and security solutions, paving the way for enhanced cybersecurity in ICS environments.**

***Index Terms*—Intrusion Detection Systems, Industrial Control Systems, Decision Support Systems, Machine Learning**

I. INTRODUCTION

Industrial Control Systems (ICS) are increasingly integrating external systems for purposes like remote monitoring and using cloud services [1], which elevates their vulnerability to cyberattacks [2]. Notable incidents like the BlackEnergy3 [3] and Triton [4] attacks highlight the urgent need for robust ICS security measures [5], bringing cybersecurity to the forefront of research in this field [6]. A critical component of ICS cybersecurity is the integration of Intrusion Detection Systems (IDS) to detect and provide a basis for mitigating attacks. However, most IDS practices rely on analyzing device logs, requiring deep system knowledge [1], and still struggle with the complexity of modern networks and fail to detect zero-day attacks. To address this, Machine Learning (ML) based techniques for automated feature learning have emerged [7].

The integration of ML-based IDS with monitoring solutions like Decision Support Systems (DSS) in industrial settings holds the promise of significant synergies. These include improved security operations, enhanced threat intelligence, automated responses, and predictive analytics, all of which facilitate incident response. However, there is a notable lack of reports on the practical implementation of ML-based IDS in industrial environments and their integration with DSSs. Implementing ML-based IDS poses several challenges, including increased false alarms and notable delays in attack detection. Additionally, integrating IDS with DSS to enhance decision-making and streamline incident response adds another layer of complexity. Bridging ML-based IDS and security solutions poses challenges like semantic gaps[1], interoperability, and real-time constraints. Therefore, more practical exploration is needed to validate the feasibility and benefits of this integration in industrial environments.

In this work, we implement a Network-based IDS (NIDS) with a simulated ICS, connecting it to a DSS to complement the NIDS functionality. This integration facilitates real-time attack detection and response, thereby strengthening cybersecurity defenses and improving attack-handling capabilities. The system continuously monitors the ICS's security status of the ICS and instantly notifies the DSS of potential cyberattacks. Additionally, the DSS provides actionable insights on mitigation strategies for the ICS, enabling effective responses to mitigate potential damages from cyberattacks. Finally, we highlight critical deployment challenges in an industrial environment by evaluating the system across various cyberattack scenarios and provide future research directions.

II. SYSTEM OVERVIEW

The goal of this work is to advance cybersecurity within ICS by analyzing the traffic of the internal network within a representative ICS environment. Through continuous monitoring of network activity, the system can pinpoint anomalies as candidates for misconfigurations, malicious behavior, or cyberattacks. Upon detection, these anomalies are then communicated to the Security Operation Center (SOC), enabling timely response and mitigation measures. We have designed and implemented seven modules to realize this goal, including ICS, data collection module, anomaly detection module, and the DSS, as depicted in Figure 1.

A. Industrial Control System

The industrial control system module simulates a realistic ICS, a bottle-filling factory that includes various sensors, actuators, and controllers, generating authentic industrial network

[1] ML models trained on generic datasets may struggle to understand the context and semantics of industrial data

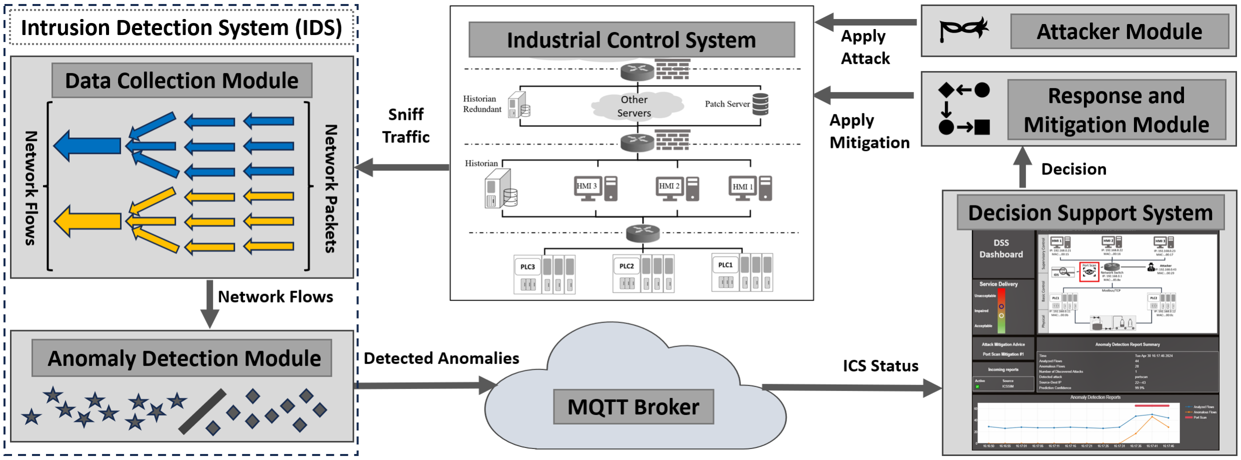


Fig. 1: The cybersecurity solution architecture and attacker module used in our evaluation.

traffic. We implemented the factory simulation testbed using the ICSSIM framework [8]. This simulation comprises ICS components and an emulated ICS network, utilizing the Modbus protocol for communication between ICS components. The simulation includes two Programmable Logic Controllers (PLCs) responsible for overseeing equipment in the bottle-filling factory. Additionally, three Human Machine Interfaces (HMIs) communicate with the PLCs to monitor the system.

B. Attacker

The attacker module implements various cyberattacks in the network, enabling us to demonstrate integrated IDS and DSS functionality. We developed the attacker module to act as a backdoor vulnerability within the ICS system. This module receives attack scenarios remotely via the MQTT broker and executes them on the ICS infrastructure. Using predefined attack types from ICSSIM tool [8] , we applied several attack scenarios, including Man-in-the-Middle (MitM), Distributed Denial of Service (DDoS), Replay, and Scan attacks.

C. Data Collection

The data collection module collects network traffic data, preprocesses collected data, and generates network flows for further investigation. To address the complexity of analyzing all network packets, we utilized a widely-used network monitoring technique: converting packets into network flows, which summarize connection details and payloads [9], using the ICSFlowGenerator tool [10]. This tool employs a fixed time interval for generating network flows. We consider this interval to be 500 milliseconds to ensure the capture of one or more rounds of ICS communication, given that loop periods of PLCs and HMIs are set at 200 and 500 milliseconds, respectively. Our captured network flows comprised 48 features after excluding address and time data, providing essential connection headers and network statistics for anomaly detection.

D. Anomaly Detection

The anomaly detection module scrutinizes network flows, identifies potential attacks, and relays the data to the DSS. However, due to the substantial influx of network flows, it is impractical to display all analyses in the DSS directly, even in a small-scale ICS. Furthermore, false alarms persist as a challenge, even with minor anomaly detection inaccuracies. To mitigate these challenges, we have implemented a filtering mechanism using a voting technique.

We developed a two-stage attack detection system. In the first stage, we deployed a pre-trained Multilayer Perceptron (MLP) model to classify incoming network flows as normal or potentially including different cyberattack categories, flagging suspicious activity. We conducted a hyperparameter grid search to find the best configuration. The optimal model was a 3-layered Multilayer Perceptron (MLP) with 48, 128, and 128 node configurations, using the ’Relu’ activation function, which yielded an accuracy of 98.1% on the test dataset. Although this accuracy is commendable for anomaly detection, it is lacking for intrusion detection, as even a 1.9% flaws in identifying anomalous network flows can result in numerous false alarms, or undetected attacks. To mitigate this issue, we implemented a filtering mechanism employing voting over an extended period in the subsequent stage.

In the second stage, we monitor network flows within specific network links (the same source, destination, and protocol) for a time window greater than the network flow period. (In our experiments, 5 seconds). If most network flows within the time window vote for a cyberattack, the detector confirms it. Every 5 seconds, it sends a JSON status message to the DSS, detailing network status, suspicious flows, and any detected attacks. Figure 2 illustrates the IDS component, which includes both the data collection module and the Two-Stage Attack Detector, while Figure 3 shows a sample message to the DSS.

E. Decision Support System

When an ICS is exposed to cyberattacks, it is important to initiate mitigation actions to preserve its behavior. Selecting the most suitable mitigating action to perform on an ICS is challenging, with respect to the current operational conditions. The relations and dependencies between the sensors, actua-

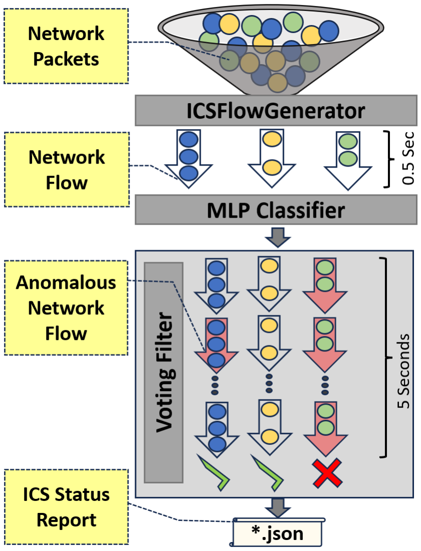


Fig. 2: Intrusion Detection component Overview.

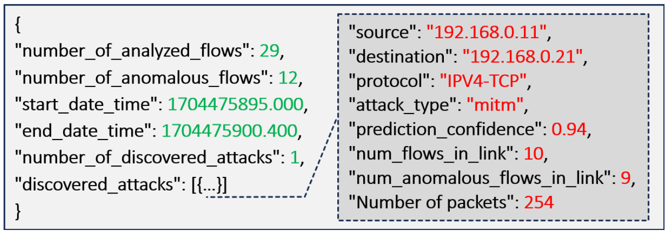

{
"number_of_analyzed_flows": 29,
"number_of_anomalous_flows": 12,
"start_date_time": 1704475895.000,
"end_date_time": 1704475900.400,
"number_of_discovered_attacks": 1,
"discovered_attacks": [{...}]
}
"source": "192.168.0.11",
"destination": "192.168.0.21",
"protocol": "IPV4-TCP",
"attack_type": "mitm",
"prediction_confidence": 0.94,
"num_flows_in_link": 10,
"num_anomalous_flows_in_link": 9,
"Number of packets": 254

Fig. 3: Sample status message sent from Anomaly Detection module to Decision Support System.

tors, controllers, and other elements comprising an ICS are complex. Any action performed on one part of an ICS to mitigate a cyberattack may have ramifications on the overall ICS behavior that are difficult to anticipate. The DSS evaluates the operational conditions and the impact of a cyberattack on the ICS and selects the mitigating action that has the best outcome for the overall ICS behavior.

A fundamental part of the DSS is the *ICS model*, containing a configurable ICS description and a set of supplementing mitigating actions from different perspectives:

- How the purpose of the ICS depends on assets. In an ICS, the assets comprise PLCs, HMIs, sensors, actuators, and other elements.
- How the purpose of the ICS is divided into discrete subsystems such as e.g., *water tank control* and *conveyor belt control*, each with dependencies to a set of assets.
- How different cyberattacks impact the state of an asset and how different mitigation actions impact the state of an asset. The asset state ranges from 'Normal Operation' via 'Partly Degraded' to 'Severely Degraded'.
- How the assets' states and how the relative significance an individual asset has on the ICS impacts the ICS behavior. The ICS behavior ranges from 'Acceptable' via 'Impaired' to 'Unacceptable'.

The *ICS model* is used in DSS to gain insight on, for example, how a mitigating action that has a positive impact on an asset under attack may have a negative impact on another asset. DSS conducts a cost-benefit analysis of the available mitigating actions, taking into account the operational conditions. This produces advice on preferred mitigating action to restore or sustain the best possible overall ICS behavior.

The DSS also contains a dashboard which presents visualizations of significant system events. One panel summarizes the status messages from the Anomaly Detection Module and a chart displays analyzed flows and identified cyberattacks over the past minute. When a cyberattack is detected the selected mitigating action is displayed. The impact on the ICS behavior caused by the cyberattack as well as the impact of the selected mitigating action is also shown in a panel illustrating the different ICS behavior levels.

F. *Response and Mitigation*

The response and mitigation module encompasses predefined mitigation strategies to combat cyberattacks. Upon receiving data from the DSS regarding a cyberattack's detection, nature, and severity, it triggers a signal to the ICS. This signal prompts the ICS to enact a predefined mitigation procedure to contain and neutralize the malicious activity.

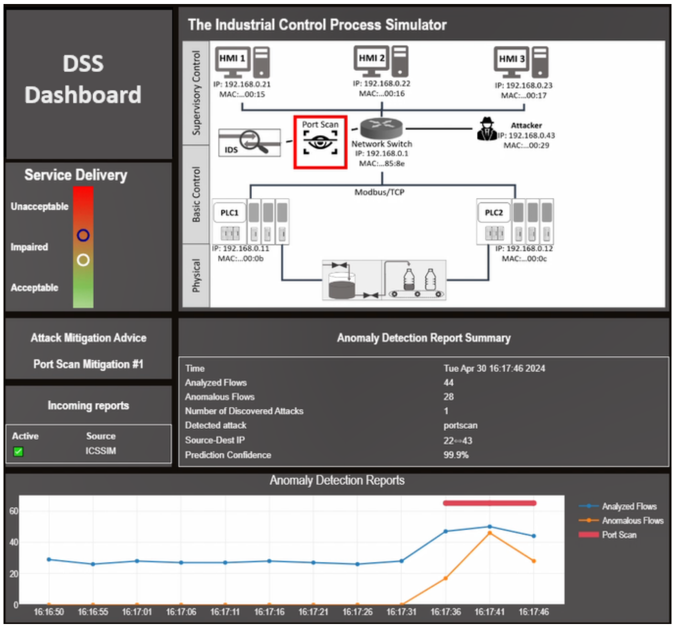
G. *MQTT Broker*

The communication between different modules within this system is facilitated using MQTT (Message Queuing Telemetry Transport), which uses the publish/subscribe paradigm. For instance, the IDS component periodically publishes the status of the ICS on a specific topic, while the DSS module, which subscribes to that topic, receives the status updates. To ensure security, this communication is password-protected and encrypted, enhancing the system's overall security.

III. Demonstration and Discussion

To assess the system, we employ three distinct physical servers. The first server manages ICS simulation within Docker containers and hosts the IDS component. The second server hosts the DSS dashboard, while the third server provides the MQTT Broker. Notably, the IDS component showed promising performance across all experiments by consistently identifying defined attack scenarios. While we acknowledge the potential existence of scenarios undetectable by our model, investigating such cases requires further exploration. Figure 4 shows the DSS dashboard, which presents insights into network flows and identified attack types via reports and diagrams. Additionally, it offers actionable mitigation strategies. Furthermore, DSS evaluates the behavior of ICS service delivery, categorizing it as acceptable, impaired, or unacceptable, and predicts the behavior post-mitigation.

Implementing this solution effectively in an industrial setting presents a few challenges. One challenge is automating the implementation of mitigation advice generated by the DSS. Currently, human intervention is necessary to execute



Attack Mitigation Advice

Port Scan Mitigation #1

Incoming reports

Active	Source
☑	ICSSIM

Anomaly Detection Report Summary

Time	Tue Apr 30 16:17:46 2024
Analyzed Flows	44
Anomalous Flows	28
Number of Discovered Attacks	1
Detected attack	portscan
Source-Dest IP	22⇔43
Prediction Confidence	99.9%

Fig. 4: Decision Support System Dashbord

mitigation strategies within the ICS, relying on expert knowledge. Automating this process would reduce the time between attack detection and performing mitigation. Another significant challenge is transitioning the system from supervised ML methods to unsupervised ones. This upgrade is essential for efficiently detecting new attack types. Currently relying on a predefined list of mitigation strategies for known attack types, the system struggles to find suitable mitigation strategies for new cyberattacks.

Furthermore, the detection delay between the onset of an attack and the system's activation of the mitigation strategy presents a notable challenge. This delay, denoted by t_{delay}, can be calculated using Eq. 1

$$t_{delay} = t_{sniff} + t_{flow} + t_{prep} + t_{vote} + 2 \times t_{link} + t_{dss} \quad (1)$$

In the above formula, t_{sniff} denotes the delay in forwarding network traffic to the data collection module, t_{flow} accounts for the 500ms aggregation time for network packets, t_{prep} represents data preprocessing time, t_{vote} indicates the delay caused by the aggregation of flows for voting, t_{link} signifies the communication time of IDS and DSS with the MQTT Broker, and t_{dss} refers to the internal computation time for DSS to analyze and recommend mitigation strategies. Our analysis shows that t_{delay} spans from 2 to 6 seconds, presenting a challenge for IDS to neutralize cyberattacks promptly. While this model may prove advantageous in combating prolonged cyberattacks such as DDoS attacks, enabling proactive measures to prevent further damage to the system, it falls short in addressing short-lived attacks. In such cases, the system can only notify administrators of the ongoing attack, necessitating rapid mitigation strategies to be deployed post-attack. Examples of strategies include backing up short-term logs or taking system snapshots to allow for restoring functionality swiftly and minimize potential damage.

IV. Conclusion

Our work designed and developed a solution to enhance cybersecurity within ICS by integrating an NIDS with a DSS. Our solution demonstrates promising capabilities in detecting and mitigating cyberattacks within a simulated ICS environment by leveraging ML algorithms for anomaly detection and real-time decision support. While our evaluation showcases high accuracy in detecting various attack scenarios, challenges such as detection delays underscore the need for further optimization and refinement. Future research directions include real-time optimization, enhanced anomaly detection techniques, improved scalability, usability improvements, and the development of adaptable IDS models.

Acknowledgment

This work has been partially supported by the H2020 ECSEL EU project Distributed Artificial Intelligent System (DAIS). DAIS (https://dais-project.eu/) has received funding from the ECSEL JU under grant agreement No 101007273, and also funded by the Knowledge Foundation within the framework of INDTECH (Grant Number 20200132) and INDTECH + Research School project (Grant Number 20220132), participating companies and Mälardalen University.

References

[1] I. Ortega-Fernandez, M. Sestelo, J. C. Burguillo, and C. Pinon-Blanco, "Network intrusion detection system for ddos attacks in ics using deep autoencoders," *Wireless Networks*, pp. 1–17, 2023.

[2] S. A. Varghese, A. D. Ghadim, A. Balador, Z. Alimadadi, and P. Papadimitratos, "Digital twin-based intrusion detection for industrial control systems," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 611–617, IEEE, 2022.

[3] T. Alladi, V. Chamola, and S. Zeadally, "Industrial control systems: Cyberattack trends and countermeasures," *Computer Communications*, vol. 155, pp. 1–8, 2020.

[4] A. Di Pinto, Y. Dragoni, and A. Carcano, "Triton: The first ics cyber attack on safety instrument systems," in *Proc. Black Hat USA*, vol. 2018, pp. 1–26, 2018.

[5] F. Sicard, É. Zamaï, and J.-M. Flaus, "An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems," *Reliability Engineering & System Safety*, vol. 188, pp. 584–603, 2019.

[6] Z. Bakhshi, A. Balador, and J. Mustafa, "Industrial iot security threats and concerns by considering cisco and microsoft iot reference models," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 173–178, 2018.

[7] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *International Journal of Critical Infrastructure Protection*, p. 100516, 2022.

[8] A. Dehlaghi-Ghadim, A. Balador, M. H. Moghadam, H. Hansson, and M. Conti, "Icssim—a framework for building industrial control systems security testbeds," *Computers in Industry*, vol. 148, p. 103906, 2023.

[9] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019.

[10] A. Dehlaghi-Ghadim, M. H. Moghadam, A. Balador, and H. Hansson, "Anomaly detection dataset for industrial control systems," *IEEE Access*, vol. 11, pp. 107982–107996, 2023.