Edge-deployed ML model in Industrial Control Systems

AP Research

Word count: 5184

Introduction

Industrial Control Systems, or (ICS) function as computing devices operating critical infrastructure facilities that society depends on daily, such as power grids, chemical processing plants, and water treatment facilities. These ICS devices are highly specialized computers and sensors allowing essential infrastructure to run safely and effectively. Recently, however, these industrial control systems have been using the internet to complete their tasks at an increased rate, making them more vulnerable to cybersecurity attacks. According to Nankya et al., an Adjunct Professor at Fitchburg State University, "Over the past decade, cyber attacks on Industrial Control Systems have notably increased due to their heightened vulnerability to off-site attacks. Previously, these systems operated in isolated environments, relying heavily on human intervention. However, the growing inter-connectivity has exposed them to potential risks from remote adversaries"(Nankya et al., 2023).

The implications that successful cyber attacks have on industrial control systems are much greater than meets the eye. As George Stergiopoulos, a researcher at the Industrial and Telecommunications Security at the University of Aegean, puts it, "Reports clearly indicate that attacks on Industrial Control Systems (ICS) of the Oil and Gas sector can have adverse effects to wide geopolitical areas and multiple countries. Even worse, the severity of some security incidents is likely to exacerbate due to cascading failures introduced by dependencies of other Critical Infrastructure on the Oil and Gas infrastructure" (Stergiopoulos et al., 2020). Due to the interconnectedness of these Industrial Control Systems, disrupting one can have a chain reaction, which could potentially affect multiple critical infrastructure facilities. Examples of negative ramifications could include, but are not limited to: power outages, delays in processing at refineries, and other disruptions in various operations. This disruption could also cause negative

effects for community members, such as higher prices, lack of supply, and greater levels of fear and uncertainty.

The current approaches to protect these ICS devices have significant technical limitations. According to researcher Riccardo Mennilli (2024) from the Politecnico di Torino Technical University, "Limitations in memory and power still represent a major challenge for the development of edge computing, especially when machine learning is involved". To clarify, Industrial Control Systems are a form of edge computing. Additionally, cloud-based security also introduces its own vulnerabilities. As Fatemeh Akbarian et al. note, "the cloud also introduces significant security challenges since moving control systems to the cloud can enable attackers to infiltrate the system and establish an attack that can lead to damages and disruptions with potentially catastrophic consequences" (Fatemeh Akbarian et al., 2023).

There is an aperture in the current research for cyber security that can effectively operate with these constraints in mind. While artificial intelligence and machine learning have the potential to improve cybersecurity, implementing those on an ICS device remains challenging. Furthermore, as Cook observes, "Future anomaly detection research could examine how the computational performance of ICS components is impacted by offline and online learning, and more importantly, what the potential trade-offs are regarding detection accuracy and speed" (Cook, 2023).

Literature Review

The research conducted for ICS hardware-based cyber security has shown exceptional performance. According to the director of the Intelligent Systems and Computer Architecture Laboratory, hardware-based machine learning can achieve up to a 98.5% accuracy rating in network anomaly detection and has a 4.6x improvement in runtime efficiency and 4.9x

improvement with energy utilization (Kornaros, 2022). To build off of this, researchers from the Department of Electrical and Computer Engineering at the University of Texas implemented hardware Trojan detection, which had a detection accuracy ranging from 85-99% (Kundu et al., 2021). To clarify, a hardware Trojan is a malicious modification to a computer's circuitry. These can cause an ICS device to behave incorrectly, and thwart certain cybersecurity protocols. Even so, these studies only focused on hardware optimization and had no decisive benchmarks for real-time threat detection.

The integration of artificial intelligence machine learning is auspicious within ICS cybersecurity. According to a professor at King Faisal University, "The KNN and DT algorithms achieved superior accuracy (100%) in binary classification and multiclass classification" (Alkahtani & Aldhyani, 2022). The authors used multiple machine-learning algorithms to detect ICS attacks in real time. Similarly, Matthew Baker and associates developed an LSTM neural network that can detect and classify network anomalies in under 14.32ms, while achieving a 96.66% accuracy rating (Baker et al., 2023). Nonetheless, these cyber security implementations demand a lot of computational power, which is typically not available in resource-constrained ICS devices.

Nankya and colleagues showed that ML approaches, such as KNN and neural networks can achieve high detection accuracies (90-100%), but they have implementation challenges within resource-constrained environments due mainly to computational demand (Nankya et al., 2023). Furthermore, findings from two postdoctoral research associates specializing in computational sciences showed that, out of 148 peer-reviewed papers on AI applications in smart grids, AI techniques show exciting results, such as a 98.67% accuracy in anomaly detection

(Omitaomu & Niu, 2021). Notwithstanding, there remains a large hurdle for real-time processing.

Three major cybersecurity approaches on ICS devices have all been researched but have yet to be overlapped: hardware-based solutions, AI/ML applications, and real-time detection systems on resource-constrained devices. To elaborate, a hardware-based cybersecurity solution is a localized cybersecurity on the hardware itself. At the same time, resource-constrained Industrial Control Systems can be defined as machines with limited computing power. While research in each field has shown promising results, there has yet to be a marriage between all three cyber security strategies.

The current gap in the body of knowledge is evident in studies that try to combine hardware constraints and ICS devices with ML systems. Both Müller et al. and Banik et al. describe a need for a comprehensive experiment, with Müller et al. referencing the need for the "accounting of physical constraints" (Müller et al., 2022) and Banik et al. stating "We identify a primary research gap in the lack of a holistic approach to simulation before deployment" (Banik et al., 2024). The current body of knowledge requires a more exhaustive framework that implements hardware optimization, machine learning, and real-time detection within a resource-constrained ICS environment. With these considerations present, it is clear that there is a gap in ICS devices that have Machine Learning systems, real-time detection, and hardware-based cybersecurity. This has prompted a question: To what extent can hardware-based Machine Learning models detect cybersecurity anomalies in resource-constrained Industrial Control System (ICS) devices while having real-time detection? The purpose of this experiment is to see to what extent an edge-deployed ML device can effectively execute anomaly detection compared to other ML cybersecurity models. The independent variable will be defined as the

configuration of the ML model, including its architecture, algorithms, training data, and hyperparameters. The dependent variable will be the ML model's temporal performance detection efficacy, with the main intervening variable being the complexity of the cybersecurity attacks, as this experiment employs two main attacks: Command Injection and Denial of Service. The goal of the study is to have an edge-deployed ML model that will be more effective at detecting cyberattacks.

Methods

Experimental Prototype

This research applies an experimental prototype methodology, which enables the development and validation of a proof of concept model, which will evaluate how effective edge-deployed machine learning models are in cyber security and industrial control systems (ICS) applications. To define the methodological approach, Camburn et al. (2017) state "Experimental studies have explored prototype requirement specification. Prototype fidelity predetermines potential for certain insights from a prototype . Requirement relaxation can be induced in design teams. Relaxed requirement prototypes cost less and require less time to build . The strategic relaxation of requirements on early prototypes does not seem to have any adverse effects on final performance. Low-fidelity prototyping fosters a sense of forward progress, learning from failure, and concept expression without fixation on detail." The research instrument will be a hardware testbed on a Raspberry Pi 4B that simulates a programmable logic controller (PLC). This instrumentation aligns with the tangential research showing that edge devices with lightweight ML models can provide "real-time threat analysis and response without relying heavily on cloud-based resources" and enable "real-time detection of security threats

analyzing network data, device logs and other parameters without introducing significant delays or latency"(Hasan et al., 2024).

The decision to use a Raspberry Pi B is justified by its proven applicability with it being "used as a replacement for an industrial-grade programmable logic controller (PLC) in a data-collection application"(Makarcheva, 2020). Python was elected as the primary coding language, aligning with previous research that notes "The information is analyzed with the help of programming language Python data analysis and the application of correlation analysis, and machine learning as well"(Yadav, 2020). As for Machine Learning, the testbed implements an Isolation Forest algorithm via TensorFlow Lite, following previous research where "we deploy a trained tensorflow model to the Raspberry Pi and convert the tensorflow ML model to the tensorflow Lite"(Hasan et al., 2024). Additionally, Scapy is applied as the packet manipulator, as existing literature has established "utilizing Python and libraries like Scapy, this project equips network administrators and security analysts with essential tools for monitoring network activity, extracting vital information from packet headers, and identifying potential security threats"(Rao, 2024).

The choice of an experimental prototype directly addresses the research question and fills the current gap in the body of knowledge concerning the application of ML for cybersecurity in ICS. The alignment between this methodology and the research is shown from several key factors:

- **Effectiveness of ML in ICS cybersecurity:** Recent work by Madupati emphasizes that "Because of the real-time operational requirements and legacy technology, traditional security methods are frequently ineffective in protecting ICS. Machine Learning (ML)

techniques are the major solution to improve Intrusion Detection Systems (IDS) or Intrusion Prevention Systems"(2025).

- **Performance constraints on resource-limited devices:** Earlier research has noted "While the cryptographic community has significantly progressed lightweight (in terms of area overhead) security primitives for low cost and power efficient hardware implementations, lightweight software implementations of security primitives for resource constrained devices are less investigated"(Su et al., 2019).
- **Importance of real-time detection:** Koay et al. (2022) note that "in real-world environments, attack detection needs to be online (real-time) to provide timely mitigation".
- **Replicability:** Mirroring a PLC with a Raspberry Pi testbed has been suggested by Fasano et al. (2020), stating, "A programmable logic controller (PLC) emulation methodology can dramatically reduce the cost of high-fidelity OT network emulation without compromising specific functionality".

While other methodologies, such as case studies or qualitative analysis could examine ICS security, an experimental prototype methodology using a Raspberry Pi testbed is the most appropriate approach for this research. This method allows the direct measurement of performance metrics on resource constrained devices, which validates machine learning models viability on resource constrained devices. Alternative approaches, like a simulation or theoretical analysis would not be able to replicate these hardware limitations. The experimental design allows testing of both detection efficacy and temporal performance, which are critical metrics that could not be accessed through the employment of other methodologies. Including this, using

a Raspberry Pi to act as a PLC provides a cost effective, replicable framework that sets a lower entry bar for any future research. Finally, this method produces concrete metrics that can be statistically analyzed and defined.

Materials

The Raspberry Pi 4 model B is a small, affordable computer that functions as a normal computer, but is also adopted as a testbed for experimental research. Its affordability and malleability makes it appealing to the research process in all disciplines. As Jolles (2021) notes, "The Raspberry Pi has been purposefully built as a highly flexible and powerful computer at the fraction of the costs of a traditional PC to be used by anyone to solve problems creatively. Its large number of assets easily outweigh its limitations and make the Raspberry Pi a great research tool that can be used for almost anything". The following components were required for setup and configuration of the Raspberry Pi:

1. Power Supply: A 5V DC power supply with minimum 2.5A output. This is used to power the Pi.
2. MicroSD Card (minimum 8GB): Essential as the Raspberry PI uses a microSD card that stores the operating system and any user data (Jolles, 2021). A MicroSD to SD card adapter was needed to flash the operating system onto the card using a standard computer.
3. Micro HDMI Cable: needed for the first-time setup to see and control the computer, which is critical for turning on settings that allow remote access to the Raspberry Pi.
4. VNC Viewer & SSH Configuration: As Jolles (2021) explains, "SSH (Secure Shell) enables control from another device on the same network using the command line" while

"Virtual Network Computing allows you to remotely control the desktop interface of the Raspberry Pi from anywhere without a monitor".

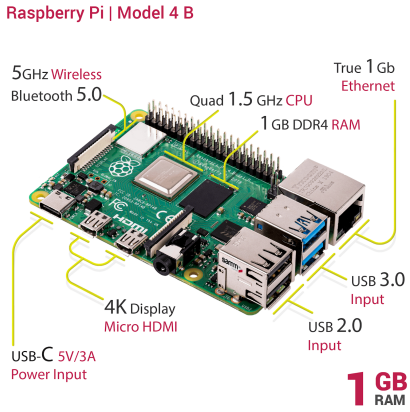

Figure 1. Raspberry Pi 4B 1GB product image

Adapted from Samm Market. (n.d.). *Raspberry Pi 4 1GB product image* [PNG image]. Retrieved February 23, 2025, from https://market.samm.com/raspberry-pi-4-1gb-en-raspberry-pi-models-raspberry-pi-5094-17-B.png

Software Architecture

This Raspberry Pi 4B testbed, emulating a PLC, incorporates python as the primary coding language, implements an Isolation Forest algorithm via TensorFlow Lite for Machine Learning (ML), and uses Scapy for packet manipulation. According to Riccardo Mennilli (2024), "TensorFlow Lite (TFLite) project was initiated in 2017, enabling much smaller binary sizes," which "opened the door to experimentation with machine learning on small, mobile, memory-constrained devices, bringing advanced inference capabilities closer to the end application". Using Tensorflow-lite enables edge deployment on Raspberry Pi's due to its lightweight nature, addressing resource constraints seen on ICS devices. The model used for anomaly detection is Isolation Forest, which is described as "a tree-based algorithm that isolates anomalies by recursively partitioning data until each data point is isolated in its tree leaf" (Nankya et al., 2023). The implementation of Isolation Forest allows real-time detection for

binary classification. In addition, the interplay between tensorflow in conjunction with isolation forest is a common occurrence, being displayed in research done by Kayan et al. (2022), stating "TF Lite models are significantly less in size than TF models. During the evaluation of the Isolation Forest, the whole dataset must be fit at once, due to [the] nature [of the] algorithm".

Software Dependencies

The Raspberry Pi 4B needs a small, selective suite of Python libraries to enable ML-based anomaly detection on a hardware restricted device. Below are the primary software dependencies used for this project:

- Python 3.11: Primary coding language for the implementation.
- NumPy 2.2.3: Scientific Computations.
- Pandas 2.2.3: Data manipulation and analysis.
- Scikit-learn 1.6.1: Machine learning - (Isolation Forest).
- Scapy 2.6.1: Packet reading and manipulation.
- Netfilerqueue 1.0.0: Network packet queuing.
- Tflite-runtime 2.14.0 ML framework.
- Python-daemon 2.3.0: Manages system daemons.
- Logger 1.4: Message logger.

While the primary python libraries used in the study are shown above, several supplementary libraries were also used; however, they are considered to have marginal effect and therefore are not mentioned.

Network Configuration

The Pi's network configuration simulates a simplified version of an industrial control system environment, focusing on solely validating machine learning based anomaly detection on resource constrained devices, following the experimental prototype methodology. The network architecture uses a socket-based communication model, composed of three main components: A simulated PLC, ICS controller, and Attack Simulator. To clarify, socket-based communication is a fundamental mechanism that allows for programs to communicate and exchange data with each other over a network. The simulated PLC listens for control commands on Port 9999, simulating the behavior of ICS's hardware with processing commands and generating responses. The ICS controller generates the control commands on Port 9999, sending requests to the simulated PLC, representing normal operational patterns seen in ICS. Finally, the attack simulator is a separate python script that generates denial of service (DOS) and command injection attacks that the ML model will attempt to detect with binary classification. Moving on to the packet monitoring, the network traffic is captured and analyzed using both Scapy and Netfilterqueue which intercept and examine the packets being sent over the 9999 port. The packet monitoring will start with capturing all network traffic using socket binding and extracting the packet features, such as the length, timestamp, TTL, and TCP window size, all of which are relevant for detecting ICS network anomalies. These extracted packets then get fed through the TensorFlow Lite deployed Isolation Forest model for anomaly detection. This configuration provides a controlled test environment for evaluating the performance of the TensorFlow Lite Isolation Forest model.

Data Collection Framework

The data collection framework is designed to find to what extent this Raspberry Pi 4B testbed can detect cybersecurity anomalies while having real-time detection. This structure

measures performance metrics for the machine learning model. The categories of performance based metrics are the detection efficacy and temporal performance. Below is a table representing the metrics and their description:

Detection Efficacy	
Category	**Description**
Detection accuracy	The percentage of correctly identified normal and anomalous network traffic patterns.
False Positive Rate (FPR)	The percentage of normal traffic incorrectly identified as anomalous.
False Negative Rate (FNR)	The percentage of anomalous data that does not get detected.

Table 1: Detection Efficacy Metrics for Edge-Deployed ML Models

Temporal Performance	
Category	**Description**
Detection Latency	The time elapsed between the introduction of an anomalous packet and its detection, measured in milliseconds.
Processing Time	The time required to extract packet information and classify it, measured in milliseconds.

Table 2: Temporal Performance Metrics for Edge-Deployed ML Models

These metrics quantify the ML model's ability to differentiate between legitimate ICS network traffic and attacks, with Alkahtani & Aldhyani (2022) displaying the importance of high

detection accuracy, achieving "superior accuracy (100%) in binary classification" in their tangential experiment. For clarification, binary classification is a machine learning technique where data is classified into one of two possible classes. In this context, detection accuracy refers to the ML model's ability to classify data into either normal network traffic or anomalous network traffic. Including this, the temporal metrics address the research gap of real time detection, as Koay et al. (2022) note that "attack detection needs to be online (real-time) to provide timely mitigation". Both detection efficacy and temporal performance are then collected using Python's time module and Logger 1.4. This data collection framework results in data that directly addresses the research question by quantifying how effective edge-deployed ML models are with detecting cybersecurity threats in ICS environments.

Variables

This research investigates how effectively edge-deployed machine learning models can detect cybersecurity anomalies and resource constrained ICS devices. There are two categories of dependent variables: detection efficacy and temporal performance. The detection efficacy measures detection accuracy, false positive rate, and false negative rate. Temporal Performance, on the other hand, measures detection latency and processing time. The independent variable employed is the ML configuration, specifically the Isolation Forest model connected to TensorFlow Lite on the Raspberry Pi 4B testbed. The control variables are the hardware environment, the network configuration, and the attack types; all of which are controlled during the experiment. The hardware environment is the Raspberry Pi 4B configuration with identical software specifications throughout each test. Similarly, all tests are performed using Scapy and socket-based communication for consistent traffic patterns, with all the attack types staying consistent. The variables are measured using Python's time module and Logger 1.4, recording

timestamps before and after detection. This approach allows the precise measurement of both detection efficacy and temporal performance, providing the quantitative data needed to address the viability of edge-deployed ML models that can detect cybersecurity anomalies in resource constrained ICS environments.

Procedures

This study uses a systematic procedure to evaluate edge-deployed ML models for ICS cybersecurity, which in this case, were executed in four specific phases: Environment setup, ICS Simulation Development, ML Implementation and Testing, and Data Analysis.

Phase 1: Environment Setup

Raspberry Pi Configuration

- Install Raspberry Pi OS on micro SD card.
- Configure SSH tunnel and use VCNviewer for remote access.
- Install the required libraries listed in the software dependencies section.

Network Configuration

- Implement socket-based communication on port 9999.
- Configure Scapy and Netfilterqueue for packet interception.
- Test and ensure proper network connectivity.

Phase 2: ICS Simulation Development

Device Simulation

- Create a listening script listening to traffic on port 9999.
- Create a control script for normal network traffic, sending it to the port.
- Test both scripts' functionality.

Attack Simulation

- Create scripts for both denial of service and command injection attacks.
- Test scripts functionality.

Phase 3: ML Implementation and Testing

Model Training and Deployment

- Train the Isolation Forest model using scikit-learn.
- Optimize Isolation Forest and deploy it on TensorFlow Lite.
- Integrate with packet monitoring scripts (ICS Simulation Development).

Attack Detection Testing

- Deploy the attack scripts.
- Capture the results using Python’s time module.
- Use Logger 1.4 for detection efficacy and temporal performance metrics.

Phase 4: Data Analysis

Performance Evaluation

- Calculate detection accuracy, FPR, and FNR.
- Calculate average detection latency and processing time.
- Analyze results.

This research addresses the proposed solution of "Machine Learning (ML) techniques are the major solution to improve Intrusion Detection Systems" suggested by Madupati (2025). This specific method proposal provides the correct quantitative evaluation of the effectiveness of edge-deployed ML models for ICS security.

Results

The experimental prototype methodology that was seen in this study produced significant findings pertaining to the effectiveness of edge deployed ML models for cybersecurity anomaly detection within resource-constrained ICS devices. The collection of data in this experiment involved using an Isolation Forest algorithm implemented with Tensorflow Lite on a Raspberry Pi 4B testbed, with two categories of results: detection efficacy and temporal performance.

Detection Efficacy

The Raspberry Pi Isolation Forest model showed strong classification abilities for specifically identifying anomalous network traffic within the simulated ICS environment. As shown in table 3 below, the model achieved a detection accuracy of 90.8%, with a FPR of 9% and a FNR at 10%.

Detection Efficacy		
Metric	**Value**	**Percentage**
Detection accuracy	0.9080	90.8%
False Positive Rate (FPR)	0.0900	9%
False Negative Rate (FNR)	0.1000	10%

Table 3: Detection Efficacy Results of Edge-Deployed ML Model on Raspberry Pi 4B testbed

These findings align with the previously stated research conducted by Nankya et al. (2023), who found that ML approaches achieve detection accuracies between 90-100% within controlled environments. However, Pelofske et al. (2024) achieved a 1.3% FNR and 3.7% FPR in a similar Random Forest ML model, displaying the need for a more robust ML model approach. Furthermore, the results extend the work done by Nankya et al. by implementing a software that

achieves the accuracy ratings on a resource constrained device mirroring a PLC. That said, previous research by Kornaros (2022) and Alkahtani & Aldhyani's (2022) reported much higher accuracy ratings, with each study earning a 98.5% and 100%, respectively. The context of the studies, however, did not include localized cybersecurity on the physical resource constrained devices.

Temporal performance

While detection efficacy is immensely important to decipher real data from anomalous data, temporal performance is just as significant. Even with a respectable detection accuracy, an extended detection latency leaves the system vulnerable, as the device continues to be compromised during the delayed response. As shown in table 4, the Raspberry Pi achieved notably low processing time and latency.

Temporal Performance	
Metric	**Value (Milliseconds)**
Average Detection Latency	1.72ms
Average Processing Time	4.96ms

Table 4: Temporal Performance Results of Edge-Deployed ML Model on Raspberry Pi 4B testbed

These results are noteworthy, especially when compared to Baker et al. 's (2023) findings, where their neural network, although more complex, achieved detection in 14.32ms. In addition, Sfaxi et al. (2023) reported an ETS model, with their best execution time at mean of 14ms. The average processing time in this study compared to Baker et al.’s research is a large improvement and addresses Koay et al.’s (2022) emphasis on having "attack detection [that] needs to be online (real-time) to provide timely mitigation" in ICS environments.

Connection to the Research Question

The methods approach and results directly address the previously stated research question: To what extent can hardware-based ML models detect cybersecurity anomalies in resource-constrained Industrial Control System (ICS) devices while having real-time detection? The findings show that an edge-deployed Isolation Forest model using Tensorflow Lite can effectively detect network anomalies with acceptable accuracy of 90.8% while having an impressive processing time of 4.96ms. These results validate the hypothesis that resource constrained edge devices are capable of achieving effective localized cybersecurity benchmarks without having the vulnerabilities associated with cloud based cybersecurity practices. As previously cited, Fatemeh Akbarian et al. (2023) note that, "moving control systems to the cloud can enable attackers to infiltrate the system and establish an attack that can lead to damages and disruptions with potentially catastrophic consequences". Due to the security being localized on the device, this testbed model mitigates these risks. Moreover, the proven temporal performance with the low value of 4.96ms builds upon Cook's (2023) study, stating that future research needs to "examine how the computational performance of ICS components is impacted by offline and online learning, and more importantly, what the potential trade-offs are regarding detection accuracy and speed". This experiment proposes the use of an optimized ML model on resource constrained devices to achieve a high accuracy and low detection latency for cybersecurity practices.

Limitations

The experimental prototype methodology used socket-based communication with a simplified network topology, which does not fully represent the complexity that ICS devices face. Critical infrastructure often has a more complex network topology and varying types of

hardware, which could affect anomaly detection performance. Additionally, Ghosh & Sampalli (2019) states that "The [] studies have research gaps that fail to address availability and secured communication channel[s]", which this research failed to establish. Also, the study focused on two attack types, being Denial of Service (DoS) and command injection attacks. This attack scope is limited, and doesn't include more sophisticated attacks, such as Advanced Persistent Threats or Man in the Middle attacks. While the Raspberry Pi provided a fair semblance of a PLC, actual ICS devices could have varying hardware specifications and performance. As aforementioned, Mennilli (2024) noted, "Limitations in memory and power still represent a major challenge for the development of edge computing, especially when machine learning is involved." Lastly, the Isolation Forest algorithm, while effective for testing, represents only one approach for machine learning anomaly detection. This study did not evaluate alternative ML algorithms that have the potential to perform better under similar constraints. Additionally, the binary classification that was used (normal vs anomalous) has the potential to be rudimentary for critical infrastructure environments that need specific attack type identification.

Implications

The shown efficacy of edge-deployed ML models opens the door to a viable alternative in ICS cybersecurity that fundamentally challenges the current dependence on cloud based security solutions. The experimental prototype methodology both validated a technical concept and established a new understanding of how cybersecurity can be implemented on edge devices, responding to the gap stated by Müller et al. (2022) regarding the "accounting of physical constraints" and Banik et al. (2024) noting the need for a "holistic approach to simulation before deployment."

The shift from online to localized cybersecurity could reduce the exposure of internet-based malware while maintaining proper cybersecurity monitoring. Moreover, the methodology chosen provides a framework for the optimization of ML models on resource constrained devices. Using Tensorflow Lite in conjunction with an Isolation Forest algorithm achieves a balance of detection efficacy (90.8% accuracy) and temporal performance (4.96ms processing time), showing that the use of both softwares is practical for future research in ICS environments. This addresses the concern stated by Madupati (2025) who expressed that Machine Learning (ML) techniques are the major solution to improve Intrusion Detection Systems (IDS). To build upon Madupati (2025), Su et al. (2019), states that "lightweight software implementations of security primitives for resource constrained devices are less investigated." This experiment implements a solution that attends to both Madupati (2025) and Su et al. (2019) concerns.

More significantly, this approach reconciles what has long been recognized as a critical challenge within ICS cybersecurity: the trade-off between detection accuracy and speed. By achieving a processing time of 4.96 milliseconds, significantly outperforming Baker et al.'s (2023) 14.32 milliseconds and Sfaxi et al.'s 14 milliseconds benchmarks, this methodological choice demonstrates that real-time security is achievable without sacrificing detection efficacy. This finding responds to Cook’s (2023) question regarding "what the potential trade-offs are regarding detection accuracy and speed." The larger implication for the cybersecurity community is the development of a new security paradigm that focuses on localized cybersecurity over cloud interconnection. This approach mitigates the previous concern by Stergiopoulos et al. (2020) that "attacks on Industrial Control Systems can have adverse effects to wide geopolitical areas and multiple countries." With the development of cybersecurity independent of the cloud, this

methodology creates a framework for Critical Infrastructure that reduces the amount of attack vectors while maintaining a high detection accuracy. Finally, this methodological experimental prototype provides a replicable and cost effective framework that has low barriers for the implementation of advanced cybersecurity in critical infrastructure.

Future research directions

This study has opened lanes for new waves of research, including:

1. **Specific attack type identification**: Creating a multi-classification model that detects both anomalous and normal data and also the type of attack being used.
2. **ICS Hardware**: Implementing real ICS devices in a study to test any limitations that cannot be seen while using a Raspberry Pi.
3. **Multi-device Coordination**: As Stergiopoulos previously stated, the interconnectedness of ICS devices can cause cascading effects. This might be solved by creating an interwoven system of ICS devices using coordinated machine learning techniques, such as federated learning.
4. **Protocol-Specific Detection**: Incorporating specific ICS protocols, such as ModBus or DNP3, could improve detection accuracy for each specific critical infrastructure industry.

Conclusion

The growing unease in the critical infrastructure sphere is partly due to the risk of cascading effects of interconnected cybersecurity systems. The experiment employed seeks a resolution to this issue, with the research question being: To what extent can hardware-based ML models detect cybersecurity anomalies in resource-constrained Industrial Control System (ICS) devices while having real-time detection?

Using a Raspberry Pi 4B acting as a ICS device, specifically a PLC, the study successfully executed an experimental prototype methodology that answers the research question. The results gathered were quite impressive, particularly the 90.8% detection accuracy and the processing time of 4.96ms. That said, the shortcomings of the False Positive Rates at 9% and a False Negative Rate at 10% are still apparent in this research. The Raspberry Pi 4B effectively detected anomalies with real time detection while having the resource constraints of an ICS device to a high extent. This is significant because localized cybersecurity has greater reliability when compared to cloud based security, which is riddled with problems such as longer latency and detection processing time.

The experiment applies a simplified network topology using socket-based communication, which does not fully encapsulate the communication done by ICS devices. Similarly, there was a limited attack scope and only one ML algorithm was tested, providing an uncomplicated experiment that does not fully express the intricate nature of the communication of ICS devices. The experimental prototype methodology has the ability to push ICS cybersecurity into focusing more on real time mitigation of attacks, versus the conventional perception that a higher accuracy is more significant. With more timely detection of attacks, the cascading effects of cyber breaches has the potential to reduce adverse effects on the very power grids that enable this computer to be powered to have this paper written. Furthermore, the lack of dependency of cloud cybersecurity in which this experiment emphasizes can reduce internet-based vulnerabilities seen in standard cybersecurity practices.

With this in mind, the study demonstrates that edge-deployed ML models can effectively detect cybersecurity anomalies in resource constrained ICS environments with excellent temporal performance. While challenges with the detection efficacy remain, this study opens a

promising direction for increasing the security of critical infrastructures to adapt with the rising rates of cyber attacks, making more resilient industrial control systems that society depends on daily. This research demonstrates the technical feasibility of edge-based ML cybersecurity for Industrial Control Systems but also establishes the paradigm that prioritizes temporal performance and a high detection accuracy, which has the potential to transform how critical infrastructure is protected against the growing sophistication and frequency of cyber attacks in our increasingly interconnected world.

References

Alkahtani, H., & Aldhyani, T. H. H. (2022). Developing Cybersecurity Systems Based on Machine Learning and Deep Learning Algorithms for Protecting Food Security Systems: Industrial Control Systems. Electronics, 11(11), 1717. https://doi.org/10.3390/electronics11111717

Akbarian, F., Tärneberg, W., Fitzgerald, E., & Kihl, M. (2023, March 20). Attack Resilient Cloud-Based Control Systems for Industry 4.0. https://ieeexplore.ieee.org/document/10076459

Baker, M., Fard, A. Y., Althuwaini, H., & Shadmand, M. B. (2023). Real-Time AI-Based Anomaly Detection and Classification in Power Electronics Dominated Grids. IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 4(2), 549–559. https://doi.org/10.1109/jestie.2022.3227005

Banik, S., Rogers, M., Mahajan, S. M., Emeghara, C. M., Banik, T., & Craven, R. (2024). Survey on Vulnerability Testing in the Smart Grid. IEEE Access, 12, 119146–119173. https://doi.org/10.1109/access.2024.3449642

Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., ... & Wood, K. (2017). Design prototyping methods: state of the art in strategies, techniques, and guidelines. Design Science, 3, e13.

Cook, M. M. (2023). Anomaly diagnosis in industrial control systems for digital forensics.

Theses.gla.ac.uk. https://theses.gla.ac.uk/83625/

Fasano, R., Lamb, C., Genk, M. E., Schriener, T., & Hahn, A. (2020). Emulation Methodology of Programmable Logic Controllers for Cybersecurity Applications. *OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information)*. https://doi.org/10.1115/icone2020-16245

Hasan, S., Alotaibi, A., Talukder, S., & Shahid, A. (2024, May 26). *Distributed Threat Intelligence at the Edge Devices: A Large Language Model-Driven Approach*. arxiv.org. https://arxiv.org/pdf/2405.08755

Jolles, J. W. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. Methods in Ecology and Evolution, 12(9), 1562-1579.

Kayan, H., Majib, Y., Alsafery, W., Barhamgi, M., & Perera, C. (2021). AnoML-IoT: An end to end re-configurable multi-protocol anomaly detection pipeline for Internet of Things. Internet of Things, 16, 100437.

Koranos, G. (2022). Hardware-Assisted Machine Learning in Resource-Constrained IoT Environments for Security: Review and Future Prospective | IEEE Journals & Magazine | IEEE Xplore. (n.d.). Ieeexplore.ieee.org. https://ieeexplore.ieee.org/abstract/document/9785622

Koay, A. M. Y., Ko, R. K. L., Hettema, H., & Radke, K. (2022). Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges. *Journal of Intelligent Information Systems*. https://doi.org/10.1007/s10844-022-00753-1

Kundu, S., Meng, X., & Basu, K. (2021). Application of Machine Learning in Hardware Trojan Detection. 414–419. https://doi.org/10.1109/isqed51717.2021.9424362

Madupati, B. (2025). Machine Learning for Cybersecurity in Industrial Control Systems (ICS). *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5076696

Makarcheva, A. (2020). Implementation of a PLC Code on Raspberry Pi in CODESYS Environment. *Theseus.fi*. http://www.theseus.fi/handle/10024/302987

Mennilli, R. (2024). Machine Learning Applications for PLC-Based Industrial Automation - Webthesis. Polito.it. https://webthesis.biblio.polito.it/secure/33090/1/tesi.pdf

Müller, N., Ziras, C., & Heussen, K. (2022). Assessment of Cyber-Physical Intrusion Detection and Classification for Industrial Control Systems. ArXiv (Cornell University), 432–438. https://doi.org/10.1109/smartgridcomm52983.2022.9961010

Nankya, M., Chataut, R., & Akl, R. (2023). Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies. Sensors, 23(21), 8840. https://doi.org/10.3390/s23218840

Omitaomu, O. A., & Niu, H. (2021). Artificial Intelligence Techniques in Smart Grid: A Survey. Smart Cities, 4(2), 548–568. https://doi.org/10.3390/smartcities4020029

Pelofske, E., Liebrock, L. M., & Urias, V. (2024, February 27). A Robust Cybersecurity Topic Classification Tool. https://arxiv.org/pdf/2109.02473

Rao, M. (2024, June). *NETWORK PACKET ANALYZER*. Researchgate. https://www.researchgate.net/profile/Madhava-Rao-9/publication/381301552_NETWORK_PACKET_ANALYZER/links/6666f614b769e7691926b876/NETWORK-PACKET-ANALYZER.pdf

Sfaxi, H., Lahyani, I., Yangui, S., & Torjmen, M. (2024, March 11). Latency-aware and proactive service placement for Edge Computing | IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/10466718/

Stergiopoulos, G., Gritzalis, D. A., & Limnaios, E. (2020). Cyber-Attacks on the Oil & Gas Sector: A Survey on Incident Assessment and Attack Patterns. IEEE Access, 8, 128440–128475. https://doi.org/10.1109/access.2020.3007960

Su, Y., Gao, Y., Kavehei, O., & Ranasinghe, D. C. (2019). Hash Functions and Benchmarks for Resource Constrained Passive Devices: A Preliminary Study. *ArXiv (Cornell University)*. https://doi.org/10.1109/percomw.2019.8730835

Yadav, V. (2020, September). *Regulatory compliance and cyber-security in healthcare: Investigating the relationship between regulatory compliance and the effectiveness of cyber-security measures in healthcare organizations*. Researchgate. https://www.researchgate.net/profile/Vivek-Yadav-29/publication/383905182_REGULATORY_COMPLIANCE_AND_CYBER-SECURITY_IN_HEALTHCARE_INVESTIGATING_THE_RELATIONSHIP_BETWEEN_REGULATORY_COMPLIANCE_AND_THE_EFFECTIVENESS_OF_CYBER-SECURITY_MEASURES_IN_HEALTHCARE_ORGANIZATIONS/links/66e00a2eb1606e24c21d905f/REGULATORY-COMPLIANCE-AND-CYBER-SECURITY-IN-HEALTHCARE-INVESTIGATING-THE-RELATIONSHIP-BETWEEN-REGULATORY-COMPLIANCE-AND-THE-EFFECTIVENESS-OF-CYBER-SECURITY-MEASURES-IN-HEALTHCARE-ORGANIZATIONS.pdf