A Heterogeneous Redundant Architecture for Industrial Control System Security

Zhihao Dai*, Matthew Leeke*, Yulong Ding†, and Shuang-hua Yang†
*Department of Computer Science
University of Warwick, Coventry, CV4 7AL, UK
Email: {zhihao.dai,matthew.leeke}@warwick.ac.uk
†Shenzhen Key Laboratory of Safety and Security for Next Generation of Industrial Internet and
Department of Computer Science and Engineering
Southern University of Science and Technology, Shenzhen, 518055, China
Email: {dingyl,yangsh}@sustech.edu.cn

***Abstract*—Component-level heterogeneous redundancy is gaining popularity as an approach for preventing single-point security breaches in Industrial Control Systems (ICSs), especially with regard to core components such as Programmable Logic Controllers (PLCs). To take control of a system with component-level heterogeneous redundancy, an adversary must uncover and concurrently exploit vulnerabilities across multiple versions of hardened components. As such, attackers incur increased costs and delays when seeking to launch a successful attack. Existing approaches advocate attack resilience via pairwise comparison among outputs from multiple PLCs. These approaches incur increased resource costs due to them having a high degree of redundancy and do not address concurrent attacks. In this paper we address both issues, demonstrating a data-driven component selection approach that achieves a trade-off between resources cost and security. In particular, we propose (i) a novel dual-PLC ICS architecture with native pairwise comparison which can offer limited yet comparable defence against single-point breaches, (ii) a machine-learning based selection mechanisms which can deliver resilience against non-concurrent attacks under resource constraints, (iii) a scaled up variant of the proposed architecture to counteract concurrent attacks with modest resource implications.**

***Index Terms*—Industrial Control System, Security, Redundancy, Programmable Logic Controller, Machine Learning**

I. Introduction

An industrial Control System (ICS) is a complex system that monitors and controls industrial processes. Unlike most computer-based systems, ICSs largely operate within the Operational Technology (OT) environment, this being the industrial counterpart to Information Technology (IT) [1]. ICSs are widely deployed across many industries, often forming part of critical national infrastructure [2]. As such, the security of ICSs is a research topic of great importance.

Notwithstanding the latest research efforts [3], ICSs are vulnerable to cyberattacks, in part due to longstanding operational issues and a recent push towards the Industrial Internet of Things (IIoT) [4], [5]. The IIoT is the integration of industrial devices into the Internet, in pursuit of better connectivity, productivity, and efficiency. The IIoT is bringing systems and devices in OT and IT closer than ever. In the case of ICSs, the IIoT enables the operators to monitor and control industrial processes remotely and manage geographically separated plants simultaneously [6], [7]. An IIoT-enabled ICS can even have the capability to adapt production rate to the market demand in real-time. However, with these new features come a larger attack surfaces and higher security risks for ICSs. In other words, the move towards the IIoT exposes the system to a wide array of attack vectors through the Internet, most of which can not easily be anticipated during ICS design.

Where OT and IT converge, conventional security measures like firewall and access control can counter most IT-oriented attacks. However, they fall short when addressing ICS-specific attacks. Indeed, some methods, such as routine software updates, can lead to the violation of the availability requirements placed on the ICS [2]. This calls for security solutions that can identify and mitigate ICS-specific attacks without disrupting control operations. A natural approach would be embedding component-level heterogeneous redundancy in ICS. Heterogeneous redundancy is different from native redundancy, or duplication, in that each version of component is implemented via software and hardware diversification techniques. When one version is compromised, the system can switch to another, thereby minimising the impact of attacks. An adversary now must uncover and concurrently exploit vulnerabilities across multiple versions of hardened components to continue along their intended attack path. Increased costs and delay are imposed on the attacker, reducing the likelihood of a successful attack being launched.

Heterogeneous redundancy thrives when applied on core components like Programmable Logic Controller (PLC). A PLC is a computer-based controller that automates process control with an interchangeable program. Its direct control over the industrial process makes it a valuable target, particularly for knowledgeable attackers. Previous incidents have underlined both the vulnerability and significance of PLCs [8]. For example, Stuxnet was the first publicly known ICS mal-

This research is supported in part by the National Natural Science Foundation of China (Grant No. 92067109, 61873119), in part by Shenzhen Science and Technology Program (Grant No. ZDSYS20210623092007023), in part by the Science and Technology Planning Project of Guangdong Province(Grant No. 2021A0505030001), and in part by the Educational Commission of Guangdong Province (Grant No. 2019KZDZX1018).

ware. Stuxnet operated by injecting malicious code into a PLC to degrade centrifuges within Iranian nuclear facilities without being detected [9], [10]. As a further example, "Kemuri", an undisclosed water company, had hundred of its PLCs open to tampering after a breach into its OT network, with the worst outcome only being averted due to the ineptness of attackers [11].

Applying redundancy to PLCs is straightforward and yet it should only be done after the consideration of resource costs. PLC hardware is expensive and implementing multiple versions of software can be costly and extended system implementation time. A high degree of PLC redundancy leads to enormous resource costs, hence it is undesirable in the context of real-world ICS deployments. This paper observes the practical resource constraints of ICSs, proposing a heterogeneous redundant architecture that allows system designers to balance security and resource costs.

A further consideration when designing a heterogeneous redundant framework is the possibility of concurrent attacks. That is, security incidents involving more than one version of the same component at the same time. To launch such an attack, an attacker needs to uncover vulnerabilities across different versions of a PLC. While this is rare and challenging, it is not impossible. Industroyer, most likely a state-sanctioned malware, includes payload components for four industrial control communication protocols [12]. If the history of ICS incidents taught us anything, it is that a well-financed actor, such as a state, is fully capable of leveraging vulnerabilities in order to breach their targets [8], [13]. This paper addresses concurrent attacks by scaling the redundant architecture whilst being mindful of resource constraints.

A. Contributions

This paper proposes a novel resource-aware ICS architecture that embeds heterogeneous redundancy in order to defend against cyberattacks on PLCs. Under a data-driven selection scheme, it purges the system of compromised components under attacks while keep resource costs under control. When resource constraints are lessened, we scale the architecture in redundancy to counter concurrent attacks beyond singe-point breaches. Our proposed solution is the first heterogeneous redundant architecture to address issues of resource constraints and concurrent attacks, which are both common in real-world ICS deployment. In this paper, we show that:

- A novel dual-PLC ICS architecture with native pairwise comparison can offer limited yet comparable defence against single-point breaches;
- A machine learning-based selection mechanism can deliver resilience against non-concurrent attacks under resource constraints;
- The proposed architecture can be scaled up to counteract concurrent attacks with modest resource implications.

B. Paper Structure

The remainder of this paper is structured as follows: Section II discusses recent work on using heterogeneous redundancy to enhance ICS security. Section III outlines a typical ICS architecture and attack modes targeting PLCs. Section IV presents our heterogeneous redundant architecture for countering PLC attacks, of which three variants are derived. Section V elaborates on our experimental setup, including the industrial process, control algorithm, disturbance scenarios, and attacks being carried out on a simulated ICS. Section VI presents the results of our experiments and discusses their resource and security implications. Section VII concludes this paper with a summary of findings and a discussion of extensions.

II. Related Work

In this section we discuss relevant research in ICS security. Section II-A addresses general attack mitigation strategies for ICSs. Section II-B covers heterogeneous redundancy and its roots in proactive rejuvenation. Section II-C considers the application of machine learning in ICS defence.

A. Attack Mitigation

Availability is a vital requirement for an ICS. An ICS is expected to maintain control over the underlying industrial process at all times. Adverse consequences will inevitably ensue where an ICS or associated resources become unavailable. As such, any cyberattack should be countered and its impact mitigated, either manually by the operator or automatically, without disrupting the operation of the ICS. In comparison to the large volume of research on attack and intrusion detection, work on attack mitigation remains relatively scarce [14], [15]. Most detection techniques assume manual intervention is to follow, leaving scant details to inform the mitigation actions. Literature on attack mitigation typically advance detection and response jointly. Measures of the latter could include restarting compromised components [16], compensating malicious signals [17], [18], modifying update period, and shutting down the whole system [19].

B. Heterogeneous Redundancy

Heterogeneous redundancy can play a central role in the mitigation of cyberattacks. When heterogeneous redundancy is implemented for critical components, a system can switch to a replica component and maintain its operation when the system becomes aware that one component has become compromised or non-functional. Heterogeneous redundancy finds its roots in proactive rejuvenation, in which diverse replicas function jointly and are restored periodically [20], [21]. Proactive rejuvenation is introduced into large-state applications, e.g., an ICS, in [22] with compiler-based software diversification and delivers high performance of state transfer, following a successful integration of intrusion-tolerant protocols into Supervisory Control and Data Acquisition (SCADA), a type of ICS, in [23].

Intrusion-tolerant architectures in [24], [25] replicate PLCs and use pairwise comparisons among replicas for intrusion discovery. Unlike proactive rejuvenation, both architectures only restore the malicious PLCs upon discovered misbehaviour. As a comparison, proactive rejuvenation methods can counter

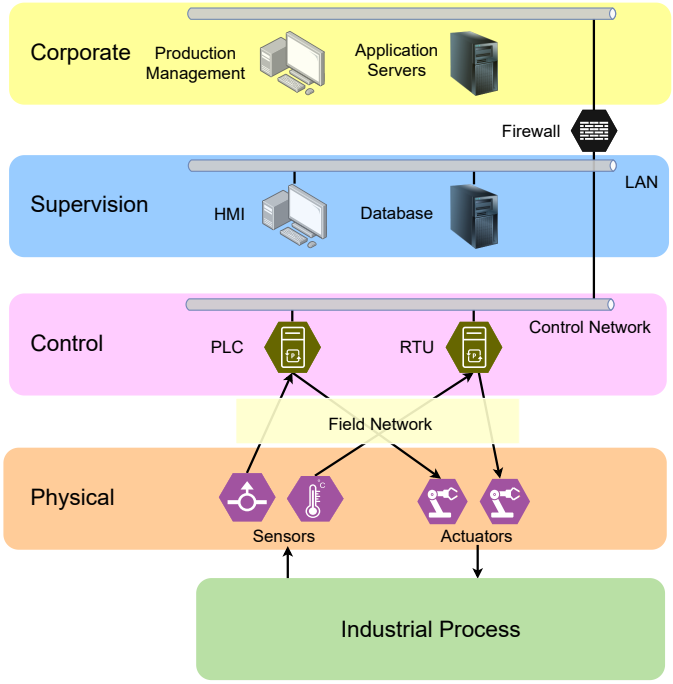


Fig. 1. System Model of an ICS.

concurrent attacks but exhibit a high degree of redundancy, leading to undesirable resource implications, while the two methods disregard the concurrent attacks in design and yet require at least three PLCs. Using a component selection approach based on machine learning, this paper addresses both issues and offers a trade-off between resource efficiency and system security.

C. Machine Learning in ICS Defence

Machine learning already has a strong foothold in research on ICS defence. An Intrusion Detection System, or IDS, in ICS is either signature-based or anomaly-based. Signature-based IDS uncovers an intrusion with pre-defined patterns, while anomaly-based with the deviation from normal behaviours learnt from data. The availability of an increasing volume of operational data has focused recently developed anomaly-based IDSs overwhelmingly on supervised machine learning techniques over statistical methods [26], [27]. Machine learning also finds its way into attack mitigation, with reinforcement learning orchestrating attack detection and response actions in place of human operators in [28]. This paper integrates machine learning techniques into diversity-based ICS defence to address limitations in previous work.

III. Models

In this section, we outline the assumed system and attack modes. These model and modes are a basis for the experimental results presented in Section VI.

A. System Model

An ICS is composed of a minimum of four layers. These are the Process, Control, Supervision, and Corporate layers, as shown in Fig. 1. The Physical layer contains a set of field devices such as sensors and actuators, which interface with the physical process. The Control layer accommodates the Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs), with local control capabilities for the former and wireless communication capabilities for the latter. The Supervision layer hosts the Human Machine Interface (HMI), data historians, and other workstations to enable real-time monitoring and control. The Corporate layer, separated by a firewall, shelters servers for production management and enterprise applications. Some literature adopt the Purdue Enterprise Reference Architecture [29], which divides the last layer further into manufacturing operations and business logistics, though this is not necessary or relevant to the contributions or scope this paper.

B. Attack Modes

We consider four types of attack that an adversary can launch against the PLCs in an ICS.

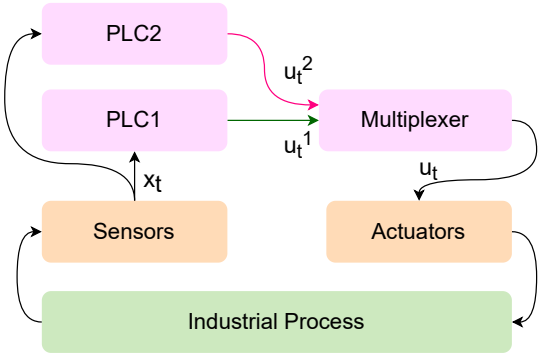
1) **Denial of Service (DoS) attacks**: DoS attacks flood the controller with requests, typically to cause the dropping of control commands. Actuators interfacing with the controller will default to the last command received.
2) **Replay attacks**: Replay attacks tamper with the control by replaying control command sequences from historical data, say, when the process is in a different scenario.
3) **Injection attacks**: Injection attacks generally seek to derail processes by sending fake control commands from compromised PLCs.
4) **Setpoint Override attacks**: Setpoint override attacks are a sophisticated form of attack that seeks to impair processes by overriding the target setpoints of the control algorithm.

For Injection and Setpoint Override attacks, the adversary is expected to possess advanced knowledge of the industrial process to launch an attack with meaningful physical impact.

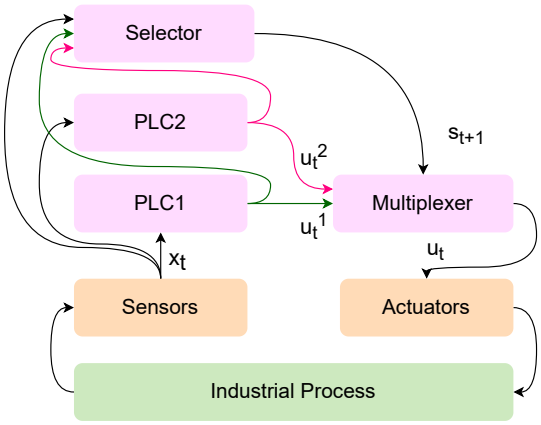
IV. Heterogeneous Redundant Architecture

Applying heterogeneous redundancy to PLCs in an ICS is founded on the premise that diversified components prevent singe-point security breaches. When one PLC replica is under attack, the system still has replica components to fall-back on. Heterogeneity hardens the system defences but it comes with increased costs with regard to system deployment. To launch a successful attack, an adversary needs to uncover and exploit vulnerabilities across multiple, if not all, replicas of the PLCs. The effectiveness of heterogeneous redundancy in defence depends on both the diversification technique and the fall-back strategy. An effective diversification scheme should decrease the probability of shared vulnerabilities across replicas, whereas a desirable fall-back strategy should promptly remove the control of compromised PLCs.

A distinction should be drawn between heterogeneous redundancy and homogeneous redundancy, or simply referred to as, duplication. While both techniques produce replicas of a component, duplication delivers the same version of hardware



(a) Pairwise Comparison-based



(b) Machine Learning-based

Fig. 2. Heterogeneous Dual-PLC Architecture.

and software across replicas. This leaves all other replicas vulnerable to the same exploit should one vulnerability exist on any one replica.

A. System Architecture

Previous research utilises diversified replicas for validating control signals from one another. The authors of [24] make three diversified versions of the PLC and a hardware-only selector to select and pass through the signal from one out of three PLCs. The selector compares outputs from each pair of PLCs and if there is a match, either one of the output is passed through to the actuators. We identify this strategy as a form of pairwise comparison. While easy to implement and promised to deliver correct outputs under non-concurrent attacks, pairwise comparison finds its limits when costs of diversification is high and resources of PLC hardware heavily constrained. Pairwise comparison is presumed to be inapplicable to having only two replicas of PLC since a pair of conflicting replicas cannot validate each other. Thus, pairwise comparison requires at least three replicas.

a) Pairwise Comparison-based Dual-PLC Architecture: Our solution observes the reality of resource constraints. We starts with only two replicas of the PLC in the ICS. We present a pairwise comparison-based dual-PLC architecture for ICS in Fig. 2a. Layers of supervision and corporate are omitted here as they remain unchanged. Two replicas of the PLC are made using diversification techniques, as discussed in Section IV-C, and output the same signals at all time despite their heterogeneous nature. Both receive the latest measurements x_t from the sensors, execute the control algorithm, and send the control commands u_t^i to a hardware-only multiplexer. Being pairwise comparison-based, the multiplexer then compares the commands and if there is a match, delivers either one of the commands to the actuators. If the commands do not match, the multiplexer will block both commands until there is a match. The latter scenario is equivalent to an DoS attack on the PLC. As a result of not receiving new commands, the actuators will default to the last command received from the multiplexer. This challenges the convention wisdom that pairwise comparison is inapplicable to any dual-PLC architecture. However, as we will show in Section VI, the pairwise comparison-based architecture delivers defence against non-concurrent attacks, albeit limited, as it promptly halts some physics-aware attacks.

b) Machine Learning-based Dual-PLC Architecture: We adopt a data-driven selection strategy, i.e., with machine learning techniques, to prune any PLC replica under attacks from control. Fig. 2b depicts our machine learning-based dual-PLC security architecture. A selector that uses machine learning algorithms is added to the system. Each replica outputs signals to both the selector and the multiplexer. The selector also interfaces with the sensors to receive latest measurements of process variables. In each round of control, the selector runs a machine learning algorithm on latest signals from sensors and both replicas to derive anomaly scores for both. It then selects the lowest-scored PLC, i.e., the PLC less likely to be under attacks, and sends the selection signal $s_{t+1} \in \{1, 2\}$ to the multiplexer. The hardware-only multiplexer passes the commands from s_t-th replica to the actuators. Unlike its pairwise comparison-based counterpart, the machine learning-based multiplexer does not block both replicas' signals at the same time and always defaults to PLC1 in absence of selection signal.

Our data-driven approach benefits from the wealth of historical data accumulated in process monitoring. Attacks, however, are of low frequency in real-world and labelling them is both difficult and costly. Unsupervised machine learning learns from unlabelled data to recognise or generate samples. We leverage anomaly detection in unsupervised learning to identify abnormal events (attacks in our case) which deviate from normal behaviours, using patterns learnt from normal operational data. Extensive coverage of this form of anomaly detection can be found in [30].

c) Machine Learning-based Tri-PLC Architecture: To counteract concurrent attacks, we scale up the dual-PLC architecture to tri-PLC by introducing a third replica. While an adversary could in theory launch attacks on all replicas simultaneously, the costs associated with such attacks could turn away even some well-financed actors. At the same time, the tri-PLC architecture also serves a testament to further scaled up solutions for defending critical infrastructure.

B. Controller Selection Problem

The goal of selector in machine learning-based architectures, either dual-PLC or tri-PLC, is to remove the controllers being compromised from controlling the process. Here we formulate the controller selection problem. Suppose $P = \{1, 2, \ldots, N\}$ is the set of PLC replicas in a heterogeneous redundant N-PLC architecture. At each round of control t, the sensors output process variables being monitored x_t. Each PLC replica, say the i-th replica, then execute the control algorithm and sends out manipulated variables u_t^i. Note that x_t and u_t^i are vectors of respective lengths. The selector, being on the receiver end on both sensors and PLC, has been collecting time series of x_t and u_t^i by each replica since the beginning. The selector also outputs a selection signal $s_t \in P$ to the multiplexer for signal selection. The manipulated variables that the actuators actually receive and execute become $u_t = u_t^{s_t}$.

Let $X_t = \{x_1, x_2, \ldots, x_t\}$ be the time series of process variables, $U_t^i = \{u_1^i, u_2^i, \ldots, u_t^i\}$ the time series of manipulated variables by i-th PLC where $i \in P$, and $C_t \subset P$ the set of compromised PLC replicas at t. Given X_t and $U_t = \{U_t^1, U_t^2, \ldots, U_t^N\}$ at any time t, the selector should always select a PLC $s_{t+1} \in P$ such that $s_{t+1} \notin C_t$.

C. Heterogeneous Redundancy

Diversified replicas form the basis of the proposed redundant architecture, distinguishing it from its traditional counterpart with only singular components. To implement diversified replicas, one could use software, hardware diversification techniques, or the combination of both.

a) Hardware Diversification: The global market for PLCs is vast, which has led to a degree of commoditisation and convergence with regard to functionality. As such, a PLC from one vendor is likely to be interchangeable with a PLC from another vendor, should the devices they are interfacing with operate on non-proprietary communication protocols. Alternatively, a replica could be a different PLC model from the same vendor. One way to reduce costs of diversification is to retain the older generation when upgrading the PLC. Retaining older PLCs could actually harden the system, as some vulnerabilities materialise on the new generation only.

b) Software Diversification: Most techniques of software diversification originated from fault tolerance and, lately, cybersecurity perspectives [31]. N-version programming, an unsophisticated form of software diversity, assigns multiple development teams a common specification to develop multiple versions of software independently. An early study in [32] showed that 27 versions of a anti-missile system exhibited several correlated faults and called into question the independence assumption. Conjoined with the costs of hiring multiple teams, N-version programming becomes unfavourable.

A more practical approach would be some form of compiler-based diversification [33]. This involves techniques such as reverse stack [34] and the randomisation of the instruction set and the address space [35]. Another solution, resembling N-version programming and yet specific to PLC, would be implementing the control algorithm using more than one of the five programming languages outlined in IEC 61131 [36], an industrial standard for PLC.

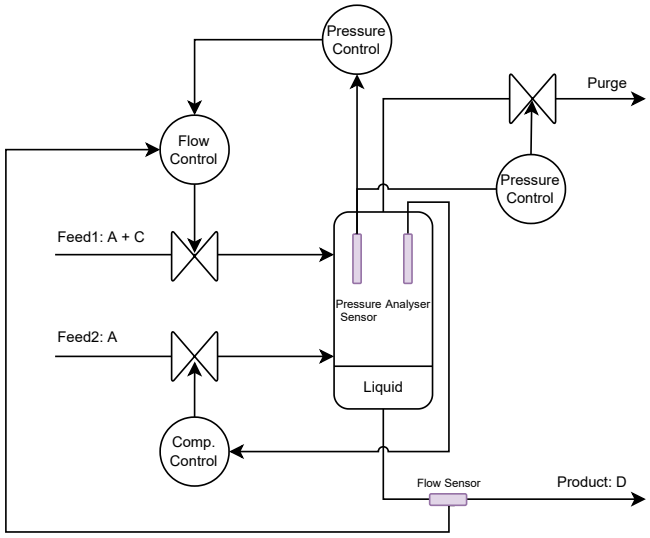


Fig. 3. Tennessee Eastman (TE) Process with Multiple Control Loops.

V. Experimental Setup

In this section we provide the details of the experiments that produced the results presented in Section VI. This includes coverage of the underlying physical process, simulated attacks, and evaluation methods.

A. Physical Process

To evaluate our proposed defence strategies, we model real-world ICS using our lightweight ICS simulator. Component-wise, it is self-contained with a built-in physical process, sensors, actuators, PLC, data historians, and user-definable devices. It is also network-adaptive via TCP Modbus protocol to facilitate inter-device communication. For the physical process, we implement Ricker's version of the Tennessee Eastman (TE) process and multiloop control [37], as showin in Fig. 3. The process is based on a real-world chemical reaction developed by the Eastman Chemical Company, hence its name, and over the years has become a benchmark for industrial control studies with its control strategies thoroughly studied. It models an irreversible reaction in a vessel with two in streams, Feed1 and Feed2, and two out streams, Purge and Product. Both Feed and Purge can be directly controlled via the openness of valve on each stream. Product is regulated using the vessel's liquid level.

B. Attacks

In line with the categorisation in Section III-B, our simulation environment natively supports modelling attack incidents of four different types. Each incident has the attributes of the target variable(s), start time, and end time. Replay attack requires an additional input of control sequence for each controlled variable manipulated. Both Injection and Setpoint Override have additional parameters of target value(s) and

drifting duration (via linear interpolation) for each variable manipulated. As a result, the parameter space for attacks is enormous and modelling all possible attacks becomes computationally intractable.

We instead define a subset of attacks that could materialise from likely attackers with diverging backgrounds and goals. For example, we consider nation state actors who might take a conservative approach by increasing Purge in the long run, or an aggressive one in active cyber engagement by opening both Feed immediately. We also consider amateur hackers who might disrupt the system by launching DDoS attacks. In the end, we implement a total of 20 attack incidents, of which four are DoS, three Replay, nine Injection, and four Setpoint Override. Users can also define new incidents with little effort using attack APIs of the simulator.

C. Evaluation

All solutions to the controller selection are to be evaluated under unsupervised settings. They are also to be compared against the two-out-of-three architecture in [24].

To enable unsupervised learning, we split the ICS simulation into two parts, normal operations without controller selection and operations under attacks with selection. The former is for collecting data to train machine learning algorithms, and the latter for testing. Notice that pairwise comparison-based solutions require no training data. We also differentiate the two parts by the conditions in which the process operate for bias prevention purposes. To mimic those changing conditions in real world, Ricker defined six scenarios of disturbances [37]. We reserve the first and third scenarios and scenario of no disturbance for testing, while using the rest for training. The combination of three scenarios and 20 attack incidents lead to a total of 59 test cases (a Replay attack is dropped due to replaying the correct sequence). Each case is also simulated three times through, mounting to 177 runs per solution evaluated. Each run lasts for 90 hours in simulated time, in which the disturbance is enacted between Hour 10 and 70 and the incident starts no earlier than the disturbance.

All PLCs send out control signals every six minutes and the selector operates at the same frequency. To leverage time series anomaly detection algorithms for selection, we constraint all time series fed into the selector with a window size $K = 10$.

VI. Results

Section VI-A and VI-B present results for the controller selection under test cases with single attacks and concurrent attacks respectively. Both success rate (SR) and accuracy (ACC) are provided for each architecture and algorithm evaluated. A test run is considered to be a success if the pressure reading inside reactor never breaches 3,000 kPa.

Accuracy is the ratio of time during attack(s) when a non-malicious PLC is being selected. Many machine learning algorithms evaluated are non-deterministic. Added to the live nature of the industrial process, accuracy score of only one iteration is not representative of the algorithm's performance. To address the issue, each test case is simulated three times through and the average accuracy across the three iterations is reported. Standard deviation (SD) for the accuracy scores is also computed in order to provide a fair indication of the robustness of each algorithm.

For machine learning-based architectures, we study six machine learning algorithms, each representative of their class. They are linear models AutoRegression [38], Principal Component Analysis [39], distance-based KDiscord [40], density-based Cluster-Based Local Outlier Factor [41], and neural-network-based Long Short-Term Memory [42] and Mingle-Objective Generative Adversarial Active Learning [43].

A. Non-concurrent Attacks

Under the non-current attack settings, the attacker always launches attacks against a single PLC. While multiple controlled variables might be targeted at the same time, only one PLC is compromised for the whole attack duration.

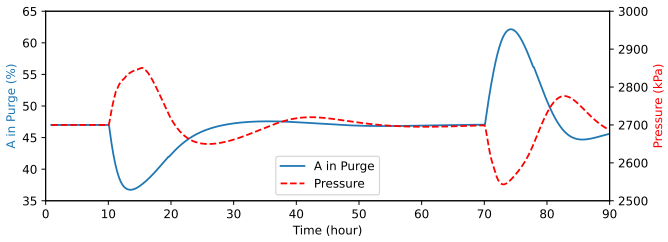
Table I presents the results of the proposed dual-PLC architectures against a baseline tri-PLC architecture. The baseline is a pairwise comparison-based tri-PLC architecutre, analogous to the two-out-of-three architecture in [24]. The results show that the baseline is near-perfect in defence but no perfect. It failed on a test case due to slight divergences in control signals towards the end of the simulation, most likely the result of abrupt changes in sensor measurements and asynchronous receipt on the controller side. This reveals the limitation to the "perfect" selector, albeit only in some rare cases.

Compared to the system without defence, all machine learning-based dual-PLC architectures improves the defence success rates of the system, except for AutoRegression. The pairwise comparison-based dual-PLC architecture has 20% improvement under injection attacks, but loses 25% under set-point override attacks. The dual-PLC architecture with CBLOF delivers the same success rates as the tri-PLC baseline, despite having less PLCs. The dual-PLC architectures with PCA and KDiscord even perfect the defence and outperform the baseline under injection attacks, despite suffering a 13% loss under replay attacks. It's worth noting that injection attacks is a major theme in many ICS security literature. In a deployment environment where injection attacks is a major concern, such architectures might be preferred over CBLOF and the baseline, or alternatively, pairwise comparison-based dual-PLC over single-PLC.

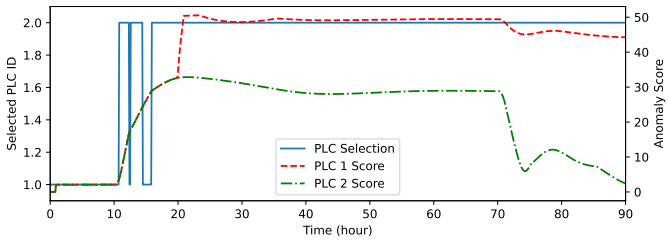
Fig. 4 depicts a test run of the machine learning-based dual-PLC architecture with CBLOF under an injection attack. A disturbance in feed composition begins at Hour 10 and causes the composition of A in Feed1 to decrease. This is evident in decreases of A in Purge. In response, the controller increases the Feed2 valve. At Hour 20, the attackers launches an injection attack to fully open both Feed valves and close Purge instantly from PLC 1. With CBLOF as the scoring algorithm, the selector always selects PLC 2 over PLC 1 from then on, as the anomaly score for PLC 1 significantly the one for surpasses PLC 2. The process remains in firm control of a genuine controller and the worst outcomes are averted.

TABLE I
SUCCESS RATES AND ACCURACIES OF CONTROLLER SELECTION UNDER SINGLE-ATTACK TEST CASES

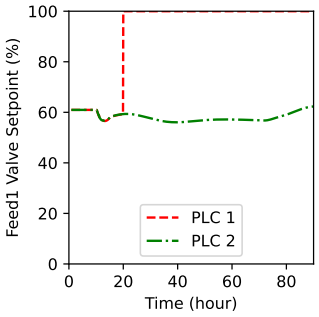
Arch.	Algo.	DoS			Replay			Injection			Setpoint Override		
		SR	*ACC*	*SD*	*SR*	*ACC*	*SD*	*SR*	*ACC*	*SD*	*SR*	*ACC*	*SD*
single	-	1.00000	0.00000	0E+00	0.75000	0.00000	0E+00	0.51852	0.00000	0E+00	0.91667	0.00000	0E+00
dual	pc	0.91667	0.00000	0E+00	0.75000	0.00000	0E+00	0.74074	0.00000	0E+00	0.66667	0.00000	0E+00
dual	autoreg	0.94444	0.07486	5E-03	0.87500	0.27226	2E-02	0.77778	0.68122	2E-02	0.86111	0.68706	3E-02
	pca	1.00000	0.31946	4E-02	0.87500	0.21250	0E+00	1.00000	0.85704	2E-02	1.00000	0.85137	3E-02
	kdiscord	1.00000	0.53645	5E-02	0.87500	0.74488	9E-04	1.00000	0.88387	3E-02	1.00000	0.93602	2E-02
	cblof	1.00000	0.44448	1E-02	1.00000	0.88557	5E-02	0.98765	0.80553	3E-02	1.00000	0.89884	2E-02
	lstm	1.00000	0.37503	5E-02	0.87500	0.76839	3E-02	0.75309	0.57905	4E-02	0.97222	0.61027	1E-01
	mogaal	1.00000	0.29585	9E-02	0.87500	0.24377	8E-02	0.64198	0.50597	5E-02	0.94444	0.70102	4E-02
tri	pc	1.00000	0.31020	4E-03	1.00000	0.95132	4E-03	0.98765	0.89433	1E-03	1.00000	0.88946	8E-04



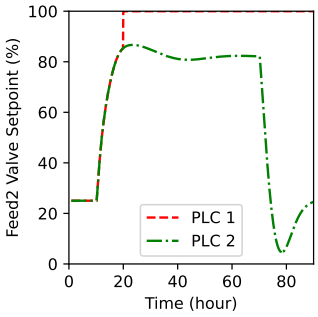
(a) Sensor Measurements



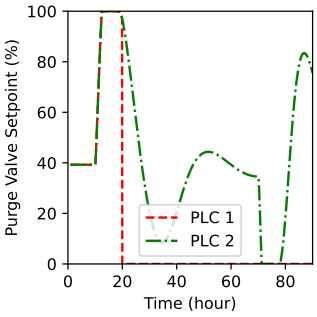
(b) Selector



(c) Feed1 Valve



(d) Feed2 Valve



(e) Purge Valve

Fig. 4. Machine Learning-based Dual-PLC Architecture (CBLOF) under an Injection Attack.

While higher accuracy scores do not guarantee higher success rates, thus safety for the system, they do help minimise disruptions caused by the attacks. In that regard, dual-PLC architectures with KDiscord and CBLOF deliver disruption mitigation closest to their pairwise comparison-based tri-PLC counterpart. DoS is a special category of attacks where even the defence-free single-PLC architecture achieves 100% success rate. The secondary task of mitigating disruptions thus becomes primary. However, no single selection algorithm, not even the baseline, fares better than a random selector in terms of accuracy. One explanation is that the last control signals which the actuator defaults to in absence of renewed signals under DoS attacks are so close to the true signals that the selection algorithm cannot tell them apart.

B. Concurrent Attacks

Under the concurrent attack settings, the attacker launches same attacks against two PLCs at the same time. As in the non-concurrent attack settings, multiple controlled variables might be targeted.

Table II shows the results of the proposed machine learning-based tri-PLC architectures against a baseline pairwise comparison-based tri-PLC architecture. The baseline does not provide any meaningful defence and has the same success rates across attack categories when compared to a dual-PLC architecture with random selection. This is because under concurrent attacks, two of the PLCs always output the same control signals, leading the pairwise comparison algorithm to always choose either one. For machine learning-based tri-PLC architectures, all except AutoRegression and LSTM outperform the baseline and the dual-PLC architecture. PCA and KDiscord deliver the best defence with only two failed cases under replay attacks and otherwise impeccable success rates. CBLOF trail both algorithms closely, having failed single cases under injection and setpoint override attacks respectively.

Under the machine learning-based defence framework, de-

TABLE II
SUCCESS RATES AND ACCURACIES OF CONTROLLER SELECTION UNDER CONCURRENT-ATTACK TEST CASES

		DoS			Replay			Injection			Setpoint Override		
Arch.	**Algo.**	***SR***	***ACC***	***SD***	***SR***	***ACC***	***SD***	***SR***	***ACC***	***SD***	***SR***	***ACC***	***SD***
dual	random	1.00000	0.00000	0E+00	0.75000	0.00000	0E+00	0.55556	0.00000	0E+00	0.91667	0.00000	0E+00
tri	pc	0.97222	0.00000	0E+00	0.75000	0.00000	0E+00	0.56790	0.00000	0E+00	0.88889	0.00000	0E+00
tri	autoreg	1.00000	0.09851	7E-03	0.87500	0.23965	3E-02	0.77778	0.62688	1E-02	0.83333	0.64089	2E-02
	pca	1.00000	0.35175	2E-02	0.87500	0.21258	1E-04	1.00000	0.82097	1E-02	1.00000	0.84442	3E-02
	kdiscord	1.00000	0.51146	7E-02	0.87500	0.74538	5E-04	1.00000	0.87582	2E-02	1.00000	0.90826	6E-03
	cblof	1.00000	0.39169	2E-02	1.00000	0.81258	9E-02	0.98765	0.79775	4E-02	0.97222	0.89363	3E-02
	lstm	1.00000	0.35656	7E-02	0.95833	0.73504	2E-02	0.77778	0.55815	4E-02	0.83333	0.51540	3E-02
	mogaal	1.00000	0.29040	5E-02	0.83333	0.20975	5E-02	0.59259	0.51198	5E-02	0.94444	0.57473	1E-01

fending against concurrent PLC attacks costs only one more PLC. In the case of pairwise comparison-based, however, we would need five PLCs and implement a three-out-of-five architecture to negate these two malicious PLCs. Another observation is that scaling up the machine learning-based architectures does not impair their defence capabilities. Comparing Table II and I side by side, the performance of machine learning-based tri-PLC architectures closely resemble the performance of their dual-PLC counterparts under non-concurrent attacks. These two observations should encourage operators of ICS to favour machine learning-based over pairwise comparison-based security architecture for the former is less resource-intensive. In the ever-changing cyber landscape, ICS operators can increase the number of PLCs in the machine learning-based architecture, either by introducing diversified replicas or retaining old ones when upgrading, to counter increasingly sophisticated attacks, without impairing the overall defence performance.

VII. CONCLUSIONS

This section summarises the contributions made in this paper and discusses future work.

A. Summary of Contributions

Applying heterogeneous redundancy to critical components such as PLCs has been proven to improve the ICS's defence against single-point security breaches. Yet the component selection approaches in existing heterogeneous redundant architectures remain native, relying on comparisons between diversified replicas. This leads to high resource usage and the system to crumble in face of concurrent attacks which target more than one PLCs. In this paper we address both issues, demonstrating a data-driven component selection approach that achieves a trade-off between resources cost and security. In particular, we propose (i) a novel dual-PLC ICS architecture with native pairwise comparison can offer limited yet comparable defence against single-point breaches, (ii) a machine-learning based selection mechanism can deliver resilience against non-concurrent attacks under resource constraints, (iii) a scaled up variant of the proposed architecture counteracts concurrent attacks with modest resource implications.

B. Future Work

This work constitutes an initial study of data-driven heterogeneous redundant PLC architecture for ICS security. Here, heterogeneous redundancy is exclusively applied to PLCs and its effect evaluated. This exclusiveness leaves open the possibility of building ICS security architecture on top of other critical components, since they might also benefit from active usage of redundancy under a data-driven component switching scheme. They should include a minimum of sensors and actuators, each of which plays an important role in process control. Looking further, we also expect a comprehensive security framework that can leverage all diversified critical components for defence and adjust dynamically the number of active replicas for performance.

REFERENCES

[1] A. Hahn, "Operational technology and information technology in industrial control systems," in *Cyber-security of SCADA and other industrial control systems*. Springer, 2016, pp. 51–68.

[2] K. Stouffer, J. Falco, K. Scarfone *et al.*, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[3] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A.-R. Sadeghi, M. Maniatakos, and R. Karri, "The cybersecurity landscape in industrial control systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1039–1057, 2016.

[4] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." *HotSec*, vol. 5, p. 15, 2008.

[5] A. C. Panchal, V. M. Khadse, and P. N. Mahalle, "Security issues in iiot: A comprehensive survey of attacks on iiot and its countermeasures," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*. IEEE, 2018, pp. 124–130.

[6] S. Schneider, "The industrial internet of things (iiot) applications and taxonomy," *Internet of Things and Data Analytics Handbook*, pp. 41–81, 2017.

[7] P. Zheng, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, X. Xu *et al.*, "Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives," *Frontiers of Mechanical Engineering*, vol. 13, no. 2, pp. 137–150, 2018.

[8] K. E. Hemsley, E. Fisher *et al.*, "History of industrial control system cyber incidents," Idaho National Lab.(INL), Idaho Falls, ID (United States), Tech. Rep., 2018.

[9] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, and T. Longstaff, "Survivable network systems: An emerging discipline," Carnegie-mellon Univ Pittsburgh PA Software Engineering Inst, Tech. Rep., 1997.

[10] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, symantec corp., security response*, vol. 5, no. 6, p. 29, 2011.

[11] Vericlave and Blueridge, "The kemuri water company hack," Vericlave, Tech. Rep., 2018.

[12] A. Cherepanov, "Win32/industroyer: A new threat for industrial control systems," *White paper, ESET (June 2017)*, 2017.

[13] T. M. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[14] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial internet of things security: requirements and fog computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[15] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on scada systems: secure protocols, incidents, threats and tactics," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020.

[16] X. Li, C. Zhou, Y.-C. Tian, and Y. Qin, "A dynamic decision-making approach for intrusion response in industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2544–2554, 2018.

[17] A. Rosich, H. Voos, Y. Li, and M. Darouach, "A model predictive approach for cyber-attack detection and mitigation in control systems," in *52nd IEEE conference on decision and control*. IEEE, 2013, pp. 6621–6626.

[18] L. F. Combita, A. A. Cardenas, and N. Quijano, "Mitigating sensor attacks against industrial control systems," *IEEE Access*, vol. 7, pp. 92 444–92 455, 2019.

[19] A. Zedan and N. H. El-Farra, "A machine-learning approach for identification and mitigation of cyberattacks in networked process control systems," *Chemical Engineering Research and Design*, vol. 176, pp. 102–115, 2021.

[20] T. Roeder and F. B. Schneider, "Proactive obfuscation," *ACM Transactions on Computer Systems (TOCS)*, vol. 28, no. 2, pp. 1–54, 2010.

[21] L. T. d. N. Brandão and A. N. Bessani, "On the reliability and availability of replicated and rejuvenating systems under stealth attacks and intrusions," *Journal of the Brazilian Computer Society*, vol. 18, no. 1, pp. 61–80, 2012.

[22] M. Platania, D. Obenshain, T. Tantillo, R. Sharma, and Y. Amir, "Towards a practical survivable intrusion tolerant replication system," in *2014 IEEE 33rd International Symposium on Reliable Distributed Systems*. IEEE, 2014, pp. 242–252.

[23] J. Kirsch, S. Goose, Y. Amir, D. Wei, and P. Skare, "Survivable scada via intrusion-tolerant replication," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 60–70, 2013.

[24] M. Denzel, M. Ryan, and E. Ritter, "A malware-tolerant, self-healing industrial control system framework," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2017, pp. 46–60.

[25] J. Luo, M. Kang, E. Bisse, M. Veldink, D. Okunev, S. Kolb, J. G. Tylka, and A. Canedo, "A quad-redundant plc architecture for cyber-resilient industrial control systems," *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 218–221, 2020.

[26] J. Suaboot, A. Fahad, Z. Tari, J. Grundy, A. N. Mahmood, A. Almalawi, A. Y. Zomaya, and K. Drira, "A taxonomy of supervised learning for idss in scada environments," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[27] 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.

[28] J. Mern, K. Hatch, R. Silva, C. Hickert, T. Sookoor, and M. J. Kochenderfer, "Autonomous attack mitigation for industrial control systems," *arXiv preprint arXiv:2111.02445*, 2021.

[29] T. J. Williams, "The purdue enterprise reference architecture," *Computers in industry*, vol. 24, no. 2-3, pp. 141–158, 1994.

[30] C. C. Aggarwal, *Outlier Analysis*. Springer, 2016.

[31] B. Baudry and M. Monperrus, "The multiple facets of software diversity: Recent developments in year 2000 and beyond," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–26, 2015.

[32] J. C. Knight and N. G. Leveson, "An experimental evaluation of the assumption of independence in multiversion programming," *IEEE Transactions on software engineering*, no. 1, pp. 96–109, 1986.

[33] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz, "Compiler-generated software diversity," in *Moving Target Defense*. Springer, 2011, pp. 77–98.

[34] B. Salamat, A. Gal, and M. Franz, "Reverse stack execution in a multi-variant execution environment," in *Workshop on Compiler and Architectural Techniques for Application Reliability and Security*, 2008, pp. 1–7.

[35] G. S. Kc, A. D. Keromytis, and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 272–280.

[36] M. Tiegelkamp and K.-H. John, *IEC 61131-3: Programming industrial automation systems*. Springer, 2010.

[37] N. L. Ricker, "Model predictive control of a continuous, nonlinear, two-phase reactor," *Journal of Process Control*, vol. 3, no. 2, pp. 109–123, 1993.

[38] J.-i. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 482–492, 2006.

[39] I. T. Jolliffe, *Principal component analysis for special types of data*. Springer, 2002.

[40] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proceedings of the international conference on very large data bases*. Citeseer, 1998, pp. 392–403.

[41] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern recognition letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[42] A. Singh, "Anomaly detection for temporal data using long short-term memory (lstm)," 2017.

[43] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1517–1528, 2019.