Alma Mater Studiorum Università di Bologna

# Cyber Threats in IIoT: Can ML Mitigate Vulnerabilities and Prevent Harm?

Gerardo Antonio Corral Ruiz
*Laurea Magistrale in Governance e Politiche dell'Innovazione Digitale*
*email gerardo.corralruiz@studio.unibo.it*

**ABSTRACT**
This paper investigates under which technical and organizational conditions a Machine-Learning-based Intrusion Detection System (ML-IDS), deployed in Industrial Internet of Things and Operational Technology (IIoT/OT) environments, can reduce cyber-risk. Rather than treating AI as a technocratic fix, the analysis is explicitly framed within cyber-risk management and governance concepts, using the NIST Cybersecurity Framework 2.0 and a classic threats–vulnerabilities–damages perspective as reference. Focusing on the *Detect* function, the study examines a single legacy service—Modbus telemetry—from the ToN_IoT testbed, using both a balanced `Train_Test_IoT_Modbus` subset and the highly imbalanced full `IoT_Modbus` dataset as a two-phase experimental setting. Decision Trees, Random Forests, and Support Vector Machines are trained on four Modbus counters to solve binary (normal vs. attack) and multiclass (normal plus five attack types) classification tasks. The results show that Modbus-only ML-IDS can provide a useful, interpretable detection layer, yet they only address a pretty narrow slice of the cyber-risk equation. The paper argues that such detectors must be complemented by richer telemetry, continuous monitoring, and governance decisions about acceptable residual risk, rather than being deployed as standalone solutions.

**Key words:** Industrial Internet of Things (IIoT); operational technology (OT); machine-learning-based intrusion detection; NIST Cybersecurity Framework 2.0; cyber-risk governance; ToN_IoT dataset.

## 1 INTRODUCTION

This paper investigates the concrete conditions under which a Machine-Learning-based Intrusion Detection System (ML-based IDS), deployed in Industrial Internet of Things (IIoT) and Operational Technology (OT) environments, can significantly reduce the risk of physical and operational damage. The analysis adopts a realistic perspective, explicitly shaped by resource constraints (data availability, computational capacity, time, human skills, lack of planning) and by the limitations of existing risk-management and governance methodologies. The goal is not to propose a technocratic solution that "solves" cybersecurity through the simple introduction of new technologies, but rather to assess to what extent such an IDS methodology can provide meaningful value within a sensible risk-management strategy.

Throughout the *Governance della Cybersecurity* course, we learned that a substantial share of cybersecurity problems in organizations stems from low maturity in the governance of "cyber" issues: incidents are not automatically eliminated by introducing more sophisticated technologies. Adding new tools without a clear strategy and a robust governance plan does not address the root of the problem. As Professor Colajanni repeatedly emphasized, organizational resources are limited: it is neither possible to address all vulnerabilities nor to deploy all conceivable countermeasures. Priorities must therefore be set, and time, budget, and competences must be allocated carefully. These decisions are not taken by engineers in isolation, but by those responsible for governance and management, who ultimately decide which level of residual risk is acceptable.

In IIoT/OT environments, security decisions are particularly delicate because operational technology is directly coupled to real-

time physical processes.[1] A security failure may lead to production downtime, equipment damage, or even risks to human safety.[2] Any serious cybersecurity reasoning must therefore start from a basic question: who are you as an organization, and only then you'll be able to identify who the most relevant attackers might be and which threats are sufficiently plausible—considering both their probability and their potential impact, as assessed through the conversion table reviewed in the course—to deserve priority. Only on this basis does it make sense to discuss the specific role that an ML-based IDS can play within the broader portfolio of available security controls.

Risk management, as discussed in class, is not a one-off decision but a continuous process that can be described through the *Plan–Do–Check–Act* (PDCA) cycle and that is reflected in compliance-oriented frameworks such as the *NIST Cybersecurity Framework 2.0*, which organizes cybersecurity into the functions *Govern*, *Identify*, *Protect*, *Detect*, *Respond* and *Recover*. In this work, I will focus primarily on the *Detect* function, analyzing to what extent an ML-based IDS, deployed in realistic IIoT/OT settings under limited resources, can contribute in a reasonable way to this risk-management cycle, and which trade-offs its adoption entails when compared to alternative countermeasures.

## 2 BACKGROUND AND RESEARCH MOTIVATION

At the beginning of this semester, I asked Professor Colajanni for advice. I had just finished reading Shoshana Zuboff's *The Age of Surveillance Capitalism*. What struck me most in this book was the way Zuboff describes the potential problems that are already emerging in the Internet of Things (IoT). We are creating an ever-growing number of networks and devices that we connect and market as "smart", [3] but that are primarily designed to maximize capital returns rather than to protect users.[4] Security and privacy are rarely the main design goals, and the systematic reduction of vulnerabilities that can affect people's lives is often treated as secondary.

With this new interest in IoT, I approached Professor Colajanni to ask whether he could recommend a more engineering-oriented book that addressed the same underlying problem and the real risks of IoT. He pointed me to Bruce Schneier's *Click Here to Kill Everybody*, a very insightful text that reinforces exactly this perspective: there is a strong trend towards connecting everything, which is not only fashionable but also potentially beneficial for society, yet this progress is not being pursued with serious attention to people's safety. Step by step, we are exposing ourselves to new risks that are increasingly hard to anticipate: from the possibility of remotely hacking a connected car and endangering the lives of its occupants, to analogous scenarios in aircraft; from a "smart" oven that could release gas, to other situations in which the boundary between digital failure and physical harm becomes thinner and thinner.[5] When this logic is extended to factories, power plants, or hospitals, the issue is no longer about convenience, but about physical safety.

Another intellectual reference that shaped this work emerged when I discussed Yuval Noah Harari's book *Nexus* with Professor

[1] Venanzi et al. observe that, historically, IT and OT have operated separately: "Historically, IT and OT have operated in isolation, with IT prioritizing scalability and interoperability, and OT emphasizing real-time control, reliability, and safety." (Venanzi et al., 2025,p. 2). However, the growing interconnection of new devices and systems has progressively changed this separation within the new I4.0 paradigm: "The maturity of the I4.0 vision has favored the spread of studies, research, surveys, and roadmaps addressing various aspects of the new industrial revolution and focusing on the technologies that are speeding up SMEs' transition to digital" (Venanzi et al., 2025,p. 10)

[2] As Bruce Schneier warns in the Introduction of *Click Here to Kill Everybody* (Schneier, 2018): "Today's threats include the possibility of hackers remotely crashing airplanes, disabling cars, and tinkering with medical devices to murder people. We're worried about being GPS-hacked to misdirect global shipping and about counts from electronic voting booths being manipulated to throw elections. With smart homes, attacks can mean property damage. With banks, they can mean economic chaos. With power plants, they can mean blackouts. With waste treatment plants, they can mean toxic spills. With cars, planes, and medical devices, they can mean death. With terrorists and nation-states, the security of entire economies and nations could be at stake."

[3] In chapter 7, Zuboff describes how Eric Schmidt's famous claim that "the internet will disappear" at the World Economic Forum Annual Meeting of 2015 really means that it will dissolve into a ubiquitous, invisible computing environment, echoing Mark Weiser's earlier vision of "ubiquitous computing" embedded seamlessly in everyday life. (Zuboff, 2019). In the IoT context, this becomes a problem because the network grows more complex and harder to analyze, giving attackers a wider surface to exploit.

[4] In her Introduction, Zuboff warns that the new digital technologies (IoT included) are filling our lives with "smart" devices whose real priority is profit, not user protection, so security, privacy, and the systematic reduction of life-impacting vulnerabilities are treated as secondary concerns. (Zuboff, 2019).

[5] In page 14, Schneier wrote: "The thing about Internet+ security is that we're all used to it. Up to now, we've generally left computer and Internet security to the market. This approach has largely worked satisfactorily, because it mostly hasn't mattered. Security was largely about privacy, and entirely about bits. If your computer got hacked, you lost some important data or had your identity stolen. That sucked, and might have been expensive, but it wasn't catastrophic. Now that everything is a computer, the threats are about life and property. Hackers can crash your car, your pacemaker, or the city's power grid. That's catastrophic."(Schneier, 2018)

Colajanni in class. In this book, Harari argues that the world's information networks are becoming increasingly complex and interdependent.[6] In his view, if we ever speak meaningfully about something like a "singularity", it will not necessarily appear as a single, isolated superintelligent system, but rather as a dense network of interconnected systems so complex that no human being can fully understand them. For me, this idea became intuitively visible in what happened with Cloudflare in November 2025: a single technical point of failure was enough to disrupt, in unexpected ways, a large number of services and organizations.[7] I do not claim that this event already constitutes a "singularity" in any strict sense, but I do see it as a warning sign of how dependent we are on opaque infrastructures and how easily a local problem can trigger cascading effects.

This paper clearly does not aim to answer questions at the vast scale raised by Zuboff, Schneier, or Harari. Doing so would require more than an individual research project and would demand sustained collective work and reflection. What I try to do instead is to bring these broader concerns down to a more concrete and operational level. In my case, this means looking at Industrial IoT and Operational Technology (IIoT/OT) environments, where digital risks rapidly translate into physical and operational consequences, and asking to what extent a Machine-Learning-based Intrusion Detection System can actually help under the real technical, organizational, and governance constraints that characterize existing organizations.

Interestingly, I already had this academic and research interest at the beginning of the semester, but I did not yet have a concrete idea of how to approach it. In mid-November, we had the opportunity to attend a lecture by the PhD candidate Isabella Marasco, who introduced us to several applications of Artificial Intelligence in cybersecurity. That session broadened my perspective because it showed, in a very clear way, that AI is already being applied to tasks such as intrusion detection (both signature-based and anomaly-based), anomalous traffic classification, and also parts of the SOC workflow, like alert filtering. At the same time, it highlighted the real limitations of these approaches: the dependence on high-quality data, the computational cost, and the constant evolution of threats over time — "A security model trained yesterday is already obsolete today!"

In that lecture, Isabella presented a number of ML and DL models that are currently used in intrusion detection systems, especially in anomaly-based approaches. On the one hand, she discussed supervised learning algorithms such as *Decision Trees*, *Random Forests*, *Support Vector Machines*, and *Gradient Boosting*, trained on labelled traffic to distinguish between normal patterns and attacks. On the other hand, she mentioned deep learning models such as *deep neural networks*, *autoencoders*, and more recent architectures based on *Graph Neural Networks* (GNNs), which are better suited to representing complex data structures. Isabella explained that in the transition of IDS from signature-based processes to anomaly-based approaches, ML and DL have been crucial for handling growing data volumes and increasingly subtle patterns, although such models require constant updates to remain effective in changing environments.

In the same lecture, Isabella mentioned a research line that is attracting increasing attention in cybersecurity: *Continual Learning* (CL). Rather than training a model once and deploying it, CL aims to update the model continuously as new data arrives and the environment changes. The promise for cybersecurity is amazing: an IDS that does not remain frozen within the boundaries of a historical dataset, thereby reducing the need for continuous retraining — a process that is costly in both time and economic resources. The difficulty, as she emphasised, is the phenomenon of *catastrophic forgetting*: as the model learns new information, it tends to "forget" part of what it previously knew, degrading its performance on older classes unless memory management and replay or regularization strategies are carefully designed.

Initially out of confusion and then out of curiosity, I began exploring the material from our same course (*Governance della Cybersecurity*) but in the *Laurea Magistrale in Artificial Intelligence*. There I found that the final project is centred precisely on these ideas. Its starting point was the observation that many *AI-based IDSs* still operate under a static paradigm: they are trained offline on a dataset that captures the threat landscape at a particular moment in time, and then deployed almost as if the threats do not evolve. This approach clashes with reality: the threat landscape is dynamic and new attack types and zero-day exploits constantly appear. A static IDS, even if powered by ML/DL, quickly becomes fragile in some contexts.

While going through the project material, I came across the *ToN_IoT* family of datasets, specifically designed to evaluate intrusion detection methods in IoT and IIoT environments.[8] This is precisely the dataset I will use in this work, albeit with a more

[6] Actually, Bruce Schneier agrees with this assumption as well, as seen in (Schneier, 2018): "According to sociologist Charles Perrow's theory of complexity, complex systems are less secure than simpler ones and, as a result, attacks and accidents involving complex systems are both more prevalent and more damaging. But Perrow demonstrates that not all complexity is created equal. In particular, complex systems that are both nonlinear and tightly coupled are more fragile. For example, the air traffic control system is a loosely coupled system. Both individual air traffic control towers and airplanes have failures all the time, but because the different parts of the system only mildly affect the others, the results are rarely catastrophic [...] The Internet is the opposite: it's both nonlinear in that pieces can have wildly out-of-proportion effects on each other, and tightly coupled in that these effects cascade immediately—characteristics that make catastrophes much more likely." (p. 197)

[7] The outage stemmed from an internal configuration error that produced an oversized Bot Management feature file, overwhelming global proxies. It shows how growing infrastructural complexity makes small internal faults capable of causing wide-scale disruption.(Cloudflare, 2025)

[8] For a detailed description of the TON_IoT testbed see (Moustafa, 2021).

"classical" focus: I will not design a Continual Learning pipeline, but rather experiment with ML models trained in a static batch, as is still common in a significant portion of the current literature on "intelligent" IDSs for IIoT,[9] as far as I know the downsides of it I still want to get familiarized with the more classic methodologies before jumping into the more advanced/exploratory ones.

In this paper, therefore, I will not explore Continual Learning, even though I consider it conceptually very promising. The reason is not only technical, but also organizational, in line with Professor Colajanni's insistence that resources are limited, and we need to choose our priorities. In my case, the scarce resource is time (and, to some extent, computational capacity and implementation complexity). Implementing and rigorously evaluating an incremental IDS would require a much larger, likely collaborative project, whereas the scope of this paper is more restricted. I am aware that this leaves out an important part of the problem. Nonetheless, at this stage I am particularly interested in further developing my skills in applied ML, and therefore I opt for a more "standard" approach.

## 3 CONCEPTUAL FRAMEWORK - IIOT/OT ARCHITECTURE, CYBER RISK AND GOVERNANCE

In this section, I introduce the conceptual framework that will support the rest of the paper. First, I clarify what is meant by Industrial IoT and Operational Technology (IIoT/OT), and I outline some key components and features that are particularly relevant for this work. I then examine these environments and the kinds of threats, vulnerabilities, and impacts that are most relevant in these settings.

Next, I discuss how my analysis of IIoT/OT and an ML-based Intrusion Detection System (IDS) fits within frameworks such as the NIST Cybersecurity Framework 2.0, particularly in relation to the *Detect* function that we are going to explore in depth in this work.

Finally, I introduce some key concepts that will help us understand the practical case study based on the ToN_IoT dataset, such as the protocols and some physical components (such as sensors), and some representative attack types.

Mapping the world of IIoT/OT risks with this conceptual framework will help us to answer to the following research question: under which technical and organizational conditions can an ML-based IDS, deployed in IIoT/OT environments, meaningfully contribute to the Detect function within a realistic cyber-risk management strategy?

### 3.1 What is IIoT and OT?

Industrial internet of things "...is a system of connected devices designed to monitor, control, and optimize industrial operations."[10]. The distinction between general IoT and Industrial IoT is mainly shown by the operational purpose of the second. Its scale and the consequences of system failure. General IoT (for public and consumers) is primarily focused on home automation and smaller-scale applications, typically connecting devices via Wi-Fi or cellular networks. IIoT devices instead usually use wired networks such as Ethernet or industrial protocols like Modbus or Profibus to connect.I am going to speak about protocols in the next subsections.[11] In terms of the "five nines" availability we discussed in class, IIoT systems operate in real time and have far less tolerance for downtime.

As IIoT is directed toward large-scale industrial applications, it demands unique operational characteristics that impose strict requirements on underlying technology. These systems require adherence to some particular protocols and standards which we are going to speak about later.

Operational Technology (OT) instead is defined in the NIST guide[12] as a broad range of programmable systems and devices that interact with the physical environment or manage devices that interact with the physical environment. This guide also tell us that IIoT is included as a specific example of an OT system type in the definition context. Both IIoT and OT systems are generally involved in the control of physical processes. OT, by definition, is a broad range of programmable systems and devices that interact with or manage devices that interact with the physical environment, detecting or causing a direct change through the monitoring and/or control of devices, processes, and events. IIoT, instead, "...consists of sensors, instruments, machines, and other

[9] On the prevalence of batch learning in ML-based IDS Miani et al. established that: "For example, the learning process of most ML-based IDS found in the literature is typically done using batch learning mode. This means that the ML algorithm is applied once to a static training dataset, and after that, the built model is used to make predictions for incoming data."(Miani et al., 2025,p. 72954)

[10] cf.(PaloaltoNetworks, 2025)

[11] cf.(Parshv, 2023)

[12] cf. The guide of Operational Technology Security made by Stouffer et. al adds a comment specifically for OT systems: "OT systems consist of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that act together to achieve an objective (e.g., manufacturing, transportation of matter or energy).".(Stouffer et al., 2023,p. 8)

devices that are networked together and use internet connectivity to enhance industrial and manufacturing business processes and applications [Berge]. As IT and OT systems continue to converge and become even more interconnected, the control of physical processes remains a relatively unique and critical concept of OT." (Stouffer et al., 2023,p. 26). Because IIoT systems are part of the OT environment, they share core security needs, specifically the need for resilience, safety, and security, as the introduction of IT capabilities into physical systems (which is how OT evolved and IIoT operates) can introduce new security implications as we saw in classes, but also as we may read in Schneier's book.[13]

### 3.2 Evolution of Security Models

IIoT is considered one of many Industrial Control Systems (ICS). ICS are integrated hardware and software systems designed to monitor and control industrial processes in sectors such as energy, manufacturing, water, and transportation.

Historically, ICS relied on Supervisory Control and Data Acquisition (SCADA) systems, which were often isolated or "air-gapped."[14] The increasing interconnectivity inherent to the IIoT paradigm exposes SCADA components to new vulnerabilities. Both SCADA and IIoT are industrial control systems (ICS).[15] The SCADA architecture is built around legacy protocols not designed for public network exposure, it struggles with the interoperability and rapid integration demands of modern industrial environments. [16]

CSE ICON explains that SCADA and IIoT are not mutually exclusive; however, SCADA systems require significant updates if they are to meet modern safety and security standards. As we discussed in Prof. Colajanni's class, choosing between these architectures depends fundamentally on who you are and what you do: the right solution must align with your sector, operational priorities, and governance model. According to CSE ICON, IIoT is preferable in scenarios where data-driven insights are essential for decision-making, where cloud-based adaptability is required to manage scalable growth and dynamic demands, where cross-device interoperability fosters collaborative ecosystems, and where predictive maintenance through data analytics is a strategic goal. All these characteristics align naturally with the implementation of ML-based IDS, reinforcing its relevance within modern industrial infrastructures. SCADA systems are used in critical infrastructures like electrical power and water distribution systems, and the increased inter-connectivity of these devices to the Internet has introduced new challenges and security vulnerabilities.[17] An important point to note is that, while detecting cyber attacks in increasingly interconnected SCADA networks is indeed complex, machine learning methods have gained significant popularity as a means of strengthening their defense. [18]

Furthermore, conventional security solutions, such as signature-based Intrusion Detection Systems (IDS), are fundamentally challenged by the complex and rapidly changing threat landscape of IIoT, as I mentioned in Section 2. These traditional methods depend on predefined patterns to detect threats and are therefore limited against previously unknown-or zero-day-attacks. Due to the enormous volume and complexity of data generated by rapidly expanding IoT ecosystems exacerbate this challenge, making it difficult for conventional systems to efficiently process and analyze network data in real time without compromising accuracy.

The transition from segregated industrial automation to fully networked IIoT represents a fundamental architectural change, moving from proprietary, centralized systems to open, decentralized, and layered frameworks.

### 3.3 The Industrial IoT Architectural Framework

SCADA systems employ a centralized architecture, where the control center dictates and monitors all activities from a single hub. This design, while suitable for historical operations, leads to issues in flexibility and interoperability, as mentioned in the last subsection.

[13] I remember his example about door locks (it is about IoT not IIoT, but still I consider it relevant): Consider a pickpocket. Her skill took time to develop. Each victim is a new job, and success at one theft doesn't guarantee success with the next. Electronic door locks, like the ones you now find in hotel rooms, have different vulnerabilities. An attacker can find a flaw in the design that allows him to create a key card that opens every door. If he publishes his attack software, then it's not just the attacker, but anyone, who can now open every lock." (Schneier, 2018,p. 33)

[14] cf. (Stouffer et al., 2023,p. 29) The NIST SP 800-82 Rev. 3 exposes this concept.

[15] A difference worth to be mentioned is that SCADA relies on a centralized architecture, where a central server controls and monitors data from multiple field devices. In contrast, IIoT employs a decentralized architecture, where devices communicate directly or through edge computing, reducing reliance on a central hub.(CSE ICON, 2023)

[16] Legacy protocols such as Modbus, DNP3, or Profinet were originally designed for isolated networks: "Legacy systems often lack resources that are common on modern IT systems. Many systems may also lack desired features, including encryption capabilities, error logging, and password protection." Traditional SCADA wasn't built-in with security features. This is explained in (Stouffer et al., 2023,p. 29).

[17] "[I]ncreased inter-connectivity of SCADA devices to the Internet has brought a new set of challenges. These interconnected SCADA networks are now more vulnerable to various cyber attacks." (Timken et al., 2023,p. 1).

[18] While individual ML methods are used, the authors' experiments demonstrate that combining multiple algorithms through ensemble methods is necessary for a more comprehensive solution when defending against cyber attacks in SCADA systems. cf. (Timken et al., 2023,p. 1).

**Table 1.** Architectural Differences: Traditional SCADA vs. Modern IIoT

| **Feature** | **Traditional SCADA** | **Modern IIoT Architecture** |
|---|---|---|
| Architecture | Centralized server controls/monitors field devices | Decentralized (Edge / Fog Computing) |
| Data Focus | Real-time adjustments/control | Big Data analytics, long-term efficiency optimization |
| Security Status | Inherently weak; relies on air-gapping and physical security | Requires strong digital security (Zero-Trust) |

In contrast, IIoT utilizes a decentralized architecture. This model allows devices to communicate directly or through local servers, significantly reducing reliance on a central server and minimizing network latency. This architectural pivot improves operational resilience, allowing applications to function even in environments with low bandwidth or where external connectivity is disconnected. [19]

In spite of the migration to modern systems, many industrial processes continue to rely on older protocols like Modbus TCP. These legacy protocols introduce substantial technological debt that must be compensated for architecturally (an example might be make it air-gapped). Modbus TCP typically lacks robust authentication mechanisms and transmits data without encryption, often in plain text. [20] This lack of basic security renders Modbus highly susceptible to common cyber threats, including network scanning and injection attacks (among others), where malicious data can disrupt or compromise industrial processes. I am going to explore some of this in my practical example analyzing the ToN_IoT dataset.

Although there are some measure to address the lack of encryption (like Transport Layer Security), its deployment is "...rarely practical due to legacy device limitations and the complexity of certificate management, especially in large-scale deployments" [21]. Defensive strategies that shift toward external mechanisms, such as anomaly detection, rather than relying solely on inherent protocol-level prevention, may prove more effective. Other great practices for mitigation strategies include strong authentication, network segmentation, continuous monitoring, and the use of robust frameworks like a Zero Trust model to secure legacy Modbus devices externally (which I will briefly revisit later). [22]

While I am going to focus on the legacy Modbus-based protocol (because of the ToN_IoT dataset), it is important to acknowledge that the industrial communication landscape is not standing still. Newer protocols and security extensions explicitly aim to address the lack of authentication and encryption in classic ICS traffic. For example, *OPC UA* integrates encryption, authentication, and authorization as part of its core design, and is promoted as a secure, interoperable alternative for Industry 4.0 architectures. [23] However, the same constraints that limit the deployment of TLS in legacy Modbus-long device life cycles, limited processing capabilities, real-time constraints, and the cost and risk of replacing running systems—also slow down the adoption of these newer stacks. As a result, many operators will continue to run mixed environments.

### 3.4 Relevant Security Frameworks

The aim of this work—as I already established before—is not only to raise awareness about the evolving risks inherent to the IoT landscape—particularly within the OT and IIoT domains—but also to critically examine the potential role of machine learning models in strengthening the cyber defense of these environments. This analysis is grounded in a multidimensional approach to security: one that goes beyond technical capabilities and takes into account organizational, strategic, and governance-related factors. Ultimately, my objective is to identify where and under what conditions these technologies can be meaningfully integrated—not as isolated technical fixes, but as components of a broader, coherent cybersecurity posture aligned with real-world constraints and responsibilities.

The NIST Cybersecurity Framework (CSF) 2.0 outlines a comprehensive structure for managing cybersecurity risks through six core Functions: *Govern, Identify, Protect, Detect, Respond*, and *Recover*. Within this architecture, the **Detect** Function plays a pivotal role in enabling the timely discovery of anomalies and potentially adverse events, thus serving as a critical of risk analysis and countermeasure execution (NIST, 2024,p. 4). This is especially relevant in OT/IIoT environments, where delayed detection can rapidly escalate into physical damage or operational disruption.

[19] cf. (CSE ICON, 2023)
[20] Veridify (a security company) established that: "Without adequate protection, malicious actors can exploit vulnerabilities in Modbus implementations to disrupt industrial processes, compromise sensitive data, and even jeopardize public safety." (Veridify Security Inc., 2025)
[21] cf. (Veridify Security Inc., 2023)
[22] cf. Idem.
[23] cf. (Nankya et al., 2023,p.21)

**Table 2.** Examples of threats, vulnerabilities, and damages in IIoT/OT environments (reorganised from Liebl et al. (Liebl et al., 2024)).

| Threat / attack goal | Typical enabling vulnerabilities | Typical damages / impacts |
|---|---|---|
| Espionage, intellectual property theft | Web-based vulnerabilities, lack of encryption, weak authentication, misconfigured remote access | Loss of confidentiality, competitive disadvantage, regulatory and contractual violations |
| Denial of Service (DoS/DDoS) | Design flaws and bugs (e.g., no rate limiting), flat network topology, insufficient resource isolation | Loss of availability, production downtime, degraded quality of service |
| Ransomware and destructive malware | Code-execution and memory-manipulation flaws, privilege escalation, weak segmentation between IT and OT, poor backup and recovery procedures | Unavailability of critical systems, data loss, halted industrial processes, financial loss |
| Spoofing and command/data injection | Communication manipulation (no integrity or authentication), insufficient input validation and sanitisation, weak identity verification | Loss of integrity of commands and telemetry, unsafe actuation, incorrect set-points in industrial processes |
| Abuse / maloperation by insiders or contractors | Misconfiguration of access control, lack of logging and non-repudiation mechanisms, physical access to interfaces and ports | Safety incidents, equipment damage, regulatory non-compliance, difficulty attributing responsibility |

It is precisely in these areas where machine learning-based intrusion detection systems (ML-IDS) offer the most tangible contribution. For this reason, I remark the importance of the practical and strategic implications of integrating ML into the *Detect* function, as a meaningful component within an adaptive cybersecurity governance strategy and not only a technocratic solution.

According to the CSF Core, "DETECT (DE) enables the timely discovery and analysis of anomalies, indicators of compromise, and other potentially adverse events that may indicate that cybersecurity attacks and incidents are occurring. This Function supports successful incident response and recovery activities." (NIST, 2024,p. 4) This is particularly relevant to my work because it is precisely at the level of detection where machine learning techniques are most actively being explored and proposed, especially in IIoT/OT environments.

Although I focused in the NIST CSF 2.0 framework and in an ML-based IDS approach, it is worth briefly mentioning the *Zero Trust Architecture* (ZTA). As emphasized in the course of *Governance della Cybersecurity*, Zero Trust is not merely a technical solution, but a strategic and organizational shift that assumes no implicit trust within any part of the network, enforcing continuous verification, least-privilege access, and strict segmentation. In the context of IIoT/OT, this approach forces organizations to rethink how access is granted to critical industrial components. While Zero Trust offers a promising conceptual reinforcement to the risk reduction logic discussed in this paper, its practical integration into IIoT/OT remains a complex endeavor—one that goes beyond the scope of the present work.

### 3.5 Vulnerability and Threat Taxonomy

In industrial environments, the priority is not only data confidentiality, but above all operational continuity and the safety of people and assets.

Many IIoT devices are designed for long life cycles and deterministic behaviour, but not for robust security; Paloalto networks highlighted that IIoT devices often lack secure communications, weak legacy systems and protocols (as I mentioned with Modbus), lack of segmentation because of the interconnectivity, or weak or absent authentication. If one such device is compromised, it can easily become a pivot into the wider OT network. Major threats observed in IIoT include Denial of Service, Espionage, Intellectual Property Theft, Maloperation, Ransomware, Injections.[24] As a compact summary of this landscape, Table 2 reports some representative examples of common threats, vulnerabilities and damages in IIoT/OT environments.

Adversaries exploit system weaknesses that fall into several categories of vulnerabilities, such as the ones presented on Table 2. Specific protocol vulnerabilities are actively exploited as well. Legacy protocols like Modbus are highly susceptible to simple, yet devastating, attacks. For instance, the absence of robust built-in security in Modbus allows for continuous data modifications

[24] Some of these threats will be analysed in my practical case study on the ToN_IoT testbed. Liebl et al. discuss and explain several of them in Section V, "Threats, Vulnerabilities and Their Impact", of their paper (Liebl et al., 2024).

to registers, which can lead to a Denial of Service (DoS) attack that overwhelms the server and prevents it from determining the correct value.[25]

I haven't answered yet: how does an attack occur? In practical terms, many IoT and IIoT attacks follow a multi-stage pattern rather than a single, isolated exploit. An attacker typically starts with reconnaissance, scanning the network to identify exposed or weakly protected devices and open services. Once a candidate device is found, initial access is gained through default credentials, outdated firmware, or protocol-level weaknesses, and a malicious payload is executed on the device. From there, the malware attempts to establish persistence (for example by creating new accounts or tampering with logs), to evade basic monitoring, and to maintain a communication channel with a remote command-and-control (C&C) server. In later stages, the compromised node can be used for lateral movement towards more critical assets, for exfiltrating sensitive data, or for directly manipulating industrial processes. The final impact may range from service degradation and data theft to full operational disruption or equipment/people damage.[26]

A particularly relevant manifestation of this pattern in IIoT/OT is the rise of large-scale IoT botnets. These botnets continuously scan the Internet for vulnerable devices—including industrial gateways, cameras, and edge nodes—and, once compromised, reuse them as part of coordinated campaigns such as distributed denial-of-service (DDoS) attacks or as stepping stones into industrial networks. This is one portion of the attack surface that an ML-based IDS can monitor.

Given that critical industrial systems cannot always undergo comprehensive security patching or protocol upgrades due to operational constraints and system longevity, I think that the defensive strategy must prioritize advanced detection and rapid response capabilities that monitor behavioral anomalies in network traffic, that is where ML-based IDS could shine. Mapping vulnerabilities and threats makes explicit that an ML-based IDS can only address a narrow slice of the cyber-risk equation: it intervenes in the detection of exploitation attempts-typically at the network and, in some cases, at the device/telemetry layer-but it does not in itself remove underlying vulnerabilities (legacy protocols, flat networks, poor authentication) nor eliminate threat actors.

### 3.6 Machine Learning and Deep Learning for IIoT Threat Mitigation

#### *3.6.1 What ML-based IDS can offer*

I found some surveys and experimental research on ICS and IIoT environments that suggested ML-IDS can offer concrete advantages over traditional approaches.

ML-IDS is particularly remarkable in some tasks like: learn normal network or process behavior and detect deviations, including some zero-day attacks that signature systems miss;[27] handle high-volume heterogeneous traffic from IIoT devices and industrial sensors—as we saw in Isabella's class; work with different organizations and networks of data: centralized or distributed.[28]

ML then, provides powerful tools for anomaly-based detection in IIoT/OT environments, but their effectiveness depends heavily on context, relevant data, and integration into broader governance and risk management strategies as I have argued throughout this work. Some of the limitations of this approach were already discussed in Section 2, as we examined during Marasco's lecture for example. The main issues can be summarized as follows:

- New constant threats and lack of dynamism.
- High sensitivity to false alarms, SOC analysts often face an overwhelming volume of alerts, many of which are false positives, leading to alert fatigue and delayed response.
- Most research relies on synthetic or testbed datasets (e.g., BoT-IoT, ToN-IoT, Edge-IIoT).[29]
- Lack of explainability and trust; opaque (black-box) models are difficult for OT engineers to rely on for safety-critical decisions.

[25] cf.(Liebl et al., 2024)
[26] According to Ivan Lee, author of the article written for Wallarm (Wallarm, 2025).
[27] Umer et. al explained: "Each approach mentioned above has its pros and cons. Unsupervised learning does not require labeled training data, therefore, the dependency on attack data gets eliminated making it capable of detecting zero-day attacks. However, it usually produces high false alarms" (Umer et al., 2022,p. 10)
[28] In his doctoral thesis, Ghadim identified a new trend known as federated learning—a distinct machine learning paradigm explored as a means to perform network intrusion detection in distributed ICS environments. However, there are multiple approaches to implementing it. (Dehlaghi Ghadim, 2025)
[29] I am going to use the ToN-IoT.

• Vulnerability to adversarial machine learning: ML models can be evaded or compromised through adversarial examples and data poisoning. [30]

I considered worth to mention that digital twins are increasingly used to generate realistic labeled data and simulate attacks without endangering the plant, which significantly improves training coverage and validation of IDS behavior. But obviously it is an expensive technology.

#### *3.6.2 The main-used models*

The application of AI in IIoT security spans a range of methodologies, categorized primarily into supervised, unsupervised, and deep learning methods. A growing trend observed in academic literature is the use of hybrid and ensemble models, which are utilized to enhance overall threat detection accuracy and resilience.[31]

Deep learning architectures usually come with a significant cost in terms of computational resources, tuning complexity, and data requirements. Training and deploying them in resource-constrained IIoT environments is not trivial. [32] In addition, their lack of transparency makes it harder for OT engineers and security officers to trust them for safety-related decisions. In this work I treat deep and hybrid architectures as an important horizon of research, but not as part of my practical analysis.

We also studied in Marasco's lesson a smaller group of "classic" ML algorithms that keeps reappearing in empirical studies of IIoT intrusion detection: Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVM), and tree-based gradient boosting. RF and DT frequently achieve high accuracy on benchmark datasets. [33] These methods are good at maintaining relatively low inference costs. SVMs, on the other hand, can reach very high accuracy on known network topologies, but they tend to be computationally expensive to train.

My next step is to focus my experimental work on RF, DT and SVM as the main ML models to be evaluated on the ToN_IoT Modbus subset. These algorithms offer a pragmatic balance between detection performance, computational feasibility, and a level of interpretability that is compatible with industrial risk-management practices. Deep neural networks and more sophisticated hybrid schemes remain deliberately out of scope: they are acknowledged as promising directions for future work, especially in combination with continual-learning strategies, as presented in Section 2, but for this time I am not going to follow that track.

## 4 CASE STUDY - THE TON_IOT IIOT ENVIRONMENT

### 4.1 The ToN_IoT datasets

The ToN_IoT datasets (Telemetry, Operating systems, and Network traffic) are new generations of Industry 4.0/Internet of Things (IoT) and Industrial IoT (IIoT) datasets. They were specifically created for evaluating the fidelity and efficiency of different cybersecurity applications based on Artificial Intelligence (AI), including Machine Learning and Deep Learning algorithms.[34]

These datasets are particularly useful for validating and testing various AI-based cybersecurity applications, such as intrusion detection systems, threat intelligence, malware detection, fraud detection, privacy-preservation, digital forensics, adversarial machine learning, and threat hunting.

The name *ToN_IoT* reflects the fact that the dataset aggregates heterogeneous data sources from three main categories:[35]

(i) **Telemetry from IoT and IIoT sensors.** Sensor readings are stored in log and CSV files and include telemetry from more than ten IoT/IIoT devices, such as Modbus-based industrial sensors and environmental (weather) sensors.

(ii) **Operating-system datasets.** Host-level data was collected from Windows 7/10 systems using the Performance Monitor tool, and from Ubuntu 14/18 LTS systems using tracing tools such as `atop`. These logs capture disk, process, CPU, memory, and network activity on the monitored machines.

[30] Ghadim presented this: "...limited availability of realistic attack scenarios, the poor quality of existing datasets, and the susceptibility of ML-based IDS to adversarial attacks."(Dehlaghi Ghadim, 2025,p. 40)
[31] cf. (Kikissagbe and Adda, 2024)
[32] cf. (Bensaoud and Kalita, 2025)
[33] RF achieved an accuracy of 99.97% in their study, which was significantly higher than previously reported models, according to the text of Eid et. al (Eid et al., 2023).
[34] A detailed description of the ToN_IoT testbed and datasets is provided in the official UNSW project documentation (`https://research.unsw.edu.au/projects/toniot-datasets`) and in Moustafa's scientific article (Moustafa, 2021).
[35] cf.(Moustafa, 2021,p. 6)

(iii) **Network-traffic datasets.** Network traces were recorded as packet captures (`pcap`), log files, and CSV exports generated by the ZEEK (formerly Bro) network monitoring tool.

### 4.2 The directories

The ToN_IoT datasets are organized into five main directories:[36]

(i) **Raw_datasets.** This dataset: "involves the entire raw packets of normal and attack".[37] events collected from the testbed using the netsniff-ng tool.

(ii) **Processed_datasets.** "This directory includes 23 CSV files that extracted using the Zeek tool. Every file includes about million records except the last file contains 329,021 records, where each file has 44 attributes and two attributes of the class label and types either normal or attack category."[38]

(iii) **SecurityEvents_GroundTruth_datasets.** This directory contains the ground-truth security events (hacking activities) and their timestamps, which are used to label the corresponding records in the other datasets.

(iv) **Train_Test_datasets.** This directory provides sampled versions of the four datasets, in CSV format, specifically: "This file is suggested to be used for evaluating new AI-based cybersecurity solutions and make fair comparisons between the new security solutions".[39]

(v) **Description_stats_datasets.** This folder includes descriptions of the features contained in the processed datasets, together with summary statistics about the number of normal and attack records.

### 4.3 IoT_Modbus

In the processed ToN_IoT telemetry, Modbus activity is represented by a dedicated service called *IoT_Modbus*. Each row in the IoT_Modbus dataset is one snapshot of sensor's activity and is described by a small set of features, according to the *IoT features description* file that is part of the Moustafa's documentation:

- **date** and **time**: indicate when the telemetry entry was logged and give temporal context to the Modbus activity.
- **FC1_Read_Input_Register**, **FC2_Read_Discrete_Value**, **FC3_Read_Holding_Register**, and **FC4_Read_Coil**: numerical features related to four Modbus function codes that are commonly used in industrial control. They describe how often the sensor reads input registers, discrete values, holding registers, and coils over time.
- **label**: a binary value that marks each record as normal (0) or attack (1).
- **type**: a categorical value that gives a more detailed class for each record (for example: normal, DoS, DDoS, backdoor). This allows both binary and multi-class intrusion detection tasks.

Beyond the feature definitions, it is important to clarify what kind of data the *Train_Test_IoT_Modbus* subset actually contains and how it relates to the full *IoT_Modbus* telemetry. In the version used in this work, the train–test subset comprises 31,106 records, with 15,000 normal observations and 16,106 attack records distributed across five attack categories (injection, backdoor, password, scanning, and XSS). This produces a dataset that is still imbalanced—but much less compared to the corresponding full *IoT_Modbus* processed dataset, where 405,904 normal records coexist with relatively fewer attack samples (40,035 backdoor, 7,079 injection, 24,269 password, 529 scanning, and 577 XSS records).[40] The full processed datasets preserve the original class distributions observed in the testbed.

### 4.4 Threats

From a threat space perspective, across the whole ToN_IoT family, the authors describe nine attack types—scanning, DoS, DDoS, ransomware, backdoor, data injection, Cross-Site Scripting (XSS), password cracking, and Man-in-the-Middle (MITM) attacks—launched against different IoT and IIoT services.[41] In the *IoT_Modbus* service, only five of these categories appear in the telemetry:

[36] cf. Idem.
[37] Idem.
[38] Idem.
[39] Idem.
[40] As seen in the *Statistics of IoT Records* a file that is part of the Moustafa's documentation.
[41] cf. All described in his scientific article.(Moustafa, 2021,p. 7)

• **injection**,"is an attacking technique that injects or inserts any fake input data from clients to applications, such as SQL injection attacks for exploiting PHP and ASP applications.";[42]

• **backdoor**, associated with unauthorised remote access paths into the service, Moustafa added: "Once hackers gain access to targeted systems using attack scenarios, for example, ransomware, DoS or DDoS, hackers keep persistence to the hacked systems using the backdoor technique to steal personal and financial data, install additional malware, and hijack devices.";[43]

• **password**, corresponding to password-guessing or brute-force activity;[44]

• **scanning**, reflecting probing and reconnaissance of the Modbus service, Moustafa wrote: "The aim of this attack is to collect information about victim systems such as finding active IP addresses and open ports in the testbed network.";[45]

• **XSS**, Moustafa described it as: "The XSS attack happens when an attacker employs a web application to transmit malicious code, generally in the form of a browser side script, to different end-users.".[46]

In this sense, `Train_Test_IoT_Modbus` offers a compact but multi-class view of Modbus-related threats, suitable for training and comparing static ML classifiers. In later sections, I will first use this subset to design and tune Decision Tree, Random Forest, and Support Vector Machine models, and then assess how these models generalise when evaluated on the much larger—and more realistically imbalanced—full `IoT_Modbus` dataset.

## 5 METHODOLOGY AND RESULTS

All experiments were implemented in Python using Google Colab notebooks. The notebooks are exported in a public GitHub repository that accompanies this work: `https://github.com/GerardoACR/iiot_ids_project`. The repository already contains the notebooks and scripts used for data preprocessing, model training, and evaluation. The README file is describes the project structure.

For practical reasons, the original ToN_IoT datasets by Moustafa and colleagues are not redistributed in this repository. In this study, I downloaded the required CSV files (processed IoT Modbus telemetry and the corresponding train–test subsets) from the official UNSW distribution and executed the experiments on local copies. Anyone wishing to reproduce the results reported in this paper must first obtain the same ToN_IoT files from the authors' website[47] and then follow the code and notebooks available in the GitHub repository.

### 5.1 Overall experimental design

I have already stressed that this practical case study represents only one of the components required for a serious cybersecurity exercise. Likewise, I have shown that this project is situated within the *Detect* function of the NIST CSF and does not aim to cover the entire security lifecycle. Now, what follows is the technical part: the application of the proposed methodology to the *ToN_IoT* Modbus dataset and the analysis of static ML models. I'd like to express that for me, this analysis should not be read as an isolated benchmark, but as an experiment explicitly embedded within a risk-management framework, where model metrics only make sense in relation to governance decisions, resource constraints, and operational priorities.

Concretely, the experimental design is structured in two phases over a single industrial service: Modbus telemetry (which are represented in the two notebooks that I created for this project and are visible in my GitHub repository). In **Phase 1**, I use the *Train_Test_IoT_Modbus* subset as a relatively balanced and manageable environment to design and tune three classical ML models for tabular data: Decision Trees (DT), Random Forests (RF), and Support Vector Machines (SVM). These families of models are plausible candidates for IIoT/OT deployments, as they can be implemented efficiently on structured data and—at least in the case of DT and RF—offer a degree of interpretability that is valuable for operators and security teams. In **Phase 2**, I take the best configurations obtained in Phase 1 and evaluate them on the full processed IoT Modbus dataset, which preserves the original, highly imbalanced distribution between normal traffic and several attack types. This second phase is closer to a realistic deployment scenario and is used to assess how much performance degrades, or remains robust, when moving from a "laboratory" subset to an imbalanced, production-like setting.

[42] Idem.
[43] Idem.
[44] cf. Idem.
[45] Idem.
[46] Idem.
[47] See `https://research.unsw.edu.au/projects/toniot-datasets`

Within this two-phase design, I formulate two related learning tasks. First, a *binary* intrusion–detection task, where the goal is simply to distinguish normal from malicious records. Second, a more informative *multiclass* task, where the model must discriminate between normal traffic and five attack categories (injection, backdoor, password, scanning, and XSS). The multiclass setting is particularly relevant for this work because it mirrors the threat taxonomy discussed in the conceptual framework: different attacks represent different risks: missing a backdoor or injection attack could directly affect safety or operational continuity.

Finally, all experiments are conducted in a static *batch*. Models are trained once on historical data and then evaluated on held–out samples, without any form of online or continual adaptation. This is a deliberate methodological choice that aligns with the discussion in Section 2: while Continual Learning techniques are arguably better suited to cope with the evolving traffic patterns and threat landscape of IIoT/OT environments, they also require additional complexity, monitoring, and resources that go beyond the scope and constraints of this project.

In the following subsections, I detail the data preparation, feature preprocessing, model configurations, and evaluation metrics used to implement this experimental design.

### 5.2 Data preparation: from ToN_IoT to Modbus ML tasks

The practical case study in this work focuses on a single sensor within the ToN_IoT family: the `IoT_Modbus` unit. This device is described as a smart TCP/IP Modbus sensor that logs telemetry about Modbus activity through a small set of counters. The corresponding telemetry file exposes eight fields: `date`, `time`, four Modbus function counters (`FC1_Read_Input_Register`, `FC2_Read_Discrete_Value`, `FC3_Read_Holding_Register`, `FC4_Read_Coil`), a binary `label` (0 = normal, 1 = attack), and a categorical `type` specifying the attack category (for example: normal, injection, backdoor, password, scanning, XSS).

I designed two supervised learning tasks on top of the Modbus telemetry:

- **Binary classification**, where the objective is simply to distinguish normal traffic from any attack. In this case, the target label is derived directly from the `label` field (0 = normal, 1 = attack). This task corresponds to a minimal IDS scenario: a SOC team wants a reliable "red flag" for suspicious Modbus behaviour, without differentiating between attack types.
- **Multi-class classification**, where the objective is to distinguish six classes: normal, injection, backdoor, password, scanning, and XSS. Here the target label is the `type` field. This task connects directly to the threat taxonomy discussed in Section 3: detecting the presence of an attack is not the same as distinguishing between scanning, brute-force access attempts (password), or more stealthy backdoors. From a governance point of view, knowing which attack is occurring matters for prioritising response and estimating potential damage.

I splitted the experiments into two phases:

- **Phase 1** uses the `Train_Test_IoT_Modbus` subset (31,106 records). In this subset, normal traffic (15,000 records) and three attack types (injection, backdoor, password; 5,000 each) are relatively well balanced, while scanning and XSS remain minority classes (529 and 577 records respectively). This phase acts as a laboratory scenario in which I can design and tune models under more stable sampling conditions.
- **Phase 2** re-uses the best models selected in Phase 1 but evaluates them on the full `IoT_Modbus` dataset, which is a much larger and more imbalanced file (around 287,000 records in my local copy). This second phase acts as a stress test: the attack types are no longer evenly represented, and the distribution of normal vs. attack traffic is closer to a realistic IIoT environment, where attacks are rare but critical.

Throughout both phases, I keep the same input features, the same basic preprocessing steps, and the same evaluation metrics. This is intentional: the goal is not to chase the best possible score, but to see how well a reasonable, tree-based ML IDS survives the transition from a controlled to a more realistic Modbus environment.

### 5.3 Feature representation and preprocessing

For both tasks and phases, I restrict the feature space to the four Modbus counters:

- `FC1_Read_Input_Register`
- `FC2_Read_Discrete_Value`
- `FC3_Read_Holding_Register`
- `FC4_Read_Coil`

The `date` and `time` fields are excluded from the predictive models. The preprocessing pipeline was executed as follows:

**Label construction**

- For the binary task, I use the original `label` field and collapse all attack categories into a single *attack* class.
- For the multi-class task, I use the `type` field as the target and encode the six classes with `sklearn.preprocessing.LabelEncoder`.

This preserves the semantic distinction between attack types introduced earlier.

**Train/validation/test splits**

- In Phase 1, I split the `Train_Test_IoT_Modbus` subset into 60% training, 20% validation, and 20% test, using a stratified split to preserve class proportions across the three sets.
- Model selection and hyperparameter tuning are performed on the training + validation portion. Once the best configuration is chosen, I retrain on 80% of the data (train+val) and evaluate once on the held-out 20% test set.
- In Phase 2, I work directly on the full `IoT_Modbus` file, using a fresh 80/20 stratified train–test split. No extra tuning is performed here: the goal is to see how the Phase 1 models behave under a more realistic class imbalance.

**Scaling**

All four Modbus counters are scaled with `StandardScaler`, fitted on the training set and then applied to validation and test sets. This avoids leakage of test information into the model and stabilises the behaviour of algorithms such as SVM.

**Data quality checks**

Before training, I verify that there are no missing values and no non-finite entries in the Modbus counters. I also inspect basic class-conditional statistics (mean and standard deviation of each counter per class). These exploratory checks show that the four counters have broadly similar ranges across normal and attack traffic, which already suggests that any classification success will come from subtle joint patterns, not from trivial thresholding on a single variable.

### 5.4 Models, hyperparameter tuning and some of the results

I focused on three classical ML models:

- **Decision Tree (DT)**;
- **Random Forest (RF)**;
- **Support Vector Machine (SVM)**;

For both the multi-class and binary tasks in Phase 1, I perform grid search over a small hyperparameter space, using the validation split for model selection:

- For the Decision Tree, I vary the maximum depth and the minimum number of samples per leaf (and, in a second grid, also `min_samples_split` and `class_weight`), seeking a balance between fit and interpretability. The splitting criterion is kept at the default.
- For the Random Forest, I tune the number of trees, the maximum depth, and the `max_features` parameter.
- For the SVM, I explore different values of `C` and `gamma` for the RBF kernel and linear.

After model selection on the validation split, the chosen models were retrained on the full training set (80% of `Train_Test_IoT_Modbus`) and evaluated once on the held-out 20% test set. For the multi-class task (normal plus five attack types), I retain a Decision Tree as an interpretable baseline and a Random Forest as the main operational model. On the test set, the Decision Tree reaches accuracy $\approx 0.966$, macro-F1 $\approx 0.918$ and weighted-F1 $\approx 0.965$. The majority classes (*normal*, *backdoor*, *injection*, *password*) all achieve F1-scores around 0.95–0.98 with false negative rates below 7%, while the minority classes behave very differently: *XSS* is still handled reasonably well (F1 $\approx 0.918$, FNR $\approx 0.086$), but *scanning* remains clearly problematic (F1 $\approx 0.73$, FNR $\approx 0.37$). False positives are low for all classes (below 3% for *normal* and close to zero for *scanning* and *XSS*), so the tree provides a strong and readable baseline, yet it already shows that reconnaissance-style attacks are intrinsically hard to capture with Modbus counters alone. The tuned Random Forest (500 trees, `max_features = 2`, `class_weight = "balanced"`) slightly improves on this picture. It attains accuracy $\approx 0.964$, macro-F1 $\approx 0.931$ and weighted-F1 $\approx 0.963$, with *normal*, *backdoor*, *injection* and *password* all in the 0.96–0.98 F1 range. The *XSS* class improves [48] to F1 $\approx 0.955$ with almost no false positives, while *scanning*

[48] Compared to the less-greedy search, see the repository for more details.

increases to F1 $\approx 0.77$ but still suffers an FNR of about $0.37$. In other words, the forest offers better macro-F1 and a somewhat fairer treatment of minority classes than the single tree, while preserving very low false positive rates, but the persistent weakness on *scanning* confirms that the limitation stems from the Modbus-only telemetry rather than from model capacity.

Speaking about the Phase 2. For the multi-class task, both models perform well overall, but Phase 2 exposes clear weaknesses on minority attacks. In Phase 2 (full, imbalanced `IoT_Modbus`), the tree drops to accuracy $\approx 0.947$ and macro-F1 $\approx 0.840$ (weighted-F1 $\approx 0.947$), whereas the forest attains accuracy $\approx 0.970$, macro-F1 $\approx 0.897$ and weighted-F1 $\approx 0.969$. In both models, *normal*, *backdoor* and *injection* remain strong (F1 roughly in the $0.90$–$0.98$ range), but *password*, *scanning* and *XSS* show significantly higher false negative rates (FNR up to about $0.26$–$0.30$). The Random Forest is clearly more robust than the single tree, yet the drop in macro-F1 and the persistent FNR for these minority classes indicate that Modbus counters alone provide only a partial view of the threat landscape.

Now, coming back to the Phase 1 binary models. For the binary task (normal versus attack), both models perform extremely well on the same test set. The binary Decision Tree achieves accuracy $\approx 0.982$ and macro-F1 $\approx 0.982$, with $\mathrm{FNR}_{\mathrm{normal}} \approx 0.008$ and $\mathrm{FNR}_{\mathrm{attack}} \approx 0.027$, meaning that it almost never misclassifies normal traffic and still detects more than $97\%$ of attacks, with a symmetric false positive rate of about 2–$3\%$. The Random Forest remains the best binary model, with accuracy $\approx 0.985$ and macro-F1 $\approx 0.985$, $\mathrm{FNR}_{\mathrm{normal}} \approx 0.008$ and $\mathrm{FNR}_{\mathrm{attack}} \approx 0.022$: in practical terms, it detects around $98\%$ of attack records while raising very few false alarms. These Phase 1 results show that, on a relatively balanced Modbus subset, tree-based models already provide very strong binary detection and reasonably good multi-attack discrimination, with the main residual weakness concentrated on *scanning*-type attacks. Support Vector Machines are consistently inferior under the same conditions and are therefore discarded from further analysis.

Finally, for the Phase 2 binary task (normal vs. attack), tree-based models remain strong but are no longer "near-perfect" under imbalance. The DT falls to accuracy $\approx 0.951$ and macro-F1 $\approx 0.931$ (with $\mathrm{FNR}_{\mathrm{normal}} \approx 0.037$, $\mathrm{FNR}_{\mathrm{attack}} \approx 0.092$), and the RF to accuracy $\approx 0.968$ and macro-F1 $\approx 0.953$, with $\mathrm{FNR}_{\mathrm{normal}} \approx 0.005$ but $\mathrm{FNR}_{\mathrm{attack}} \approx 0.123$. Thus, the forest remains the best binary model and protects normal traffic extremely well, but at the cost of letting a non-trivial fraction of attacks go undetected.

### 5.5 Evaluation protocol and metrics

The evaluation protocol is designed to connect ML metrics with what I have being highlighting about the risk-management and governance concepts discussed in Sections 2 and 3.

For each model, task, and phase, I compute:

- **Accuracy**, for a complete landscape.
- **Macro-averaged F1**, which gives equal weight to all classes and is therefore more sensitive to performance on minority attack types such as *scanning* or *XSS*.
- **Weighted-averaged F1**, which weights each class by its support. This metric can look very good even when rare attacks are poorly detected, especially on the full, imbalanced `IoT_Modbus` dataset.
- **False negative rate** (FNR) and **false positive rate** (FPR) for each class.

From a risk to governance perspective, FNR and FPR are the key quantities:

- A high FNR for a given attack type means that those attacks will often pass unnoticed by the IDS. This is particularly dangerous for reconnaissance attacks (such as *scanning*) or credential attacks (*password*), which can silently prepare more disruptive incidents. So, as risk analysts, we should give more attention of the possible risks of this undetected attacks.
- A high FPR on normal traffic means that operators and SOC analysts will be flooded with false alarms, leading to alert fatigue and potential neglect of real incidents.

In Phase 2, as I already explained, I deliberately reuse the Phase 1 hyperparameters, precisely to observe how a model tuned under "laboratory" conditions behaves when exposed to a more realistic environment.

## 6 LIMITATIONS AND FUTURE WORK

From a technical point of view, the most evident limitation is the focus on a single service and a single type of telemetry. All experiments are built around the `IoT_Modbus` sensor, using only four Modbus counters as input features. Although I am reasonably satisfied with this work—especially considering that it is my first project of this kind and my first in-depth exploration of the ToN_IoT dataset—I would like to continue studying it in the future, both to develop new research directions and to better understand

its implications for organisational practice and real-world cyber-risk. The case study was a little bit minimalist, but useful as a first step, it also constrains what the IDS can "see": attacks that are only weakly reflected in these counters (such as *scanning* or some *password*-related behaviour) remain hard to detect even for well-tuned tree-based models. In addition, the study is restricted to one dataset family (ToN_IoT) and one laboratory environment. Phase 2 moves closer to a realistic setting by using the full, imbalanced `IoT_Modbus` dataset, but it still reflects a specific testbed rather than diverse industrial deployments.

I think that a natural extension would be to expand the practical analysis beyond Modbus. When designing this project, I originally considered the Modbus sensor as a starting point, with the intention of later exploring other ToN_IoT services such as `Motion_Light`, `Thermostat`, `Weather`, or `Network`. Time constraints prevented this broader exploration, but revisiting ToN_IoT with a multi-sensor perspective would allow the design of IDS models that exploit correlations between different data sources and get closer to the "layered" view of risk implied by industrial architectures. This would also make it possible to test whether the weaknesses observed on specific attack types (e.g. *scanning*) can be mitigated by integrating network-level or process-level information instead of relying solely on Modbus telemetry.

Another important limitation is that all models are static, batch-trained classifiers. As discussed in the course and in the lecture by Marasco, this is precisely one of the main problems of many ML-based IDS currently proposed in the literature: they are trained once on a historical dataset and then deployed in an environment where threats, behaviours, and even devices evolve over time.

In Section 3.6.1 I briefly mentioned digital twins as a promising tool to simulate realistic industrial scenarios and generate richer, labelled data. I would love to be able to combine digital-twin environments with ML-based IDS, in order to (i) train and validate models under a wider variety of operating conditions and attack scenarios, and (ii) study more advanced paradigms such as Continual Learning, where the IDS is updated over time instead of remaining frozen in a batch-trained state.

Finally, there are clear limitations on the governance side. Throughout the paper, I have tried to situate the Modbus IDS within frameworks such as NIST CSF 2.0 and to reason in terms of threats, vulnerabilities, and damages. However, the analysis remains necessarily generic: it is not anchored in a specific industrial organisation, it does not attach concrete economic or safety costs to false negatives and false positives, and it does not explicitly simulate how ML-IDS alerts would be integrated into SOC workflows, incident-response procedures, or compliance reporting.

From an organisational perspective, a Modbus-focused ML-IDS like the one explored here only makes sense under a set of explicit preconditions that future research should model more concretely. At a minimum, the organisation must have (or plan to build) a Security Operations Centre or equivalent function capable of interpreting ML-generated alerts and embedding them into well-defined playbooks, rather than treating model outputs as opaque warnings. It must also articulate a clear risk appetite in terms of false positives and false negatives—for example, accepting a very low FPR to protect SOC workload while explicitly acknowledging that certain attack types (such as scanning or password abuse) will remain under-detected and must therefore be mitigated through other controls. In addition, the IDS needs to be embedded within a broader control stack—network segmentation, access control, backup and recovery, and compliance processes—so that its detections can trigger concrete technical, but mainly, organizational actions.

Future work should therefore not only refine models and features, but also formalize these organizational preconditions and develop richer governance scenarios, treating ML-IDS deployment as a sociotechnical design problem rather than a purely algorithmic choice. Only in that broader context can the kind of Modbus-focused IDS explored here be properly evaluated as one component—necessary but not sufficient—of a mature IIoT/OT cybersecurity strategy.

## 7 CONCLUSION

In this paper, I have examined a very specific slice of the IIoT/OT security problem: what a static ML IDS can realistically contribute. Across the two experimental phases, the results show that Decision Trees and Random Forests can deliver excellent binary discrimination between normal and attack traffic and reasonably good multi-attack classification for the most frequent categories. At the same time, the persistent blind spots on minority but operationally relevant threats—especially *scanning*, *password* and *XSS* attacks—make clear that Modbus-only telemetry provides, at best, a partial view of the threat landscape.

A Modbus-focused ML detector can be a valuable component of the NIST CSF 2.0 *Detect* function, yet it neither removes underlying vulnerabilities nor eliminates threat actors, and its apparent strength (high accuracy, high weighted-F1) can mask non-negligible residual risks for specific attack types. In other words, ML-IDS should be understood as a complementary capability: they act as additional "sensors" for decision-makers rather than a self-sufficient solution that can be deployed in isolation, because on their own they do not solve any problem—they only provide analytical signals that must be interpreted and turned into action.

Interpreted in this way, the metrics analyzed in this work become inputs for governance rather than end goals. A low false positive rate, for instance, can meaningfully support SOC teams by reducing alert fatigue and helping them concentrate scarce human

attention on the most credible signals, which is crucial under real resource constraints. Conversely, the high false negative rates observed for minority but operationally relevant classes (such as *scanning*, *password* or *XSS* attacks) act as a warning light in a risk-management framework: they indicate that certain threat paths will systematically remain under-detected by a Modbus-only IDS and therefore require other technical and organizational measures—additional telemetry, different controls, specific playbooks—to be addressed and to work as countermeasures. Without that surrounding layer of action, contextualization and resource management, even the best ML-based detector remains just an elegant model watching a narrow slice of an industrial reality it cannot, by itself, secure.

The analysis therefore reinforces the central claim of the paper: ML-IDS should not be deployed as technocratic fixes, but as carefully scoped building blocks within broader risk-management strategies that combine richer telemetry, continuous monitoring, and explicit governance decisions about acceptable residual risk. Only in that wider organizational context can the kind of IDS explored here contribute meaningfully to mitigating cyber threats in IIoT/OT environments.

## REFERENCES


Bensaoud, A., and J. Kalita, 2025, Optimized detection of cyber-attacks on iot networks via hybrid deep learning models: arXiv preprint arXiv:2502.11470.

Cloudflare, 2025, 18 november 2025 outage: https://blog.cloudflare.com/18-november-2025-outage/. (Accessed: 2025-12-01).

CSE ICON, 2023, Scada vs. industrial iot: Which one should you choose?: https://www.cse-icon.com/scada-vs-industrial-iot/. (Accessed: 2025-12-06).

Dehlaghi Ghadim, A., 2025, Machine learning-based network intrusion detection for industrial control: Doctoral dissertation, Mälardalen University, Västerås, Sweden.

Eid, A. M., A. Bou Nassif, B. Soudan, and M. Injadat, 2023, Iiot network intrusion detection using machine learning: 2023 6th International Conference on Intelligent Robotics and Control Engineering (IRCE), IEEE, 196–201.

Kikissagbe, B. R., and M. Adda, 2024, Machine learning-based intrusion detection methods in iot systems: A comprehensive review: Electronics, **13**, 3601.

Liebl, S., L. Lathrop, U. Raithel, M. Söllner, and A. Aßmuth, 2024, Threat analysis of industrial internet of things devices: arXiv preprint arXiv:2405.16314.

Miani, R. S., G. D. G. Bernardo, G. W. Cassales, H. Senger, and E. R. d. Faria, 2025, A survey of data stream-based intrusion detection systems: IEEE Access, **13**, 72953–72990.

Moustafa, N., 2021, A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets: Sustainable Cities and Society, **72**, 102994.

Nankya, M., R. Chataut, and R. Akl, 2023, Securing industrial control systems: Components, cyber threats, and machine learning-driven defense strategies: Sensors, **23**.

NIST, 2024, The nist cybersecurity framework (csf) 2.0: NIST Cybersecurity White Paper NIST CSWP 29, National Institute of Standards and Technology, Gaithersburg, MD.

PaloaltoNetworks, 2025, What is industrial internet of things (iiot) security?: https://www.paloaltonetworks.com/cyberpedia/what-is-iiot-security. (Accessed: 2025-12-05).

Parshv, J., 2023, Iiot or iot?: https://www.telecomhall.net/t/iiot-or-iot/22684. (TelecomHall Forum post, accessed 2025-12-05).

Schneier, B., 2018, Click here to kill everybody: Security and survival in a hyper-connected world: W. W. Norton & Company.

Stouffer, K., M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson, 2023, Guide to operational technology (ot) security: NIST Special Publication 800-82r3, National Institute of Standards and Technology.

Timken, M., O. Gungor, T. Rosing, and B. Aksanli, 2023, Analysis of machine learning algorithms for cyber attack detection in scada power systems: Presented at the 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), IEEE.

Umer, M. A., K. N. Junejo, M. T. Jilani, and A. P. Mathur, 2022, Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations: International Journal of Critical Infrastructure Protection, **38**, 100516.

Venanzi, R., G. Di Modica, L. Foschini, and P. Bellavista, 2025, Towards it/ot integration in industry digitalization: A comprehensive survey: Journal of Network and Computer Applications, **245**, 104373.

Veridify Security Inc., 2023, Modbus security issues and how to mitigate cyber risks: www.veridify.com/article/modbus-security-issues-and-how-to-mitigate-c. (Technical brief, accessed 2025-12-06).

———, 2025, Ot security: Cybersecurity for modbus: https://www.veridify.com/ot-security-cybersecurity-for-modbus/. (Technical brief, accessed 2025-12-06).

Wallarm, 2025, What are iot attacks? vectors, examples, and prevention: https://www.wallarm.com/what/iot-attack. (Accessed 2025-12-06).

Zuboff, S., 2019, The age of surveillance capitalism: The fight for a human future at the new frontier of power: PublicAffairs.