

Cyber Threats in IIoT

Can ML Mitigate Vulnerabilities and Prevent Harm?

Gerardo Antonio Corral Ruiz

Laurea Magistrale in Governance e Politiche dell'Innovazione Digitale
Course: Governance della Cybersecurity

Let's explore the next question...



Under which technical and organisational conditions can an ML-based IDS on Modbus telemetry (the subset that I chose) meaningfully contribute to risk reduction?

Why IIoT Security Matters to Me (Background and Research Motivations)



- In *The Age of Surveillance Capitalism*, Zuboff argues that many “smart” devices are primarily instruments of data extraction and profit, so security, privacy, and the systematic reduction of harmful vulnerabilities are not treated as primary design goals.
- In *Click Here to Kill Everybody*, Schneier warns that once “everything is a computer”, cyber attacks no longer concern only data theft, but can directly affect life and property—from cars and medical devices to power grids and industrial plants.
- Harari’s description of increasingly complex (presented in his book *Nexus*), tightly coupled information networks, combined with events such as the **Cloudflare** outage, illustrates how a single technical failure can trigger cascading effects across many organizations.
- In IIoT/OT, this fragility is amplified: we have digital systems that directly **control physical processes**, and **insecure legacy protocols such as Modbus** can lead to **production downtime, equipment damage, and safety incidents**.
- For these reasons, IIoT security has become a central concern in my work, I wanted to understand how **ML-based IDS** can realistically support attack detection under real technical and governance constraints.

What is OT and IIoT?



- **Industrial Internet of Things (IIoT)** is a system of connected devices designed **to monitor, control, and optimize industrial operations**.
- **Operational Technology (OT)** refers to a broad range of programmable **systems and devices that interact with the physical environment or manage devices that interact with the physical environment inside industry**.

The IIoT environment is considered to be **part** of the OT environment. Historically, this phenomena happened due that IT and OT systems **increasingly converge and become interconnected**. Security failures in IIoT/OT are delicate because they are directly coupled to **real-time** physical processes. The primary security priority is **operational continuity and the safety of people and assets**.



The key differences between general IoT and IIoT are:

- **1. Purpose and Scale:** General IoT focuses on **home automation and smaller-scale applications**, while IIoT is directed toward **large-scale industrial applications and demands unique operational characteristics**.
- **2. Real-Time/Availability:** IIoT systems operate in **real time** and have far **less tolerance for downtime (the 9s example)**, requiring adherence to strict availability standards. **General IoT not necessarily.**
- **3. Connectivity:** General IoT devices typically connect via **Wi-Fi or cellular networks**, whereas IIoT devices usually use **wired networks such as Ethernet or industrial protocols like Modbus or Profibus to connect**.
- **4. Consequences and Priority:** In industrial environments (IIoT/OT), the security priority is above all **operational continuity and the safety of people and assets**. Unlike **general IoT failure which typically involves data loss or identity theft** but is generally not catastrophic physical harm.

Legacy protocols?



The purpose of legacy protocols was to provide a simple, open, and reliable communication standard for devices such as sensors and automation systems, allowing them to communicate and share data within industrial environments. Legacy protocols (such as Modbus, DNP3, or Profinet) are older protocols originally designed for isolated networks. They typically lack modern security features common in IT systems, such as encryption, error logging, and password protection

Why will I speak about Modbus?

The Modbus protocol is considered **outdated** because it **was not built with modern security in mind**.

1. **Lack of Security**: Modbus typically lacks robust authentication and transmits data without encryption, often in plain text.
2. **Vulnerability**: This exposes it to threats like **network scanning and injection attacks**. The absence of built-in security allows for **continuous data modifications to registers**, which is a clear threat.
3. **Deployment Difficulty**: Implementing **external security measures is rarely practical** because of **limitations of the devices that uses Modbus**.



Cyber risk must be analyzed mainly from a **governance perspective**, because decisions about security are **taken by people, not by technology**. My ML-based IDS can meaningfully **reduce risk if used as a tool for analysis and flag generator**, but this depends on **structure, strategy, and priorities planning**.

Frameworks and principles: the **NIST Cybersecurity Framework (CSF) 2.0**, the **PDCA cycle**, and the **principles of Zero Trust**

Two steps back: Cyber Risk & Governance



NIST Cybersecurity Framework (CSF) 2.0: It organizes cybersecurity into six core functions: **Govern, Identify, Protect, Detect, Respond, Recover.**

My case study lives mainly in the **Detect function**. It reinforces the idea that an **ML-based IDS is one control inside a wider governance strategy**, not an isolated technical gadget.

PDCA cycle (Plan–Do–Check–Act): Cybersecurity is a process, so PDCA works as a **continuous-improvement loop.**

Zero Trust principles: “Never trust, always verify”: **no device, user, or network segment is implicitly trusted.** This is crucial in **IIoT/OT**, where **continuous verification is needed and nothing can be taken for granted**; in my experiment it translates into closely monitoring **FNR** and **FPR** as alarm signals, instead of assuming that our models or our strategies are **infallible**.



Threats

The threats aim for **loss of availability** (**DoS/DDoS**), **data manipulation** (**spoofing/injection**), or **system compromise** (**ransomware, espionage, insider abuse, scanning, backdoor, password cracking**). Even **people safety**, as shown in Schneier's book.

Vulnerabilities

Caused by **unsafe legacy protocols** (e.g., Modbus, it does not have **encryption** nor **robust authentication**), **weak IIoT device security**, **misconfigured remote access**, **weak segmentation**, and **insufficient logging**.

Damages

Could be really severe due to integration with **physical/real-time processes**: **risks to human safety, operational continuity loss, production downtime, equipment damage, and financial loss**.

«We cannot eliminate threats from our systems, but we can systematically reduce vulnerabilities and design robust plans to limit potential damage.»

Under which technical and organisational conditions can an ML-based IDS on Modbus telemetry (the subset that I chose) meaningfully contribute to risk reduction?

Modbus telemetry from ToN_IoT

Focus on detection (**binary & multi-class**),

Two phases: balanced lab setting [*Train_Test_IoT_Modbus*] vs. realistic stress test [*IoT_Modbus*]

Limitations: **Static batch, limited feature set, limited time, testbed scenario.**

Dataset & Experimental Design



Datasets

- 8 fields: date, time, four Modbus counters (FC1–FC4), binary label, attack type
- Train/Test subset: 31,106 records (15K normal, 5K injection/backdoor/password, 529 scanning, 577 XSS)
- Full dataset: ~287K records; heavily imbalanced.

Phase 1

Tune models, choose the best

Phase 2

Stress-test realistic

- Same features (FC1–FC4) and preprocessing across phases
- Phase 1: 60/20/20 split; Phase 2: 80/20 split
- Binary: normal vs. attack; Multi: 6 classes (normal, injection, password, backdoor, xss, scanning).



Decision Tree

Interpretable & simple. It performed really decent.



Random Forest

Top performance was achieved thanks to the model's ability to capture **feature interactions**. The data showed a **strong overlap between normal and attack traffic** in feature space.



SVM (RBF)

Explored but **discarded** – poor performance in general due to its **difficulty handling the highly overlapping Modbus feature space and strong class imbalance**.

Training Pipeline

- **Stratified splits to save class distribution:** 60/20/20 (Phase 1) & 80/20 (Phase 2)
- **StandardScaler** on training set only
- **5-fold stratified cross-validation** for hyperparameters validation test
- **Macro-F1** as primary metric because: “[it] gives **equal weight to all classes** and is therefore **more sensitive to performance on minority attack**”.
- **FNR/FPR** were analyzed to find red flags. Useful for governance strategies.

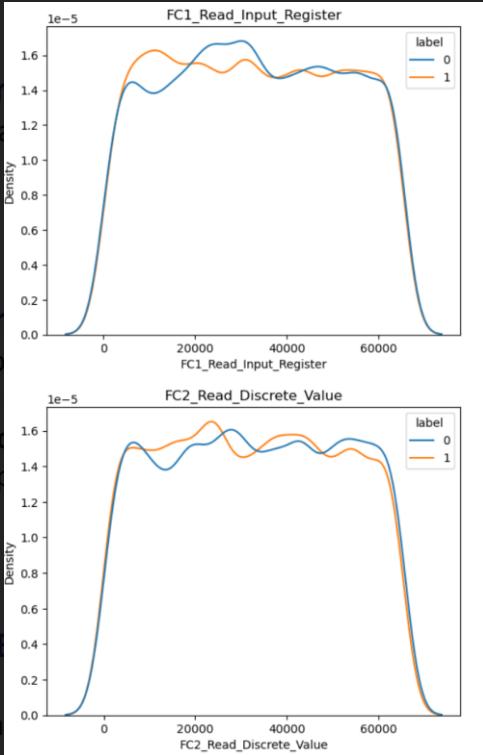
Features' overlap Pipeline



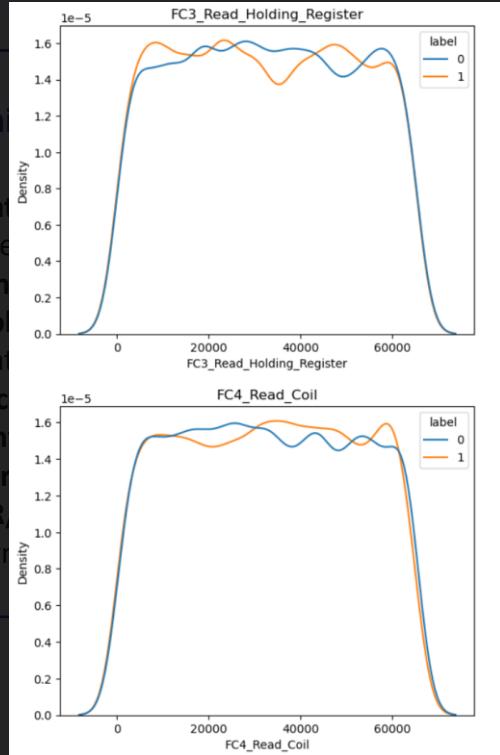
Decision
Interpretation

Random
Forest
Top performance
ability to handle
a strong correlation
feature selection

SVM (RF)
Explored
performance



Training
• Stratified
(Phase 1)
• Standard
• 5-fold
validation
• Macromodel
weights
• FNR
• government



50/20/20
parameters
gives equal
sensitive to
useful for

Phase 1: Balanced Setting



Binary detection:

Metric	Decision Tree (DT)	Random Forest (RF)
Macro-F1	≈0.982	≈0.985
FNR normal/FPR attack	=0.0083	=0.0083
FNR attack/FPR normal	=0.0267	=0.022

Multi-class detection:

Metric	Decision Tree (DT)	Random Forest (RF)
Macro-F1	≈0.918	≈0.931
Majority Classes (F1)	≈0.95–0.98 (Normal, Backdoor, Injection, Password)	≈0.96–0.98 (Normal, Backdoor, Injection, Password)
XSS (F1)	≈0.918	≈0.955
Scanning (F1)	≈0.73	≈0.77
Scanning (FNR)	≈0.37	≈0.37



Phase 2: Realistic Setting



Task	Model	Accuracy	Macro-F1	Weighted-F1	Key Weakness/FNR
Binary Detection (Normal vs. Attack)	Random Forest (RF)	≈0.968	≈0.953	N/A	FNR (attack)≈0.123 (12.3% of attacks undetected)
	Decision Tree (DT)	≈0.951	≈0.931	N/A	FNR (attack)≈ 0.092
Multi-class Detection (6 classes)	Random Forest (RF)	≈0.970	≈0.897	≈0.969	Password, Scanning, and XSS show significantly higher false negative rates (FNR up to about 0.26–0.30) ≈1/3 undetected
	Decision Tree (DT)	≈0.947	≈0.840	≈0.947	Clear weaknesses on minority attacks: password: 0.2581 scanning: 0.3208 xss: 0.2400



Low FPR reduces alert fatigue, helping SOC teams focus on credible signals.



High FNR on rare classes exposes residual risk; complementary controls (segmentation, process monitoring) are needed. It depends on the priorities of each organization, but ML-based IDS give us tools to PDCA.



Note: Metrics are inputs for governance decisions (risk appetite, SOC capacity) – not end goals.



Technical limitations

- Single sensor/service (Modbus only)
- Small feature space (four counters)
- Single dataset family and testbed environment
- Static, batch-trained models (no continual learning)

Organisational preconditions

- Requires SOC or equivalent function to interpret alerts
- Must define risk appetite for FNR/FPR trade-offs
- IDS needs integration with network segmentation, backup & recovery measures that every organization need to set.
- Without governance maturity, models are generally useless



Key Learnings

- ML-IDS is a useful detection layer within the NIST CSF Detect function not the whole scheme.
- It neither removes vulnerabilities nor eliminates attackers; it serves as an additional tool for decision-makers.

Thanks!

Here my repository:

https://github.com/GerardoACR/iiot_ids_project/tree/main

Gerardo Antonio Corral Ruiz

Laurea Magistrale in Governance e Politiche dell'Innovazione Digitale
Course: Governance della Cybersecurity