National College of Ireland

Can an AI-Based Behavioural Scanner Improve the Detection of Protocol-Level Misconfigurations and Anomalies in Industrial IoT Environments

MSc Research Project
MSc in Cybersecurity

Conor Lacey
Student ID: X23287047

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Conor Lacey
Student ID:	X23287047
Programme:	MSc Cybersecurity **Year:** 2
Module:	MSc (Research) Practicum Part 2
Supervisor:	Ross Spelman
Submission Due Date:	September 15th, 2025
Project Title:	Can an AI-Based Behavioural Scanner Improve the Detection of Protocol-Level Misconfigurations and Anomalies in Industrial IoT Environments
Word Count:	7178 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Conor Lacey
Date:	September 15th, 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

Assignment title - Research Practicum

Your Name/Student Number	Course	Date
X23287047	Conor Lacey	15/09/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
Grammarly	Tool for assisting with spell-checking, grammar and plagiarism	https://support.grammarly.com/hc/en-us

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Grammarly	
When you write a paragraph, Grammarly can suggest better wording, correct spelling and check if it is plagiarised. It also helps with punctuation in general grammar. It also assists in general sentence restructuring and suggests better ways to convey the paragraph. Through the software, you can set goals such as “Academic and Formal” to restrict suggesting and structuring options.	
For example - This is spealt wrong	Would be - This is spelt wrong.

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Can an AI-Based Behavioural Scanner Improve the Detection of Protocol-Level Misconfigurations and Anomalies in Industrial IoT Environments

Conor Lacey
X23287047

Abstract

Industrial Internet of Things systems are being increasingly deployed in smart manufacturing, enabling automation and efficiency under Industry 4.0. However, many protocols such as Modbus were never designed with security in mind, leaving them vulnerable to misconfigurations and misuse. Current vulnerability scanners, such as OpenVAS rely on Common Vulnerabilities and Exposures and therefore struggle to detect behavioural anomalies and protocol-level threats.

This research presents the development of a prototype AI-based behavioural scanner for Modbus traffic, designed to detect protocol misuse and anomalies that current CVE-driven scanners overlook. Using Conpot, a honeypot, to emulate a Modbus-enabled SCADA system, a custom Python logger was built to generate both benign and malicious traffic, producing a labelled dataset of 15,500 records. Two machine learning models were then evaluated, using Random Forest for supervised and Isolation Forest for unsupervised.

The results show that the Random Forest achieved a high accuracy (97% overall and a recall of 0.87 for malicious traffic) while the Isolation Forest flagged the anomalies without labels, although with a reduced recall. In contrast, OpenVAS scans detected only generic vulnerabilities while failing to detect any Modbus misuse.

The key contribution of this work is a reproducible dataset pipeline and demonstration that feature-level behavioural analysis can complement signature-based scanners. The findings highlight both the potential and limitations of AI-driven scanners in anomaly detection in IIoT and suggest a pathway for future research using real-world datasets, protocol expansion and scalable deployment.

1 Introduction

The integration of Internet of Things technologies into manufacturing has transformed how industrial systems are managed, maintained, and work together as part of Industry 4.0. Modern environments such as these rely heavily on connected devices and protocols such as Modbus to automate everything from machine control to environmental monitoring offering significant advantages while also expanding the attack surface of industrial networks in ways that some cybersecurity tools struggle to protect.

Most industrial devices were never designed with security in mind or built in and Modbus is still one of the most widely used and least secure industrial electronic device communication protocols. It was initially designed in 1979 by a company called Schneider Electric, which was

Modicon at the time. Since then, it has become a standard in the field and has been adapted for TCP/IP networks without combining basic security mechanisms such as authentication, encryption, or integrity checks. This lack of built-in protection leaves it exposed and open to various kinds of cyber-attacks, which will be discussed throughout the paper.

Vulnerability scanners, such as OpenVAS, play an important role in identifying known issues in systems. However, they rely heavily on CVE databases and can only detect what is already known. This could potentially be a significant blind spot in IIoT networks, where misconfigurations and protocol misuse are just as dangerous. Tools such as OpenVAS are not designed to understand the behaviour of industrial protocols like Modbus, nor are they capable of detecting the misuse of valid commands in unintended contexts. Instead, preliminary testing with OpenVAS during this research showed that it did not detect Modbus anomalies. This is particularly worrying as attackers do not always need to exploit a known vulnerability; often, a misused function code or an unauthorised code write is enough to simply disrupt a system.

This gap is what this project aims to highlight and potentially fill. The report explores whether machine learning techniques can be used to detect abnormal Modbus activity, by learning what "normal" behaviour looks like and identifying deviations from that baseline. Throughout the research, a prototype behavioural scanner was developed that uses supervised and unsupervised learning models to flag potentially malicious Modbus traffic. The scanner was trained on a custom simulated dataset generated in a controlled environment using Conpot, which can function as an open-source SCADA honeypot, along with a custom traffic generation script, developed using Python, that simulates both benign and malicious activity.

The motivation behind this work comes from personal interests in manufacturing environments and cybersecurity in general, as well as hobbyist interests in automation and improving business processes. Having worked in automotive manufacturing companies for my whole career so far, I have seen firsthand how operational technology networks are often treated as "set and forget" environments, where security is assumed rather than verified and may be overlooked or misunderstood. This attitude could potentially leave systems open, particularly as more devices are added to networks that were not built to accommodate them securely in the first place.

Therefore, my research question for this project is – "Can an AI-Based Behavioural Scanner Improve the Detection of Protocol-Level Misconfigurations and Anomalies in Industrial IoT environments?".

Rather than attempting to build a full CVE-detection engine or vulnerability scanner this scanner focusses specifically on behavioural anomalies – for example, unauthorised write commands, suspicious access patterns, or malformed function codes, which are randomised by the script to add a degree of realism and boost the models learning. These elements did not show up in the OpenVAS report but can pose serious security risks in industrial settings anyway.

To assess the scanner's performance, I generated 15,500 labelled traffic records and trained both a Random Forest and an Isolation Forest model. These were assessed using standard metrics found online, such as accuracy, F1-Score, ROC-AUC, and false positive rates. The models were also tested against OpenVAS outputs from a standard scan of ports open on the virtual machine to highlight what behavioural threats the models could potentially catch that OpenVAS missed, thus acting as a complementary feature more than a whole new separate tool.

In doing so, this project demonstrates how machine learning can play a complementary role in security. By combining behavioural insights generated using a tool like the proposed one with automated detection, organisations could potentially catch threats earlier and reduce reliance on static databases of known vulnerabilities if used correctly.

Following the guidance, Section 2 reviews current literature in the field of Industrial Internet of Things security, Modbus vulnerabilities specifically and AI applications in modern network threat detection. Section 3 then outlines the research methodology and describes how the data was generated and labelled, as touched on in the introduction and other sections. Section 4 then investigates the technical design and architecture of the scanner. Section 5 discusses the implementation and Section 6 presents the evaluation metrics and results of the testing. Finally, Section 7 discusses the implications of the findings, limitations of the current approach, opportunities for future research and conclusions.

2 Related Work

The following literature review has been divided into six sub sections, each focusing on different aspects of the research area. The first section, on Modbus Protocol Security, explores vulnerabilities surrounding the Modbus protocol itself, delving into other researcher's proposed strategies, methods and enhancements which could help mitigate or enhance security in the protocol. The second section builds on this by reviewing the applications of AI and ML in Industrial Network Security to overcome or enhance current tools of the trade. The third section analyses Behavioural Anomaly Detection specifically, looking at detection methods that focus on deviations from normal protocol behaviour. The fourth section looks at the Limitations of CVE-driven Scanners, to evaluate why current tools fail to detect protocol misuse in IIoT systems. The fifth section reviews Relevant Machine Learning Approaches, looking at algorithms and model architectures, as well as assessing both supervised and unsupervised learning and their applicability in the field. Finally, the six section, Data Challenges in IIoT Research addresses the lack of high-quality datasets, the role of data simulation and traffic generation to enable better model training. Together, all the papers supply a foundation for the identified research gap in the design of AI-Driven behavioural scanners.

2.1 Modbus Protocol Security

Throughout the research, several studies examined the weaknesses described in the Modbus protocol. Nardone et al. published a paper where they performed a security assessment if the

Modbus protocol, using Dynamic State Machines and model checking to prove if Man In The Middle attacks could manipulate process values while not triggering any alarms (Nardone et al., 2016). The paper referenced and proved a scenario where the communication between a Modbus master, Alice and slave, Bob, is threatened by Mallory, an attacker performing a MITM attack. The same paper also highlighted not only how easy it was to exploit, but also how little had been done in the design to prevent such attacks.

Following this, Katulić et al. introduced a protection method for Modbus/TCP using message authentication codes embedded directly in Programmable Logic Controller logic (Katulić et al., 2023). Their method was effective in identifying any modifications, replay and insertion attacks targeting PDUs specifically while maintaining system performance. It also acknowledges that MITM attacks ae an issue with Modbus and highlights recent attacks in the area, such as the Russo-Ukrainian conflict, where power grids have been affected.

On the simulation side, Al Dalky et al. designed a traffic generator like the proposed one in this paper to emulate malicious Modbus activity for SCADA systems so they could test intrusion detection and anomaly detection methods (Al Dalky et al., 2014). The generator they designed used Python also but differed in the modules used to build the script; in their case, they used Scapy and Snort rules, which could create realistic attacks; however, it looked like it relied on a Snort rules file to act as input. In a more up to date paper, Srivastava et al. integrated Modbus communication into a real time digital simulator to model cyber physical systems under attack (Srivastava et al., 2024). The specific attack explored in the paper was again a MITM. They showed that Modbus traffic could be manipulated to turn off distributed energy resources, like the goals of this paper. Their setup enabled them to test both offensive and defensive techniques, including fuzzing and protocol manipulation, which are also relevant to the paper's proposed methods.

Finally, Ferst et al. investigated how integrating Transport Layer Security into Modbus could achieve secure industrial control system communication. Their research showed that cryptographic protection was feasible but had its own trade-offs in the form of latency, which was still within acceptable limits and compatibility with older resource constrained systems (Ferst et al., 2018).

2.2 AI or ML in Industrial Network Security

As with most fields with massive amounts of data, integrating AI and ML into industrial cybersecurity is a promising solution to address the shortcomings of CVE-driven detection tools while leveraging the available data. Unlike CVE-driven scanners, which rely on known vulnerabilities and large databases, AI models can identity deviations in protocol behaviour that often occur before an attack, as shown through this research's artefact. This type of integration becomes important in the light of the previously discussed flaws in the Modbus protocol.

Suchorab et al. presented an anomaly detection system using Zeek and a Discrete Time Markov Chain model. Their approach combined ideas not included in the proposed tool, such as timing analysis, process parameter inspection, and online learning to detect attacks like MITM without relying on predefined signatures. Their focus was on keeping their system open source so that it could be easily adopted in the field (Suchorab et al., 2025). Their research also supported the idea that behavioural analysis is key to finding threat sand can detect protocol level anomalies, with the potential to complement a scanner such as the one proposed in this paper.

Similarly, Kotsiopoulos et al. tackled the challenge by combining both AI-driven detection and software-defined networking. Their system used a ResNet50 neural network trained with Active and Transfer Learning, allowing it to adapt to new threats over time. They also used STRIDE, Attack Defence Trees, and Common Vulnerability Scoring System scoring to map out the vulnerabilities in Modbus traffic. Then by using software defined networking to respond dynamically to attacks and Thompson Sampling to keep mitigation both effective and low cost. Their research could again be used to complement the proposed scanner, and the idea again supports the proposed research that AI-driven behavioural scanning can be effective when used for anomaly detection (Kotsiopoulos et al., 2025).

Another challenge was encountered while broadly reviewing trends in operational technologies when Gupta et al. found that while AI adoption is growing, the scarcity of realistic datasets remains an issue. Throughout their paper, they argued that without proper traffic simulation platforms, it is challenging to train models to handle the high level of complexity in industrial environments (Gupta et al., 2024). Something echoed in this paper, where simulated traffic generation was used to build a dataset due to the lack of easily accessible data online through platforms like Kaggle.

While investigating software capabilities to run the models, Holasova et al. proposed a modular approach using AI driven security modules that can be deployed without disrupting existing infrastructure. The modules used neural networks and logged Modbus traffic to detect incidents. Their approach achieved high accuracy on Modbus protocols, while also avoiding the need for costly hardware, an ideal that mimics the approach taken throughout this project and it's complementary style add-on approach (Holasova et al., 2021).

Finally, Nankya et al. combined ML models like Random Forest with protocol-aware preprocessing to improve intrusion detection. They highlighted how critical the preprocessing stage is, turning raw network traffic into structured, meaningful features, which significantly impacts the model accuracy (Nankya et al., 2023). Their insights directly supported this paper's focus on feature extraction and labelling during dataset generation, showing that good data preparation is just as important as the choice of model, as demonstrated in the proposed research.

2.3 Behavioural Anomaly Detection

Following on from the more general research before, it was decided to look at the behavioural anomaly detection specifically. This type of detection has started to become more popular as it

focuses on deviations from the norm, rather than static vulnerability lists. This type of approach is valuable in the context of industrial control systems, where protocol misuse or abnormal device behaviour often precedes an incident.

Eyeleko and Feng offered an insight into a layer-based hacking scenario study. Their research showed how attacks often unfold gradually across different parts of the system. They highlighted how multi-layer behaviours reveal compromises through their cross-layer scenario and showed how tools such as the one proposed would be helpful but are required across the various layers to categorise malicious activity properly, something which could complement the proposed tool (Eyeleko and Feng, 2023).

Similarly, Jayalaxmi et al.'s scoping review of existing solutions suggests that many weaknesses only surface as behavioural deviations. They argued that anomaly detection should work alongside traditional defences to address these types of threats, further supporting the case for the proposed tool to complement existing tools such as OpenVAS (Jayalaxmi et al., 2021).

To complement Jayalaxmi et al.'s scoping review, Siwakoti et al. mapped vulnerabilities, attacks and other criminal services and noted that behavioural analysis is key to finding these hidden threats. (Siwakoti et al., 2023). The insights from both papers reinforce the approach taken throughout this paper and the proposed tool, which has the potential to catch threats that only become visible when interactions deviate from the norm.

Moving on to the paper by El Gharbaoui et al., the researchers highlight that supervised models suit finding known patterns while unsupervised methods help find unknown patterns (El Gharbaoui et al., 2024). This aided the proposed research as it led to the inclusion of both types, thus offering a way to detect unknown anomalies without needing labelled data. The researchers here reiterated the need for proper data to train the models on, which again aligned well with the simulated data approach taken in this paper.

Finally, in the last paper for this topic, Lazaridis et al. examined how attackers exploit Modbus TCP in industrial systems. Their work discussed the various threats, vulnerabilities and how they could be exploited while noting that spotting behavioural anomalies in real time was key to protecting modern environments. So, the insights gathered from the paper supported the traffic generation elements of this research, focusing on random coil writes and function code related traffic sent to the Conpot (Lazaridis et al., 2024).

2.4 Limitations of CVE-driven Vulnerability Scanners

The goal of the research paper was to improve CVE-driven scanners by enhancing them with tools capable of behavioural analysis as highlighted in the previous section, which led me to look more closely at the specific limitations of CVE-driven scanners and how they could be fixed.

Sachin et al. offered a domain-specific case study in digital wine manufacturing, using the CVSS framework to assess IIoT sensor vulnerabilities. While the CVSS framework helped quantify known risks, its analysis showed that many real-world issues, like insecure default settings, are not reflected in vulnerability databases. That means scanners often miss them. These findings again supported the need for better scanners, such as the approach taken in this project, where feature extraction and labelling are used to find overlooked but impactful weaknesses (Sachin et al., 2024).

Tsiknas et al. reinforced the limitations in their survey of IIoT cyber threats by showing that many exploits arise from architectural flaws and insecure integrations, not just patchable vulnerabilities. These weaknesses are hard for scanners to detect because they require an understanding of how systems interact, not just a list of known CVEs. This supported the need for more context-aware approaches to threat detection, like the one taken in this project (Tsiknas et al., 2021).

In their paper, Martins and Oliveira proposed authentication and authorisation functions for ModbusTCP. While this approach was promising, it required changes to the protocol itself, which is something CVE-driven scanners cannot enforce or test for, and left a gap in environments where legacy equipment cannot be modified, highlighting the need for detection methods that account for the real-world constraints of legacy equipment and environments (Martins and Oliveira, 2022).

In addition to that, Rodríguez et al. developed the MOSTO toolkit to audit ICS devices, with a focus on ModbusTCP traffic. Unlike generic scanners, MOSTO analysed message patterns and detected anomalies at the protocol level. However, one weakness of their approach was relying on predefined rule sets, which means it struggled to identify novel attack patterns. This highlighted the same adaptability gap seen in scanning tools and shows the need for more flexible detection methods, like what this research proposes (Rodríguez et al., 2023).

Finally, Aslam and Akinrolabu examined deeper architectural vulnerabilities in ICS environments, pointing out that scanning tools often miss weaknesses like what I found. They noted that such flaws cannot be captured by point-in-time scans alone, as they require a combination of architectural assessment and behavioural monitoring (Aslam and Akinrolabu,2022). This supported the approach taken in this project, which aims to find these hidden risks through more context-aware analysis.

2.5 Relevant ML Approaches

Looking at some similar ML approaches to the topic, Karacayılmaz and Artuner presented a hybrid expert system for detecting IIoT attacks, combining rule-based reasoning with both supervised and unsupervised ML, like the proposed approach. Their system targeted threats like MITM, using domain-specific features such as "dup and retransmission" rates to boost detection accuracy, highlighting the value of thoughtful feature engineering (Karacayılmaz and Artuner, 2024).

Zarzycki et al. took a different approach and explored generative approaches to IIoT security by applying GAN-based neural networks to simulate cyberattacks on industrial control networks. Their method addressed the challenge of data scarcity by generating synthetic attack patterns, enabling better training of anomaly detection models. This approach was interesting and could serve as a complement to tools like the proposed tool by introducing diverse, hard-to-detect traffic (Zarzycki et al., 2023) for better training.

Finally, Alfahaid et al. provided a comprehensive survey of ML-based security solutions across IIoT. Their analysis highlighted a tradeoff between model accuracy and computational cost, noting that in constrained deployments such as in IIoT, lightweight algorithms were more practical unless edge or fog computing resources were available. This insight informed the design choices in this project, where model selection must balance detection performance with real-time feasibility,staying lightweight (Alfahaid et al, 2025).

2.6 Data Challenges in IIoT Research

One of the main challenges with AI-driven security is the need for good-quality data. Having been highlighted in various subtopics, it was decided to look at papers specifically on the topic. Biswas et al. reflected on a significant challenge in smart grid and SCADA security, namely that real-world datasets are often heavily imbalanced, with attack traffic making up less than 10% of records. This forces researchers to either oversample synthetic attacks or risk building models that do not hold up in the real world. Their use of the WUSTL-IIOT-2018 dataset also showed that even well-known sources can lack the protocol diversity needed for generalisable models. This supports the approach taken in this project, where a fully simulated environment is used to generate a balanced dataset for model training (Biswas et al., 2024).

Finally, Mubarak et al. addressed the data scarcity challenge by developing a "cyber kit" testbed capable of generating realistic ICS traffic across multiple protocols, including Modbus, DNP3, and IEC 61850. Their approach embedded attack scenarios within the traffic, again reinforcing the synthetic traffic generation approach used in this research (Mubarak et al., 2021).

In general, the literature helps to identify some of the challenges in and ways we can secure IIoT environments, particularly surrounding the Modbus protocol. The fact that Modbus is basic and widely used also exposes it to more vulnerabilities, especially as its use increases as we transition to Industry 4.0 and 5.0 in the future. Across the various topics, there is a consensus that CVE-driven vulnerability scanners are not well-suited to detecting the malicious Modbus behaviour. AI and ML can help fill this gap, with evidence supporting the use of behaviour-based detection, feature engineering, and model architectures that balance performance with operational feasibility. Behavioural anomaly detection, in particular, is positioned as essential for finding new attacks, though its success depends heavily on high-quality datasets. Data scarcity remains one of the most significant constraints, driving the need for simulation environments, synthetic data generation, and labelled training records. In summary, there exists a research gap in the need for adaptable, protocol-aware AI models capable of detecting anomalies in IIoT traffic without relying exclusively on static vulnerability signatures, a gap

that directly informs the design and implementation of the AI-driven behavioural vulnerability scanner developed in this project.

3 Research Methodology

This research adopted the following methodology, intending to build and validate a working prototype of an AI driven vulnerability scanner for IIoT environments. This combined controlled simulation using software such as VirtualBox, Docker and Conpot, simulated data generation, and supervised/unsupervised ML. The setup ensured everything was kept isolated and repeatable for future research.

3.1 Research Approach

As discussed in the literature review, gaps in scanning tools were identified, and the strengths of behaviour-based detection were evident. To approach the task, the project was divided into three parts, namely environment setup, data collection and model development/evaluation.

3.2 Environment Setup

The virtualised environment was made using Ubuntu 24.04 running through VirtualBox on a bridged network adapter. Then Docker was used to create the Conpot, which was configured to emulate a Modbus/TCP endpoint. This setup ran on a host system with an AMD Ryzen 5800x3d CPU, 64 GB RAM, of which 8GB of RAM and 4 CPUs were assigned to the virtual machine and used Python 3.12.3. Various modules were installed using a separate Python virtual environment. The choice of Conpot was informed by prior work showing its suitability for safe ICS attack simulation.

3.3 Data Collection Procedure

Two traffic profiles were generated using the logger script modbus_ai_logger.py, which could operate in two modes. In benign mode, traffic simulated normal Modbus operations, issuing function codes such as reads and occasional coil writes at realistic intervals. In malicious mode, traffic was generated with deliberate anomalies, such as repeated coil writes and non-standard function codes. These reflect common Modbus misuse cases identified in the literature. The number of packets per run was set manually, and they were labelled based on the mode they came from. In the original tests, this generated 6,000 records with equal benign and malicious traffic. However, I noticed that it over perfected the results making class separation unrealistically obvious for the models. I replaced the script to add more realistic scenarios, which generated overlapping malicious patterns and a greater imbalance between the records, all based on probability. I also expanded the dataset to include 15,500 records, of which 3,000 were malicious and the rest benign to more closely emulate a real environment. The modbus_ai_logs.csv file, included fields such as function code, address, value, etc.

3.4 Data Preparation

The records were loaded in Python's pandas library in ai_scanner_analysis.ipynb. This step involved verifying if columns were present and selecting various features for the model inputs. In this case, no data normalisation was performed because the data was already clean coming from the script. The dataset was roughly 80% benign and 20% malicious, introducing

imbalance and overlaps between the two, providing a more realistic challenge for the models. Stratified train test splits were then used to maintain the distribution.

3.5 Model Selection and Training

A random forest was used for the supervised model and was trained on labelled data to classify future traffic as benign or malicious. It was configured with stratified splits and class weights to account for imbalanced data. Isolation Forest for unsupervised learning was then trained only on benign data to detect previously unseen anomalies without pre-labelling. Contamination was changed from the true malicious ratio, around 20%, to "auto" which let the algorithm estimate anomalies independently. The dataset was split 70/30 for training and test data. I chose this approach based on literature indicating that combining supervised and unsupervised techniques increases accuracy. I also didn't need to perform any sort of manual encoding as most of the data was in numeric format already, except for the labels column which could be handled by the module scikit-learn automatically.

3.6 Evaluation Methodology

The Random Forest's performance was assessed in terms of precision, recall, F1, support, accuracy, as well as a confusion matrix through seaborn and extended evaluation metrics including ROC with AUC, precision-recall and feature importance analysis. For the Isolation Forest, anomalies flagged by the model were compared to the actual malicious distribution. With contamination set to "auto", the model flagged 11.6% of traffic as anomalous, which was lower than the true malicious part of the dataset, demonstrating the challenges faced with more realistic data in IIoT settings. The results highlighted how supervised and unsupervised methods can complement each other.

4 Design Specification

Three main requirements guided the design of the script. It needed to be able to understand Modbus/TCL behaviour at a feature level, such as the function codes used or the patterns, so that it could pick out the malicious attempts. It also had to combine supervised and unsupervised learning to work on both known and unknown traffic. Lastly, it needed to be modular and lightweight, since it is built for a research project where this was paramount, and so it could be updated without having to rebuild everything. In addition to this, the script needed to create realistic traffic, where the malicious attempts weren't easily separable from the benign. To address this, the script was enhanced to generate overlapping malicious traffic and the resulting dataset was imbalanced to ensure the models dealt with more realistic conditions.

4.1 System Architecture

As touched on in the methodology, the system follows a pipeline-based architecture starting with the traffic generation layer which was implemented through via modbus_ai_logger.py, capable of generating both benign and malicious Modbus traffic, depending on the command used and outputting structured CSV records with labelled data points. Once completed, data is fed into the notebook, ai_scanner_analysis.ipynb, as part of the data processing layer. The notebook was responsible for loading CSV records, verifying the columns and selecting the features for model training. Once complete, the models are trained on the data, utilising random

forest and isolation forest. Both models were implemented in scikit-learn. Finally, as part of the evaluation layer, performance metrics are generated to show how the models are performing with the aid of visual outputs created using matplotlib and seaborn.

4.2 Tools and Modules Used

Conpot	An industrial control system honeypot used for simulating Modbus/TCP endpoints.
Docker	Acted as deployable containers like virtual machines for running applications in an isolated environment. Used for both running Conpot and Greenbone Security Assistant separately.
VirtualBox	Used to run Ubuntu 24.04 OS to host scripts and docker images keeping it off the host.
Python 3.12.3	Used to create all the scripts and notebooks used in the project.
Jupyter Notebook	Used for creating interactive training and evaluation pipeline.
Scikit-learn	Used to create, train and evaluate the ML models used in the project.
Pandas/Numpy	Used in the notebook for data analysis, processing and feature selection.
Matplotlib/Seaborn	Used in the notebook for data visualisation and model evaluation, as well as ROC, PR and feature importances.

4.3 Design Rationale

The architecture was selected based on the literature review, which consistently found that supervised learning is great at classifying known threats while unsupervised learning is better suited for unknown threats. The choice of Random Forest was influenced by its popularity in the literature, and the Isolation Forest was chosen for its efficiency, versatility and scalability. The system's modular design means parts can be swapped out or extended without altering mass sections of the scripts. The rework on the script to make it more realistic was motivated by the literature again citing the lack of realistic data, so the models needed to be tested under more challenging conditions. I decided to use "auto" for the contamination rate to reflect the uncertainty of the rate of anomalous behaviour in real industrial settings.

5 Implementation

The first part was the traffic generator and logger script called modbus_ai_logger.py. It could operate in two modes: malicious or benign. The malicious mode was used to try to inject nonstandard Modbus behaviour such as misuse patterns; repetitive coil writes and nonstandard calls while the benign mode generated the standard calls. Each run of this script generated a manually set number of log entries in the CSV, called modbus_ai_logs.csv. So, initially this CSV dataset consisted of 6,000 records of Modbus traffic, with equal numbers of records for benign and malicious traffic. This resulted in perfect performance for the models and made it easy to distinguish between the modes. As a result, to add more realism, overlapping malicious traffic was added, such as using valid function codes at abnormal addresses or values, as well as class imbalance, producing a final dataset of 15,500 rows with 12,500 benign and 3,000 malicious. This better reflected proper IIoT traffic where malicious traffic may be a lot more subtle and rare.

5.1 Data Processing Outputs

The next stage involved using the CSV data in the Python notebook called ai_scanner_analysis.ipynb. The records were loaded into pandas, validated for completeness, and filtered to only retain features relevant to the Modbus experiment and model training. Since the generator was designed to produce clean data, minimal preprocessing and cleanup was needed. There was also no need to encode the data as mentioned above, so the result was a structured dataset ready for direct use in model training, with labels separating both types of records. Stratified splits ensured class proportions discussed earlier were preserved in training and test data.

5.2 Model Development

The next stage involved building two machine learning models using scikit-learn. As discussed, the two models I chose were Random Forest and Isolation Forest. The Random Forest was trained on the labelled dataset to classify benign vs malicious records. Its performance was evaluated using a classification report, a confusion matrix, a ROC curve, a precision recall curve and using feature importance. The Isolation Forest was trained on the benign records only to determine anomalies without prior attack labels. Initially, the contamination was set to the true malicious ratio but was later changed to "auto" to reflect realistic scenarios where it's an unknown value. This resulted in around 11.6% of traffic being flagged as anomalous, showcasing the strengths and limitations of unsupervised methods. Both models were implemented and evaluated in the python notebook.

6 Evaluation

This evaluation set out to determine whether the proposed AI-driven scanner could reliably detect anomalous Modbus activity in IIoT environments using both supervised and unsupervised approaches. It includes both quantitative metrics and visual outputs supported by screenshots from the Python notebook implementation.

6.1 Dataset Overview

The dataset used in the research contained 15,500 records generated using the traffic logger script, modbus_ai_logger.py, consisting of 3,000 malicious entries and 12,500 benign. Unlike earlier experiments where malicious and benign classes were easily separable, the new script introduced more over to make the data more realistic. Each entry was labelled and contained the function code, register address, and value transmitted, along with its benign or malicious classification. It also included a timestamp and response field. Below is a sample of the top of that file.

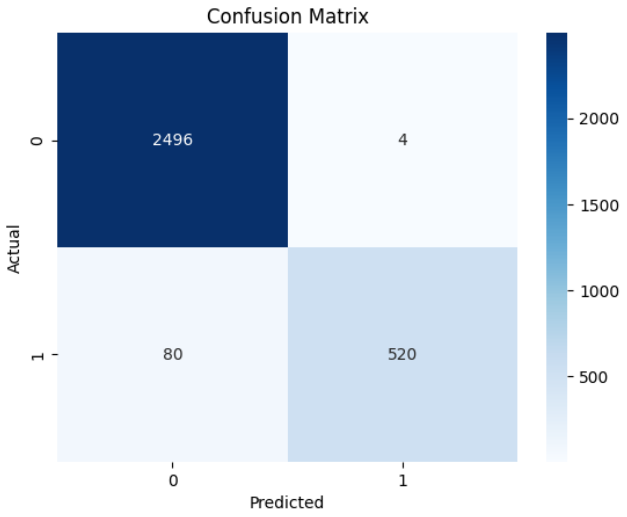
	timestamp	function_code	address	value	success	label	response
0	2025-08-20T01:42:31.075182	3	500	2539	True	malicious_write	b'\x97\x8e\x00\x00\x00\x03\x01\x83\x03'
1	2025-08-20T01:42:31.299105	6	328	2000	True	malicious_write	b'\xb0\x03\x00\x00\x00\x03\x01\x86\x02'
2	2025-08-20T01:42:31.504786	4	6114	453	True	malicious_write	b'\xb6\n\x00\x00\x00\x03\x01\x84\x03'
3	2025-08-20T01:42:31.706500	3	356	52489	True	malicious_write	b'M\xb3\x00\x00\x00\x03\x01\x83\x03'
4	2025-08-20T01:42:31.908118	5	24	4908	True	malicious_write	b'\xc2\xbb\x00\x00\x00\x03\x01\x85\x03'

6.2 Supervised Model Performance

The Random Forest classifier was trained on 80% if the dataset, with the remaining 20% left for testing the performance. The classification report below shows that the model achieved 97% accuracy, with the benign records hitting a recall of 1.00 and the malicious records slightly lower at 0.87. Overall, this shows the classifier was very accurate, though it did still miss some malicious records.

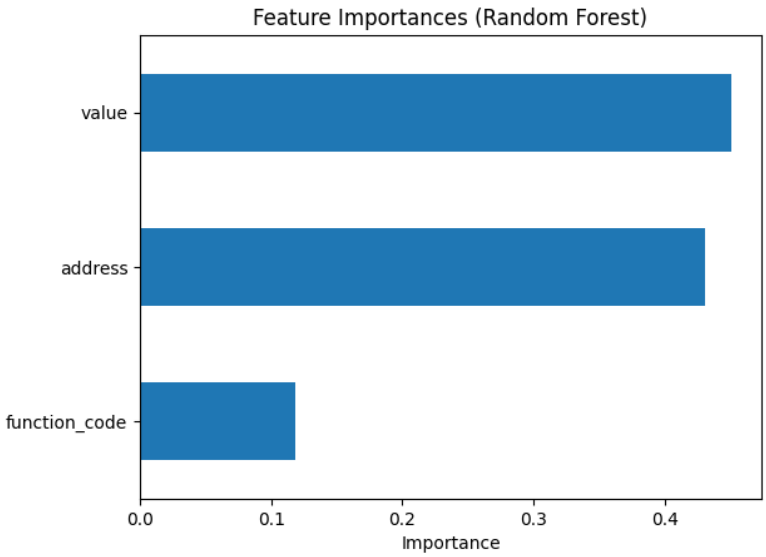
```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      2500
           1       0.99      0.87      0.93       600

    accuracy                           0.97      3100
   macro avg       0.98      0.93      0.95      3100
weighted avg       0.97      0.97      0.97      3100
```

The confusion matrix visualised the model's predictions against true labels. This resulted in 2496 true negatives, 520 true positives with only 4 false positives and 80 false negatives. So, we can deduce that while benign detection is nearly perfect, there is still some malicious traffic that is incorrectly classified as benign.

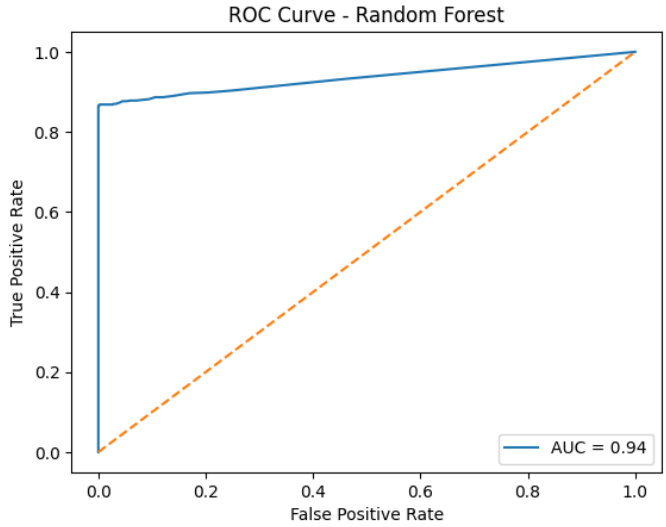


Looking at feature importances, we can see the Random Forest highlighted value (0.45) and address (0.43) as the strongest predictive features, with function_code (0.12) coming in as less influential overall. This suggests that anomalous values and addresses are stronger indicators of malicious behaviour than function codes alone.

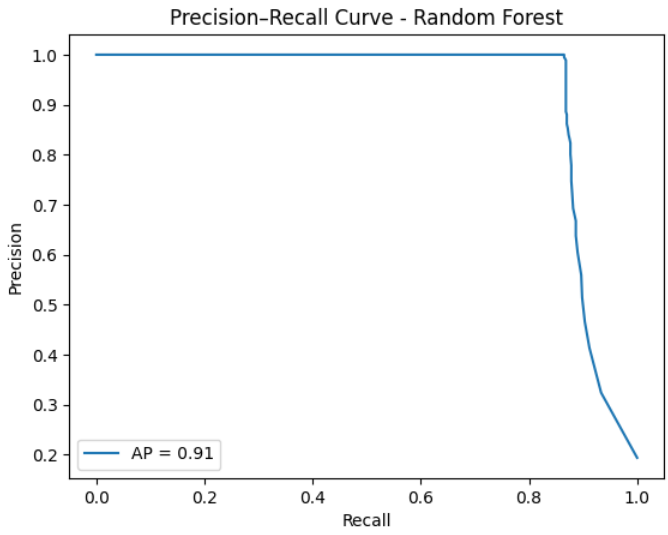
```
function_code: 0.1181
address: 0.4306
value: 0.4513
```



Moving on to the ROC curve, we see the model achieved an AUC curve of 0.94. This shows a high capability to distinguish between benign and malicious records across thresholds. The curve shows strong performance, although not perfect separation.



Finally, looking at the Precision Recall curve, we see the model achieved an average precision score of 0.91. This is a strong score and shows the balance between precision and recall, which is particularly important in this context, where catching as many attacks as possible is a necessity.



6.3 Unsupervised Model Performance

To complement supervised learning with Random Forest, the Isolation Forest algorithm was chosen and applied to the same dataset without using labels during the training. The model flagged 1,800 anomalies out of the 15,500 records, around 11.6%, which is in line with the malicious traffic ratio in the dataset. In total the dataset was known to have 3,000 malicious records, which was about 19% of the full dataset, meaning that the model still missed a considerable amount of them. While this does prove that the model can identify previously unseen and suspicious Modbus activity, even when explicit labels are unavailable, it does highlight the known limitations of unsupervised approaches, which often prioritise minimising false positives at the cost of reduced recall.

	function_code	address	value	anomaly_label
0	3	500	2539	0
1	6	328	2000	0
2	4	6114	453	1
3	3	356	52489	1
4	5	24	4908	0

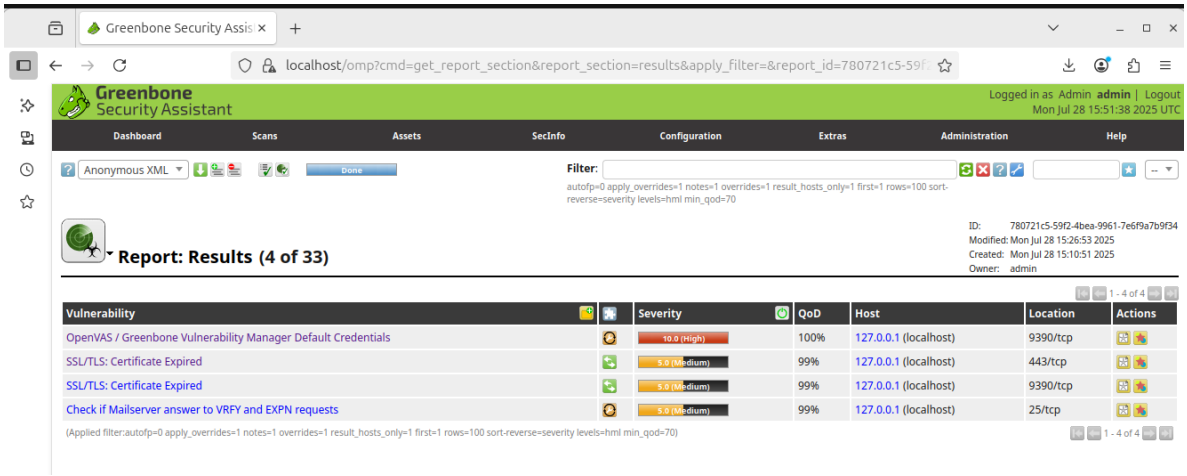
```
Anomalies flagged: 1800 / 15500 (11.6%)
```

	function_code	address	value
2	4	6114	453
3	3	356	52489
6	43	341	1308
8	6	7139	749
11	4	417	59976
12	4	4955	56348
13	4	99	61147
16	3	2995	2434
17	15	370	2844
18	3	3	42286

6.4 CVE-driven Scanner Comparison - Greenbone Security Assistant

To compare with a known scanner, I chose to use OpenVAS as discussed previously. This was bundled into Greenbone Security Assistant, and I found a version of the tool on Github from a

user called mikesplain, who bundled the tool into a docker container for use on Ubuntu, which suited my project and required very little configuration unlike the Conpot docker. I wanted to test whether a signature-based scanner could detect the type of Modbus traffic generated by the script and found using the models. The standard scan revealed 33 vulnerabilities in total, most of which related to the scanner container itself, such as using default OpenVAS/Greenbone credentials, expired SSL/TLS certificates and mail service checks on port 25. It also detected other services that were exposed on 25/tcp, 443/tcp, and 9390/tcp. These were to do with the OpenVAS docker and not the Conpot docker. Modbus was running on 5020/tcp through the Conpot, and was therefore not flagged by the scan, this highlighted the fact that CVE-driven scanners excel at finding known misconfigurations and outdated services but don't detect anomalous Modbus behaviour generated for this research, such as fuzzed function codes or repetitive coil writes. In contrast, the AI scanner was able to find such behaviour and is therefore able to fill the detection gap, acting as a complementary tool rather than a direct replacement.



6.5 Discussion

Throughout the evaluation, the AI-driven scanner performed strongly in both supervised and unsupervised detection, but with its own limitations. The Random Forest classifier achieved high accuracy scores of 97% with strong precision and recall, however its performance was still biased towards classifying records as benign, with a perfect detection rate of recall 1.00, whereas malicious detection was lower, with a recall of 0.87. This imbalance reflected a recurring challenge in intrusion detection, particularly in IIoT environments where attack traffic is often underrepresented. For example, similar recall drop off was recorded by Biswas et al, who observed degraded performance on imbalanced SCADA datasets despite using resampling techniques (Biswas et al., 2024). Nankya et al. also highlighted the risk of false negatives in industrial anomaly detection, noting that undetected threats pose significant operational and safety risks (Nankya et al., 2023).

The feature importance also confirmed that function codes alone are not sufficient indicators of malicious activity in Modbus. My model showed that address ranges and values were better for identifying malicious attempts; however, this could lead to attackers adapting by carefully mimicking legitimate ranges and values, which the current model may fail to deal with.

The unsupervised model also highlighted the potential and limitations of anomaly detection. While it successfully flagged 11.6% of traffic as anomalous, it did not catch all the malicious traffic, which accounted for roughly 19% of the dataset. In this case, the model under-reported malicious activity, trading recall for fewer false positives, a limitation consistent with findings by Suchorab et al. and El Gharbaoui et al., who noted that reducing false positives often weakens overall detection (Suchorab et al., 2025) (El Gharbaoui et al., 2024).

The OpenVAS baseline comparison highlighted how the scanner could work as a complementary tool rather than a complete replacement. This reinforces the evaluation finding that OpenVAS overlooked Modbus anomalies, confirming the critique of signature-based approaches and highlighting the value of a combined approach using both types of tools.

From a design perspective, one limitation of this project was the reliance on synthetic traffic, as discussed previously. Throughout the project, the generation script was completely reworked to make it more realistic with attempts to replicate the unpredictability of real industrial network traffic; however, nothing will replace the real thing. As noted in the literature, the lack of realistic datasets remains one of the most significant barriers to progress in this field. If deployed in production, the models trained here would likely require retraining with live industrial data.

Another limitation could be scalability. The experiments were run on a relatively small dataset of 15,500 records. Although the models trained quickly and performed well at this scale, it remains to be seen how they would perform on an industrial level. In industrial settings, millions of records are generated every day, to handle such volumes down the line, future work could look at batch or stream processing of records or deploying lightweight models on edge gateways, which would reduce the central processing load on the scanner and help it maintain its detection accuracy.

In summary, the evaluation showed that the AI-driven scanner could enhance the detection of anomalous Modbus activity where CVE-driven scanners, such as OpenVAS, fail but also highlighted that this is not a complete solution since malicious recall and the reliance on artificial datasets need to be addressed, with performance tested under realistic conditions.

7 Conclusion and Future Work

The project set out to answer the following research question: “Can an AI-Based Behavioural Scanner Improve the Detection of Protocol-Level Misconfigurations and Anomalies in Industrial IoT environments?”. To address this, I developed and evaluated an AI-driven scanner built around supervised and unsupervised machine learning models. The solution was tested on traffic generated by a custom Modbus logger script designed to produce realistic, overlapping and imbalanced datasets.

The evaluation showed that the supervised Random Forest model achieved strong performance and was effective in identifying anomalous traffic patterns, while the unsupervised Isolation Forest further complemented this by flagging anomalies without labels, although it under-

reported malicious traffic compared to the real ratio. Finally, the tool was compared with OpenVAS, which confirmed that CVE-based scanners could not detect the Modbus activity presented in this research.

In my view, the research contributes three main outcomes. First, it provides a reproducible traffic generation and dataset creation pipeline, tailored to the Modbus protocol, which fills a gap in available IIoT security datasets. Second, it shows that feature-level insights such as address and value ranges are stronger predictors of malicious behaviour than function codes alone, an observation with practical relevance for my model's design. Third, it validates that behavioural analysis and machine learning models can act as complementary tools to traditional scanners, strengthening detection where CVE-driven scanners are proven to fall short.

To summarise, the promising findings indicate that an AI-based behavioural scanner can indeed improve detection of protocol-level anomalies in IIoT environments but also highlight key limitations. The reliance on synthetic datasets, reduced recall for malicious records, and uncertainty about scalability mean that further research is required before practical deployment.

Future work should focus on testing with real industrial traffic datasets like WUSTL-IIOT-2021, which was also discussed in the literature (M. Zolanvari et al., 2021), or with proprietary datasets from industry partners. Real world validation is an essential next step, as it addresses the limitations of simulated data and ensures that the models generalise beyond laboratory conditions. A secondary priority ideally would be looking at improving recall for malicious traffic, for example by exploring ensemble approaches or hybrid supervised-unsupervised pipelines and approaches. Beyond this, expansion to other protocols such as BACnet and Profinet would test the generalisability of the approach, while scalability research should consider edge deployment or streaming models to handle large amounts of production-level traffic. In terms of commercialisation, some opportunities could exist in integrating the scanner as a plugin to OpenVAS/Greenbone, or as a lightweight edge module for operations teams, which would make anomaly detection more accessible.

Finally, a reflection on the learning journey is important. Early experiments with a perfectly balanced dataset gave "too perfect" and unrealistic results, forcing a redesign of the logger script to introduce overlaps and imbalance. This shift mirrored the realities of industrial environments, where malicious traffic is both subtle and rare, and it taught me the importance of critical evaluation and iteration in research. Developing the logger, tuning the models, and benchmarking against OpenVAS gave me a deeper understanding and appreciation of both the technical and practical challenges of IIoT security research. This process strengthened my ability to adapt methods, question results, and design tools that are both scientifically valid and operationally relevant.

References

Al-Dalky, R. *et al.* (2014) 'A Modbus traffic generator for evaluating the security of SCADA systems', in *2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*. *2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*, pp. 809–814. Available at: https://doi.org/10.1109/CSNDSP.2014.6923938.

Alfahaid, A. *et al.* (2025) 'Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey', *Sensors*, 25(11), p. 3341. Available at: https://doi.org/10.3390/s25113341.

Aslam, M.M. *et al.* (2024) 'Scrutinizing Security in Industrial Control Systems: An Architectural Vulnerabilities and Communication Network Perspective', *IEEE Access*, 12, pp. 67537–67573. Available at: https://doi.org/10.1109/ACCESS.2024.3394848.

Biswas, H. *et al.* (2024) 'Cyber Security in Smart Grid/SCADA: Use of Artificial Intelligence/Machine Learning', in *Global Trends in Water Resources, Power and Renewable Energy Technologies*. Indian Institute of Technology Roorkee, India.

Eyeleko, A.H. and Feng, T. (2023) 'A Critical Overview of Industrial Internet of Things Security and Privacy Issues Using a Layer-Based Hacking Scenario', *IEEE Internet of Things Journal*, 10(24), pp. 21917–21941. Available at: https://doi.org/10.1109/JIOT.2023.3308195.

Ferst, M.K. *et al.* (2018) 'Implementation of Secure Communication With Modbus and Transport Layer Security protocols', in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*. *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pp. 155–162. Available at: https://doi.org/10.1109/INDUSCON.2018.8627306.

Gharbaoui, O.E., Kiyadi, I. and Boukhari, H.E. (2024) 'Evaluating AI and ML in Network Security: A Comprehensive Literature Review', *Procedia Computer Science*, 251, pp. 727–733. Available at: https://doi.org/10.1016/j.procs.2024.11.176.

Gupta, H. *et al.* (2024) 'Operational Technologies in Industrial Control System: Cybersecurity Perspectives and Research Trends', in *2024 17th International Conference on Security of Information and Networks (SIN)*. *2024 17th International Conference on Security of Information and Networks (SIN)*, pp. 01–08. Available at: https://doi.org/10.1109/SIN63213.2024.10871534.

Holasova, E. *et al.* (2021) 'Security Modules for Securing Industrial Networks', in *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*. *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, pp. 1125–1132. Available at: https://doi.org/10.1109/CECIT53797.2021.00199.

Jayalaxmi, P. *et al.* (2021) 'A Taxonomy of Security Issues in Industrial Internet-of-Things: Scoping Review for Existing Solutions, Future Implications, and Research Challenges', *IEEE Access*, 9, pp. 25344–25359. Available at: https://doi.org/10.1109/ACCESS.2021.3057766.

Karacayılmaz, G. and Artuner, H. (2024) 'A novel approach detection for IIoT attacks via artificial intelligence', *Cluster Computing*, 27(8), pp. 10467–10485. Available at: https://doi.org/10.1007/s10586-024-04529-w.

Katulić, F. *et al.* (2023) 'Protecting Modbus/TCP-Based Industrial Automation and Control Systems Using Message Authentication Codes', *IEEE Access*, 11, pp. 47007–47023. Available at: https://doi.org/10.1109/ACCESS.2023.3275443.

Kotsiopoulos, T. *et al.* (2025) 'Defending industrial internet of things against Modbus/TCP threats: A combined AI-based detection and SDN-based mitigation solution', *International Journal of Information Security*, 24(4), p. 157. Available at: https://doi.org/10.1007/s10207-025-01076-2.

Lazaridis, G. *et al.* (2024) 'Unraveling the Threat Landscape of CPS: Modbus TCP Vulnerabilities in the Era of I4.0', in *2024 IEEE International Conference on Cyber Security and Resilience (CSR). 2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 593–598. Available at: https://doi.org/10.1109/CSR61664.2024.10679453.

M. Zolanvari *et al.* (2021) 'WUSTL-IIOT-2021 Dataset for IIoT Cybersecurity Research'. Washington University in St. Louis, USA. Available at: https://www.cse.wustl.edu/~jain/iiot2/index.html (Accessed: 25 August 2025).

Martins, T. and Oliveira, S.V.G. (2022) 'Enhanced Modbus/TCP Security Protocol: Authentication and Authorization Functions Supported', *Sensors*, 22(20), p. 8024. Available at: https://doi.org/10.3390/s22208024.

Mubarak, S. *et al.* (2021) 'ICS Cyber Attack Detection with Ensemble Machine Learning and DPI using Cyber-kit Datasets', in *2021 8th International Conference on Computer and Communication Engineering (ICCCE). 2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, pp. 349–354. Available at: https://doi.org/10.1109/ICCCE50029.2021.9467162.
Nankya, M., Chataut, R. and Akl, R. (2023) 'Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies', *Sensors*, 23(21), p. 8840. Available at: https://doi.org/10.3390/s23218840.

Nardone, R., Rodríguez, R.J. and Marrone, S. (2016) 'Formal security assessment of Modbus protocol', in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST). 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 142–147. Available at: https://doi.org/10.1109/ICITST.2016.7856685.

Rodríguez, R.J. *et al.* (2023) 'MOSTO: A toolkit to facilitate security auditing of ICS devices using Modbus/TCP', *Computers & Security*, 132, p. 103373. Available at: https://doi.org/10.1016/j.cose.2023.103373.

Sen, S.K., Karmakar, G.C. and Pang, S. (2024) 'Assessment of IIoT Sensor Security Vulnerabilities in Digital Wine Manufacturing Leveraging the CVSS', *IEEE Access*, 12, pp. 141489–141513. Available at: https://doi.org/10.1109/ACCESS.2024.3467248.

Siwakoti, Y.R. *et al.* (2023) 'Advances in IoT Security: Vulnerabilities, Enabled Criminal Services, Attacks, and Countermeasures', *IEEE Internet of Things Journal*, 10(13), pp. 11224–11239. Available at: https://doi.org/10.1109/JIOT.2023.3252594.

Srivastava, A. *et al.* (2024) 'Development of Cyber-Physical Security Simulation Testbed in RTDS using Modbus Communication', in *2024 IEEE Power & Energy Society General Meeting (PESGM)*. *2024 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5. Available at: https://doi.org/10.1109/PESGM51994.2024.10689146.
Suchorab, J., Plamowski, S. and Ławryńczuk, M. (2025) 'Anomaly detection system for Modbus data based on an open source tool', *Computers & Security*, 157, p. 104572. Available at: https://doi.org/10.1016/j.cose.2025.104572.

Tsiknas, K. *et al.* (2021) 'Cyber Threats to Industrial IoT: A Survey on Attacks and Countermeasures', *IoT*, 2(1), pp. 163–186. Available at: https://doi.org/10.3390/iot2010009.

Zarzycki, K. *et al.* (2023) 'GAN Neural Networks Architectures for Testing Process Control Industrial Network Against Cyber-Attacks', *IEEE Access*, 11, pp. 49587–49600. Available at: https://doi.org/10.1109/ACCESS.2023.3277250.