Attacks on the Siemens S7 Protocol Using an Industrial Control System Testbed

Nicolai Kellerer
Karlsruhe Institute of Technology (KIT)
KASTEL Security Research Labs
Karlsruhe, Germany
nicolai.kellerer@kit.edu

Gustavo Sánchez
Karlsruhe Institute of Technology (KIT)
KASTEL Security Research Labs
Karlsruhe, Germany
sanchez@kit.edu

Hermenegildo Alberto
Karlsruhe Institute of Technology (KIT)
KASTEL Security Research Labs
Karlsruhe, Germany
hermenegildo.alberto@kit.edu

Veit Hagenmeyer
Karlsruhe Institute of Technology (KIT)
KASTEL Security Research Labs
Karlsruhe, Germany
veit.hagenmeyer@kit.edu

Ghada Elbez
Karlsruhe Institute of Technology (KIT)
KASTEL Security Research Labs
Karlsruhe, Germany
ghada.elbez@kit.edu

Abstract

The stability of critical infrastructure depends on a secure energy supply, highlighting the need for robust cybersecurity in Smart Grids (SGs) as they increasingly integrate renewable energy sources. Unlike traditional power plants, modern renewable facilities are distributed and rely on remote control, broadening the attack surface and requiring enhanced resilience against cyber threats to ensure availability. Testing cyberattacks on physical power plants poses risks of outages and financial losses. To address this, we developed a Hardware-in-the-Loop (HIL) testbed simulating three renewable energy plants, offering a safe, adaptable, and cost-effective platform for analyzing cyberattack impacts. Our novel testbed architecture enables simulation of unsafe states under cyberattack scenarios. We advance the field by demonstrating an attack using a S7 data modification technique, rarely explored in previous research. Additionally, we contribute a comprehensive dataset for SG cybersecurity, using it to create a baseline Machine Learning (ML) based Intrusion Detection System (IDS).

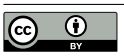
CCS Concepts

• **Security and privacy** → **Network security**; *Intrusion detection systems*; • **Hardware** → **Power and energy**.

Keywords

Data Modification Attack, Siemens S7, IDS, Smart Grid, Energy Generation

ACM Reference Format:
Nicolai Kellerer, Gustavo Sánchez, Hermenegildo Alberto, Veit Hagenmeyer, and Ghada Elbez. 2025. Attacks on the Siemens S7 Protocol Using an Industrial Control System Testbed. In *The 16th ACM International Conference on Future and Sustainable Energy Systems (E-ENERGY '25), June 17–20, 2025, Rotterdam, Netherlands.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3679240.3734645



E-ENERGY '25, Rotterdam, Netherlands

ACM ISBN 979-8-4007-1125-1/25/06
https://doi.org/10.1145/3679240.3734645

1 Introduction

Attacks like Stuxnet [26] have shown that Industrial Control System (ICS) networks are a strategic target for cyberattacks as they can result in physical damage. In addition, the operator and any dependent customers will incur considerable financial losses until the production system is repaired and back online. Therefore, power grids are a particularly high-impact target because of their importance to other critical infrastructure. The Ukraine power grid was attacked in 2015 with the BlackEnergy malware [12] and in 2016 with Industroyer (aka Crashoverride) [40]. Furthermore, legacy systems were previously air-gapped, and thus cyber defense received little attention. The distributed generation environment of the Smart Grid (SG) significantly increases the attack surface, because the power plants are connected to the remote Supervisory Control and Data Acquisition (SCADA) system over public networks. SCADA systems operate by monitoring and controlling physical processes through a network of Programmable Logic Controllers (PLCs), which in turn control sensors and actuators. The SCADA system and the PLCs are interconnected at the control layer and communicate via protocols such as IEC 60870-5-104 and Siemens S7. The S7 protocol is of particular concern, which, despite its age and security weaknesses, remains widely deployed, including in modern power plants.

The hierarchical structure of SCADA, typically described by the Purdue model, places SCADA systems at level 2, where they control physical processes running at levels 0 and 1 [20]. The primary vulnerability of SCADA systems lies in their integration with IT networks, which exposes them to cyber threats that exploit this connectivity. Attackers can gain access through IT means, such as phishing, and then leverage the SCADA system to send malicious commands to ICS devices [20]. This connectivity also means that attacks can be launched remotely, significantly increasing the attack surface.

In some countries, private networks are used instead to connect power plants to the control center in a more secure way, but as

the Kaspersky ICS report [23] from Q1 2024 has revealed, such a network is still vulnerable through infection of PCs and servers indirectly connected to the Internet. The report indicates that 25.1% of all computers within ICS networks in the energy sector were attacked, with the Internet as the primary source (12.24%). In most cases, the initial infection of ICS computers occurs through phishing attacks originating from the Web or E-Mail. Therefore, improving network security and hardening the Internet-facing components is essential in the ICS environment to defend against initial infections.

In this paper, we present a testbed specially designed for cybersecurity research, utilizing a Hardware-in-the-Loop (HIL) approach. The power plants are simulated with Python to allow experiments with the power plant in an unsafe state. Furthermore, this architecture allows rapid modifications to the layout of the power plants to implement different control behaviors of the microgrid. The Python simulation is connected to industrial controllers and communicates with a Siemens SCADA solution. This testbed implements a microgrid with a photovoltaic power plant, a wind power plant, and a battery power plant. Furthermore, we implement a data modification attack and release a dataset that can be used by the community to advance research, particularly for Intrusion Detection Systems (IDSs).

The main contributions of this paper are as follows:

- We present a comprehensive description of a cybersecurity testbed for distributed generation.
- We implement a data modification attack on the Siemens S7 protocol for inter-controller communications.
- We show the suitability of our dataset by training a Machine Learning-based Intrusion Detection System (IDS) prototype as a baseline for future research.

Additional contributions include the following:

- We release attack scripts and the dataset [25] based on network packets, SCADA events and log messages from field level devices[1].
- We also release the Python implementation of our simulator, which integrates a model for a photovoltaic power plant, a wind power plant, and a battery power plant.
- We propose a framework to taxonomize unsafe states, assigning a score based on severity, likelihood, duration, and controllability.

The remainder of the paper is organized as follows. In Section 2, we discuss related work. In section 3 we explain the Siemens S7 protocol. Section 4 describes the architecture of the testbed. Next, we explain the design of our data modification experiment, and the collection of the dataset in section section 5. In section 6 we present our method for multi-sensor data fusion and detection of the attacks with an Machine Learning (ML)-based IDS. Then we conclude the paper in section 7.

2 Related Work

This section provides an overview of related work focusing on two key areas. First, we examine existing literature on attacks targeting the S7 protocol, comparing them against our data modification attack detailed in section 5.2. Considering the extensive existing literature [21, 37, 39] on anomaly detection for Modbus/TCP in ICS networks, our investigation instead focuses on the S7 protocol. Second, we explore Multi-Sensor Data Fusion techniques. As our dataset consists of three sources, data fusion is essential for our IDS to enhance its detection accuracy. Compared to the Electra [32] dataset, we conducted S7 data modification attacks in the generation environment instead of a traction substation used to supply railways. Furthermore, our dataset provides additional features such as the binary packets, the parsed packet as JSON, and a list of the signals that were modified during each attack.

2.1 Attacks on the Siemens S7 Protocol

Siemens developed four S7 protocol revisions. Wael *et al.* [6] provide an overview of the cryptographically secured version of S7 called S7CommPlus. S7CommPlusV1 lacks integrity protection, S7CommPlusV2 includes integrity and anti-replay protection, and S7CommPlusV3 improves the integrity protection mechanism. They also demonstrate that an attacker with sufficient information can bypass the integrity mechanism of S7CommPlusV3 and spoof S7 Function packets to perform a replay attack. Finck *et al.* [13] reverse engineered the firmware of the S7-1500 software controller. They analyze the legacy handshake and cryptography used by S7CommPlusV3 and conclude that the PLC communication is only secured through a secret algorithm and not by cryptographic guarantees. The newer Transport Layer Security (TLS) handshake is not affected by this attack. Authors in [4, 5] showed that the PLC can be infected with malicious code at runtime, which runs at a specified time. Consequently, a cyberattack can be coordinated across multiple power plants. Hui *et al.* [19] build on works in [9, 18, 27] to circumvent the anti-replay mechanism even for a password-protected PLC. Ghaleb *et al.* [14] showed a stealth command modification attack. Furthermore, attacks on S7 between the Engineering software and the PLC have been investigated by [36].

Much attention has been given to the S7 connection between engineering software and PLC, but the S7 protocol is very flexible. This paper investigates the subtypes of the S7 protocol used for PLC to PLC communication. Therefore, the presented data modification attack demonstrates a new attack vector.

2.2 Multi-Sensor Data Fusion

The architecture of data fusion algorithms is divided into three categories [16]: in a centralized architecture the raw or derived data is first fused, before it is sent to the classifier. In contrast, in an autonomous architecture, the extracted features are first sent to the classifier before the fusion algorithm combines the probability distributions. The hybrid fusion architecture combines the centralized and autonomous approach.

The authors in [34] used the centralized fusion architecture to combine physical features from the DNP3 master, Snort alerts, and network packets captured with Packetbeat. Their approach utilizes co-training for inter-domain data fusion and performs 15-20% better than the baseline. [1] showcases a scalable data fusion methodology with the Stream-to-Stream Full Outer Join from Apache Spark. They achieve 99.98% accuracy with a Multi-Layer Perceptron Classifier on the SWaT, WADI and Edge-IIoT datasets.

[1] https://github.com/nbke/s7-attacks

3 Interfacing with the S7 Protocol

The S7 protocol is widely used in industrial automation systems. Effective interfacing with this protocol is essential for implementing robust cybersecurity measures and conducting authorized penetration testing. This section details our approach to manipulating S7 communication packets by utilizing custom extensions based on the s7scan project [31]. Our method focuses on packet filtering, modification, and interaction, specifically in simulated Man-in-the-Middle (MITM) scenarios, to underscore the critical manipulation techniques applicable to our testbed but generalizable to other setups.

3.1 Methodology

Our methodology leverages the Scapy library [10] alongside our custom "s7.py" module, which is designed to intercept and manipulate data transmitted between industrial components. This module defines a variety of packet structures and behaviors specific to the S7 protocol, such as job requests, write commands, and data exchange formats, thereby enabling detailed manipulation of communication processes.

3.2 Code Implementation

The attack script [24] utilizes the Scapy framework to define, dissect, and construct packets specific to the S7 protocol. This includes a series of packet definitions that mimic S7 communication frames for detailed network interaction and manipulation. These handle everything from basic packet structure to complex operations like cyclic data reads and writes, error handling, and reading System Status List (SZL) packets, which are used for diagnostics and PLC monitoring.

Furthermore, each packet class includes methods for padding extraction, used to align data fields correctly within the byte streams transmitted over the network. This meticulous packet manipulation facilitates extensive security testing and analysis of industrial networks, offering tools to simulate various network conditions and potential attack scenarios.

This Python-based handling of S7 communications enables deep integration into network services, supporting sophisticated interaction and control over automation processes, in this case beneficial from an adversarial viewpoint. In addition, the Scapy implementation of S7 is used during the dataset creation step to extract process values from the BSEND/BRECV packets.

4 Testbed Architecture

There are testbeds for the energy sector like [2, 17, 22, 33], but they do not provide labeled datasets usable for cybersecurity research. Park *et al.* [30] also introduce a testbed for distributed energy generation with a real-time simulator, but their setup does not include industrial controllers or the S7 protocol.

4.1 Hardware-in-the-Loop Setup

The proposed system integrates a photovoltaic power plant, wind power plant, and battery storage into a comprehensive hybrid energy solution. The architecture is designed with a modular approach, ensuring a clear separation between component models. This modularity facilitates system scalability and adaptability to diverse use cases. Real-time simulation capabilities are incorporated, allowing configurable update frequencies to adapt to dynamic operational conditions. The *PlantSim* simulator communicates with the three controllers at the field level via Modbus/TCP. Additionally, an S7 Get/Put connection is used to integrate the *PowerFlowSim* simulator with the control center for the power flow algorithm. Siemens WinCC runs on the Operator Station as a SCADA solution and maintains an S7 connection to the Process Control System (PCS) 410E in the control center, referred to as master PCS.

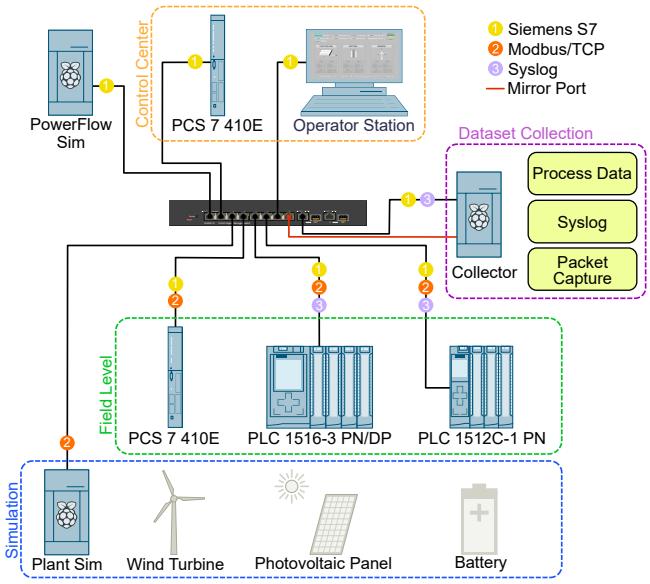


Figure 1: Architecture of testbed including data collection and conversion to dataset (contains icons from [38]).

Power Flow Management. The simulator running on the *Plant Sim* server is built on top of the *pandapower* [41] library, which integrates the three power plant models and provides an accurate simulation of the energy grid. This simulator also models the effects of a power deficit and surplus on the grid frequency. The power flow algorithm running in the control center has the task of maintaining a stable grid frequency by dynamically adjusting the distribution of generated energy. The control algorithm has the ability to charge and discharge the battery, regulate the power output of the wind turbine, and switch the photovoltaic power plant on or off to satisfy the demand. Pandapower integrates the Temperature-Dependent Power Flow (TDPF) [28] algorithm for balanced AC power flow from the *PYPOWER* library, which is a Python port of the *MATPOWER* library.

In our lab environment, the power flow algorithm is not implemented natively in the PCS running in the control center, but is rather offloaded to the *PowerFlow Sim* server, which is running a second instance of the *pandapower* simulation. The *PowerFlow Sim* retrieves the current state of the PCS. After running the simulation, the result is written to the PCS, after which it is relayed to the three controllers of the power plants.

The power flow algorithm must be run on a second instance of the simulation that is isolated from the *Plant Sim*, because in the case of a data modification attack, the state of the control center and the power plants will diverge. Therefore, it is required that the decisions of the power flow algorithm are solely based on the state retrieved from the SCADA system.

Technical Features. Real-time data logging and analysis are incorporated, providing continuous insights into the state of the simulated power plants. RESTful API endpoints enable remote system control and monitoring, for better accessibility and user interaction.

4.2 Monitoring and Visualization

Real-time monitoring is facilitated through an interactive dashboard. The dashboard provides multi-layered data visualization, including metrics for power generation and consumption, environmental parameters, system status indicators, and battery performance.

The system includes advanced analysis capabilities, such as historical data tracking and performance metric calculations. Efficiency monitoring tools provide insights into the operational effectiveness of the system. The platform also supports data export functionality, facilitating further analysis and reporting.

5 Dataset Generation

In this section, we first introduce our threat model for cyberattacks on power plants. Then we describe the data transmitted between the components of the testbed during normal operation and during the attack scenario. Furthermore, we explain the data pipeline for the capture of the dataset.

5.1 Threat Model for Power Plants

Every generator in a power plant is protected against physical damage with a Safety PLC. This special PLC is also responsible for avoiding human damage even in the event of a malfunction of the controlling PLC. A safety PLC reads directly the sensor values and is wired to the generator shutdown. While this device is networked in some installations, it is air-gapped in a separate network to defend against vulnerabilities in the network protocols. The presence of a Safety PLC limits the possible impact of a successful cyberattack on a power plant.

In our testbed, the PLC 1512, PLC 1516 and PCS 410E simulate the protocol conversion done by Remote Terminal Unit (RTUs). This approach is still representative of the real devices, as firmware attacks, reprogramming attacks and configuration changes are out of scope.

We assume an insider attack, where the attacker is already part of the network and has physical access to the control center and field level of each power plant. Furthermore, the attacker has already performed reconnaissance steps and therefore has knowledge of the network layout.

5.2 Proposed Experiment

In this section, we describe the data fields that are sent between the Siemens devices during normal operation and which other protocols can be observed on the network. Next, we implement a prototype of the data modification attack for the S7 connection between a Siemens 1500 PLC and a Siemens S7 PCS.

We use our HIL testbed introduced in section 4 to capture network packets, log messages and process data during normal operation and during cyberattacks. Afterwards we preprocess the raw data, assign labels during the autolabel step, and convert the dataset into the Apache Parquet format. This process is explained in detail in section 5.3.

Normal Operation. The control center implemented by the Master PCS instructs each power plant to generate power at a certain level. Hence, the control center sends the boolean variable *on_off* and the floating point value *target_power*. In fact, the power plants try to meet the power output defined by *target_power* and report back their actual *generated_power*. The control center then uses this information in the power flow algorithm to ensure that the output of the power plants meets the demand in the SG. Figure 2 shows the power generated by the PV and Wind Power Plant to meet the demand.

The PLCs and PCS, which represent RTUs, do not perform any processing on the transmitted values. Instead the model of each power plant receives the control commands from the RTU and calculates the output values, which are then send back to the control center. The simulation models can ramp up or down depending on the *target_power* and derive output voltages and currents. Furthermore, the battery power plant implements a State of Charge (SoC), such that the control center can store excess power in the battery or use the stored energy during spikes of high demand.

WinCC uses tags to refer to values in the Human Machine Interface (HMI) screens. These WinCC tags refer to values stored in the datablocks of the Master PCS. For each tag, WinCC sends a subscription to the Master PCS over a S7 connection. The Master PCS then sends an indication packet back on each value change. The HMI screen contains a button to switch on/off each power plant and an input box to configure the target power. Additionally, it displays the values that are received by the control center from the tree power plants.

The configuration of the devices was not changed during the dataset capture. Therefore the dataset does not contain packets from the engineering tools Simatic Manager and TIA Portal.

Attack scenario on PLC. We performed a data modification attack on the S7 connection between the master PCS and the PLC for the photovoltaic power plant. The operator station connects to the master PCS via S7, with the master PCS as the server and the operator station as the client. Additionally, the PLC and WinCC both connect to the master PCS via S7. Before the attack occurs, the traffic flows directly between the master PCS and the PLC. After the attacker performs the Address Resolution Protocol (ARP) spoofing attack, the packets are relayed between the master PCS and the PLC via the attacker controlled PC. To perform the MITM attack, the attacker must be connected to the switch or infect a computer in the network. In our testbed a separate attacker PC is used to invoke the *arpspoof* tool from the *dsniff* package and later execute the data modification script. After the ARP spoofing attack, the network packets are sent to the attacker PC instead of the intended destination, where the attacker can inspect the packet contents, modify them, and then forward the packets to their destination. This

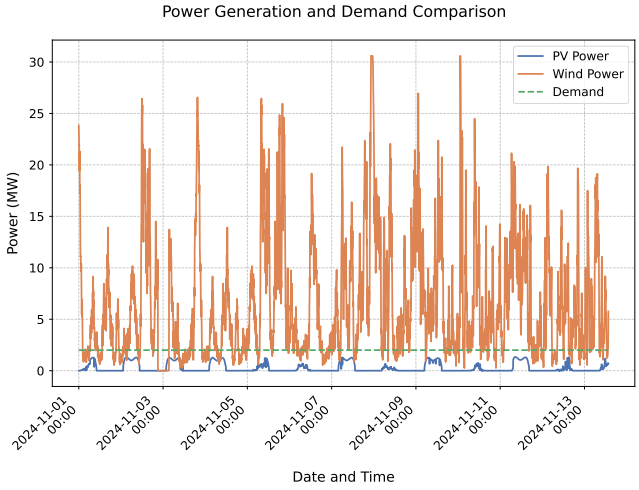


Figure 2: Power Generation and Demand.

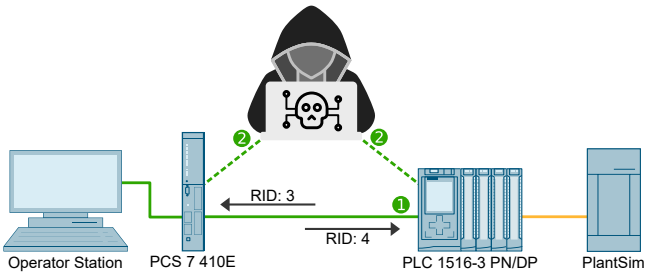


Figure 3: Data modification attack on S7 protocol between a PLC and PCS (contains icons from [38]).

type of attack is possible because the legacy version the S7Comm protocol used in this scenario lacks authentication and encryption.

Due to reduced energy generation, the demand can no longer be met in this microgrid scenario with only one remaining power plant, leading to a blackout.

5.3 Data Collection

The dataset consists of network packets, process data and log messages. The process data is collected every second by *PowerFlowSim* from the PCS 410E used in the control center and stored in a database.

A mirror port is configured on the switch that sends a copy of all packets on the network to a server for further processing. On the server *tcpdump* is used to collect all packets and store them as `.pcap` files.

The PLC 1516 and PLC 1512 monitor their signals and trigger an alarm, which results in a syslog message, when the value of a signal passes the thresholds defined in table 1 and 5. For capturing the Syslog messages of the PLCs *rsyslog* is used. Syslog requires at least version 16 of TIA Portal and the installation of the Siemens Communication library, which provides the Syslog function blocks.

Table 1: Syslog rules on PLC 1516 (PV).

Name	Lower Threshold	Upper Threshold	Type
Air Temperature	1.0 °C	30.0 °C	Monitor
Wind Speed	0.5 m/s	10.0 m/s	Monitor
Plane of Array (POA) Direct	100 W/m^2	675 W/m^2	Monitor
POA Diffuse	50 W/m^2	300 W/m^2	Monitor
Cell Temperature	2.0 °C	40.0 °C	Monitor
Inverter AC Power	250 kW	2000 kW	Monitor
Inverter DC Power	250 kW	2000 kW	Monitor
On Off	N/A	N/A	Control

The data modification script introduced in 5.2 was integrated into a fuzzer to generate a large dataset with high variation. The fuzzer operates as follows: Every 30 seconds it randomly chooses an S7 connection between a power plant and the control center. The fuzzer then decides on the direction of the connection for the attack and executes the *arpspoof* tool to establish a MITM position. This allows the attacker to intercept all packets from the targeted S7 connection. Following this, the fuzzer uses a custom Scapy parser for S7 together with a definition for the datablocks used by the control center to decode the S7 BSEND/BRECV packets. It randomly chooses one or more signals from the decoded datablock for the data modification attack. Next, it randomly chooses a delta for every attacked signal and finally forwards the modified packet. Additionally, the fuzzer implements a feature where it sometimes reuses a previous decision to reduce the jitter in the modified signals.

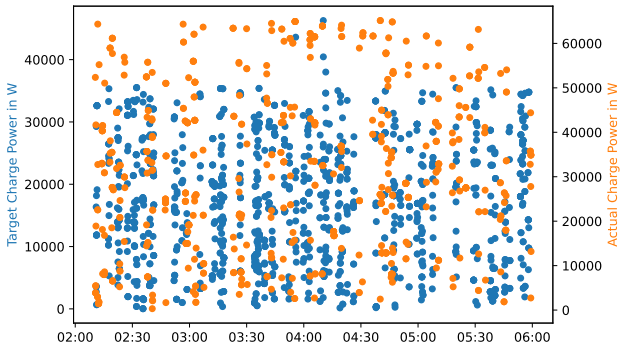


Figure 4: Largest difference between original value and modified value for the signals target charge power and actual charge power.

With this approach we created a dataset that consists of 10,002,832 packets, 6,459 log messages and 13,912 process data entries over a span of 3 hours and 52 minutes. The figure 4 shows the largest difference between the original value of the signal and the value after the data modification attack for the target charge power signal from the Remote ID (RID) 2 and actual charge power from RID 1. The S7 connection between the PLC 1512 (Battery) and the master PCS is identified as RID 1. RID 2 represents the reverse direction.

During the data modification attacks, a list of modified signals are identified. For each signal, both its original value and the changed value are saved to a database. Additionally, we store the original

packet and the modified packet for later analysis. With this information the Autolabel script can match the recorded packet in the `.pcap` files with the modified packet from the attack database. This way the dataset can be automatically labeled as *normal* or *data_modification*. If the Autolabel script is not able to find a packet in the `.pcap` files because *tcpdump* was stopped before the attack script, then it applies the label *no_match*. The dataset contains 9,991,107 normal packets and 11,683 modified packets.

6 Preliminary Attack Detection Results

In this section, we show preliminary attack detection results with a proof-of-concept implementation of an IDS. We describe how the presented dataset can be used to train ML models.

As a pre-processing step, we remove feature columns that exhibit a single unique value (e.g., *facility*, *tag*, and *severity* from the log messages) or that may lead to spurious correlations [8, 35] (e.g., *hostname*, *IP address*, *MAC address*) to ensure our models are trained on meaningful and non-redundant information. Our IDS instead leverages process values extracted from the S7 application layer protocol, log messages, and directly from the SCADA system for detection of the data modification attacks.

Our proposed IDS aggregates the data from the three sources into fixed time windows and labels each window as an attack if any packet within the window is abnormal (i.e., not labeled as `normal`). We explored multiple candidate time windows (5, 10 and 15 seconds) as a hyperparameter, and through grid search, we identified an optimal aggregation window of 10 seconds, which yielded the best F1 macro score. The F1 macro score takes the F1 scores of each class and averages them, treating all classes equally:

$$F_1^{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} \frac{2\,\text{Precision}_i\,\text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{1}$$

where K is the number of classes.

In parallel, we tuned the hyperparameters of three baseline classifiers – Logistic Regression, Decision Tree, and Random Forest (RF) – using Grid Search with the F1 macro score as the evaluation metric. These tuned models were then combined into a hard-voting ensemble, which, when evaluated on the optimal 15-second aggregation configuration, achieved an overall F1 macro score of 90% on the test set (99% on attack and 80% on normal samples). However, the RF model was able to achieve a F1 macro score of 95% on the test set (100% on attack and 91% on normal samples), outperforming the ensemble. These results are presented in table 2.

Table 2: Confusion matrix for each model using a 15s time window (Positive: Attack, Negative: Normal).

Model	FN	FP	TN	TP
Logistic Regression	0	6	0	187
Decision Tree	0	2	4	187
Random Forest	0	1	5	187
Ensemble	0	2	4	187

Now, we use explanations to better interpret the best model (RF) via feature importance. The built-in feature importance method for RF is Gini Importance, also known as Mean Decrease in Impurity (MDI) [11]. This method relies on the reduction in impurity achieved by each feature during the construction of the decision trees. The average importance in table 3 is scaled by 10^3 for clarity.

The feature ranking reveals the most critical features for detecting cyberattacks in our SG dataset. Notably, the feature `X_new_value` from the logs data achieved the highest average normalized importance. Following this, features `X_old_value` (also from logs data) and `in_batt_actual_charge_power` from process data rank highly.

Table 3: Top 10 Features for the best-performing model (Random Forest)

Feature	Avg. Importance	Source
X_new_value	97.0	Logs
X_old_value	95.9	Logs
in_batt_actual_charge_power	87.6	Process
in_batt_temperature	86.8	Process
in_pv_cell_temperature	83.5	Process
in_pv_wind_speed	76.1	Process
in_pv_poa_direct	54.8	Process
in_batt_current	45.3	Process
in_batt_state_of_charge	44.2	Process
in_pv_inverter_ac_power	43.3	Process

Overall, the RF feature importance ranking indicates that a combination of energy measurements and log metadata collectively contributes to a robust detection mechanism, providing an advanced view of system behavior under potential cyberattack scenarios.

7 Conclusion and Future Work

We introduced our Hardware-in-the-Loop (HIL) testbed for cybersecurity research in Industrial Control System (ICS) networks, which offers a scalable and modular approach to integrating renewable energy sources with advanced monitoring and control mechanisms. The testbed integrates PV, wind power, and battery storage to enable real-time renewable energy management. We captured a dataset that includes S7 attacks, which are designed to cause the unsafe states identified in section B. Our S7 data modification attack modifies the control commands that are sent to the power plant or the monitoring values that are transmitted to the SCADA system, thereby influencing the decisions of the power flow algorithm.

This work investigates the lack of security of the Siemens S7 protocol as no TLS encryption is used, which was only introduced in firmware version 2.9 for the S7-1500 controllers and firmware version 4.3 for the S7-1200 controllers [3]. However, many deployments still rely on perimeter security instead of encryption. Thus, an attacker can gain control over the ICS network through physical access to the network components or through the office network by using phishing emails.

Furthermore, an ICS network can be hardened against data modification attacks through a multifaceted approach. First, implementing advanced IDS systems that are capable of recognizing unusual command or data patterns is crucial for early detection of anomalies. Equally important is the practice of regular updates and patch management, particularly for software components connected to the Internet, to close off known vulnerabilities. Comprehensive network monitoring should be employed to examine traffic to critical components such as PLCs and protection relays, ensuring any irregular behavior is promptly identified. In parallel, robust authentication and access control measures must be established, especially

for remote access and maintenance activities, to prevent unauthorized intrusion. Finally, the deployment of redundant security measures, including a variety of firewalls and anomaly detection systems, further reinforces the network's resilience by providing additional layers of defense.

In the future, we want to extend our testbed with Virtual Private Network (VPN) Gateways to model IPsec connections between the control center and remote power plants. Furthermore, we want to improve the simulation of the battery power plant by limiting the maximum charge and discharge rate depending on the temperature of the environment. In addition, future research could explore host-based features extracted from the Windows Event log of the Operator PC for the detection of malware attacks.

Acknowledgments

This research is supported in part by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs (structure 46.23.02).

Disclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Ahlem Abid, Farah Jemili, and Ouajdi Korbaa. 2024. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. *Cluster Computing* 27, 2 (2024), 2217–2238. doi:10.1007/s10586-023-04087-7

[2] Sridhar Adepu, Nandha Kumar Kandasamy, and Aditya Mathur. 2019. EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security. In *Computer Security*, Sokratis K. Katsikas, Frédéric Cuppens, Nora Cuppens, Costas Lambrinoudakis, Annie Antón, Stefanos Gritzalis, John Mylopoulos, and Christos Kalloniatis (Eds.). Springer International Publishing, Cham, 37–52.

[3] Siemens AG. 2022. *WinCC V7 TLS Certificate Guide.* Accessed: 2024-08-27.

[4] Wael Alsabbagh and Peter Langendöerfer. 2022. A New Injection Threat on S7-1500 PLCs - Disrupting the Physical Process Offline. *IEEE Open Journal of the Industrial Electronics Society* 3 (2022), 146–162. doi:10.1109/OJIES.2022.3151528

[5] Wael Alsabbagh and Peter Langendörfer. 2022. No Need to be Online to Attack - Exploiting S7-1500 PLCs by Time-Of-Day Block. In *2022 XXVIII International Conference on Information, Communication and Automation Technologies (ICAT)*. IEEE, 1–8. doi:10.1109/ICAT54566.2022.9811147

[6] Wael Alsabbagh and Peter Langendörfer. 2023. You Are What You Attack: Breaking the Cryptographically Protected S7 Protocol. In *2023 IEEE 19th International Conference on Factory Communication Systems (WFCS)*. IEEE, 1–8. doi:10.1109/WFCS57264.2023.10144251

[7] Kevin S. Anderson, Clifford W. Hansen, William F. Holmgren, Adam R. Jensen, Mark A. Mikofski, and Anton Driesse. 2023. pvlib python: 2023 project update. *Journal of Open Source Software* 8, 92 (2023), 5994. doi:10.21105/joss.05994

[8] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and Don'ts of Machine Learning in Computer Security. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 3971–3988. https://www.usenix.org/conference/usenixsecurity22/presentation/arp

[9] Eli Biham, Sara Bitan, Aviad Carmel, Alon Dankner, Uriel Malin, and Avishai Wool. 2019. Rogue 7: Rogue Engineering-Station attacks on S7 Simatic PLCs. *Blackhat Conference USA* (2019).

[10] Philippe Biondi, Pierre Lalet, Gabriel Potter, Guillaume Valadon, and Nils Weiss. 2024. *Scapy: the Python-based interactive packet manipulation program & library.* https://github.com/secdev/scapy

[11] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[12] Defense Use Case. 2016. Analysis of the cyber attack on the Ukrainian power grid. *Electricity Information Sharing and Analysis Center* (2016).

[13] Colin Finck and Tom Dohrmann. 2023. A Decade After Stuxnet: How Siemens S7 is Still an Attacker's Heaven. *Blackhat Conference Europe* (2023).

[14] Asem Ghaleb, Sami Zhioua, and Ahmad Almulhem. 2018. On PLC network security. *International Journal of Critical Infrastructure Protection* 22 (2018), 62–69. doi:10.1016/j.ijcip.2018.05.004

[15] Sabine Haas, Uwe Krien, Birgit Schachler, Stickler Bot, Velibor Zeli, Florian Maurer, Kumar Shivam, Francesco Witte, Sasan Jacob Rasti, Seth, and Stephen Bosch. 2024. *wind-python/windpowerlib: Update release.* doi:10.5281/zenodo.10685057

[16] D.L. Hall and J. Llinas. 1997. An introduction to multisensor data fusion. *Proc. IEEE* 85, 1 (1997), 6–23. doi:10.1109/5.554205

[17] Eman Hammad, Mellitus Ezeme, and Abdallah Farraj. 2019. Implementation and development of an offline co-simulation testbed for studies of power systems cyber security and control verification. *International Journal of Electrical Power & Energy Systems* 104 (2019), 817–826. doi:10.1016/j.ijepes.2018.07.058

[18] Henry Hui and Kieran McLaughlin. 2018. Investigating Current PLC Security Issues Regarding Siemens S7 Communications and TIA Portal. *ICS-CSR'18.* doi:10.14236/ewic/ICS2018.8

[19] Henry Hui, Kieran McLaughlin, and Sakir Sezer. 2021. Vulnerability analysis of S7 PLCs: Manipulating the security mechanism. *International Journal of Critical Infrastructure Protection* 35 (2021), 100470. doi:10.1016/j.ijcip.2021.100470

[20] Moses Ike, Kandy Phan, Keaton Sadoski, Romuald Valme, and Wenke Lee. 2023. Scaphy: Detecting Modern ICS Attacks by Correlating Behaviors in SCADA and PHYsical. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 20–37. doi:10.1109/SP46215.2023.10179411

[21] Jehn-Ruey Jiang and Yan-Ting Chen. 2022. Industrial Control System Anomaly Detection and Classification Based on Network Traffic. *IEEE Access* 10 (2022), 41874–41888. doi:10.1109/ACCESS.2022.3167814

[22] Nandha Kumar Kandasamy, Sarad Venugopalan, Tin Kit Wong, and Nicholas Junming Leu. 2022. An electric power digital twin for cyber security testing, research and education. *Computers and Electrical Engineering* 101 (2022), 108061. doi:10.1016/j.compeleceng.2022.108061

[23] Kaspersky. 2024. Threat Landscape for Industrial Automation Systems: Q1 2024. Accessed: 2024-08-02.

[24] Nicolai Kellerer, Gustavo Sánchez, Hermenegildo Alberto, Veit Hagenmeyer, and Ghada Elbez. 2025. *S7 Data Modification Attack Scripts and IDS.* https://github.com/nbke/s7-attacks

[25] Nicolai Kellerer, Gustavo Sánchez Collado, Hermenegildo Alberto, Veit Hagenmeyer, and Ghada Elbez. 2025. *S7 Data Modification Attacks using an Industrial Control System Testbed.* doi:10.5281/zenodo.15373938

[26] Ralph Langner. 2011. Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security & Privacy* (2011).

[27] Cheng Lei, Li Donghong, and Ma Liang. 2017. The spear to break the security wall of S7CommPlus. *Blackhat Conference EU* (2017).

[28] Bonface Ngoko, Hideharu Sugihara, and Tsuyoshi Funaki. 2019. A Temperature Dependent Power Flow Model Considering Overhead Transmission Line Conductor Thermal Inertia Characteristics. In *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. IEEE, 1–6. doi:10.1109/EEEIC.2019.8783234

[29] National Renewable Energy Laboratory (NREL). 2024. *NSRDB: National Solar Radiation Database.* https://nsrdb.nrel.gov/

[30] Kyuchan Park, Bohyun Ahn, Jinsan Kim, Dongjun Won, Youngtae Noh, Jinchun Choi, and Taesic Kim. 2021. An Advanced Persistent Threat (APT)-Style Cyberattack Testbed for Distributed Energy Resources (DER). In *2021 IEEE Design Methodologies Conference (DMC)*. IEEE, 1–5. doi:10.1109/DMC51747.2021.9529953

[31] Danila Parnishchev. 2018. *s7scan: The tool for enumerating Siemens S7 PLCs through TCP/IP or LLC network.* https://github.com/klsecservices/s7scan

[32] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, Félix J. García Clemente, Cristian Cadenas Sarmiento, Carlos Javier Del Canto Masa, and Rubén Méndez Nistal. 2019. On the Generation of Anomaly Detection Datasets in Industrial Control Systems. *IEEE Access* 7 (2019), 177460–177473. doi:10.1109/ACCESS.2019.2958284

[33] Stephan Ruhe, Steffen Nicolai, and Peter Bretschneider. 2018. Modelling and simulation of electrical phenomena in a real time test bench. In *2018 53rd International Universities Power Engineering Conference (UPEC)*. IEEE, 1–6. doi:10.1109/UPEC.2018.8542092

[34] Abhijeet Sahu, Zeyu Mao, Patrick Wlazlo, Hao Huang, Katherine Davis, Ana Goulart, and Saman Zonouz. 2021. Multi-Source Multi-Domain Data Fusion for Cyberattack Detection in Power Systems. *IEEE Access* 9 (2021), 119118–119138. doi:10.1109/ACCESS.2021.3106873

[35] Gustavo Sánchez, Ghada Elbez, and Veit Hagenmeyer. 2024. Attacking Learning-based Models in Smart Grids: Current Challenges and New Frontiers. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems* (Singapore, Singapore) *(e-Energy '24)*. Association for Computing Machinery, New York, NY, USA, 589–595. doi:10.1145/3632775.3661984

[36] G. P. H. Sandaruwan, P. S. Ranaweera, and Vladimir A. Oleshchuk. 2013. PLC security and critical infrastructure protection. In *2013 IEEE 8th International Conference on Industrial and Information Systems*. IEEE, 81–85. doi:10.1109/ICIInfS.2013.6731959

[37] Yatish Sekaran, Tanmoy Debnath, Taeesh Azal Assadi, Sai Dileep Suvvari, and Shubh Oswal. 2023. Using Machine Learning to detect abnormalities on Modbus/TCP Networks. In *Proceedings of the 4th International Conference on Information Management & Machine Intelligence* (Jaipur, India) *(ICIMMI '22)*. Association for Computing Machinery, New York, NY, USA, Article 56, 6 pages. doi:10.1145/3590837.3590893

[38] Siemens. 2024. *Industry Bilddatenbank*. https://www.automation.siemens.com/bilddb/

[39] Ilias Siniosoglou, Panagiotis Radoglou-Grammatikis, Georgios Efstathopoulos, Panagiotis Fouliras, and Panagiotis Sarigiannidis. 2021. A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1137–1151. doi:10.1109/TNSM.2021.3078381

[40] Joe Slowik. 2018. Anatomy of an attack: Detecting and defeating crashoverride. In *VB'2018*.

[41] Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. 2018. Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems* 33, 6 (2018), 6510–6521. doi:10.1109/TPWRS.2018.2829021

A Encoding and Decoding Data

Encoding transforms data types like integers, floats, and booleans into byte sequences that conform to the S7 protocol's specifications, which dictate byte order and data type size. Decoding reverses this process, allowing us to interpret these byte sequences. We implement these processes through Scapy's custom packet definitions in the "s7.py" module, ensuring each packet type is equipped with fields tailored to the data it carries.

For numeric and boolean data, the S7 protocol prescribes specific byte representations, requiring precise byte order and data size, such as 16-bit integers or 32-bit floating points. Strings and other arbitrary data are encoded using length-specified fields to ensure the decoder accurately defines the data length.

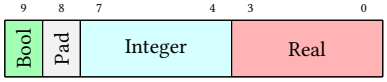


Figure 5: Datablock structure of the S7 protocol including *boolean, integer* and *real* fields.

For example, encoding a boolean value in the S7 protocol transforms the logical 'true' or 'false' into the byte sequence '0x01' or '0x00', respectively. The byte for the boolean field is always followed by a padding byte, as shown in 5. For a 32-bit integer, such as 256, the encoding process would yield a byte sequence like '0x00, 0x00, 0x01, 0x00' when employing a big-endian format. A floating point value (aka real) is encoded using the IEEE 754 format and uses 4 bytes of space. The decoding function then takes these byte sequences and reconstructs the original integer or boolean values.

These functions ensure that data not only remains consistent and accurate across different network segments but also adheres to the stringent performance and safety standards required in industrial settings. Our system also integrates error checking and validation mechanisms to ensure data adherence to protocol specifications.

The S7 protocol supports multiplexing multiple connection channels within a single TCP connection. The remote ID is used to identify a channel and must be unique, because it can only be used for a single connection. Furthermore, it must be the same on both ends of the connection, e.g. the *BSEND* block in the PCS must use the RID 4 and the *BRECV* block in the PLC must also be defined with the RID 4. The RID 3 is used in our testbed for the inverse direction to establish a bidirectional connection.

B Investigating Unsafe States Within Power Plants

In this section, we provide an analysis of the potential unsafe states that can occur in our power plant testbed as a result of attacks.

B.1 Manipulation of Control Commands

Modifying control data sent to PLCs from a master PCS could wrongly alter the operation of photovoltaic panels, wind turbines, or battery storage systems, risking operational disruptions. An attacker with access to the ICS network could use a MITM attack to intercept packets between SCADA and field devices and send commands to turn on/off the power plants.

$$\text{Command}_{\text{real}} = \text{Command}_{\text{intended}} + \Delta\text{Command} \tag{2}$$

where ΔCommand represents the unauthorized modification.

Impact: Cause grid instability, damage components, and depending on the SoC of the battery, a possible complete blackout.

B.2 Manipulation of Monitoring Data

The modification of monitoring data by an attacker can lead to incorrect control decisions. An attacker can reduce the production of a power plant below the operational levels of the grid. The power flow algorithm running on the control center in this case would respond by incorrectly attempting to compensate the shortfall by ramping up other power plants production.

Incorrect energy production data can lead to improper grid balancing decisions, risking grid instability.

$$P_{\text{grid}} = \sum_{i=t_0}^{T} (P_{\text{generated}} + \Delta P) \tag{3}$$

where ΔP represents the tampered increment or decrement in reported power output.

Impact: Cause grid instability, operation inefficiency and frequency fluctuation with cascading effect that can damage grid components and blackout.

B.3 Compromise of Remote Connections to Control Center

The communication between the remote control center and field devices are critical for monitoring and controlling the status of the devices in the grid. Any disruption or delay in the communication can compromise the safety operations of the grid. An unauthorized access to the ICS network by an attacker can compromise the availability of the grid components by launching a volumetric DOS attack. Remote communications in ICS are time sensitive, and an attacker could target the clock synchronization of the devices causing a voltage/frequency anomaly due to delayed commands sent to the PLCs.

Impact: Safety system compromised, control loop disruption with a cascading impact.

Unsafe State	(S)	(L)	(D)	(C)	Explanation and Risk Score (RS)
Manipulation of Control Commands	5	3	4	3	Potential for complete blackout and equipment damage. Requires access to ICS network. Could persist until detection and intervention. Can be mitigated with existing safety PLCs but delayed response. **RS:** $5 \times 3 \times 4 \times (6-3) = 180$
Manipulation of Monitoring Data	4	4	3	2	Grid instability and component damage. Moderately likely with sufficient network access. Could be mitigated once inconsistencies are detected. Detection systems could help mitigate quickly. **RS:** $4 \times 4 \times 3 \times (6-2) = 192$
Compromise of Remote Connections to Control Center	4	5	4	4	Safety system compromised, cascading impact. High likelihood due to vulnerabilities in remote access. Persistent until communication is restored. Difficult to mitigate during an active attack. **RS:** $4 \times 5 \times 4 \times (6-4) = 160$
Battery Storage Mismanagement	5	3	3	2	Thermal runaway, fire hazards, grid imbalance. Requires detailed knowledge of battery systems. Could be detected via abnormal SoC patterns. Safety PLCs can intervene. **RS:** $5 \times 3 \times 3 \times (6-2) = 180$
Compromise of Maintenance Capabilities	5	4	5	5	Potential for persistent backdoors and incorrect operation. Moderate to high likelihood depending on maintenance access. Can persist until firmware is restored. Extremely difficult to mitigate without full reconfiguration. **RS:** $5 \times 4 \times 5 \times (6-5) = 100$

Table 4: Scoring of Unsafe States with respective explanations.

B.4 Battery Storage Mismanagement

Battery storage systems play a critical role in modern energy grids, providing support for load balancing, peak shaving, and backup power during outages. However, these systems are highly sensitive to accurate control and management, particularly regarding their charge and discharge cycles. Successful data manipulation attacks targeting these cycles can have serious consequences, including reduced battery lifespan, safety risks, and overall system instability, even though the safety PLC can prevent it.

$$SOC_{\text{new}} = SOC_{\text{old}} + \int_{t=t_{init}}^{T} I(t)\,dt - \Delta SOC \tag{4}$$

where ΔSOC indicates falsified SoC changes.

Impact: Overcharging or deep discharging, increased cycle count, thermal runaway and fire hazards, and imbalanced on cells.

B.5 Compromise Maintenance Capabilities

Maintenance sessions pose a serious risk to the security and stability of power grid operations. During maintenance, attackers can introduce malicious firmware into critical components such as inverters, controllers, or PLCs, tampering with these devices to alter voltage or frequency regulation and potentially embedding backdoors for persistent access. They could also modify the firmware of critical safety components leading to incorrect operation of power plants and destabilizing the grid. Additionally, unauthorized changes to inverter configurations and control parameters could result in improper power output, grid instability, or failures in critical grid support functions, risking significant damage to the infrastructure.

$$V_{\text{actual}} = V_{\text{nominal}} + \Delta V \tag{5}$$

$$f_{\text{actual}} = f_{\text{nominal}} + \Delta f \tag{6}$$

where ΔV and Δf are unauthorized modifications in voltage and frequency. It is worth noticing, that an attacker must investigate the permissible threshold of Δ that can be introduced without activating fault detection mechanisms, particularly for frequency protection systems, which exhibit high sensitivity to such changes.

Impact: Improper voltage regulation, frequency instability, cascading effects that can lead to component damage and blackout.

B.6 Framework for Scoring Unsafe States

To taxonomize the unsafe states resulting from the described attacks, we propose a framework that assigns a score to each unsafe state based on four key metrics: Severity, Likelihood, Duration, and Controllability. The combination of these metrics will provide an overall risk score, which allows categorizing attacks and determining their potential impact on grid stability and safety.

- Severity (S): The potential impact of an unsafe state on the power plant and grid stability. This metric ranges from 1 (low severity) to 5 (high severity), representing the extent of the damage, ranging from operational inefficiencies to catastrophic blackouts and equipment damage.
- Likelihood (L): The probability of an unsafe state occurring as a result of a successful attack. This metric ranges from 1 (unlikely) to 5 (highly likely), taking into account the complexity of the attack and the security controls in place.
- Duration (D): The duration for which an unsafe state can persist. It ranges from 1 (momentary state) to 5 (long-term state). Longer durations generally imply greater difficulty in recovery and increased risk to the system.
- Controllability (C): The ability of the grid operators or control systems to mitigate or respond to an unsafe state. This metric ranges from 1 (easily controllable) to 5 (difficult to control), considering factors such as response time, available countermeasures, and automation capabilities.

Scoring Formula: The overall risk score (RS) for each unsafe state can be calculated as follows:

$$RS = S \times L \times D \times (6 - C) \tag{7}$$

The controllability factor is subtracted from 6 to ensure that higher controllability reduces the risk score.

The described unsafe states are scored using this framework and presented in Table 4.

B.7 Discussion and Analysis

The proposed framework provides a structured approach to evaluating unsafe states induced by S7 attacks. Higher risk scores indicate scenarios that require more immediate attention in terms of mitigation and monitoring. For instance, the manipulation of monitoring data and battery mismanagement present high risks due to their potential to destabilize the grid and the difficulty in controlling their

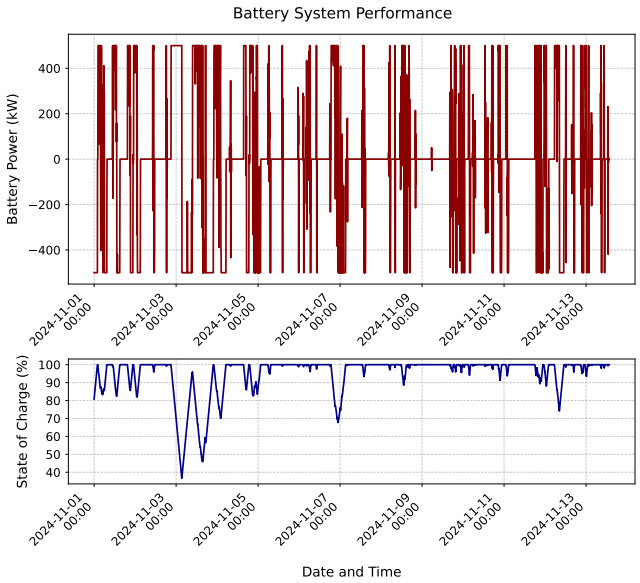


Figure 6: Battery storage model.

effects. On the other hand, compromised maintenance capabilities, while severe, may have a lower risk score due to their controllability through firmware restoration.

C Component Models

C.1 Photovoltaic Power Plant

The solar PV system model integrates real weather data through the National Solar Radiation Database (NSRDB) [29]. Detailed modeling is performed using the *pvlib* library [7], which includes advanced solar position calculations and Plane of Array (POA) irradiance modeling. The PlantSim [24] simulator uses the validated models of the `First_Solar__Inc__FS_4117_3` panel and `TMEIC__PVL_L1833GRM` inverter. The model accounts for temperature effects on cell performance, ensuring accurate simulation under varying environmental conditions. Additionally, comprehensive inverter modeling is included to provide a realistic estimate of power output. The system supports multi-string configurations, offering customizable parameters to cater to diverse deployment scenarios.

C.2 Wind Power Plant

The wind power system is modeled using the *windpowerlib* [15], enabling advanced simulations of wind farms. The model supports heterogeneous turbine fleets, allowing the inclusion of various turbine types within a single simulation. The PlantSim [24] simulator within our testbed uses a fleet of six custom wind turbines and three Enercon E-126/4200. Environmental parameters are integrated comprehensively into the simulation, including wind speed at different heights, air density calculations, temperature variations, and surface roughness effects. These considerations ensure precise modeling of power output based on real-time weather conditions, enabling robust and accurate simulations for wind energy generation (see figure 2).

C.3 Battery Storage

The battery storage model incorporates a comprehensive framework for simulating energy storage and management. It includes SoC management to ensure efficient utilization of the battery system (see figure 6). The model enforces power flow constraints and charge/discharge rate limitations, accounting for safety and operational boundaries. Temperature effects are also integrated into the simulation, along with detailed current and voltage calculations. Safety features, such as minimum and maximum SoC limits, are implemented to prevent system failures and extend the battery lifespan.

D Signal Table and Syslog rules

Name	Lower Threshold	Upper Threshold	Type
State of Charge	20.0 %	90.0 %	Monitor
Voltage	390.0 V	410.0 V	Monitor
Current	−1500.0 A	1500.0 A	Monitor
Target Charge Power	350 W	50 kW	Control
Actual Charge Power	350 W	50 kW	Monitor
Temperature	20.0 °C	30.0 °C	Monitor
On Off	N/A	N/A	Control

Table 5: Syslog rules on PLC 1512 (Battery).