Real-Time Testbed for Studying Cyberattacks and Defense in DER-integrated Smart Inverter Systems

M. Maliha*, A. Oluyomi†§, M. Booge‡ S. Bhattacharjee*, N Braasch‡, P. Gomez‡, and S. K. Das†,
* Computer Science Department, Western Michigan University, Kalamazoo, USA
† Computer Science Department, Missouri University of Science and Technology, Rolla, USA
‡ Electrical and Computer Engineering Department, Western Michigan University, Kalamazoo, USA
{maisha.maliha, shameek.bhattacharjee}@wmich.edu; {aoonzb, sdas}@mst.edu, pablo.gomez@wmich.edu

***Abstract*—In this paper, we propose a Hardware-in-the-Loop (HIL) simulation testbed suitable for the implementation and testing of realistic cyberattacks on grid-tied smart inverter systems integrated with Distributed Energy Resources (DER) that use the Distributed Network Protocol-3 (DNP3) protocol for communications between grid components. Specifically, our testbed combines a Real-Time Digital Simulator (RTDS) NovaCor device, outfitted with GNETx2 network interface cards, a grid-tied DER topology implemented via the RTDS software package RSCAD, and a custom virtual network that emulates a man in the middle attacker. The Man-in-the-Middle (MITM) attacker captures DNP3 traffic and falsifies telemetry data in DNP3 packets to trigger unwarranted commands from a DNP3 controller that exploit smart inverter grid support functions. We choose DNP3 and implement grid support functions according to the IEEE Std. 1547-2018 mandated for the interconnection and interoperability of DER power systems with associated power components.**

Furthermore, we develop a protocol payload agnostic attack detection framework that leverages the round-trip time (RTT) anomalies between DNP3 requests and responses and can detect the presence of attacks without having to analyze the payload's contents, while balancing trade-offs between false alarm counts, missed detections, and time to detection. To facilitate further research, we publicly release benign and attack network traffic exchanged between various sensors, controllers, and actuators in our grid-tied inverter testbed.

***Index Terms*—DNP3, DER, Smart Inverters, Cybersecurity, Smart Grid Communications**

I. Introduction

Smart grids support communication of various signals between sensors, controllers, and actuators in addition to power flows. This communication enables supervisory control and data acquisition (SCADA) control systems to effectively monitor and manage DERs (e.g., wind turbines, photovoltaic systems, battery storage units) using control and communication components such as grid-tied smart inverters, remote terminal units (RTUs) and intelligent electronic devices (IEDs). Smart inverters specifically facilitate the seamless integration and management of DERs [1]. In a smart inverter supported grid, real-time telemetry data (e.g. voltage, current, and tripping signals) are typically sent from field-level measurement devices (e.g., IEDs, RTUs) to the SCADA controller. The SCADA controller processes such telemetry data and issues control commands to the smart inverter. The smart inverter participates in the reactive power control and stabilizes an unbalanced grid. Hence, a change in the telemetry data can change the state of the grid to affect the stability of the grid and power efficiency [2]. The communication between RTUs, SCADA controllers, and smart inverters is supported by network communication protocols.

Motivation Communication protocols such as Distributed Network Protocol-3 (DNP3) have been recently mandated by the IEEE Std. 1547-2018 as one of the protocols for grid tied smart inverter communications [3]. DNP3 clients/outstations (field-level devices) collect and send telemetry data to the DNP3 master station co-located in the SCADA system. Naturally, cyber attackers would seek to exploit DNP3 vulnerabilities to breach the communication network to impact the grid-tied smart inverter. While there are implementations on generic DNP3 attacks, currently there is no detailed documented research on the realization of Hardware in the Loop (HIL) testbeds for cybersecurity research in DER-integrated smart inverter systems that use DNP3 [7].

Real-time simulation is a high-fidelity approach that produces realistic datasets for network security assessment and to test intrusion detection methods. There is a lack of attack and benign datasets using real-time simulation of DNP3 communication in the context of DER-integrated grids. Attacking a real grid is not feasible because of the cost and dangers associated. Therefore, a real-time testbed is the closest approximation that mimics the behavior of a real grid in a controlled, safe environment. This paper aims to fill this gap.

While previous work on DNP3 testbeds and datasets [12] is available, they commonly provide traditional cyber network emulation and implement traditional attacks without an attempt to emulate a DER-integrated smart inverter system, real time HIL, or attacks that target a certain smart inverter functionality. Since all the aforementioned factors affect network traffic patterns and DNP3 payloads, any intrusion detection model trained on such datasets will lack the specificity required to capture unique operational behaviors inherent to DER-integrated grids, as observed by [13].

Contributions This paper addresses the aforementioned needs by providing a HIL testbed design to simulate both cyber and physical components of grid-linked smart inverters, and an attacker in the loop that intercepts communication between

This material is based upon work supported by the U.S. Department of Energy's Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Award no. DE-CR0000029. Additional support has been provided by Factory Mutual (FM) Insurance Co. through contract no. 7036430.

DNP3 outstations and DNP3 master station. Specifically, we make the following salient contributions:

(i) We developed an RSCAD model that emulates a DER-connected balanced grid topology containing a wind turbine, a smart inverter, two transformers, and a Thevenin equivalent grid connected by a point of common coupling (PCC).

(ii) We integrated this model with external devices to establish a realistic HIL environment. Specifically, we used the RTDS NovaCor platform equipped with GNETx2 network interface cards to facilitate real-time communication between the simulated grid components and the SCADA controllers. Furthermore, we used a network simulator 3 (NS-3) to create a virtual network to emulate a realistic communication channel that implements a Man-in-the-Middle (MITM) attack between DNP3 master and outstations.

(iii) We propose a proof-of-concept false data injection attack that aims to exploit the fault ride-through functionality as defined in IEEE Std. 1547-2018. Specifically, we devise multiplicative attack strategy on the sinusoidal instantaneous voltage at the PCC that emulates a fake three phase to ground fault to induce unwarranted actions from the controller.

(iv) We collected and publicly released the benign and attacked DNP3 datasets using multiple polling intervals, scan commands, and time durations that capture a diverse range of communication patterns which effect the performance of data-driven attack detection methods.

(v) We developed a detection method based on timing patterns in the DNP3 network traffic to identify anomalies without inspecting the packet payloads.

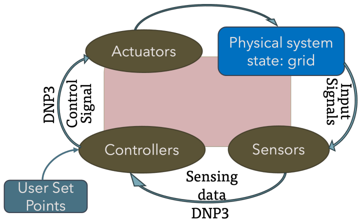


Fig. 1: Overview of CPS operation.

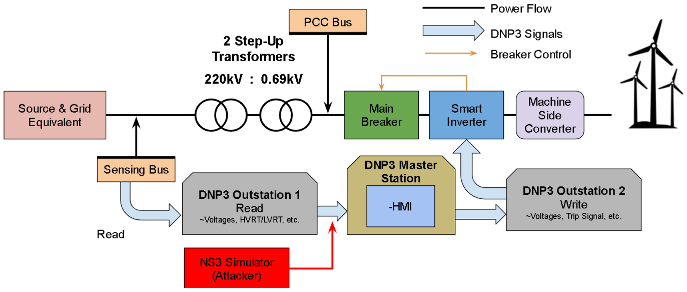


Fig. 2: Functional topology of a DER-interconnected smart inverter system

II. Background and Preliminaries

Here we discuss DNP3's operations and vulnerabilities.

DNP3: This protocol supports communication between a DNP3 master and outstation devices like Remote Terminal Units (RTUs) [4] that send telemetry sensing data and receive commands from the DNP3 master. Despite its functionality and widespread adoption, DNP3 is an application layer protocol that runs over TCP/IP/Ethernet and inherits vulnerabilities such as MITM attacks.

Although the DNP3 protocol popularly does not support encryption, recent versions such as DNP3 Secure Authentication Version 5 (SAv5) [5] provisions for symmetric encryption, key distribution, and challenge-response mechanisms to add a layer of security using the following sub-protocols: (1) Session Key Update, (2) Critical Application Service Data Units (ASDU) Authentication Protocol, and (3) Update Key Change Protocol (in symmetric and asymmetric modes).

DNP3 Vulnerabilities to Data Falsification Attacks: One significant threat facing smart grids is the risk of data falsification attacks, which can severely affect critical infrastructure operations. In smart grid environments, due to stringent real-time operational requirements, encryption is often not adopted/preferred due to latency and processing overheads that conflict with the performance constraints of grid operations [6]. Even if encryption is adopted, it does not give complete security. For example, even with the advancements in DNP3SaV5, which provisions for encryption, a vulnerability still exists that keeps the door open for data falsification by a MITM attacker. Specifically, the "Update Key Change (UKC) Protocol" is responsible for distributing update keys to outstations. In its symmetric mode, the Update Key Change message lacks an explicit identification of the intended outstation. An attacker can impersonate one outstation and intercept update keys meant for another, thus compromising the UKC protocol's integrity. By manipulating this flaw, an attacker can force one outstation to accept a fake key. Then, by means of an ARP spoof (or similar) to intercept this data, it uses the fake key for unauthorized decryption of DNP3 packets [5]. Once this is done, the attacker can falsify such data and then re-send it to the DNP3 master. Consequently, the DNP3 master would issue inaccurate commands. Finally, the most successful attacks are insider attacks or social-engineering exploits that gain privileged access to bypass encryption controls and then launch false data. As such, the true essence of cyber-physical security, i.e., what happens to the operations and how we can develop detection methods and prevent attacks when the first line of defense (encryption) is bypassed, is the main concern. However, before detection methods can be developed, we need realistic testbeds and datasets that emulate real DNP3 traffic exchange for DER-integrated smart inverter systems.

III. Testbed Design and Implementation

In this section, we describe the design of a grid-tied smart inverter topology, followed by our HIL simulation testbed design that emulates DNP3 in the grid. A logical grid topology of our testbed is summarized in Figure 2, while the physical network topology is given in Fig. 3.

A. Grid-tied Smart Inverter DER Model

Here we discuss the electrical, cyber, and communication aspects of a typical grid-tied smart inverter system.

1) Electrical Model: The main components are below:
(i) Wind farm and conversion to regulated AC, including smart inverter: The wind farm generates an unregulated alternating current (AC) from the varying rotational energy of a generator depending of wind conditions. Then an AC to AC converter regulates the frequency, voltage, and phase so that the AC power is suitable for the grid. This converter includes a Ac-Dc rectifier, a DC link, and a DC-AC smart inverter.
(ii) Main DER Breaker: This actuator interrupts or restores the DER connection to the grid. DER Disconnection usually happens when the terminal node senses abnormal voltage, current or frequency measurements, signifying a fault or malfunction in the system.
(iii) Point of Common Coupling (PCC): It represents the point where the inverter's AC voltage output couples to the grid.
(iv) Transformers: Two transformers step up the voltage between the inverter and the grid. Transformer 1 steps up from 0.69 kV to 35 kV, and transformer 2 further steps it up to 220 kV, which is the transmission voltage level of the grid.
(v) Grid Equivalent: This is modeled as a Thevenin equivalent and emulates the impedance and voltage of the rest of the grid.

2) Cyber and Communication Model: The various cyber components of the DER-integrated smart inverter system and the communication between them are described below.

Cyber Network Devices: (a) Outstation: This is a networked device that sends/receives DNP3 messages over TCP/IP, enabling real-time data exchange between grid components and the SCADA control center. The outstation typically sends telemetry sensing data to a DNP3 Master Device. (b) Master Station: This holds the algorithms for real-time monitoring and control. It receives and processes the telemetry data received from one or more outstations or sensors. Based on the analyzed grid conditions from the data received, the master station issues control commands via the DNP3 protocol back to the designated outstations. These outstations then translate and forward these commands to actuators (e.g., smart inverts) to execute specific operational adjustments.

Data Communication The testbed carries the following data between DNP3 master and outstations: (i) Sensing Data: The DNP3 Outstation responds with sensing data to the DNP3 master upon request. In our testbed, the DNP3 outstation sends the *line to neutral instantaneous voltages of each phase.* (ii) Control Commands: Control commands are sent from the DNP3 master to the DNP3 Outstation to manage essential functions. Specifically, we implement the enabling/disabling Low Voltage Ride-Through (LVRT) control capabilities and adjusting reactive power setpoints.

B. Testbed Design

Here we discuss our proposed testbed and how it emulates the cyber-physical and communication aspects of a DER-integrated smart inverter. An overview of the testbed is shown in Figure 3. The hardware and their corresponding software components are defined below.

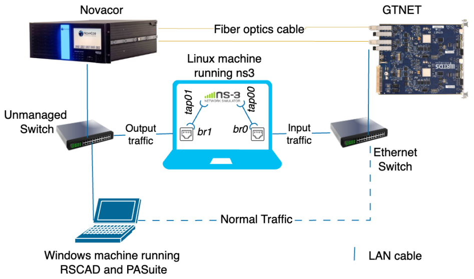


Fig. 3: Testbed Physical Topology: This includes the sensing and communication flows in the form of DNP3 packets. Parallel to the physical flow, a cyber communication flow monitors and controls the grid's operation.

RSCAD Software and RTDS is a real-time power systems modeling software for emulating a desired grid topology. We implemented the physical DER, inverter and grid topology of Fig. 2 using RSCAD software. An RSCAD representation of the physical grid model is called a "draft case". The draft case is exported from the RSCAD software (running on the Windows machine) to the NovaCor hardware.

NovaCor is the hardware that emulates power system models [8] to replicate the real-time response of a physical grid in a controlled environment, which can be captured from output sinks within the RSCAD model and emulate the networking layer via the GTNETx2 card. The RTDS hardware allows for interfacing with various network hardware devices as needed (via a network chassis) and protection and control devices to realistically simulate closed-loop cyber-physical systems.

The GTNETx2 Card is the hardware stack that packetizes and allows for the transmission of DNP3 data over IP and remote port connections. The GTNETx2 card is analogous to a Remote Terminal Unit (RTU), which is the digital communication gateway between the DER-integrated smart inverter model and the DNP3 master in the SCADA system. In addition to the above hardware, there are two essential components within the RSCAD software: (i) the Protection and Automation Suite (PASuite), and (ii) the DNP3 Database.

The PASuite is used to virtualize a DNP3 master station, facilitating execution of control commands and gathering telemetry data. This part of RSCAD communicates with the physical GTNETx2 card. With this configuration, the PASuite can send control commands and display telemetry data including, but not limited to, analog and binary input/output data with timestamps from the outstations.

DNP3 Database is an organized collection of analog and binary inputs/outputs sent to/from a DNP3 outstation inside the RSCAD model. The database represents signals for monitoring and control. Specifically, we collect the voltage and current signals on each phase (See sec IV-A) into the DNP3 database to facilitate the monitoring and falsification of signals.

Wireshark is the network protocol analyzer we use to capture both benign and attacked network traffic during the MITM attack on our DER operations. The Wireshark is installed on the Windows machine and captures the DNP3 network traffic between the GTNETx2 cards and the RSCAD model.

Linux Machine with NS-3 emulates a MITM attacker in our testbed. Additionally, it is necessary to host the DNP3 master/outstations and the attacker on different machines to simulate realistic attacks to the grid.

Ethernet Switches & Cables are used to connect the NovaCor, GTNETx2 cards, and the Linux and Windows machines into one cyber-physical network with the attacker in the loop. There are two Ethernet switches in the testbed. The first switch routes traffic between the GTNETx2 (emulating outstation hardware) and the Linux (attacker) machine, while the unmanaged switch routes traffic from the Linux machine (attacker) to the Windows machine (emulating the DNP3 master within the PASuite) and the NovaCor, ensuring real-time communication throughout the simulation process. In Fig. 3, the connection of an unmanaged switch to the Windows machine allows the Wireshark packet analyzer to observe all network traffic traversing this switch, including attack data transmitted from the Linux machine. This configuration facilitates the RSCAD software in receiving signal transmissions and also enables Wireshark to capture the network traffic.

Integration of NS-3 for Network and Attack Simulations We installed NS-3 software, Python (for running NS-3 scripts), GNU Compiler Collection (GCC), and CMake (for building NS-3) on a Linux-based desktop computer running Ubuntu 22.04.4 LTS. We created a virtual Carrier-Sense Multiple Access (CSMA) network within NS-3 to route DNP3 traffic through the RSCAD and various physical devices. This setup relies on NS-3's Tap and Bridge components to enable communication between the virtual attacker node within NS-3 and the external physical devices. Specifically, we configured two Tap devices (tap00 and tap01) and two Bridge devices (br0 and br1) on the Linux host running NS-3.

A Tap device is a virtual network kernel driver that can send/receive Layer 2 packets. We add a Tap on a Linux machine to create a virtual network adapter that interacts with user-space program (i.e., the NS-3). In Fig. 3, the tap00 captures the telemetry data and connects the physical network (where the DNP3 Outstation resides) to the NS-3 (shown in Fig 3). Inside NS-3, the data is manipulated by an attacker node for an MITM attack. The modified DNP3 data exits NS-3 through the tap01 and is forwarded to the DNP3 Master. Once Tap devices are created they need to be bridged with physical Ethernet interfaces via the bridge interfaces (br0 and br1). Bridge br0 connects tap00 to the rest of the testbed linking NS-3 with the DNP3 outstation. In contrast, the Bridge br1 connects tap01 to the rest of the testbed facilitating communication between the DNP3 master and the NS-3. This bridge interface ensures that data can be processed within the NS-3 virtual network and transmitted back and forth to the physical network, emulating a MITM attacker in a real system.

IV. Cybersecurity Attack Simulation

In this section we outline the objective of the attack and the methodology. The goal is to compromise the communication between outstations and the master station by altering telemetry data via a MITM to trigger an unwarranted control command from the DNP3 master.

We performed an MITM attack by ARP spoofing. The attacker node within NS-3 launches an ARP spoofing attack by linking its MAC address to the IP of the GTNETx2 card. This poisoned the ARP caches of outstation devices, causing them to send data to the attacker instead of the intended recipient. NS-3 was able to capture all network traffic regardless of the destination MAC address. Once we capture the traffic, we modify the packets before sending them to the master station. Next, we discuss the parameters modified to impair a certain inverter functionality. It should be noted that a malicious insider (e.g., compromised employee, operator or technician) within the SCADA network could carry out a similar MITM attack with their privileged position within the internal network without relying on external ARP spoofing techniques.

A. Targeted Telemetry Variables

We seek to examine the impact of falsifying one or more instantaneous voltage measurements on the PCC, which causes the smart inverter to react. However, the testbed has the ability to be further extended to study the impact of current measurement falsification too, which would exploit over and undercurrent protection. In general, the DNP3 communication between the outstation and the master carries continuous valued analog signals representing set-points to the smart inverter or sensed values from terminal nodes, and often digital or discrete signals such as trip signals, enabling FRT, or reactive power compensation. In our model, WF1VGA, WF1VGB, and WF1VGC are the sensed voltage measurements for phases A, B, and C of the terminal node, respectively.

B. Attack Strategy

After setting up the testbed, the simulation begins by executing the RSCAD model, which replicates the system dynamics. Simultaneously, NS-3 was launched to simulate the network containing the attacker node. The initial step involved capturing and disassembling the DNP3 packets to extract the analog signals transmitted within the communication framework. Once the signal values were identified, we launched the following attack model.

Attack Strength refers to the extent of data manipulation applied to each phase. In our work, we applied a multiplicative attack of strength 0.5, which reduces the instantaneous voltage magnitudes by half of their original value. The rationale behind a multiplicative attack as opposed to a additive/subtractive attack is that additive/deductive types applied on sinusoidal signals will show a DC bias in the final signal, which is not a typical reaction seen in transmission systems. Hence, it can be more easily identified by model driven/bad data detectors.

IEEE Std. 1547-2018 [3] mandates that DERs ride through certain high and low voltage conditions. The DER control system will delay ceasing grid supply until specific voltage (or frequency) thresholds and time durations defined by the standard are exceeded. If these conditions persist, the DER breaker may trip. The allowable ride-through time before DER tripping is inversely related to the severity of voltage deviation from its nominal value. This ride-through time decreases with the increase in the magnitude of the voltage deviation, as defined by the voltage-time characteristic curves in the standard shown in Figure 4. This figure summarizes the LVRT functionality in IEEE-Std. 1547-2018 that is being exploited by our attack. While IEEE Std. 1547-2018 allows configurable parameters within specified bounds, the default value of 50% in the deviation from the nominal voltage will cause the controls to trip the breaker within 160 ms according to the standard. This means that if the voltage at the PCC does not fall to 50% of the nominal voltage, the attack will be easier to detect by an operator. A multiplicative attack of strength 0.5 will cause the breaker to trip, so the attacker does not have to emulate the transient condition into the attack to feign a true fault. If the above multiplicative attack type is created, the operator has no easy way of distinguishing the attack from a real fault.

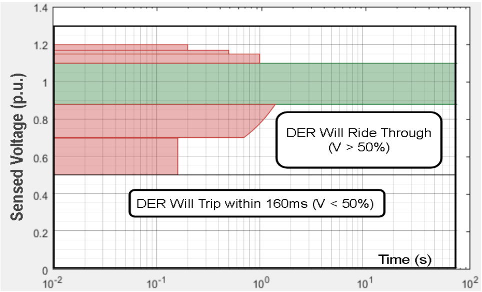


Fig. 4: IEEE Std. 1547-2018 Trip and Ride-Through Zones [3]

Potential Attack Impacts The impact of this attack may vary depending on the system configuration and grid dependence on DERs. In this testbed scenario, the attack results in the DER disconnecting from the grid and ceasing all support functions. This grid support includes reactive power compensation, which helps maintain voltage stability and improve the system's power factor. Without reactive power support from the DER, voltage regulation may worsen, leading to decreased power quality and increased transmission losses. These effects can result in financial impacts for utilities and consumers. Furthermore, without the DER contributing to voltage and frequency support, the grid's overall resilience may decrease, particularly in systems with high DER penetration.

C. Key Configuration Settings

The following are the three configuration settings that affect the dataset collected.

Polling Interval: The polling interval refers to how often the DNP3 Master sends data requests to the outstations. According to IEEE Std. 1547-2018 [3] it can be any value less than 30 s. In reality, it is up-to the administrator to decide the actual polling interval. However, the polling interval affects the timing and frequency of read and response packets. To account for this diversity, we collected data under two possible average polling intervals: 1 s and 10 s, as indicated in the released dataset.

Polling Command: The data patterns and the dataset depend on the polling command used. The dataset currently includes data collected using both *Integrity Polling*, which returns static data, and the *Scan* command in the PASuite, which returns event data. [10].

Duration of Study: The longitudinal duration of the data impacts the statistical patterns. For example, the accuracy of probability distributions for features from benign and attack traffic depends on the total study time. Longer durations provide more data for reliable patterns but may delay detection. To help researchers and engineers account for this trade-off and design a secure control mechanism, we publicly release our dataset over different durations. First, we have 16 hours of benign only data. Furthermore, we release segments of 1 hour, 20 min, 10 min, and 1 min durations of benign and attack data.

V. Network Traffic Time-based Detection Method

In this section we discuss the method for attack detection.

A. Dataset Description

We first ran the simulation without attacks and then executed the same simulation with the MITM attack. We captured the DNP3 packets using Wireshark under both benign and attack conditions. The captured packets include timestamp of packet, source and destination IP address, protocol, packet length, and payload. The dataset consists of raw .pcap and csv files, with different variations publicly available for research [9].

For our detection model design and evaluation, we used 16.31 hours of benign data and 20.7 minutes of attack data. We split the benign and attack datasets into training, cross-validation, and test sets. Training set includes only benign data from the first 8.2 hours. A long duration benign dataset helps build an accurate model of normal behavior.

The remaining benign data (of 8.11 hours) is split into two parts: 4.09 hours for cross-validation and 4.02 hours for test set. The attack data is also divided in 10.92 minutes for cross-validation and 9.86 minutes for testing. As a result, cross validation and test sets contain both benign and attack data. Test set contains 4.02 hours of benign data and 9.86 minutes of attacks, with a benign to attack ratio split of 25:1. The smaller attack portion reflects the real world scenario that attacks happen less often in real life compared to benign data. This imbalance between benign and attack traffic datasets allows us to avoid base rate fallacy during security performance evaluation of our detection method.

B. Extracting the Time Interval Data

We measure the time interval between each DNP3 *Read request* and its corresponding *Response* from .pcap files collected via Wireshark. These time intervals are computed using a Python script executed outside the RSCAD environment. Since the attacker needs to capture, read, and modify data, subtle delays will be introduced in the timing patterns, which we exploit for detection purposes.

Let $\{(t_k, I_k, \ell_k)\}$ be the ordered DNP3 packet data, where t_k is the timestamp of the k-th packet, I_k is the packet type (i.e., "Read" or "Response"), and $\ell_k \in \{0, 1\}$ is the label indicating benign ($\ell_k = 0$) or attack ($\ell_k = 1$).

For each packet k where $I_k = Read$ with the corresponding time index t_k, we find the immediate next packet where $I_k = Response$ (in chronological order) whose time is index $t'_k > t_k$. We term this as a matched read-response pair. For each such matched read response pair, we calculate the time interval between a read and response packet as $X_i = t'_k - t_k$, where i denotes the i-th matched Read–Response pair.

C. Modeling Benign Timing Behavior of the Packets

In this section, we discuss how we model benign behavior.

a) Window Size: We start by dividing the serial traffic into windows of a certain number of X_i intervals. Let w donate a window size, i.e., the number of matched read-response pairs consider in a single window. Let j denote the j-th window number over the entire traffic. The process of obtaining the optimal window length w is given in later section V-E.

b) Ratio of Means ($Q(j)$): For each window j, we calculate the Harmonic Mean (HM) and Arithmetic Mean (AM) of all time intervals $X_i(j)$ within that window. We define the invariant $Q(j)$ as the ratio of HM to AM. Under benign conditions, time intervals within a window remain relatively similar, resulting in a stable $Q(j)$ value. However, a MITM attack introduces additional communication delay, increasing the time intervals. This disrupts the stability of the timing pattern, causing a deviation in $Q(j)$. This invariant was first proposed by [11], where they presented a lightweight framework for the detecting false data injection (FDI) in smart meters for energy usage, but we adopt the idea and show it works when we use this ratio metric where the inputs are the round trip times instead of raw energy usage.

$$Q(j) = \frac{HM(j)}{AM(j)} = \frac{\left(\frac{1}{w}\sum_{i=1}^{w} X_i(j)^{-1}\right)^{-1}}{\frac{1}{w}\sum_{i=1}^{w} X_i(j)}. \tag{1}$$

D. Learning Thresholds with Quantile L_1 Regression

In this section, we describe the methodology for determining the attack detection threshold that characterizes benign behavior. We can view this problem as learning a best fit line on the observed $Q(j)$ where we only need to learn the bias term of a regression problem to get a time invariant threshold. However, we cannot use ordinary regression (that uses L2 norm loss) because we found that the regression errors are non-Gaussian (See the Q-Q plot (Fig. 5). Specifically, the Q-Q plot indicates that the tails are far away from Gaussian indicating non-malicious outlying points which need to be accounted for during the learning. Hence, we use a Quantile L_1 regression learner. Unlike L_2 norm which minimizes squared errors and is sensitive to large errors, L_1 regression minimizes absolute errors, making it more resistant to large non-malicious outliers in benign data, while adding quantile weights helps to achieve a balance between false alarm and missed detection [11].

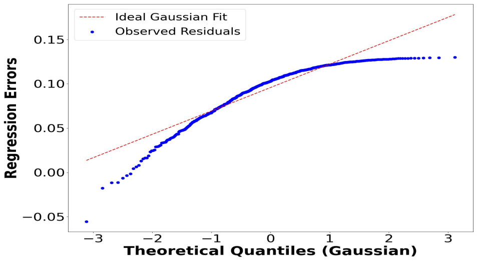


Fig. 5: Non-Gaussian Distribution of Regression Errors

The optimal threshold τ_{opt} is determined by minimizing the empirical risk over the observed ratio values $Q(j)$, defined in Eqn. 2 and 3. For each candidate threshold $\tau \in \tau^*$, a risk function $R_j(\tau)$ is evaluated per time window j. If $Q(j) < \tau$, a cost is applied with weight α_1; otherwise, a penalty is applied with weight α_2. The choice of weights allows for tuning the sensitivity of the threshold selection process to ensure a balance between false alarms & missed detections. The optimal threshold τ_{opt} is then selected that minimizes the total empirical risk, calculated as the sum of costs and penalties normalized by the number of observations η.

$$R_j(\tau) = \begin{cases} \alpha_1 \cdot |\tau - Q(j)|, & \text{if } Q(j) < \tau \\ \alpha_2 \cdot |Q(j) - \tau|, & \text{if } Q(j) \geq \tau \end{cases} \tag{2}$$

$$\tau_{\text{opt}} = \arg\min_{\tau \in \tau^*} \left[\frac{1}{\eta} \sum_{j=1}^{\eta} R_j(\tau) \right] \tag{3}$$

E. Obtaining the Optimal Hyperparameters

Now we discuss how we find the optimal parameters—α_1, α_2, and w. Our approach balances all three key considerations for real time detection, i.e. false alarms (FA), time to detection(TTD), and missed detection (MD). We formulate this as a preference based multi-objective optimization problem with the following priorities: minimize FA count first, then TTD, and finally MD rate. This prioritization reflects real-world needs. Minimizing the total number of FA is the most important since the prior probability of an actually attack is much lower (i.e., the base rate fallacy) [14]. The next priority is to minimize TTD since delayed detection lead to undesirable consequences in smart grid. MD rate is treated as a lowest-priority objective, since the system raises an alarm as soon as an attack is detected, even if some intervals are missed afterward. Based on this, we model the problem as a multi-objective optimization problem shown below.

Let f_{FA}, f_{TTD}, and f_{MD} denote the objective functions for false alarms count, time to detection, and missed detection rate, respectively. The optimization problem is formulated as:

$$\begin{aligned} &\min\{f_{FA}, \quad f_{TTD}, \quad f_{MD}\} \quad \text{subject to:} \\ &0 < \alpha_1 < 1, \quad \forall\, \alpha_1 \in \mathbb{R} \\ &0 < \alpha_2 < 1, \quad \forall\, \alpha_2 \in \mathbb{R} \\ &w > 1, \quad \forall\, w \in \mathbb{I} \end{aligned} \tag{4}$$

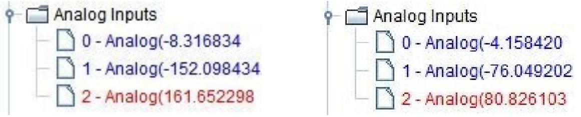
We use the classical ϵ-constraint method to solve the above problem. The ϵ-constraint method allows optimizing one primary objective while enforcing user defined preferred constraints on other objectives. We enforce the following constraints: keep the time to detection (f_{TTD}) below 2 seconds and the missed detection rate (f_{MD}) at or below 0.2. This ensures the system focuses on minimizing false alarms while maintaining fast and accurate detection. From the cross-validation set, we solve the MOOP, to get the following optimal parameters were $\alpha_1 = 0.999$, $\alpha_2 = 0.001$, and window size $w = 10$, resulting in an optimal detection threshold of $\tau_{\text{opt}} = 0.870$.

VI. Results and Discussion

In this section, we examine the results of implementing an MITM attack and we show the results of our detection model.

A. Modification Demo

Figure 6 illustrates the results of the executed attack, wherein it is apparent that the three analog (instantaneous) input signals have been halved with an attack strength of 0.5.


Analog Inputs
0 - Analog(-8.316834
1 - Analog(-152.098434
2 - Analog(161.652298

(a) Pre Attack

Analog Inputs
0 - Analog(-4.158420
1 - Analog(-76.049202
2 - Analog(80.826103

(b) Post Attack

Fig. 6: Analog input signals from outstations in PASuite

B. Evaluation of the Detection Mechanism

The optimal hyperparameters were applied to the detection model. The model was subsequently evaluated on the test set. A visualization of the entire test set is shown in Figure 7. The results are given in Table I.

TABLE I: Performance of Detection Mechanism

FA Count	TTD (seconds)	MD Rate
0	0.99906	0.1

The zero FA count during the test confirms that the effectiveness of our model along with the ϵ-constraint method. It also shows that normal operational behavior is not misclassified as an attack, demonstrating the operational reliability of our model. Additionally, the TTD and the MD rates signify a timely attack detection. Overall, the evaluation results demonstrate the practical viability of the detection mechanism in a real-world DER-integrated smart inverter.

VII. Conclusions

In this research, we developed a real-time HIL testbed for simulating MITM false data injection attacks in DER-integrated smart inverter topologies. Our testbed supports the study of smart inverter vulnerabilities and facilitates further research on advanced attack strategies and detection methods. To aid continued research, we publicly release DNP3 traffic captures under both benign and attack conditions.

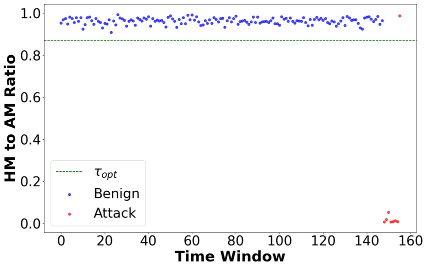


Fig. 7: Visualization of detection model in the test set showing detection metric below threshold after attacks

Furthermore, we designed and evaluated an attack detection mechanism an anomaly detection mechanism based on round-trip time (RTT) between DNP3 requests and responses. Our results show that timing deviations are effective indicators of MITM attacks, even without payload inspection. Using a multi-objective optimization framework, we tuned detection hyperparameters to balance false alarms, time to detection, and missed detections. This approach provides a practical method for real-time intrusion detection in DNP3-based industrial control systems.

Acknowledgement

We thank *Chamara Devanarayana* and the RTDS team for technical inputs, as well Alfred Batu, Badr Semia and Lucas Ling for their efforts in developing the initial RSCAD testbed, which laid the foundation for this project.

References

[1] E. Ghiani & F. Pilo."Smart inverter operation in distribution networks with high penetration of photovoltaic systems",*Journal of Modern Power Systems and Clean Energy*,2015.

[2] P Jain and P Tripathi, "SCADA security: a review and enhancement for DNP3 based systems", *CSI transactions on ICT*, pp. 301-308, 2013.

[3] "IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003), pp. 1–138, 2018.

[4] I. Darwish, O. Igbe & T. Saadawi,"Experimental and theoretical modeling of DNP3 attacks in smart grids",*36th IEEE Sarnoff symposium*, 2015.

[5] C. Cremers, M. Dehnel-Wild, and K. Milner, "Secure Authentication in the Grid: A Formal Analysis of DNP3 SAv5," Journal of Computer Security, vol. 0, 2018.

[6] S. East, J. Butts, M. Papa, & S. Shenoi,"A Taxonomy of Attacks on the DNP3 Protocol", *In Critical Infrastructure Protection III: Third Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection*, Hanover, New Hampshire, USA, 2009.

[7] M. H. Cintuglu, O. A. Mohammed, K. Akkaya and A. S. Uluagac, "A Survey on Smart Grid Cyber-Physical System Testbeds," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 446-464, Firstquarter 2017, doi: 10.1109/COMST.2016.2627399.

[8] "RTDS Simulator — Powerful Hardware for World-Class Simulation." Accessed: Oct. 13, 2024. [Online]. Available: https://www.rtds.com/technology/simulation-hardware

[9] M. Maliha, A. Oluyomi and N. Braasch-Cyberattack-Data" [Online]. Available: https://github.com/maisha29/-RTDS-Testbed-Cyberattack-Data/tree/main

[10] "IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)," in IEEE Std 1815-2012 (Revision of IEEE Std 1815-2010) , vol., no., pp.1-821, 10 Oct. 2012.

[11] S. Bhattacharjee and S. K. Das, "Detection and Forensics against Stealthy Data Falsification in Smart Metering Infrastructure," IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 1, pp. 356–371, Jan. 2021.

[12] P. Radoglou-Grammatikis, V. Kelli, T. Lagkas, V. Argyriou, P. Sarigiannidis, November 22, 2022, "DNP3 Intrusion Detection Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/s7h0-b081.

[13] M. H. Cintuglu, O. A. Mohammed, K. Akkaya and A. S. Uluagac, "A Survey on Smart Grid Cyber-Physical System Testbeds," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 446-464, Firstquarter 2017.

[14] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Transactions on Information and System Security, vol. 3, no. 3, pp 186–205, 2000.