Stealthy PTP Delay Attack Mitigation through Machine Learning-Based Detection in Smart Grids

Olivier Cabana, Amr M. Youssef, Mourad Debbabi, El-Nasser S. Youssef, Marthe Kassouf

***Abstract*—Smart grids are critical infrastructure that rely on accurate timing to operate effectively, which makes time synchronization very important. The Precision Time Protocol (PTP) is a standard used to synchronize clocks with high accuracy, making it one of the best choices for smart grid time synchronization. However, the latest version of PTP, IEC 61588:2021, is vulnerable to stealthy delay attacks. In this work, we implement several types of delay attacks on a co-simulation testbed to demonstrate the impacts of these attacks on grid operations and test the performance of multiple machine learning models for detection. We have achieved a mean False Positive Rate of 0.03 and a mean False Negative Rate of 0.12 per sample with a Gated Recurrent Unit (GRU) model.**

I. Introduction

Power grids rely on extremely precise timing, with time inaccuracies less than one microsecond, to coordinate critical control processes [1]. A time synchronization error can cause measurements to have incorrect timestamps [2], [3], leading to erroneous control decisions that can result in blackouts or damaged equipment. To prevent this, grids use high-precision clocks, such as Global Positioning System (GPS) clocks that act as references for the time signals distributed to the network devices through synchronization protocols like the Power Utility Profile (PUP) from the Precision Time Protocol (PTP). However, PTP has inherent vulnerabilities [4], especially to delay attacks, that an attacker can exploit.

Motivation. Incorrect time synchronization affects networks at every level. Measurements with incorrect timestamps risk being dropped or lead to incorrect state estimation [5], without being detected by bad data detection [6]. Processes involving multiple Intelligent Electronic Devices (IEDs) that are not correctly synchronized cannot properly cooperate, leading to failures such as line disconnections [7] and blackouts [8]. Finally, logs and security alerts with incorrect timestamps make it more difficult to reconstruct the sequence of events in the event of a fault or an attack [8]. To prevent the exploitation of the PTP protocol for time synchronization attacks, in this paper we intend to propose a machine learning-based detection model to address delay attacks against time synchronization.

Challenges. Current research on PTP security proposes security mechanisms such as thresholds, guardian clocks, or path asymmetry evaluation to prevent delay attacks. These methods, however, can be bypassed by incremental shifts in the delay that simulate normal network jitter, or impose structural changes to PTP. Thus, there is a need for another method of detection, which comes with several challenges:

1. Finding a security solution for PTP delay attacks that can be generalized to any PTP implementation without modifying the protocol. Any change in PTP requires a thorough review process, which does not protect currently vulnerable systems.

2. Detecting different types of stealthy delay attacks against PTP. Attacks that introduce incremental, periodic, or random delays can be hard to distinguish from random jitter, making them difficult to identify.

3. Detecting attacks before the gap between the time reference and the compromised clock affects network performance.

Contributions. This article improves the security of PTP. It demonstrates that current path, guardian, and threshold-based detection methods are insufficient for stealthy delay attacks and provides an alternative solution. Specifically, we identify the following contributions of our work:

- Generating a stealthy delay attack scenario against a Network and System Management (NSM) substation.
- Evaluating the performance of threshold-based detection methods against stealthy delay attacks.
- Testing several machine learning-based detection models on the different PTP delay attack types to determine which model performs best.

The rest of this article is organized as follows: in Section II, we present an overview of the main concepts used in this work as well as the state of the art. Section III explores the threat model, while Section IV details the implementation of our attack scenario. Section V presents the machine learning-based detection results. Finally, in Section VI, we summarize our findings and discuss future research avenues.

II. Background

In this section, we describe the PTP standard and delay attacks, and explore the current state of the art.

A. Precision Time Protocol (PTP)

PTP is a time synchronization protocol that allows a set of clocks to synchronize their time with an accuracy of sub-microseconds [4]. Clocks in the network are organized in a hierarchy and are synchronized to a grandmaster clock occupying its top level and acting as the network time reference. This clock is selected with the Best Master Clock Algorithm (BMCA) and synchronizes the other clocks with the synchronization message exchange [9]. The PUP profile

Olivier Cabana, Amr M. Youssef, and Mourad Debbabi are with the Security Research Centre (SRC) of Concordia University, Concordia University, Montreal, QC, H3G 1M8, Canada (e-mail: o_cabana@concordia.ca; amr.youssef@concordia.ca; mourad.debbabi@concordia.ca).

El-Nasser S. Youssef Abdelhafez and Marthe Kassouf are with l'Institut de recherche d'Hydro-Québec, Varennes, QC J3X 1S1, Canada (e-mail: {Last name}.{First name}@hydroquebec.com).

979-8-3315-8138-1

defines specific requirements for PTP in power grids. It requires a maximum uncertainty of no more than 1000 ns for time-sensitive devices [10]. The profile mandates that synchronization messages, delay measurements messages, and Announce messages carrying the parameters of each clock for the BMCA are sent every 1 sec [10], and that the PTP network includes redundant paths. The delay between two clocks (A and B) is measured as follows:

$$Delay = \frac{((t_4 - t_1) - (t_3 - t_2))}{2} \tag{1}$$

where t_1 is the time at which the message is sent, t_2 is the timestamp at which the message is received, t_3 is the timestamp at which the response message is sent, and t_4 is the timestamp at which it is received. The delay measurement relies on the assumption that the delay from A to B is the same as from B to A, i.e. half the round-trip time. When receiving a PTP synchronization message, the clock adjusts its time by calculating the offset from the timestamp received in the message using the measured delay.

1) PTP Delay Attacks: Delay attacks occur when an adversary introduces a delay during the transmission of PTP messages, which in turn affects the calculated clock offset:

$$Delay' = \frac{((t_4 - t_1') - (t_3 - t_2))}{2} \tag{2}$$

where t_1' is the modified delay, which can be represented as:

$$t_1' = t_1 + d \tag{3}$$

where d is the delay introduced by the adversary. There are four types of delay attacks in the literature:

- **Single Value Delay Attacks:** the adversary adds a single value to the delay (e.g. see [11]).
- **Linear Delay Attacks:** the attacker increases the clock offset in small increments, reducing the risk of detection, but prolonging the attack duration [11].
- **Symmetrical Delay Attacks:** the adversary adds equal delays on both sides of the exchange. Once the delay reaches the desired value, the attacker reduces the delay on one side, causing desynchronization [12].
- **Random Delay Attacks:** the attacker introduces random delays to the messages, causing desynchronization [13].

B. Related Work

There are three main PTP delay attack detection methods proposed in the literature. Threshold-based detection methods monitor some of the values of the clocks, using static or dynamic thresholds to detect an attack. For example, the PTP standard [9] imposes that the mean of the interarrival time of the messages is within 30% of the configured interval (i.e. one second). Giorgi [11] proposed a strategy to calculate the offset of clocks, which is then leveraged in a detection algorithm for delay attacks. Likewise, the authors of [13] suggested a set of thresholds based on the maximum offset, the required accuracy of the PTP implementation, and the average recorded delay to detect attacks. Threshold-based detection, however, is vulnerable to linear delay attacks and random delay attacks that introduce incremental changes to the delay that are indistinguishable from random noise. Guardian clock-based detection introduces an additional device into the PTP network that monitors the clocks for discrepancies. For example, Moussa *et al.* [14] proposed a new type of clock that would receive the offsets of the slave clocks and calculate its own offset based on the mean. The problem with these approaches is that they require modifying the network and the implementation of the protocol. Finally, path asymmetry-based detection such as [15], uses redundant paths to harden the network against delay attacks by comparing the delay of both paths. This solution, however, can be bypassed by compromising nodes that are on both paths, or two nodes instead of one. Most of the proposed solutions involve changing PTP, by adding new messages [15], or additional clocks [14]. Changes to the protocol involve a lengthy review process, leaving the current system implementations relying on PTP exposed to this threat until a new standard is published, adopted by manufacturers, and implemented by asset owners. Our proposed solution leverages existing mechanisms to gather the necessary information about clocks, and uses machine learning to learn the underlying patterns of the network traffic to identify attacks despite the network jitter. It can detect attacks that bypass threshold-based methods, can be deployed in any network, and can be retrained to adapt to changing network conditions.

III. Attack Framework

In this section, we present the threat model and two attacks.

A. Attack Model

In this section, we discuss the threat model for our attack. We present different aspects of the model in Fig. 1. As stated in the figure, the goal of the attacker is to disrupt grid operations by desynchronizing the clocks in the network. To accomplish this goal, the attacker leverages their understanding of PTP, and knowledge about the topology of the network to understand where to launch their attack, and secures access to the infrastructure of the network. They do so by either tricking the operator into installing a fake update, or by exploiting vulnerabilities in the switches/routers/gateways (i.e., by using vulnerabilities such as CVE-2025-20316, CVE-2025-7851, or CVE-2025-20339 [16]). The attack introduces a malicious delay that results in an incorrect offset calculation by the slave clocks, increasing the difference between the slave clock time and the reference time. The result is the disruption of critical processes [1], incorrect measurement timestamps [2], dropped measurements, monitoring gaps [17], and possibly blackouts [8]. We only consider three Industrial Control System (ICS) network levels from the Purdue model framework for ICS [18]: the supervisory-control level (level 2), the process control level (level 1), and the physical process level (level 0).

IV. Implementation

In this section, we describe the implementation of attack scenarios on a Hypersim testbed with Hardware-in-the-Loop.

979-8-3315-8138-1

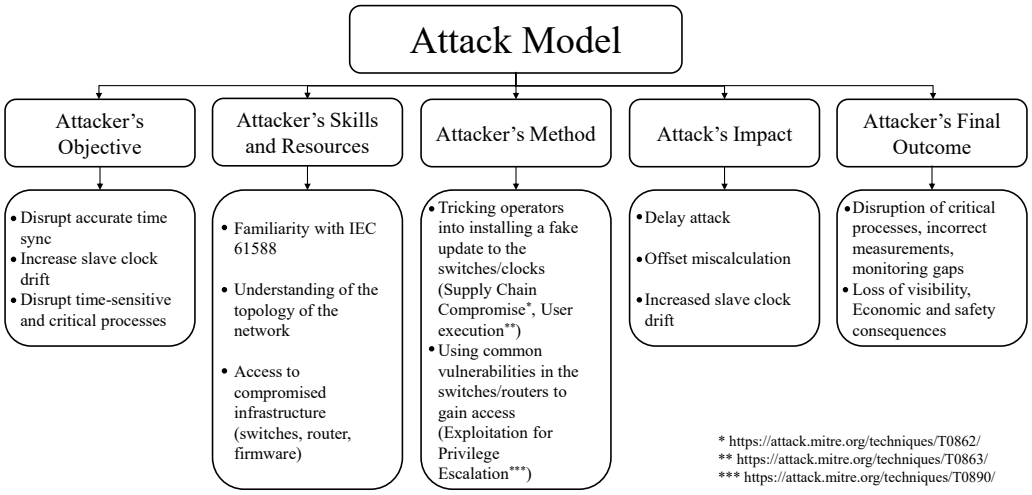


Fig. 1: Threat Model against a PTP Smart Grid.

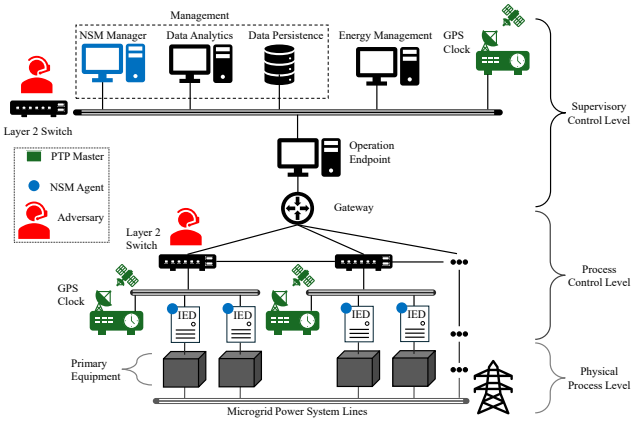


Fig. 2: Overview of the microgrid with PTP and NSM.

A. Hypersim IEC 61870 Microgrid Testbed

We demonstrate the effect of desynchronization attacks against a microgrid testbed, using a Hypersim co-simulation framework [19], in which we implemented PTP v2.1. The testbed can simulate the power system with a timestep of 50 microseconds. It has a microgrid central controller (MGCC), three loads, three energy storage systems (ESS), three wind turbines, combined heat and power (CHP), and a photovoltaic array (PV). The testbed employs IEC 62351–7 Network and System Management (NSM) [19], with agents distributed across critical elements of the grid, such as lines and the point of common coupling (PCC). The NSM agents monitor the condition of the grid, with the manager polling the agents every 5 seconds to capture statistical network traffic data, which includes five synchronization exchanges. The full details of the microgrid are described in [19]. We have implemented PTP v2.1 on the NSM agents and manager and disrupted synchronization among the NSM devices by using random attacks. The schema for our setup is presented in Fig. 2.

V. Experimental Results

We present the results of the attacks and test the performance of machine learning algorithms and contrast their performance against a threshold-based approach.

1) Attacks against the Microgrid: We test two attacks against the synchronization of the NSM. NSM relies on polling agents at a consistent rate to collect statistical information about the state of the network. If the polling rate changes due to a shift in the clock's time, but the manager continues to assume it is polling at the same rate as before, the collected values, such as the number of packets received, will show a discrepancy, causing alerts that are not reflecting the real source of the attack, potentially obscuring those triggered by other attacks on the grid [19]. Additionaly, when used in conjunction with other attacks, the generated alerts can cause the operator to incorrectly identify the underlying attack, leading to wrong corrective actions. However, if the PTP delay attack is detected early enough, these issues can be mitigated.

Increased Polling Rate Attack. Our first attack targets the clock of the NSM manager. We use a delay attack that introduces a positive clock offset, accelerating the NSM manager's polling rate. Our goal is to demonstrate how time manipulation can be used to disrupt the monitoring system. We launch the attack after 20 minutes, achieving a clock shift of 0.5 seconds and causing a rapid increase in the events generated by the manager. The results of the attack are shown in Fig. 3, which shows the number of events over time, aggregated by the minute. In the figure, we can see a sharp increase in events generated the longer the attack lasts. This is caused by the NSM monitoring security platform, described in [19], which uses both rule-based and anomaly-based detection models. These models count events such as the number of GOOSE and SV messages received. The change in the polling rate causes a discrepancy in these numbers, triggering alerts that will need to be addressed by the operators, potentially slowing their response time to other threats. The last blue bar is smaller, because the simulation lasted slightly less than 30 minutes.

979-8-3315-8138-1

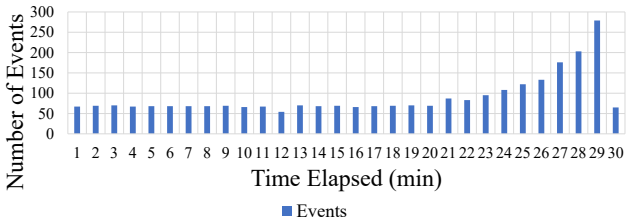
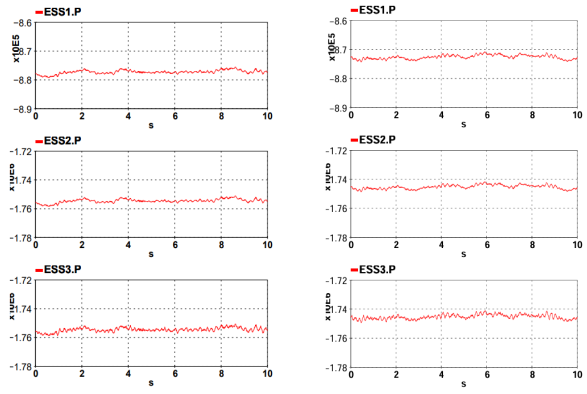


Fig. 3: Delay attack on NSM monitoring system.



(a) Normal grid conditions (b) Frequency delay attack

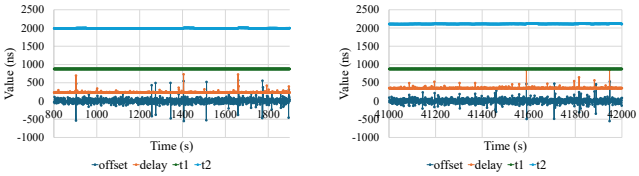
Fig. 4: Delaying frequency measurements to trick the ESS into supplying excess power. The y-axis is the supplied power (W), and the x-axis is the time (s).

Combined NSM and Frequency Delay Attack. In our second attack, we combine the previous desynchronization attack on the supervisory level with an attack on the process-control level. We launch a frequency delay attack as shown in [19], where the packets containing frequency measurements from the PCC are delayed, which deceives the energy storage system into supplying excess power, as shown in Fig. 4 taken from Hypersim ScopeView. The excess power causes frequency instability. The attack would normally be detected due to the agent at the MGCC recording a reduced number of received frequency packets, which generates an alert on the manager's side, but the synchronization attack on NSM masks the frequency delay attack amidst other anomalies, risking aggravating the impact of the attack. This approach could be used to hide other physical attacks on the grid as well.

A. Threshold-Based Detection

We evaluate the performance of a threshold-based approach for PTP delay attack detection. Thresholds can be implemented without modifying the protocol, but struggle with incremental delay increases. We use the threshold proposed in [13], where two values, k_1 and k_2, are calculated as:

$$k_1 = T_A - |O_M| \tag{4}$$



(a) Before the Attack (b) During the Attack

Fig. 5: Performance of the threshold-based detection against a linear delay attack.

where T_A is the accuracy required by the PTP profile, and O_M is the maximum offset captured during the learning period.

$$k_2 = 2 \times k_1 + \sigma \tag{5}$$

where σ is an arbitrary value chosen by the operator to reduce the sensitivity of the detection. The threshold triggers when the offset is larger than k_1 while the difference between the delay and the average delay over a time window W is smaller than k_2. We use a T_A of 1 microsecond, matching the requirements of the PUP profile, a σ of 0. We demonstrate the performance of the threshold in Fig. 5, where the behavior of the system before the attack and during it are almost identical. The attack increments at a rate of $2 \times W$, to avoid tripping the threshold, thus avoiding detection.

B. Machine Learning-Based Detection

Machine learning is frequently used in anomaly-based detection systems. We implement a total of eight machine learning models, which were hypertuned through grid-search, and use them to classify the offset of the clocks as normal or anomalous. We use a training set containing the delay and offset of network clocks under normal conditions and during different PTP delay attacks, with a equal split of samples, the Adam optimizer and binary cross-entropy loss. We test a set of algorithms that perform well in anomaly detection tasks: the Generative Adversarial Network (GAN) model learns to recognize fake (malicious) data, the Variational Autoencoder (VAE) will have a larger reconstruction error for malicious data, and the Temporal Convolutional Neural Network (TCNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models because we are using time series data. We implement several variations of these models to find the best performing model for detection. We find that the performance of the ML models change depending on the PTP delay attack scenario, and that overall, the GRU model has the best performance. It is a binary classifier with one hidden layer, that takes in a time window of measurements and identifies whether the last entry is normal or compromised. We compare its performance against the threshold-based detection in Table I. Each attack lasts 400 seconds, and reaches a maximum delay of approximately 3.6 microseconds. The table reports the GRU model's performance on each type of delay attack, then its average performance over all attacks, according to the False Positive Rate ($FPR = FP/(FP+TP)$) per sample, the False Negative Rate ($FNR = FN/(FN+TN)$) per sample, and

979-8-3315-8138-1

TABLE I: Summary of the performance of the GRU model for the detection of different PTP delay attacks, compared to threshold-based detection.

Attack	GRU model FPR	GRU model FNR	GRU model MTTD	Threshold FPR	Threshold FNR	Threshold MTTD
Sing.	0.092	0.000	1.0	0.014	0.987	162
Lin.	0.018	0.200	81.0	0.0	1.0	–
Sym.	0.017	0.095	3.0	0.005	0.997	120
Rand.	0.002	0.188	78.0	0.000	0.933	0
Avg.	0.032	0.121	40.75	0.003	0.983	94

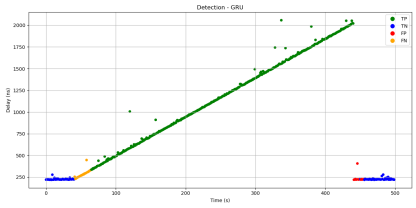


Fig. 6: Performance of the GRU model on a linear attack.

the Mean Time To Detection (MTTD), which is a measure of how fast and accurate is the detection. We find that the hardest attack to detect is the Linear increase attack, as suggested by the FNR of 0.2. However, this larger MTTD is because the impact of the attack is more gradual, as shown in Fig. 6. In contrast, the threshold-based detection has a large FNR and MTTD. It was also unable to detect the linear attack.

C. Feature Explainability

We use the SHapley Additive exPlanations (SHAP) python library [20], which estimates the change in the expected model prediction for each feature. The importance of our features is presented in Table II. We find that the most important feature is the delay variance, followed by the offset mean and variance.

TABLE II: Importance of the features used for detection.

Feature	Importance
var_delay	0.005638
std_offset	0.003358
std_delay	0.003269
mean_delay	0.002103
delay	0.000807
mean_offset	0.000506
var_offset	0.000234
offset	0.000023

VI. Conclusion

In this work, we define an attack scenario for desynchronization attacks against the smart grid. We demonstrate the effects of the PTP delay attacks on a Hypersim testbed, and test multiple machine learning-based algorithms for PTP delay attack detection, achieving an average FPR of 3% and an FNR of 12% with the GRU algorithm. We also identify the need for training models to detect different types of attacks, as each causes a different type of disturbance to the network. In future work, we will explore the use of machine learning to study the impact of complex delay attacks involving multiple devices on power grid operation.

VII. Acknowledgment

This research is supported by the Concordia Hydro-Québec Hitachi Partnership Research Chair on Smart Grid Security with support from NSERC and PROMPT Québec.

References

[1] P. Fei, Z. Chen, Z. Wang, and S. A. Jafar, "Communication-efficient clock synchronization," in *ICC 2022 - IEEE International Conference on Communications*, pp. 2592–2597, 2022.

[2] Y. Li, C. Li, G. Wu, and C. Zhang, "Research on high-precision time distribution mechanism of multi-source power grid based on mec," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–5, 2019.

[3] B. M. Raju, P. Risbud, G. Gajjar, and S. Soman, "Mitigation of gps spoofing attacks in linear state estimation using principal angle computation," in *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2023.

[4] B. Moussa, M. Debbabi, and C. Assi, "Security assessment of time synchronization mechanisms for the smart grid," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1952–1973, 2016.

[5] D. Macii and S. Rinaldi, "Time synchronization for smart grids applications: Requirements and uncertainty issues," *IEEE Instrumentation & Measurement Magazine*, vol. 25, no. 6, pp. 11–18, 2022.

[6] M. Delcourt and J.-Y. Le Boudec, "Security measures for grids against rank-1 undetectable time-synchronization attacks," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 1, pp. 231–244, 2022.

[7] W. Alghamdi and M. Schukat, "Cyber attacks on precision time protocol networks—a case study," *Electronics*, vol. 9, no. 9, 2020.

[8] Y. Liu, B. Sun, Y. Wu, Y. Zhang, J. Yang, W. Wang, N. L. Thotakura, Q. Liu, and Y. Liu, "Time synchronization techniques in the modern smart grid: A comprehensive survey," *Energies*, vol. 18, no. 5, 2025.

[9] *IEC/IEEE International Standard - Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. 2021.

[10] *IEC/IEEE International Standard - Communication networks and systems for power utility automation – Part 9-3: Precision time protocol profile for power utility automation*. 2016.

[11] G. Giorgi, "Measurable quantities in a synchronization system under attack: A first step to implement a detection system," in *2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, 2024.

[12] J. Neyer, L. Gassner, and C. Marinescu, "Redundant schemes or how to counter the delay attack on time synchronization protocols," in *2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pp. 1–6, 2019.

[13] H. Li, D. Li, X. Zhang, G. Shou, Y. Hu, and Y. Liu, "A security management architecture for time synchronization towards high precision networks," *IEEE Access*, vol. 9, pp. 117542–117553, 2021.

[14] B. Moussa, M. Debbabi, and C. Assi, "A detection and mitigation model for ptp delay attack in a smart grid substation," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 497–502, 2015.

[15] A. Finkenzeller, O. Butowski, E. Regnath, M. Hamad, and S. Steinhorst, "Ptpsec: Securing the precision time protocol against time delay attacks using cyclic path asymmetry analysis," in *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*, pp. 461–470, 2024.

[16] MITRE, "Common vulnerabilities and exposures (cve)." https://cve.mitre.org, 2025. [Online; accessed 2025-11-07].

[17] B. Dickerson, *Time in the Power Industry: How and Why We Use It*. 2021.

[18] D. Garton, "Purdue model framework for industrial control systems & cybersecurity segmentation," *US Department of Energy*, vol. 14, pp. 2022–10, 2022.

[19] M. Karanfil, D. E. Rebbah, M. Debbabi, M. Kassouf, M. Ghafouri, E.-N. S. Youssef, and A. Hanna, "Detection of microgrid cyberattacks using network and system management," *IEEE Transactions on Smart Grid*, pp. 1–1, 2022.

[20] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," Curran Associates, Inc., 2017.

979-8-3315-8138-1