Towards Secure and Resilient Synchrophasor Networks Using P4 Programmable Switches

Zhiyao He*, Yanfeng Qu*, Gong Chen†, Reuben Samson Raj*, Hui Lin‡, Dong Jin*
*Electrical Engineering and Computer Science, University of Arkansas, Fayetteville, AR, USA
{the, yqu, rs077, dongjin}@uark.edu
†Computer Science, Illinois Institute of Technology, Chicago, IL, USA
gchen31@hawk.iit.edu
‡Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA
huilin@uri.edu

2024 IEEE Green Technologies Conference (GreenTech) | 979-8-3503-0858-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/GREENTECH58819.2024.10520393

Abstract—**Phasor Measurement Units (PMUs) enable high-speed and high-precision power quality measurements, but their vulnerability to cyber-attacks poses substantial risks to the stability and reliability of power systems. This paper explores the application of programmable networks, specifically P4 switches, to enhance the cybersecurity of PMU systems. These switches can be programmed to enable the creation of custom forwarding and processing logic at the data plane level. We analyze PMU-specific attacks through attack trees, propose effective mitigation strategies using programmable switches, and present a case study demonstrating the recovery of missing synchrophasor data by implementing a matrix completion algorithm on the switch data plane. The recovery scheme is evaluated in a container-based P4 network emulator using real PMU data from a campus microgrid. For instance, in the 10% missing data scenario, the recovery scheme can achieve a mean absolute percentage error of 0.038% for voltage magnitude and a phase angle error discrepancy of around 0.08 degrees.**

Index Terms—**Phasor Measurement Unit, Cyber Security, Programmable Network, P4, Smart Grid**

I. Introduction

Synchrophasors in Wide Area Monitoring Systems (WAMS) are time-synchronized phasor measurements from Phasor Measurement Units (PMUs), providing insights into the power system's current state and dynamics in real time [1]. They contribute to the efficiency, reliability, and resilience of electricity delivery, aligning seamlessly with the goals of promoting sustainable and environmentally friendly energy solutions. However, the vulnerability of PMU systems to cyber-attacks poses substantial threats to the stability and reliability of power systems. Adversaries could manipulate PMU measurements, launch denial-of-service attacks to disrupt normal operations, engage in spoofing to inject false data, and exploit vulnerabilities in communication protocols or software. Additionally, the risk of physical tampering, susceptibility to malware infections, and other cybersecurity challenges further accentuate the need for effective protective measures.

The specific configurations of PMUs may not be fully supported by commercial off-the-shelf security tools, such as firewalls and intrusion detection systems. Also, the real-time operational requirements of PMUs can render traditional security measures ineffective by introducing unacceptable delays. Furthermore, traditional security solutions, designed for more resource-abundant environments, may not align with the resource-constrained PMU systems (e.g., limited processing power and bandwidth). This mismatch results in significant protection gaps, leaving these critical systems vulnerable to cyber-attacks.

To address these cyber-security challenges, we incorporate cutting-edge network data plane programmability, specifically leveraging P4 (Programming Protocol-independent Packet Processors) [2], into PMU networks. P4, as an advanced data plane programming language, empowers real-time and precise control over packet processing and transmission. This programmability is utilized to deploy customized security solutions tailored for PMU networks.

Incorporating P4 into PMUs yields significant cybersecurity enhancements. Firstly, it facilitates the direct integration of security features at the communication network device level, offering an in-network and decentralized security solution. Secondly, P4's real-time deep packet inspection capability enables proactive detection and mitigation of potential attacks [3]. Thirdly, the flexibility and programmability of P4 make it an ideal platform for developing adaptive security applications capable of evolving in response to changing threats and security requirements.

This study explores the enhancement of PMU system security using P4-based solutions. We first conduct a thorough analysis of cyber-attacks on PMU systems, leading to the creation of comprehensive attack trees focusing on data, software, communication networks, and physical devices. These attack trees systematically identify potential attack vectors and their impacts, providing a structured insight into vulnerabilities and the threat landscape surrounding PMU systems. Next, we introduce a design framework tailored for P4 programmable switches, mapping security applications to PMU-specific attacks outlined in the attack trees. Utilizing a P4-enabled data plane, we formulate a parser conforming to the PMU communication protocol specification, facilitating deep packet inspection and event detection. Furthermore, the controller on the P4 switch can be harnessed to broaden the scope of attack analysis and enhance mitigation techniques.

To demonstrate the feasibility and effectiveness of P4-based solutions for PMU system security. We employ the proposed P4 switch design framework to develop a PMU missing data recovery application. Utilizing Spectral Regularization Algorithms (SRA) for matrix completion, this application efficiently recovers missing PMU data on the P4 switch. Our approach leverages multiple data channels from PMUs to form a matrix, processed with SRA to effectively recover lost or corrupted data. The case study not only reveals potential vulnerabilities in PMU networks but also highlights P4-based strategies for recovering missing data. With 10% missing PMU data, the mean absolute percentage error for voltage magnitude is 0.038%, and the phase angle error discrepancy is approximately 0.08 degrees.

The main contributions of this work are summarized as follows.

- We construct a comprehensive taxonomy of potential cyber threats to PMU systems, structured visually as attack trees and categorized into four main types: Software Attack, Physical Attack, Communication Network/Protocol Attack, and Data Manipulation.
- We create a specialized design framework for P4 programmable switches, empowering PMU system security applications.
- We pioneer the development of a P4 application for recovering missing PMU data, utilizing a convex optimization algorithm on a P4 software switch.

The remainder of the paper is organized as follows. Section II presents attack trees specifically tailored for PMU systems. Section III elucidates a P4-enabled PMU network architecture for security enhancement. Section IV provides an in-depth exploration of a case study focused on missing data recovery through matrix completion. Lastly, Section V summarizes the paper with future work.

II. Cybersecurity Threats in PMU systems

We classify potential attacks on PMU systems into four main types: Software Attack, Physical Attack, Communication Network/Protocol Attack, and Data Manipulation, illustrated in Fig. 1. The categorization utilizes a color-coding system to differentiate between attack categories, targets, and methods, as well as the potential impact on the system's security. Blue boxes represent the attack categories, green boxes indicate the specific targets within the PMU system, and orange boxes denote the methods of attacks. Additionally, the impact of each attack is symbolized by red circles with 'C' for Loss of Confidentiality, circles with 'I' for Loss of Integrity, and circles with 'A' for Loss of Availability. These symbols provide a reference to understand the consequences that each type of attack might have on the PMU system.

Software attacks exploit vulnerabilities such as buffer overflows , where attackers overwhelm the system memory, potentially allowing for unauthorized code execution. Malicious firmware updates can introduce backdoors or sabotage PMUs by replacing legitimate firmware with compromised versions. Masquerading attacks involve impersonating trusted devices or users, and bypassing security measures to gain unauthorized access.

Communication attacks intercept or disrupt the data flow between devices. Eavesdropping [4] allows attackers to capture sensitive information, while packet spoofing deceives systems into accepting false data. Route manipulation [5] manipulates network traffic, potentially leading to communication delays or blockages, disrupting PMU operations.

Data manipulation attacks involve false data injection [6], where incorrect information is fed into the PMU data stream, leading to misinformed decisions. Command injection [7] involves sending unauthorized commands to control systems, which can cause improper actions within the power grid.

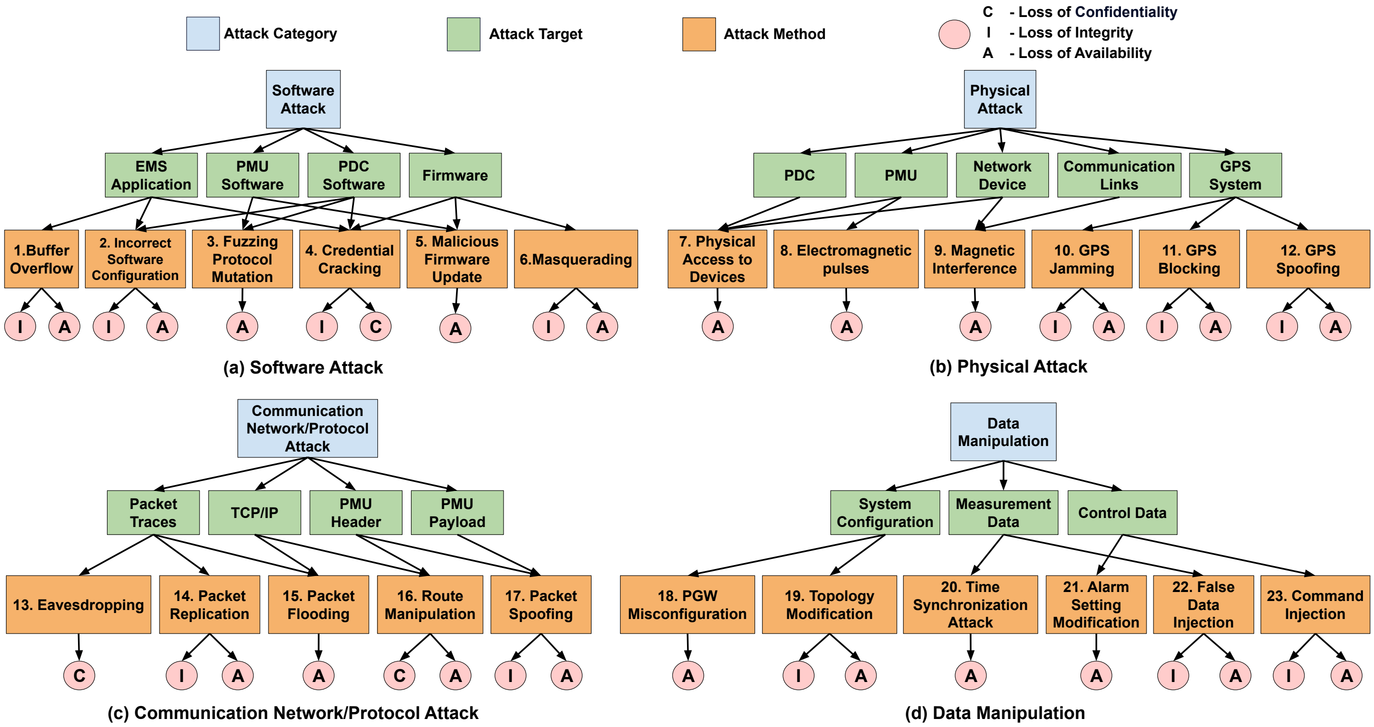
Physical attacks directly harm PMUs with methods like physical damage to devices, which can be as blatant as destruction of hardware or as subtle as electromagnetic pulses (EMPs) [8], which can incapacitate electronic components. Magnetic interference specifically targets PMUs' sensitivity to magnetic fields, leading to erroneous measurements or operational failure.

For instance, in scenarios where GPS spoofing [9] is categorized under physical attack in Fig. 1 (b), which directly impacts the functionality of PMUs. The primary consequence of a GPS spoofing attack, as indicated in the pink box, is the injection of incorrect data into the power grid due to falsified timestamps. PMUs, which are crucial for delivering accurate and synchrophasor measurements across the power grid, rely heavily on the precision of GPS signals for their timing. When these signals are compromised, the resultant data can be inaccurately timestamped, leading to a lack of synchronization with other measurements. This is a critical step in maintaining the integrity and availability of the overall dataset within the PMU system.

III. PMU Security Enhancement Using Programmable Switches

We aim to explore the P4-based data-plane programmable network to tackle these security concerns, leveraging its distinct advantages: (i) P4 allows for the decentralized implementation of security features, offering a scalable, in-network solution for securing WAMS; (ii) P4 enables the monitoring and inspection of network traffic at the packet level, up to the application payload, aiding in early attack detection; and (iii) P4 provides a flexible, customizable platform for developing applications that can evolve in response to changing security needs and threats.

The P4 switches offer a versatile defense against various cyber-attacks on PMU systems as highlighted in Table I. It caters to software attacks through detailed packet header and payload inspections, ensuring that malicious activities like masquerading or buffer overflows are detected before they can cause harm. While P4 may not directly prevent unauthorized physical access to devices, its ability to reconfigure network traffic on the fly allows for quick isolation and mitigation of compromised devices. For network-based threats,



(a) Software Attack

(b) Physical Attack

(c) Communication Network/Protocol Attack

(d) Data Manipulation

Fig. 1. Attack trees on PMU system vulnerabilities. Note: Loss of Confidentiality (e.g. Information gathering and Loss of Privacy), Loss of Integrity (e.g. Manipulation of Control and Manipulation of View), Loss of Availability (e.g. Loss of View, Loss of Control, and Service interruption)

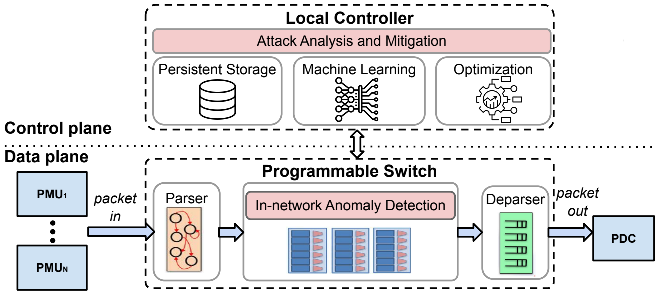


Fig. 2. Architecture Design of In-Network Security Applications on a P4 Programmable Switch

TABLE I
TRANSPOSED P4-BASED SOLUTIONS FOR CYBER-ATTACKS ON PMU SYSTEMS

P4-based Solutions	Attack Index (see Fig. 1)
Packet header inspection	3,12-17,20
Payload inspection	1,13-15,17,22
Stateful Filtering	3,10,14-16,20,22
Traffic engineering	14,15
Hashing	20,22,23
Encryption	2,13,16,18,19,21-23

P4's granular traffic engineering and filtering capabilities are crucial for warding off packet flooding and route manipulation, and maintaining the correctness of communication channels. Against data manipulation, P4 employs hashing to verify data integrity and encryption to safeguard data privacy, addressing concerns of unauthorized data alteration. P4's distributed nature, high-performance processing, and customizable data plane collectively form a robust framework, enhancing the resilience of PMUs against a wide spectrum of cyber threats.

To enhance the security and resilience of a PMU network, we propose a cross-plane framework in a programmable network, supporting new P4-based security applications as illustrated in the Fig. 2. This framework is two-tiered. The first tier is an in-network anomaly detection system in data plane, utilizing a parser to examine incoming PMU traffic, including headers and payloads. The P4-switches' programmable nature allows for custom inspection of specific packet types. This thorough inspection enables rapid detection while maintaining line-rate packet forwarding. However, the limited resources of P4 switch hardware, such as computational power and memory, pose challenges in deploying complex models or optimization algorithms for comprehensive analysis or optimization on the data plane. To address this, we utilize the programmability of the control plane to collaborate with the data plane for more complex tasks, like attack analysis and mitigation. In case of anomalies, the local controller, along with the switch, performs attack analysis and may initiate mitigation strategies. This controller, which could also be a PC or a powerful server, has more computational resources and operates on a full OS kernel, enabling the development of sophisticated applications, including machine learning models and complex optimization algorithms. It also possesses more resources, including CPU, GPU, memory, and persistent storage such as databases. These advantages empower the controller to handle challenging tasks. After completing its analysis, the controller communicates decisions to the data plane switch, such as altering packet routes or correcting invalid packet fields. Once the packets are processed and secured, they are reassembled by the deparser, ensuring the integrity of headers and payloads for onward transmission. These packets then exit the P4 switch and proceed to their next destination, like a Phasor Data Concentrator (PDC) or a WAMS server, thus passing through the security system.

The adaptability of the P4 language is evident in its ability to establish robust security measures against cyber-attacks on PMUs, enabling proactive detection and mitigation strategies for diverse threat factors. Subsequently, the next section will delve into strategies for missing data recovery, serving as a pioneering demonstration application. This ensures that, even in the event of an attack or system fault, the quality of power data can be maintained.

IV. Case study: Missing PMU data recovery

Attacks targeting the PMU network, such as packet flooding (see Index 15 in Fig. 1), cause communication delays, missing data, and degraded power system observability and situational awareness. To mitigate the attack and address the missing data issues, we develop real-time detection and recovery applications on the proposed P4-based framework. **Missing PMU data detection** is developed in the data plane, identifying anomalies by closely monitoring time differences between PMU packets. This process begins with the parsing of incoming PMU packets at the P4 switch, where the timestamps are compared. Data from multiple PMU channels is systematically pushed onto corresponding stacks. When the time difference between expected data points exceeds a predefined threshold, a missing data event is flagged.

Missing PMU data recovery is developed in the controller of the P4 switch. Upon detecting missing data, the switch promptly forwards the preceding PMU measurements to the local controller. Leveraging higher computing power and low communication latency, the local controller efficiently executes complex recovery algorithms. Since PMU data has low-rank features, we use matrix completion for missing data recovery [10]. Given multiple PMU data as low rank matrix Ω of size $n_1 \times n_2$, to tackle the challenge of a missing value x, we use Spectral Regularization Algorithms [11] [12], which are designed for learning large incomplete matrices.

The objective is to estimate the missing entries by finding a matrix $\hat{M}$ that minimizes the function:

$$\min_{\hat{M}} \frac{1}{2} \sum_{(i,j)\in\Omega} (M_{ij} - \hat{M}_{ij})^2 + \lambda \|\hat{M}\|_*, \tag{1}$$

where Ω denotes the indices of observed entries, $\|\hat{M}\|_*$ is the nuclear norm (the sum of singular values) of $\hat{M}$, and λ is a regularization parameter. The algorithm updates $\hat{M}$ iteratively to 'fill in' the missing data. Once $\hat{M}$ is recovered, the P4 switch will further read the missing data point and generate the complete PMU packet to the destination host.

Evaluation. We set up a star network topology compromising four hosts and a central P4 switch. Three hosts function as PMUs and one serves as the PDC. The three PMUs transmit data traffic to the PDC. This scenario is evaluated using two P4 programs on the P4 switch, a program for simple forwarding, and a program implementing our detection mechanism. We carried out the performance evaluation by introducing varying missing data rates, from 1% to 10%, within a sample of 5,000 data packets based on the real data from a campus microgrid. In each experimental iteration, a proportionate number of packets (determined by the specified missing data rate) were randomly eliminated. For example, at a 5% data missing rate, we randomly dropped 250 of the 5,000 transmitted packets. The recovery scheme's effectiveness was gauged by measuring the precision of the data recovered at different data missing rates. All experiments were carried out on the Mininet [13] network emulation testbed, incorporating a P4 software switch, BMv2 [14]. Each experiment was repeated three times.

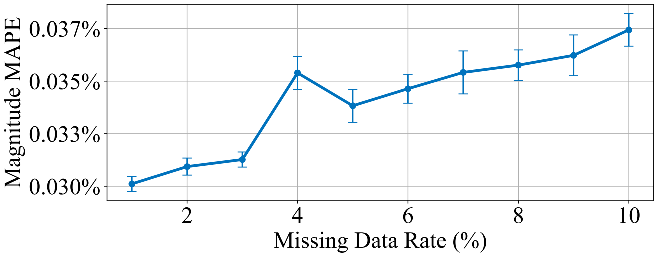


Fig. 3. Magnitude Mean Absolute Percentage Error vs Missing Data Rate

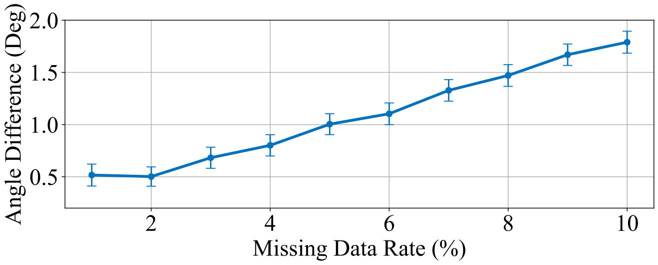


Fig. 4. Phase Angle Error vs Missing Data Rate

To evaluate the accuracy of the recovery scheme, we display the magnitude Mean Absolute Percentage Errors (MAPEs) and the phase angle differences with various missing data rates as depicted in Fig. 3 and 4. The results indicate a high fidelity between the reconstructed and actual phase angle data, with discrepancies ranging narrowly from 0.04 degrees to 0.16 degrees. The magnitude MAPEs also exhibit minimal fluctuation, with values spanning from 0.030% to 0.038%. Fig. 3 indicates an increase in the percentage of missing data, there is a corresponding upward trajectory along the Y-axis, showing a growth in the MAPE. This pattern reveals that as more data is missing, the average error in predictions increases. However, the upward trajectory on the Y-axis progresses at a moderate pace, which implies the effectiveness of the matrix completion method employed for estimating missing data with a controlled error margin.

Fig. 4 shows the plot of the missing data rate (x-axis) and the phase angle discrepancy, along with the corresponding standard deviation error bars. The graph illustrates an increasing trend of phase angle differences as the missing data rate increases. Initially, when the missing data rate is between 1% and 2%, the average difference between the original and recovered data is 0.5 degrees. As the missing data rate reaches 10%, the maximum difference is 1.8 degrees. In contrast to the trend in Fig. 3, the reason for this behavior lies in the fact that the phase angle alters relatively more compared to magnitude. The standard deviation remains around 0.2 overall, which indicates a consistent variability in the data regardless of the increasing missing data rate. This consistency in standard deviation suggests that while the phase angle errors increase with higher missing data rates, the predictability of these errors remains stable.

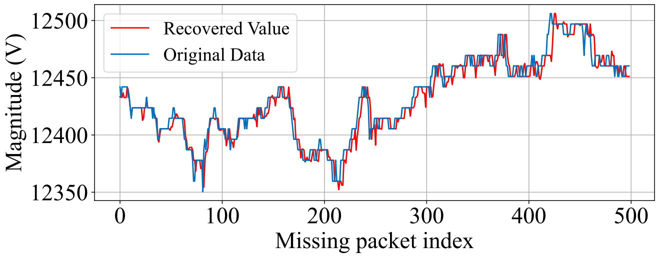


Fig. 5. Plot showing Original Data and Recovered value for each packet index (10% Missing Data Rate)

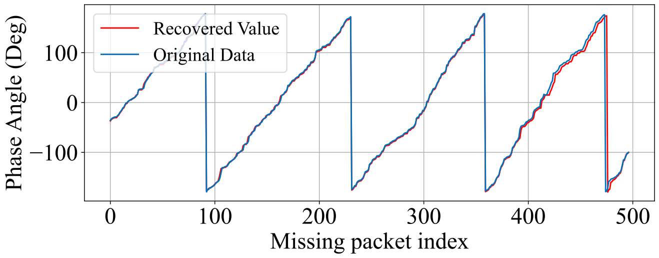


Fig. 6. Actual Measurement vs Recovered PMU Phase Angles (10% Missing Data Rate)

Fig. 5 and 6 depict a comprehensive analysis of the relationship between the number of missing packets (x-axis) and the recovered data generated by the controller. The graphs show a comparison between the original PMU data and the data recovered through the matrix completion method. The comparison highlights the remarkable accuracy achieved by the proposed recovery scheme. The close alignment of the recovered data and the original data under various missing data scenarios, ranging from 0 to 500, illustrates the method's effectiveness and reliability in handling missing PMU data.

V. Conclusion and Future Works

We leverage P4 switches to bolster the cybersecurity of PMU systems. We study PMU-specific attacks using attack trees, suggest mitigation strategies employing P4 switches, and provide a case study showcasing the recovery of missing PMU data. Future work includes the implementation of additional security measures tailored for PMU systems on P4 switches. Additionally, we plan to transit these implementations from BMv2 software switches to P4 hardware for enhanced performance and fidelity.

Acknowledgment. The authors are grateful for the support of the National Science Foundation (NSF) under Grant CNS-2247721, CNS-2247722, and CNS-2144513, and the NSF Center for Infrastructure Trustworthiness in Energy Systems (CITES) under Grant EEC-2113903, and the Office of Naval Research under Contract no. N000142412129.

References

[1] P. Nanda, C. K. Panigrahi, and A. Dasgupta, "PMU implementation for a wide area measurement of a power system," in *Proceedings of 2017 Devices for Integrated Circuit (DevIC)*, 2017, pp. 690–694.

[2] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, p. 87–95, Jul 2014.

[3] Z. Hu, H. Lin, L. Waind, Y. Qu, G. Chen, and D. Jin, "Industrial network protocol security enhancement using programmable switches," in *Proceedings of 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2023, pp. 1–7.

[4] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *Proceedings of the 10th International Conference on Mobile Computing and Networking (MobiCom'04)*, 2004, pp. 112–117.

[5] S. Basumallik, R. Ma, and S. Eftekharnejad, "Packet-data anomaly detection in PMU-based state estimator using convolutional neural network," *International Journal of Electrical Power Energy Systems*, vol. 107, pp. 690–702, 2019.

[6] Y. Li and Y. Wang, "False data injection attacks with incomplete network topology information in smart grid," *IEEE Access*, vol. 7, pp. 3656–3664, 2019.

[7] T. H. Morris, S. Pan, and U. Adhikari, "Cyber security recommendations for wide area monitoring, protection, and control systems," in *Proceedings of 2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–6.

[8] D. Wang, Y. Li, P. Dehghanian, and S. Wang, "Power grid resilience to electromagnetic pulse (EMP) disturbances: A literature review," in *Proceedings of 2019 North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[9] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 146–153, 2012.

[10] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1006–1013, 2016.

[11] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, p. 2287–2322, 2010.

[12] A. Rubinsteyn and S. Feldman, "fancyimpute: An imputation library for python." [Online]. Available: https://github.com/iskandr/fancyimpute

[13] "Mininet: An instant virtual network on your laptop (or other pc)," http://www.mininet.org/.

[14] p4Lang, "p4lang/behavioral-model: The reference P4 software switch." https://github.com/p4lang/behavioral-model.