SAND2019-10288C

Implementation of Intrusion Detection Methods for Distributed Photovoltaic Inverters at the Grid-Edge

C. Birk Jones
Renewable, Distributed Systems Integration
Sandia National Laboratories
Albuquerque, U.S.A
cbjones@sandia.gov

Adrian R. Chavez
Autonomous Cyber
Sandia National Laboratories
Albuquerque, U.S.A
adrchav@sandia.gov

Rachid Darbali-Zamora
Renewable, Distributed Systems Integration
Sandia National Laboratories
Albuquerque, U.S.A
rdarbal@sandia.gov

Shamina Hossain-McKenzie
Cyber Resilience
Sandia National Laboratories
Albuquerque, U.S.A
shossai@sandia.gov

***Abstract*—Reducing the risk of cyber-attacks that affect the confidentiality, integrity, and availability of distributed Photovoltaic (PV) inverters requires the implementation of an Intrusion Detection System (IDS) at the grid-edge. Often, IDSs use signature or behavior-based analytics to identify potentially harmful anomalies. In this work, the two approaches are deployed and tested on a small, single-board computer; the computer is setup to monitor and detect malevolent traffic in-between an aggregator and a single PV inverter. The Snort, signature-based, analysis tool detected three of the five attack scenarios. The behavior-based analysis, which used an Adaptive Resonance Theory Artificial Neural Network, successfully identified four out of the five attacks. Each of the approaches ran on the single-board computer and decreased the chances of an undetected breach in the PV inverters control system.**

I. Introduction

Monitoring and classifying the cyber activity of smart photovoltaic (PV) inverters helps maintain desired confidentiality, integrity, and availability. This is done using a signature-, behavior-, or hybrid-based Intrusion Detection System (IDS). However, the advantages and disadvantages of the signature and behavior-based approaches for PV inverter applications is unknown. The experiment, described in this paper, focused on the implementation of the signature and behavior methods in a real-world setting, and provided evidence, not proof, of each method's abilities. Identifying the optimal detection system is important as more and more PV inverters are connected to the internet while providing critical services to the grid.

Smart PV inverters connected to the internet are subject to interference, modifications, and damage through cyber-attacks. At one time, the sole intent of PV inverters was to convert direct current (DC) output from the solar electric generation system to a grid compatible alternating current (AC). Now, PV inverters connect to the internet through WiFi and ethernet connections to enable communications that allow for external control commands to alter settings and prompt system output. This interoperability and functionality offers opportunities for electric utilities to improve grid performance by injecting or absorbing reactive power (VAR), using Volt-VAR control operations. However, the connections create attack vectors for adversaries to probe, penetrate, and infect PV inverter control systems. Counter measures, that include IDS are necessary to properly protect the critical infrastructure [1].

Protecting PV inverters is important for maintaining appropriate operations on the grid and monitoring system health. For instance, advanced analytics leverages remote data flows to identify performance issues [2]. Or utilities use different functions, such as Volt-Var control, fixed and variable power factor, to address voltage problems [3]. These interconnections, if left unchecked, can be exploited by adversaries intent on creating harm. Tools that detect adversary and support remediation control are currently development stages.

The accurate discovery of malicious actions targeting PV inverters connected to the internet requires the evaluation of both signature and behavior-based features. To highlight the needs for each method, this paper deploys the two approaches into a small, single-board computer connected in a Bump-in-the-Wire (BitW) configuration with an actual PV inverter at the grid-edge. The computers are programmed to capture, inspect, store network packets, read PV inverter electrical performance values, run the signature [4] and behavior-based detection software [5]. Snort, the signature-based software, capture packets and identifies irregularities [6]. For the behavior-based software, the Adaptive Resonance Theory (ART) Artificial Neural Network (ANN) learns and finds network traffic anomalies [7]. The two approaches ran on the computer to detect several prominent cyber attacks. This work adds to the current state-of-the-art that includes the identification of PV physical faults [8], detection of cyber-attacks using physical data [9]; however minimal work, in the past, developed and/or evaluated methodologies for performing IDS of PV inverters at the grid-edge

II. Methodology

A. Test Environment

The present work deployed two IDS methodologies at the grid-edge to monitor PV inverter communications and identify cyber-attacks. The IDS, embedded inside a single-board computer, monitored traffic in-between an actual grid-tied inverter and an aggregator as shown in Figure 1.

1) Grid-Tied Inverter: The grid following PV inverter used in this work has a nameplate rating capacity of 3kVA operating at 240 V and a maximum current of 16 A. The PV inverter

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

adheres to recent communications standards, such as [10], that permit aggregators to access the sensor points and control of the Volt-Var and Frequency-Watt functions.

2) Intrusion Detection Device: A BitW device, with embedded packet capture tools and analytics, monitors and evaluates the traffic to and from the PV inverter as depicted in Figure 1. In this case, a Raspberry Pi (RPi) 3 B+ single-board computer acted as the BitW device and supported the collection and analytics of the network traffic. For this experiment, the computer specifications included a 1.4 GHz CPU, 1 GB of SDRAM, Gigabit Ethernet, and a 16 GB Micro-SD card for storage. Past work used the RPi devices of this size successfully to detect and respond to malicious behavior [11] [12].

3) Aggregator Control: The aggregator turned on or off the PV inverters Volt-Var Curve (VVC). The control logic used a basic *if* statement shown in Equation 1:

$$\text{VVC}_{\text{status}} = \begin{cases} \text{Enable}, & \text{if } \text{V}_{bus} > 243.9V \\ \text{Disable}, & \text{otherwise} \end{cases} \quad (1)$$

where the VVC was enabled when the bus voltage (V_{bus}) was greater than 243.9 V and disabled when it was less than or equal to 243.9 V. This control required the aggregator to perform both read and write requests using the Modbus protocol. First, the aggregator sent a read request to retrieve the V_{bus}. If the V_{bus} was above the defined threshold, the aggregator sent a write request to turn the VVC on.

4) Adversary: The adversary, shown in Figure 1, was assumed to have access to the aggregator/PV inverter data flows, and could send signals to the PV inverter. Visibility of data flows allowed the adversary to view and potentially modify control packet communication en route. And, its ability to access the PV inverter's controller enabled unwanted attempts to penetrate the device.

B. Experiment

The experiment used the test environment to perform two key evaluations that focused on the single-board computers ability to (1) host and (2) perform sensing and analysis functions when subjected to different cyber-attacks.

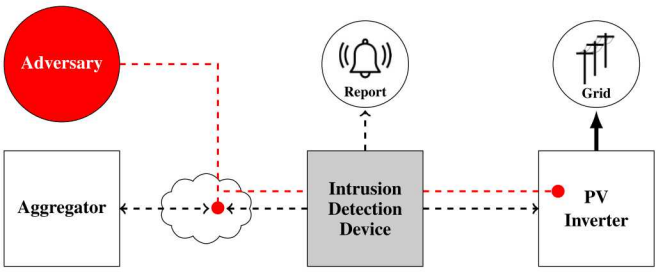


Fig. 1. The intrusion detection device monitored traffic to and from the grid connected PV inverter's control board. The device intent is to host signature or behavior-based analysis tools that detect adversary penetration attempts or attacks on data in transit.

1) Computer Performance: Small RPi devices present a low-cost option for performing sensing and analytics at the grid edge. However, computations are limited by the amount of available resources. Therefore, this paper evaluates the computers RAM, CPU temperature, and CPU usage while performing different types of analysis. The signature-based approach that uses the readily available Snort software. The behavior-based algorithms include the Support Vector Machine (SVM) [13] using Python Scikit-learn [14], Deep Autoencoder using Python TensorFlow [15], and the ART ANN.

2) Detection Comparison: Seperate from the computer performance evaluations, the identification of irregularities in network packets is performed by analyzing signatures or through the identification of anomalies in traffic behavior. This experiment deployed both approaches into the RPi device to perform intrusion detection analytics on the traffic moving to and from the inverter as depicted in Figure 2. Each approach captures the packets moving through the ethernet connections (eth0 & eth1) of the RPi device.

C. Signature-Based Analysis

Irregular network behavior is commonly identified by analyzing the signatures within a network packet. The content of a network packet includes multiple layers; the layers of a TCP/IP packet, for example, are divided into the application, transport, network, and network access layers. The analysis dissects the packets and uses rule-based evaluation methodology to identify malicious activities.

1) Detection Software: Snort is an open source IDS that has been freely available since 1998 [16]. It is a signature-based IDS that alarms on unique signatures of known malware or trends that are indicative of an attack. The software, available for download using the commands defined on the website [17], was installed on the Debian-based operating system of the RPi. Once installed, the rules were set to match the application and then programmed to run in the background to perform real-time traffic analysis. ++++++++++++++++++

2) Rules: The user-defined rules are setup to detect various types of cyber-attacks, including buffer overflows, Stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts,

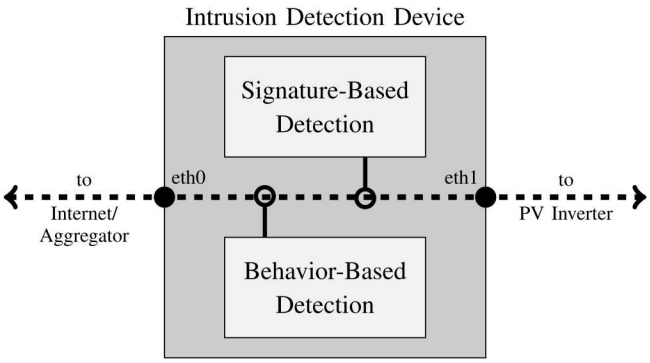


Fig. 2. The Raspberry Pi computer hosted both signature and behavior-based detection software. Each approach captures network packets and performs analysis in real-time.

etc. In this work, the setup used the default rules that came with the software that are designed to identify exploits, file executables, port scans, malware, malicious imap packets, and many others. The experiment also added Modbus rules from the the QuickDraw suite [18]. The QuickDraw rules were developed to discover denial-of-service (DOS) attacks, malformed Modbus packets, and Modbus packets that have the incorrect source and destination IP addresses.

D. Behavior-Based Analysis

Analysis of traffic patterns at the PV inverter requires a behavior-based methodology. In this case, the ART neural network algorithm used the lengths and number of packets destined for a certain device to detect issues.

1) Detection Algorithm: The ART ANN [19] works well in on-line learning applications and can operate on a small, single-board computer. The algorithm learns data features by first normalizing between zero and one and then performs a complement coding of the data. The data the category choice values are calculated using Equation 2:

$$c = \frac{|X \wedge T_j|}{\alpha + |T_j|} \tag{2}$$

where X was the preprocessed input matrix, α was the choice parameter that was set to 0.001, T_j was the template (weight) vector, and $\wedge$ is the fuzzy set theory conjunction or minimum operator. Then, a vigilance test, that used 0.95 for the free parameter ρ, was computed to find the T_j that passed Equation 3:

$$\frac{|X \wedge T_j|}{|T_j|} \geq \rho \tag{3}$$

The template with the highest choice value from Equation 2 that passed vigilance test (Equation 3) was picked and updated based on Equation 4:

$$T_j^{update} = \beta(X \wedge T_j^{old}) + (1 - \beta)T_j^{old} \tag{4}$$

where β was the learning rate set to the commonly used value of 1. In the case where no templates passed the vigilance test, the inputs were used to create a new template.

2) Algorithm Inputs: The ART algorithm analyzed various types of network packets and used key features from each type to learn and detect anomalies. The different types of protocols included the Address Resolution Protocol (ARP), Transport Control Protocol (TCP), and Modbus TCP/IP. The features included the average total length and the number of packets in a 10 second period to a specific device.

E. Attack Scenarios

The experiment exposed the two intrusion detection methodologies (signature and behavior) to five different attack scenarios. The attack scenarios, outlined in Table I, potentially impact the confidentiality, integrity, and availability of the PV inverter control system. In the first and second scenarios, the adversary attempts to send its own Modbus signals to the inverter to read or write system settings. The third attack scenario attempts to flood the PV inverter controller with TCP/IP requests in hopes of denying service to the aggregator. The fourth and fifth attack scenarios perform a Man-in-th-Middle (MitM) to either deny service or spoof data by modifying packet payloads in transit.

TABLE I
ATTACK SCENARIO OVERVIEW

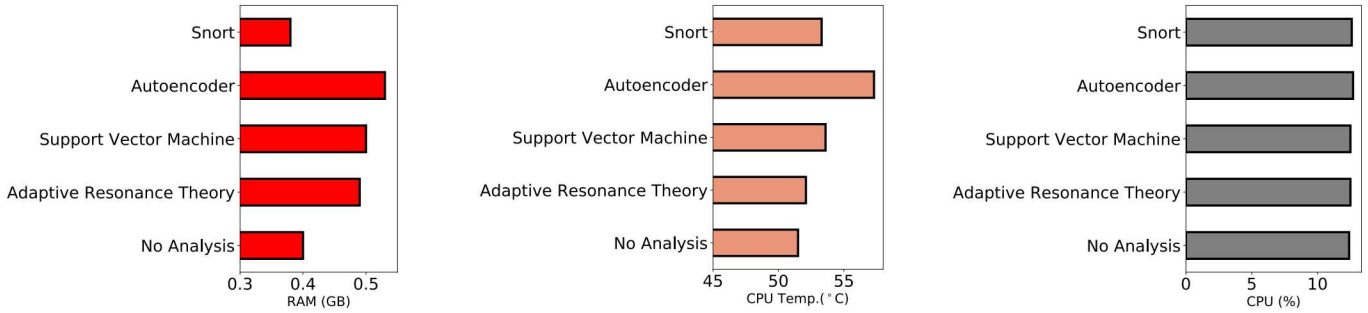
Number	Name	Description
1	Unauthorized Client	An unknown client attempts to communicate with the inverter using the Modbus protocol.
2	Invalid Packet	Modbus packet that is not a typical length and may contain malicious content.
3	Spoof TCP Handshake	Generate large amounts of synchronization requests that initiate a face Modbus TCP/IP connection from a spoofed IP address.
4	Man-in-the-Middle Denial-of-Service	Perform ARP spoof to get in-between the aggregator and the inverter and stop all traffic between the two devices.
5	Man-in-the-Middle Data Spoof	Perform ARP spoof to get in-between the aggregator and the inverter. Then, intercept answers to read requests and modify the value.

III. RESULTS

A. Computer Performance

The RPi computer, acting as an intrusion detection device, performed multiple tasks at once to sense and analyze the network traffic without reaching the maximum Random Access Memory (RAM) or exceeding a Central Processing Unit (CPU) temperature of 60°C. Under baseline operations, with no network sensor or analytics running, the computer used 0.22GB of RAM and 12.4% of the CPU, and its CPU temperature did not exceed 50°C. After turning on the Snort, signature-based network tool, the RAM increased to 0.32GB, the temperature rose to 53.3°C, and the CPU usage remained the same. The network sensor, written in Python programming language, supports the ART behavior-based analysis tool and leverages the Scapy packet sniffer package [20]. The Python-based sensor operations resulted in a RAM usage of 0.4GB, and a CPU temperature that reached 51.5°C.

The analytics provided by the Snort and three behavior-based tools used computer resources that increased the RAM, and CPU temperature as shown in Figure 3. Assessing the packet signatures using the Snort software did not impact the RAM and CPU significantly. Whereas, each of the behavior-based tools increased the RAM (Figure 3a) and CPU temperature (Figure 3b); the CPU usage, however, did not change for any of the analytics (Figure 3c). The Autoencoder produced the most significant change in RAM and temperature, followed by the SVM, and finally the ART algorithm. In addition, the amount of time to train and test for the three behavior-based algorithms on the RPi was highest for the Autoencoder (Train = 9.01sec., Test = 1.58sec.), then the ART algorithm (Train = 0.033sec., Test = 0.014sec.), and the fastest was the SVM (Train = 0.015sec., Test = 0.009sec).



(a) Random Access Memory (RAM) Usage (b) Central Processing Unit (CPU) Temperature (c) Central Processing Unit (CPU) %

Fig. 3. The signature and behavior-based tools did not exceed the limits of the Raspberry Pi computer. The signature-based tool, Snort, had the least amount of impact on the computer, while the Autoencoder behavior-based algorithm demanded the most resources to operate.

B. Intrusion Detection Performance

Overall, the behavior-based approach slightly outperformed the signature-based software for the five attack scenarios described in Section II-E as indicated in Table II.

1) Attack 1: Unauthorized Client: The two approaches both identified unauthorized clients attempting to perform Modbus read and write requests. To accomplish this, Snort required a user to input the client and server IP address; the ART algorithm did not require user input and instead learned the allowed interconnections.

2) Attack 2: Invalid Packet: The signature-based approach detected modified Modbus packets with inappropriate content that did not change the length of the packet. But, the behavior-based approach did not flag malformed packets that had typical lengths. The ART algorithm could, however, detect malicious Modbus packets where the length was altered, because its analysis considered the length as a feature in the training and testing process.

3) Attack 3: Spoof TCP/IP Handshake: The fake TCP/IP handshakes intent on flooding the Modbus controller of the PV inverter did not cause the signature-based tool to generate a flag. The behavior-based method noticed the strange behavior and produced an alarm within 10 seconds of sensing the malicious network traffic. This result is shown in Figure 4a, where the active and reactive power, and voltage remained normal, but the TCP/IP traffic increased significantly.

4) Attack 4: Man-in-the-Middle Denial-of-Service: The MitM DOS attack performed an ARP spoof that tricked the Aggregator into thinking that the adversary's computer was the PV inverter. This caused any Modbus control signals from the Aggregator to be blocked by the adversary's machine. The behavior-based analysis identified the ARP traffic as abnormal and generated a warning flag as shown in Figure 4b. However, the signature-based tool did not recognize any bad packets and therefore did not detect malicious activity.

5) Attack 5: Man-in-the-Middle Data Spoof: The final attack tested in this work created a MitM situation similar to Attack 4, but included the spoofing of values within Modbus packets. In this case, the adversary artificially lowered the voltage reading to 238 V, which was below the threshold defined in Equation 1. The altered message caused the aggregator to send a signal that turned off the VVC. This caused the reactive power to be zero until an operator could intervene and fix the issue as shown in Figure 4c. Fortunately, the two detection methods identified that unusual behavior had occured as shown in the bottom of Figure 4c. The signature-based approach noticed that a packet was altered. And, the behavior-based approach was tipped off by the unusual ARP traffic.

TABLE II
INTRUSION DETECTION RESULTS

Number	Name	Signature-Based Analysis	Behavior-Based Analysis
1	Unauthorized Client	✓	✓
2	Invalid Packet	✓	✗
3	Spoof TCP Handshake	✗	✓
4	Man-in-the-Middle Denial of Service	✗	✓
5	Man-in-the-Middle Data Spoof	✓	✓

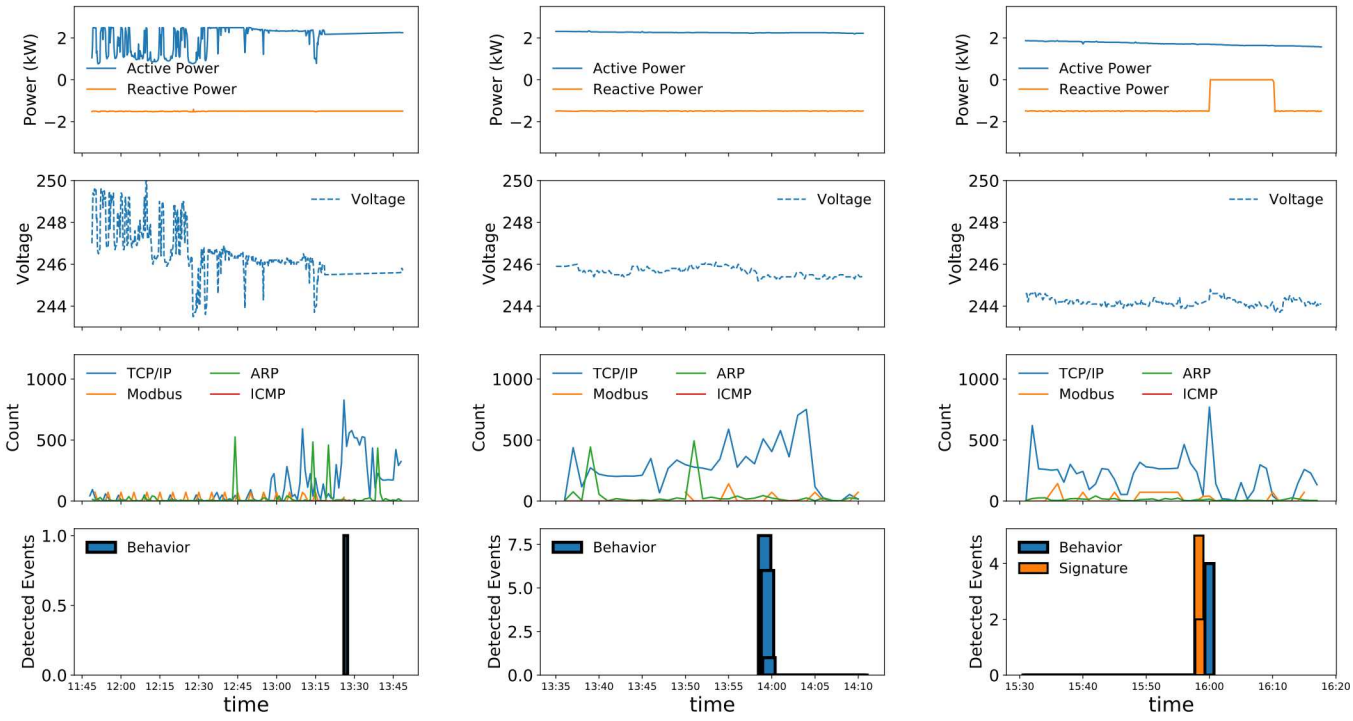
IV. CONCLUSION

In conclusion, the signature and behavior-based analysis tools successfully ran and detected malicious network activity. The two approaches ran on the small, RPi computer in a BitW configuration and generated alarms in real-time. This initial and limited analysis found evidence that the behavior-based method identified more attacks than the signature-based approach. However, future work is important for providing a statistically significant overview of the two approaches, and investigate the potential to merge signature- and behavior-based approach to create a more accurate detection system.

ACKNOWLEDGMENT

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technology Office, Award Number DE-EE00034234.



(a) Attack 3: Spoof TCP/IP Handshake

(b) Attack 4: MitM Denial of Service

(c) Attack 4: MitM Data Spoof

Fig. 4. Three attacks, TCP/IP handshake spoof, denial-of-service using a Man-in-the-Middle, and data manipulation using a Man-in-the-Middle, were detected by at least one of the techniques explored in this work. The behavior-based approach detected the TCP/IP handshake and denial-of-service. Both approaches noticed the data manipulation attack that resulted in an alteration in system performance.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

[1] J. Johnson, "Roadmap for photovoltaic cyber security," Sandia National Laboratories, Sandia Technical Report SAND2017-13262, 2017.

[2] A. S. Spanias, "Solar energy management as an Internet of Things (IoT) application," in *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, Aug. 2017, pp. 1–4.

[3] J. W. Smith, W. Sunderman, R. Dugan, and B. Seal, "Smart inverter volt/var control functions for high penetration of PV on distribution systems," in *2011 IEEE/PES Power Systems Conference and Exposition*, Mar. 2011, pp. 1–6.

[4] T. Crothers, *Implementing Intrusion Detection Systems: A Hands-On Guide for Securing the Network*, 1st ed. Wiley Publishing, Inc, 2003.

[5] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[6] R. Martin, "Snort: Lightweight intrusion detection for networks," in *Lisa*, vol. 99, Nov. 1999, pp. 229–238.

[7] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, pp. 54–115, Jan. 1987.

[8] A. Mellit, G. M. Tina, and S. A. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, Aug. 2018.

[9] D. M. Shilay, K. G. Lorey, T. Weiz, T. Lovetty, and Y. Cheng, "Catching Anomalous Distributed Photovoltaics: An Edge-based Multi-modal Anomaly Detection," Sep. 2017. [Online]. Available: https://arxiv.org/abs/1709.08830v1

[10] T. Basso, S. Chakraborty, A. Hoke, and M. Coddington, "IEEE 1547 Standards advancing grid modernization," in *2015 IEEE 42nd Photovoltaic Specialist Conference (PVSC)*, Jun. 2015, pp. 1–5.

[11] C. B. Jones and C. Carter, "Trusted Interconnections Between a Centralized Controller and Commercial Building HVAC Systems for Reliable Demand Response," *IEEE Access*, vol. 5, pp. 11 063–11 073, 2017.

[12] C. B. Jones, C. Carter, and Z. Thomas, "Intrusion Detection Response using an Unsupervised Artificial Neural Network on a Single Board Computer for Building Control Resilience," in *2018 Resilience Week (RWS)*, Aug. 2018, pp. 31–37.

[13] N. Cristianini, J. Shawe-Taylor, and D. o. C. S. R. H. J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Mar. 2000, google-Books-ID: _PXJn_cxv0AC.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, 2011.

[15] G. Zaccone and M. R. Karim, *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python, 2nd Edition*. Packt Publishing Ltd, Mar. 2018, google-Books-ID: zZlUDwAAQBAJ.

[16] R. U. Rehman, *Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID*. Prentice Hall Professional, 2003, google-Books-ID: 1WKrLbh23LAC.

[17] "Snort - Network Intrusion Detection & Prevention System," 2019. [Online]. Available: https://www.snort.org

[18] J. Nivethan and M. Papa, "Dynamic rule generation for SCADA intrusion detection," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016, pp. 1–5.

[19] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, Jan. 1991.

[20] T. J. O'Connor, *Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers*. Newnes, Dec. 2012, google-Books-ID: 2XliiK7FKoEC.