Performance Evaluation of Vertical Federated Machine Learning Against Adversarial Threats on Wide-Area Control System

Preprint

Ethan Tucker, Rakib Hossain, and Vivek Kumar Singh

National Renewable Energy Laboratory

Presented at Resilience Week 2024
Austin, Texas
December 3-5, 2024

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5T00-89951
December 2024

Performance Evaluation of Vertical Federated Machine Learning Against Adversarial Threats on Wide-Area Control System

Preprint

Ethan Tucker, Rakib Hossain, and Vivek Kumar Singh

National Renewable Energy Laboratory

Suggested Citation
Tucker, Ethan, Rakib Hossain, and Vivek Kumar Singh. 2024. *Performance Evaluation of Vertical Federated Machine Learning Against Adversarial Threats on Wide-Area Control System: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5T00-89951. https://www.nrel.gov/docs/fy25osti/89951.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5T00-89951
December 2024

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed herein do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Performance Evaluation of Vertical Federated Machine Learning Against Adversarial Threats on Wide-Area Control System

Ethan Tucker
National Renewable Energy Laboratory
Golden, Colorado
Email: ethan.tucker@nrel.gov

Rakib Hossain
National Renewable Energy Laboratory
Golden, Colorado
Email: rakib.hossain@nrel.gov

Vivek Kumar Singh
National Renewable Energy Laboratory
Golden, Colorado
Email: vivekkumar.singh@nrel.gov

Abstract—**Federated machine learning (FL) is gaining significant popularity to develop cybersecurity solutions in power grids because of its advanced capability to support decentralized data handing at local devices, its privacy preservation, and its low-bandwidth requirement. However, the evolving adversarial machine learning (AML) threats raise significant concerns for the cybersecurity of FL architectures. The FL-based split neural network (SplitNN) achieves high performance through the decentralized training of local neural network models while preserving data privacy across multiple entities. In this paper, we propose a methodology for evaluating the performance of a vertical FL-based anomaly detector against different types of AML attacks, including denial-of-service attacks, adversarial data injection attacks, and replay attacks on the trained local models deployed in the grid network. For a case study, we consider the modified IEEE 13-bus system, and we develop SplitNN-based binary and multiclass classification models to detect, locate, and identify different types of data integrity attacks on the volt-watt control with two pooling layers: maximum pooling and AvgPool. Our experimental results, computed through performance metrics, reveal that the severity of these AML attacks varies with the integrated pooling mechanism, the type of classification model, and the nature of the cyberattack. Further, the AML attacks negatively impacted the prediction time per sample for the pretrained SplitNN during the online testing.**

Index Terms—**Federated machine learning, split neural network, adversarial threats, cybersecurity, power grid.**

I. Introduction

The modern power grid is a highly complex and interconnected cyber-physical system that combines digital technology with physical infrastructure. The substantial advancement in cyber technologies aimed at enhancing grid intelligence has propelled the energy industry into a new era characterized by improved reliability, sustainability, and efficiency. This progress necessitates a greater reliance on robust communication infrastructure. While the rise of Cyber-Physical Systems (CPS) is central to the functioning of the modern power grid, it has also made the grid network more susceptible to various cyberattacks [1], [2]. Some cybersecurity incidents, including the notorious Stuxnet worm attack in 2010 [3] and the Ukraine power grid hacks in 2015 [4], have severely impacted industrial control systems (ICS) and affected public and private sector customers. These cybersecurity incidents show the interdependencies between ICS and digital infrastructure and reveal how cyberattacks can lead to compromises in regular operations, leading to financial losses [5].

Federated machine learning (FL) has emerged as a promising approach to bolster the cybersecurity defenses in power grid networks by addressing challenges related to data distribution, privacy preservation, and communication bandwidth optimization. The authors of [6], [7] showed that the performance of FL models was comparable to conventional machine learning models for detecting anomalies in smart meter and synchrophasor data. Split neural network (SplitNN) is one of the promising vertical FL (VFL) approaches that preserve data privacy by substituting sensitive data communications for intermediate neural network activations. Further, it offers significant advantages in terms of reducing the computational expenses associated with training deep neural networks while ensuring the privacy-aware use of data shared among the clients. It accomplishes this task by dividing a neural network across the local clients, deployed at substations, and control center-based global model. In 2020, the authors of [8] showed the benefits of integrating several pooling functions, including max pooling, average pooling, etc. at the global model to combat client dropouts during the model training. These pooling layers significantly contribute to make SplitNN more efficient, robust, and effective by reducing dimensionality, preserving important features, and improving generalization while managing computational resources effectively.

As with any evolving technology, however, is not immune to adversarial machine learning (AML) threats [9], which pose unique challenges to the security and privacy of the SplitNN architecture. For example, adversaries can manipulate input features, poison training datasets, and compromise model updates, and hence affect the confidentiality, integrity, and availability of distributed machine learning models. A recent survey [10] highlighted the existing vulnerabilities in communication protocols and discussed different types of poisoning attacks, inference attacks, and generative adversarial network attacks in the FL architecture.

In this paper, we propose a methodology for evaluating the performance of trained VFL against different types of AML threats. The proposed approach consists of several steps, including generating datasets for different cyber-physical scenarios for a given wide-area controller, training VFL with a global pooling layer using generated datasets, testing the trained model against different AML attacks, and, finally, evaluating the pooling functions using various performance measures. In this work, we consider SplitNN-based VFL, and we perform adversarial attacks during the transfer of the local model outputs to the global model during the testing. Further, a voltage-watt controller is applied to regulate the voltage in the IEEE 13-bus system and to generate datasets through various cyber-physical scenarios. Finally, several performance metrics—including accuracy, precision, and F1 score—are applied to evaluate and compare different pooling functions in the SplitNN-based VFL.

II. Related Work and Overview

Previous research efforts have applied several FL algorithms to detect cybersecurity attacks in distributed energy resources (DERs)-integrated grids. The authors of [11] showed the better performance of decentralized FL compared to centralized machine learning algorithms against false data injection attack in photovoltaic (PV) systems. In [12], the authors combined the federated k-means clustering algorithm with variational mode decomposition and SecureBoost to improve the prediction performance for the day-ahead load forecasting. The authors of [13] showed the benefits of applying both horizontal and VFL in protecting user privacy, securing power traces, and preventing data leakage for power consumption datasets. A distributed deep learning-based VFL method, SplitNN, was introduced in [14], which showed efficient performance compared to other federation algorithms in terms of validation accuracy and computational resources. Further, several configurations of SplitNN using various pooling functions—including element-wise average, maximum (max), sum, multiplication, and concatenation—were evaluated, where element-wise average and max showed consistent and efficient performance across various datasets.

Yet none of these works focused on addressing the evolving vulnerabilities in FL architectures that might have impacted the performance of the aforementioned proposed models. A recent survey on FL threats [15] highlighted how this new learning paradigm is subjected to several adversarial machine learning (AML) attacks and emphasized the importance of developing state-of-the-art defense solutions to ensure the secure operation of FL architectures. In this work, we perform two layers of cyberattacks in the distribution grid network while considering a wide-area controller. In Layer 1, cyberattacks are performed on measurement and control signals of a wide-area controller to develop binary and multiclass classification models using the SplitNN algortithm. In Layer 2, adversarial attacks are performed during the online testing of the trained SplitNN models, and the performance is evaluated for different configurations (pooling functions). To the best of our knowledge, this is the first work to discuss the performance of a trained vertical SplitNN against AML threats in the context of grid cybersecurity.

A. Overview of Vertically Partitioned SplitNN

SplitNN is a VFL paradigm wherein a machine learning model is divided into different segments that are distributed across several devices or parties [16]. In this work, the portion of the SplitNN model is at grid node c, where $c \in \{c\}_{c=1}^{N_c}$ is denoted by the local model, f^c_{local}. Here, N_c represents the total number of grid nodes. The portion of the SplitNN placed at the control center is referred to as the global model, f_{global}. Each local data batch D^c is passed through its corresponding local model f^c_{local}, as shown by equation 1, to create a total of N_o local outputs. The local outputs from f^c_{local}, represented by $\{a^c_o\}_{o=1}^{N_o}$, are then transmitted to the control center. Note that N_o must be equal across each f^c_{local}. Further, for each batch of data, the control center receives a set of successful local outputs from N_s local clients, where $N_s \leq N_c$:

$$\{\{a^s_o\}_{o=1}^{N_o}\}_{s=1}^{N_s} = \{f^s_{\text{local}}(D^s)\}_{s=1}^{N_s} \tag{1}$$

Upon receiving a successful batch of local outputs, f_{global}, deployed at the control center, passes the local outputs through a pooling layer, $P : (\mathbb{R}_{N_o}, \mathbb{R}_{N_s}) \rightarrow \mathbb{R}_{N_o}$, to reduce the dimension of the input to a consistent form. The pooling layer mitigates communication failures by allowing N_s to vary per batch. The detailed operation of the pooling layers is provided in the following subsections. Next, the global model does a forward pass to create an event prediction, $\hat{y} = f_{\text{global}}(\{\{a^s_o\}_{o=1}^{N_o}\}_{s=1}^{N_s})$.

Two different global pooling functions for developing SplitNN-based classification models are discussed here:

1) Output-wise average pooling

Average pooling (AvgPool), introduced in [17], is a common pooling function wherein a region of data are averaged. At the control center, we perform pooling operations neuron-wise on the local output values to create a consistent number of inputs to the global model. In particular, AvgPool performs the following computation on the successful transfer of local outputs from (1) to create the global model inputs, $\{\text{Avg}_o\}_{o=1}^{N_o}$:

$$\{\text{Avg}_o\}_{o=1}^{N_o} = \{\frac{1}{N_s}\sum_{s=1}^{N_s} a^s_o\}_{o=1}^{N_o} \tag{2}$$

2) Output-wise max pooling

AvgPool is compared against another common pooling function, known as max pooling (MaxPool). MaxPool, first used in [18], selects the maximum value from a region of data rather than the average value. MaxPool is also applied neuron-wise, and it creates the global model inputs, $\{\text{Max}_o\}_{o=1}^{N_o}$:

$$\{\text{Max}_o\}_{o=1}^{N_o} = \{\max_s(\{a^s_o\}_{s=1}^{N_s})\}_{o=1}^{N_o} \tag{3}$$

B. Overview of Volt-Watt Control

Volt-watt control (VWC) addresses the voltage fluctuation and violation problem of the grid by dynamically adjusting the active power output of the inverters based on the grid voltage level. When the grid voltage exceeds a certain upper threshold, indicating high grid voltage conditions, the inverters reduce their power output to help stabilize the grid voltage. Conversely, when the grid voltage exceeds a certain lower threshold, indicating low grid voltage conditions, the inverters increase their power output to support the grid voltage. In our approach, we use a VWC curve to determine the amount of active power that needs to be absorbed by or injected into the inverters to control the network voltage. Fig. 1 represents the volt-watt curve used to control the active power set point of the inverters. The slope of this curve determines how the active power output changes in response to variations in voltage at the point. The mathematical presentation of the curve is shown in (4).

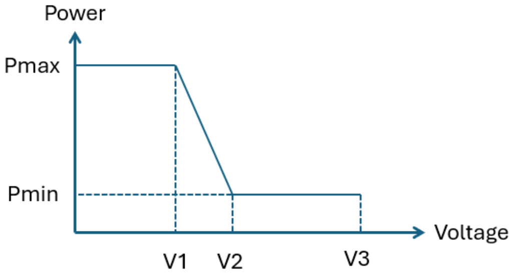

Fig. 1: Graphical representation of the volt-watt curve

$$P(V_i) = \begin{cases} P_{max} & \text{if } V_i \leq V_1 \\ \frac{V_i - V_1}{V_1 - V_2} \times (P_{max} - P_{min}) + P_{min} & \text{if } V_1 < V_i \leq V_2 \\ P_{min} & \text{if } V_2 < V_i \leq V_3 \end{cases} \tag{4}$$

where P_{max} and P_{min} represent the maximum and minimum power of the DERs, and V_1 and V_3 indicate the minimum and minimum node voltages, respectively. We set V_1 and V_2 to 0.95 p.u and 1.05 p.u, respectively, according to the American National Standards Institute (ANSI) [19] standard.

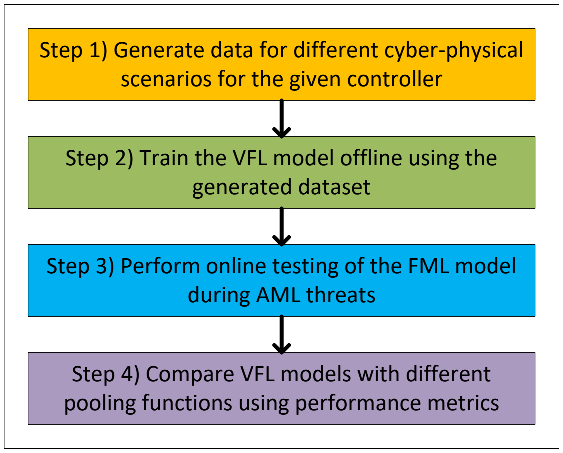
Step 1) Generate data for different cyber-physical scenarios for the given controller
Step 2) Train the VFL model offline using the generated dataset
Step 3) Perform online testing of the FML model during AML threats
Step 4) Compare VFL models with different pooling functions using performance metrics

Fig. 2: Proposed methodology for evaluating the FL architecture against adversarial threats

III. Impact Analysis of Adversarial Attacks

Fig. 2 illustrates the detailed steps required for evaluating VFL in the presence of adversarial threats in real time. These steps ensure that any kind of VFL could be evaluated for different types of AML threats as necessary to performance quantitative risk assessments in the context of artificial intelligence cybersecurity in the grid network.

Step 1: Generate datasets for different cyber-physical scenarios for the given controller.

In this step, we create several operating points using the load profile that varies every 15 minutes for the given distribution grid. Further, different types of data integrity attacks (pulse, ramp and scale) are simulated on both measurement and control signals while considering periodic and regular operation of the VWC. The generated datasets are labeled to facilitate the training of the supervised learning-based SpilitNN. The classification schemes used by SplitNN are binary, attack location, and attack type classification. Binary classification detects any attack that happened or not on wide-area signals, location-based classification identifies the location of the attack, and attack type classification differentiates between different types of attacks and normal operation.

Step 2: Train SplitNN for anomaly detection system.

SplitNN is trained offline to classify the cyberattacks described in the previous section. Each f^z_{local} is initialized with an equal number of local model outputs, N_o. During training, a loss function, L, compares a true label, $\mathbf{y}$, against $\hat{y}$ to determine the classification error. The loss function is either binary or categorical cross-entropy, depending on the number of classes for the task. For each batch, global weight gradients, ∇_{global}, are computed by taking the partial derivative of the global weights with respect to the loss. The Adam optimizer, as introduced in [20], then updates the global model weights using ∇_{global}. Next, the control center sends each local client, s, that contributed to the batch its respective gradients, ∇_s, which are then used to update f^s_{local}:

$$\nabla_s = \frac{\partial L(\mathbf{y}, \hat{y})}{\partial \{a^s_o\}^{N_o}_{o=1}} \tag{5}$$

After each training epoch, the validation loss is computed using a reserved dataset. When the validation loss fails to decrease for a prespecified number of consecutive epochs, the learning rate is decreased to fine-tune the SplitNN. Training is terminated when the validation loss fails to decrease for a larger number of consecutive epochs. At the end of the training process, the weights of SplitNN's local and global models are restored to the epoch that yielded the minimum validation loss.

Step 3: Perform online testing of FL to detect attacks with AML threats.

In this step, two levels of cyberattacks are considered on the pretrained SplitNN over the wide-area network (WAN). In the Level 1 (primary) attack, we assume that the adversary is able to sniff the communication traffic and perform data integrity attacks, as defined in Step 1, targeting wide-area signals for the VWC. For the Level 2 (secondary) attacks, we assume that the attacker can perform single AML cyberattacks targeting one of the local machine learning models as well as multiple AML attacks where multiple local models are compromised at the same time. In particular, we consider three AML attacks:

Denial-of-service attack: A denial-of-service (DoS) attack is a communication failure attack on the network infrastructure to disrupt the communication between the client and the sever. In this work, we aim to disable a local client from communicating with the global server [21], [22]. Specifically, the attacker intercepts and destroys a communication, $\{a_o^z\}_{o=1}^{N_o}$, thereby decreasing the number of successful communications, N_s, by one per attack.

Adversarial data injection attack: In an adversarial injection attack, the attacker attempts to subvert the global pooling layer by replacing local outputs with large values. During a first epoch over the test set, the attacker listens to the entire set of local outputs for the target client and records the largest values neuron-wise (N_o in total). The largest local output values, $\{L_o^s\}_{o=1}^{N_o}$, for a given target client are represented by (6):

$$\{L_o^s\}_{o=1}^{N_o} = \{\max_i\{a_{o,i}^s\}_{i=1}^{N_\text{test}}\}_{o=1}^{N_o} \tag{6}$$

During a second epoch over the test set, the attacker substitutes the true local outputs from the target client with the recorded values. This is done to ensure that the injected values come from the real population distribution of the local outputs from the targeted f_local^z while exploiting the global model's pooling function.

Replay attack: During a replay attack, the adversary loops the local outputs of the target client with real values from a previous time [23]. Similar to the adversarial data injection (ADI) attack, the adversary first observes a first epoch over the test set. The attacker records the *entire* set of local outputs from the target client, denoted by $\{\{a_{o,i}^s\}_{o=1}^{N_o}\}_{i=1}^{N_\text{test}}$. At each time, t, during the second epoch over the test set, the attacker replaces the target local outputs, $\{a_{o,t}^s\}_{o=1}^{N_o}$, with a different randomly chosen recorded set (i.e., $i \neq t$). The set of injected values for the target client at time i are given by (7):

$$\{R_{o,i}^s\}_{o=1}^{N_o} = \{a_{o,t}^s \mid i \neq t\}_{o=1}^{N_o} \tag{7}$$

Step 4: Compare SplitNN models with different pooling functions. Performance metrics for the SplitNN are computed during online testing using the performance metrics based on the generated predictions. These metrics include accuracy, precision, F1 score, and prediction time per sample. Further, we consider two different pooling mechanisms, AvgPool and MaxPool, while evaluating the performance of SplitNN against AML threats.

IV. Simulation Setup

For a case study, we simulate the modified IEEE 13-bus test system using the OpenDSS software (see Fig. 3).

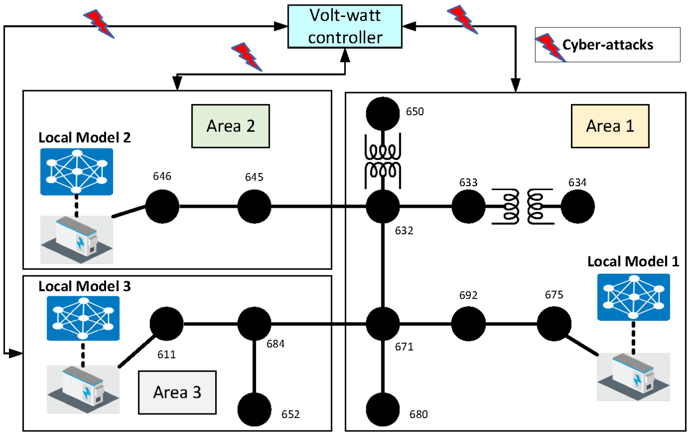

Fig. 3: BESS-integrated IEEE 13-bus system during primary attacks

This system is divided into three regions, and three BESS are connected at three nodes (675, 646, 611). The system loads are varied every 15 minutes based on the aggregated load profile of a physical site in Henderson, Nevada [24], to create several operating points. To develop the SplitNN-based anomaly detection system, three types of single data integrity attacks—pulse, ramp, and scale, with defined parameters, as shown in Table I—are injected into the measurement and control signals of the VWC (see Fig. 3). The generated datasets are used for training the local models in Area 1, Area 2, and Area 3 to perform three types of classification: binary, attack location, and attack types. Binary classification includes two labels: attack and no attack; attack location classification includes four labels: no attack, Area 1, Area 2, and Area 3; and attack type classification includes four labels: no attack, pulse, ramp, and scale attacks.

TABLE I: Attack scenarios for training and testing models

Attack Type	Attack Parameters	Area
Pulse	Magnitude =[0.9, 1.1, 1.4], Duty Cycle = [0.3, 0.5, 0.8]	A1, A2, A3

Ramp	Ramp = [0.5, 0.7, 0.3],
	Direction = [-1, 1]
		A1, A2, A3

Scale	[1.1, 0.9]	A1, A2, A3

Fig. 5 presents the simulation setup for training SplitNN using the data aggregators from Area 1, Area 2, and Area 3. Hyperparameter and model architecture tuning is performed using random search, introduced by [25], using 30 trials each. The chosen architecture and hyperparameters are selected by the minimum validation loss. The model architectures and

TABLE II: SplitNN architectures and hyper-parameters

	A	B	C	D	E
Binary Classification					
MaxPool	2	110	2	96	2.344e−3
AvgPool	2	71	3	65	3.332e−3
Attack Location Classification					
MaxPool	3	91	1	111	8.885e−3
AvgPool	2	109	1	26	3.682e−4
Attack Type Classification					
MaxPool	2	110	1	95	1.713e−3
AvgPool	1	56	2	118	1.345e−3

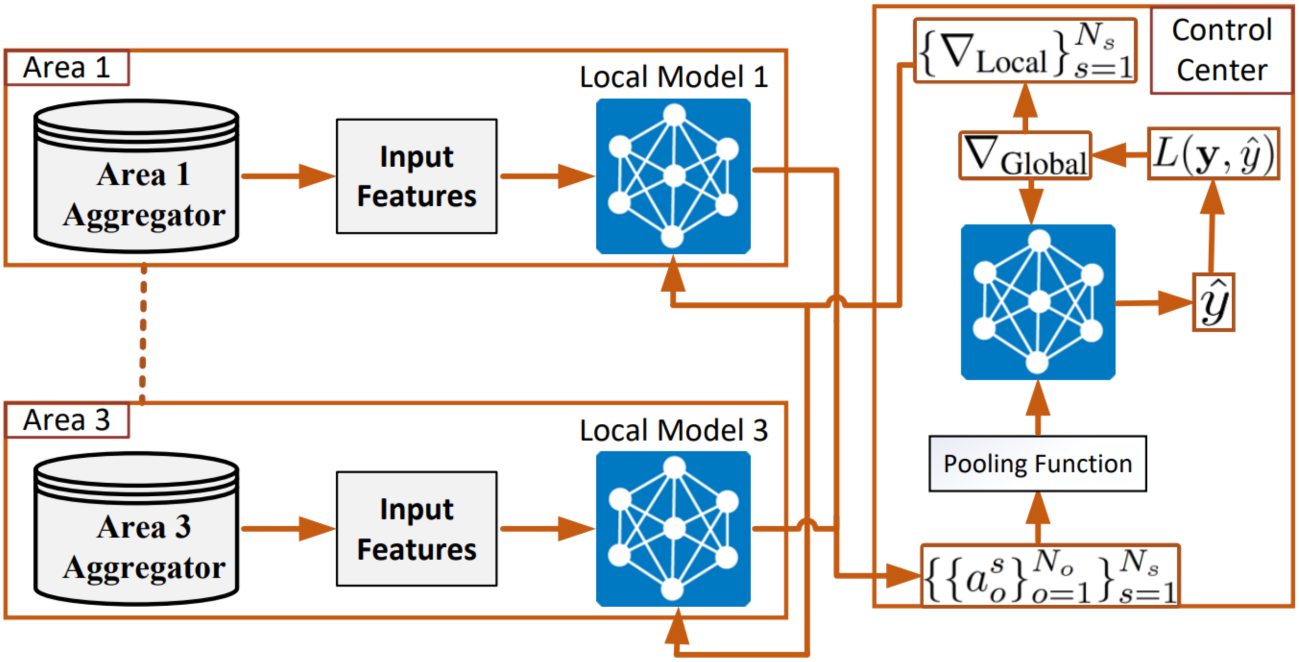

Fig. 4: Simulation setup for training SplitNN

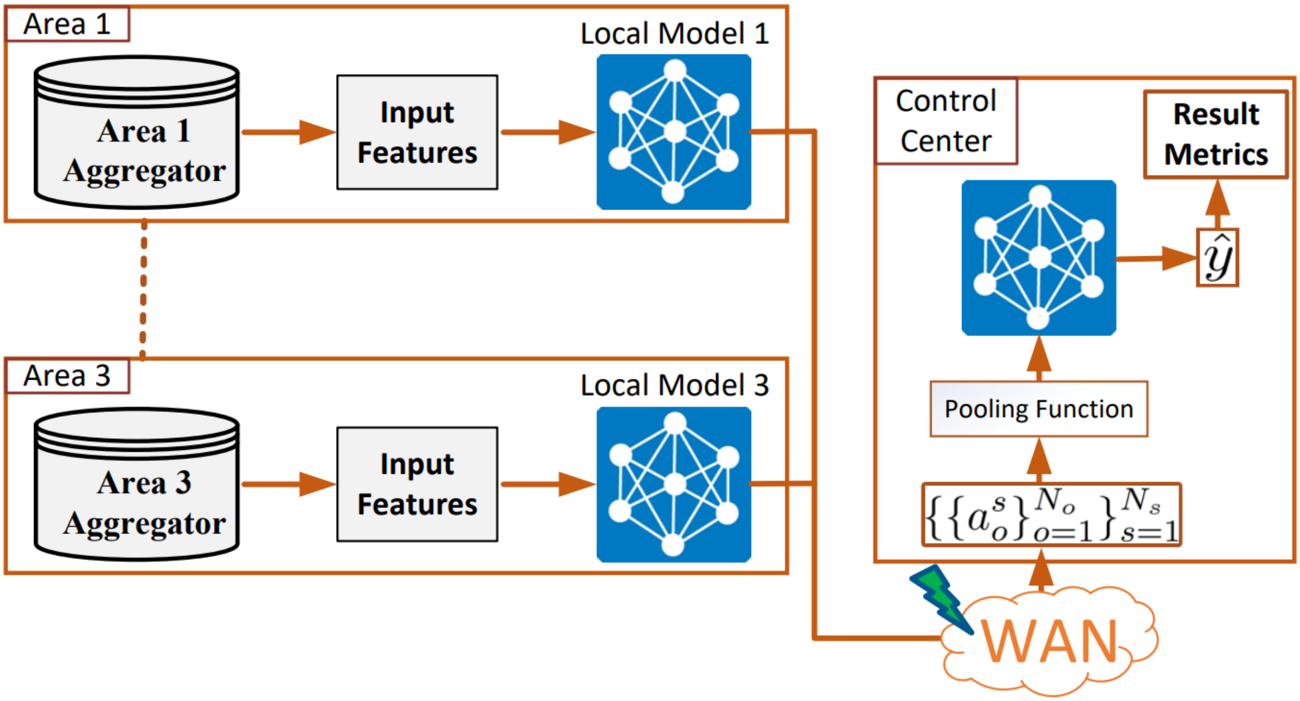

Fig. 5: Simulation setup for testing SplitNN against AML threats

hyperparameters used for each classifier are given in Table II. The columns of Table II are as follows: (A) number of hidden layers per f_{local}^c, (B) number of neurons per hidden layer in f_{local}^c, (C) number of hidden layers in f_{global}, (D) number of neurons per hidden layer in f_{global}, and (E) initial learning rate. Each local model is initialized with $N_o = 6$. The learning rate decay factor is set to 50% after 3 consecutive unsuccessful training epochs, down to a minimum of 10^{-7}, and training is terminated after 15 consecutive unsuccessful training epochs.

During the online testing, both the primary attacks against the VWC and the secondary attacks against the local model outputs are launched over the WAN. The testing setup is shown in Fig. 5. The pretrained SplitNN is trained to perform only one of the binary, attack location, or attack type classification for the primary attacks. We compare the performance of the

SplitNN with two global pooling functions, AvgPool and MaxPool, during various AML scenarios.

V. Result Analysis

The performance of the proposed SplitNN-based approach with two different global pooling layers is tested and evaluated during real-time testing against three AML attacks. Several case studies for these three attacks are evaluated using performance metrics, as shown from Fig. 6 to Fig. 9. In particular, we consider seven cases, where Case 0 indicates normal operation of the SplitNN with no AML threat; Case 1 through Case 3 indicate a particular AML attack on Local Model 1, Local Model 2, and Local Model 3; and Case 4 through Case 6 indicate coordinated AML attacks on any of the two local models, respectively.

Fig. 6 and Fig. 7 show the accuracy metric and F 1 scores of the SplitNN for detecting primary attack scenarios when DoS attacks are performed on local models for different cases. The SplitNN shows robust performance during the binary classification compared to the attack location and attack type classifications with both the MaxPool and AvgPool layers. Because the attack location and attack type classifications have multiple labels, the DOS attacks have a more severe impact in different cases. It also means that the SplitNN is more resilient against DoS attacks while performing binary classification than the other classifications.

Fig. 8 and Fig. 9 represent the accuracy and F 1 scores during the ADI attacks. In this case, both pooling layers are vulnerable to high-valued fake data because MaxPool passes the highest valued neuron element-wise across clients, and AvgPool computes an element-wise average. During the binary classification, both AvgPool and MaxPool are entirely subverted by the ADI attack. During the attack type classification, MaxPool mitigates the ADI *slightly* better than AvgPool during cases 1, 3, and 5. Note that outside of Case 1, both AvgPool and MaxPool are heavily subverted by the ADI attack during the attack type classification. In the attack location classification, MaxPool performs much better during Case 3 and slightly better in Case 6, but it worse or equal during the other experimental cases. In general, MaxPool displays a more volatile mitigation ability than AvgPool during the ADI attack.

Fig. 10 and Fig. 11 depict the accuracy and F1 score during the replay attacks. Like the ADI attacks, the replay attack is a data injection. Moreover, MaxPool displays the same volatility in mitigation ability compared to AvgPool. The binary classification performances are similar to the ADI scenario, with MaxPool outperforming in cases 3 and 5 but falling behind or approximately equal in the other cases. The attack type performances exhibit the same comparison. During the attack location classification, the MaxPool mitigation volatility is even more drastic. This is highlighted by outperforming AvgPool during cases 3, 5, and 6 while performing much worse during cases 1, 2, and 4.

Table III summarizes the SplitNN performance during the AML attacks. It averages each computed performance metric by the number of clients targeted by the AML threats. Table III shows that AvgPool has better performance than MaxPool during DoS attacks during all three classification schemes. In contrast, MaxPool is more resilient than AvgPool against ADI attacks during the attack location and attack type classifications. Both MaxPool and AvgPool are entirely subverted into the null model during ADI attacks on binary classifiers, as the accuracy for each is 50%. For the replay attack, MaxPool

TABLE III: Averaged performance metrics of VFL models during testing with various AML scenarios

Scenario	# Targets	AvgPool				MaxPool			
		Accuracy	Precision	F1 Score	Time/Sample (ms)	Accuracy	Precision	F1 Score	Time/Sample (ms)
Binary Classification									
No AML Attack	0	0.9878	1.0	0.9877	0.1818	0.9773	1.0	0.9768	0.1658
DoS Attack	1	0.9543	0.9975	0.9522	0.1252	0.9274	1.0	0.9206	0.1045
	2	0.8494	0.9816	0.8250	0.0945	0.7960	1.0	0.7213	0.0884
ADI Attack	1	0.5000	0.5000	0.6667	0.0846	0.5000	0.5000	0.6667	0.0825
	2	0.5000	0.5000	0.6667	0.0854	0.5000	0.5000	0.6667	0.0789
Replay Attack	1	0.7930	0.7278	0.8192	0.8434	0.8037	0.7585	0.8230	0.7918
	2	0.6520	0.6061	0.7140	1.6147	0.6520	0.6156	0.6999	1.5166
Attack Location Classification									
No AML Attack	0	0.9576	0.9578	0.9575	0.2504	0.8911	0.8978	0.8902	0.1461
DoS Attack	1	0.8713	0.8836	0.8703	0.0872	0.6550	0.5698	0.5879	0.0938
	2	0.5884	0.5840	0.5491	0.0721	0.4430	0.2949	0.3282	0.0744
ADI Attack	1	0.3634	0.4044	0.2486	0.0624	0.4814	0.4254	0.3940	0.0656
	2	0.2500	0.0625	0.1000	0.0665	0.2677	0.1470	0.1306	0.0663
Replay Attack	1	0.6911	0.7036	0.6872	0.9667	0.6574	0.6666	0.6556	0.7927
	2	0.4496	0.4567	0.4407	1.4649	0.4397	0.4472	0.4375	1.6186
Attack Type Classification									
No AML Attack	0	0.7055	0.7535	0.6815	0.2289	0.5901	0.6932	0.5545	0.1278
DoS Attack	1	0.5062	0.5865	0.4645	0.0792	0.4123	0.4089	0.3488	0.0816
	2	0.3136	0.2276	0.2451	0.0695	0.2659	0.1848	0.1785	0.0660
ADI Attack	1	0.3292	0.2031	0.2159	0.0628	0.3608	0.2952	0.2702	0.0661
	2	0.1910	0.0427	0.0687	0.0619	0.2117	0.0710	0.0973	0.0654
Replay Attack	1	0.4684	0.4865	0.4489	1.1536	0.4317	0.4921	0.3991	0.7753
	2	0.2810	0.2855	0.2668	1.5152	0.2810	0.3069	0.2534	1.4783

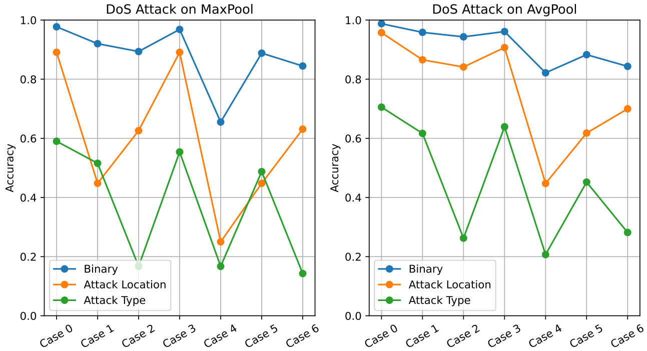

Fig. 6: Accuracy metric against the DoS attack

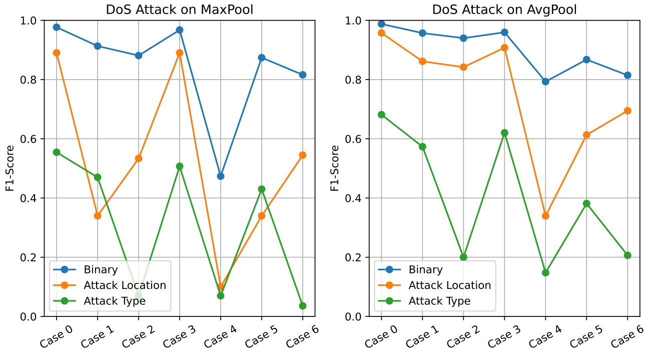

Fig. 7: F 1 score against the DoS attack

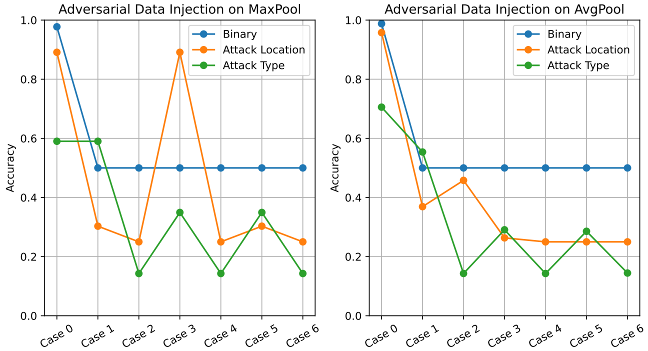

Fig. 8: Accuracy metric against the ADI attack

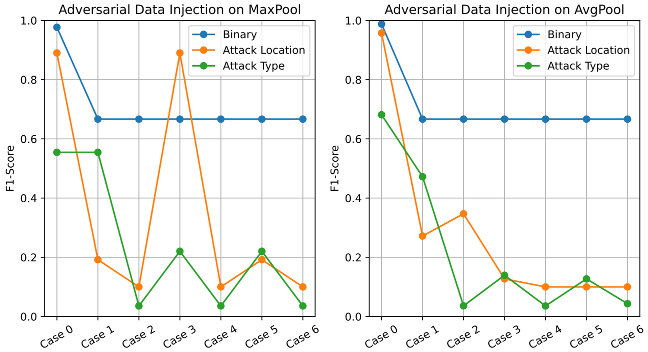

Fig. 9: F1 score against the ADI attack

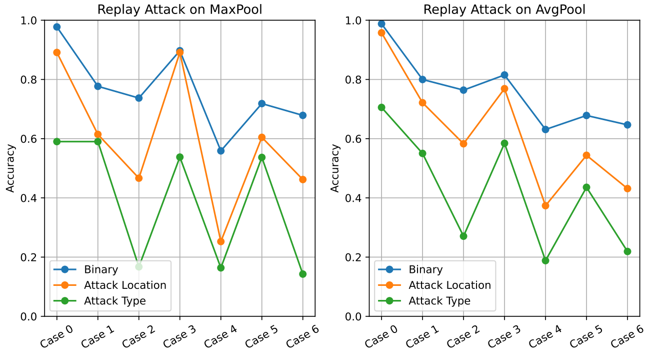

Fig. 10: Accuracy metric against the replay attack

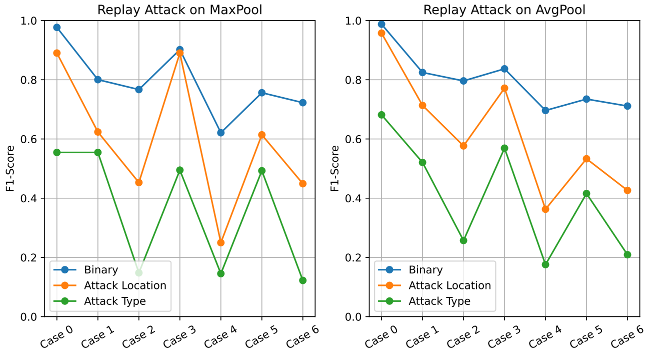

Fig. 11: F1 score against the replay attack

performs better during cases 1, 2,and 3 during the AML attacks on the binary classifier but worse or equal to AvgPool during all other scenarios. Finally, for each classification scheme, AvgPool has a higher baseline performance than MaxPool at detecting primary attacks when no AML threats are present.

Note that we also compute the average processing time/sample for both pooling layers during the cyberattacks and normal operation. We observe that the average processing time/sample during normal operation is higher for AvgPool than MaxPool because of the added computational complexity during the averaging process. In the case of AML-based DoS and ADI attacks, the average processing time/sample decreases in both pooling layers for all three classification models; however, it increases to approximately 1.5 milliseconds during replay attacks. Note that calculating replay attacks has a higher computational complexity than calculating the other two AML attacks.

VI. Conclusions and Future Work

This paper proposed a methodology for evaluating the performance of VFL against different types of AML attacks during online testing in the grid network. Initially, the SplitNN-based VFL was considered for developing three classification models for detecting, locating, and identifying different types of (primary) cyberattacks on wide-area signals of the VWC in the grid network. These cyberattacks include pulse, ramp, and scale attacks, which were simulated across three different zones in the distribution grid. Later, we described AML-based cyberattacks, including DoS, ADI, and replay attacks, which were launched during the testing phase of the trained SplitNN network, despite being absent during the training phase.

Two pooling layers, AvgPool and MaxPool, were considered during the simulation-based testing of the applied SplitNN network. Our simulation-based evaluation showed that case studies related to applying multiple AML attacks at the same time severely impact the regular operation of the trained SplitNN compared to single AML attacks. Also, the impact of these AML attacks varies with the nature of the attack and the pooling layers. Further, the computed performance metrics showed that the SplitNN is more capable of reducing the impact of a DoS attack than ADI and replay attacks. We also observed that the processing time/sample is higher during replay attacks because of the added computational complexity during testing.

For future work, we plan to develop an attack-resilient pooling layer by incorporating concatenation and stochastically chosen order statistic functions. We will further evaluate the performance of VFL models under various AML threats, incorporating multiple local models in large-scale power grid networks.

Acknowledgments

This work was authored by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

[1] N. Kshetri and J. Voas, "Hacking power grids: A current problem," *Computer*, vol. 50, no. 12, pp. 91–95, 2017.

[2] V. K. Singh and M. Govindarasu, "A cyber-physical anomaly detection for wide-area protection using machine learning," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3514–3526, 2021.

[3] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, 2011, pp. 4490–4494.

[4] A. Shehod, "Ukraine power grid cyberattack and us susceptibility: Cybersecurity implications of smart grid advancements in the us," *Cybersecurity Interdisciplinary Systems Laboratory, MIT*, vol. 22, pp. 2016–22, 2016.

[5] V. Kumar Singh, A. Ozen, and M. Govindarasu, "Stealthy cyber attacks and impact analysis on wide-area protection of smart grid," in *2016 North American Power Symposium (NAPS)*, 2016, pp. 1–6.

[6] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: A federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.

[7] V. K. Singh, E. Tucker, and S. Rath, "Federated machine learning-based anomaly detection system for synchrophasor network using heterogeneous data sets: Preprint." [Online]. Available: https://www.osti.gov/biblio/2331418

[8] I. Ceballos, V. Sharma, E. Mugica, A. Singh, A. Roman, P. Vepakomma, and R. Raskar, "Splitnn-driven vertical partitioning," *arXiv preprint arXiv:2008.04137*, 2020.

[9] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial learning in the cyber security domain," *CoRR*, vol. abs/2007.02407, 2020. [Online]. Available: https://arxiv.org/abs/2007.02407

[10] J. Zhang, M. Li, S. Zeng, B. Xie, and D. Zhao, "A survey on security and privacy threats to federated learning," in *2021 International Conference on Networking and Network Applications (NaNA)*, 2021, pp. 319–326.

[11] L. Zhao, J. Li, Q. Li, and F. Li, "A federated learning framework for detecting false data injection attacks in solar farms," *IEEE Transactions on Power Electronics*, vol. 37, no. 3, pp. 2496–2501, 2021.

[12] Y. Yang, Z. Wang, S. Zhao, and J. Wu, "An integrated federated learning algorithm for short-term load forecasting," *Electric Power Systems Research*, vol. 214, p. 108830, 2023.

[13] H. Liu, X. Zhang, X. Shen, and H. Sun, "A federated learning framework for smart grids: Securing power traces in collaborative learning," *arXiv preprint arXiv:2103.11870*, 2021.

[14] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *ArXiv*, vol. abs/1812.00564, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:54439509

[15] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, vol. 90, pp. 148–173, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522001439

[16] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, 2018.

[17] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[18] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[19] N. E. M. Association *et al.*, *American National Standard for Electric Power Systems and Equipment-Voltage Ratings (60 Hertz)*. National Electrical Manufacturers Association, 1996.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] G. Carl, G. Kesidis, R. R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *IEEE Internet computing*, vol. 10, no. 1, pp. 82–89, 2006.

[22] V. K. Singh, E. Vaughan, and J. Rivera, "Sharp-net: Platform for self-healing and attack resilient pmu networks," in *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2020, pp. 1–5.

[23] T.-T. Tran, O.-S. Shin, and J.-H. Lee, "Detection of replay attacks in smart grid systems," in *2013 International Conference on Computing, Management and Telecommunications (ComManTel)*. IEEE, 2013, pp. 298–302.

[24] "Solar Forecast Arbiter," accessed: 2021-08-30. [Online]. Available: https://dashboard.solarforecastarbiter.org/

[25] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.