

RESEARCH ARTICLE

Model-Free Cyber-Resilient Coordinated Inverter Control in a Microgrid

MILAD BEIKBABAEI^{ID}, (Graduate Student Member, IEEE),

CAROLINE LARSEN^{ID}, (Graduate Student Member, IEEE),

AND ALI MEHRIZI-SANI^{ID}, (Senior Member, IEEE)

Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

Corresponding author: Ali Mehrizi-Sani (mehrizi@vt.edu)

This work was supported in part by the National Science Foundation (NSF) under Award ECCS-1953213, in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award under Grant 38637 (UNIFI Consortium led by NREL), in part by the Department of Defense, in part by the Commonwealth Cyber Initiative (www.cyberinitiative.org), and in part by Virginia Tech's Open Access Subvention Fund (OASF).

ABSTRACT The increasing number of inverter-based resources (IBR) in the grid introduces new challenges due to the fast transient response and low inertia of IBRs. Set point automatic adjustment with correction enabled (SPAACE)-based techniques smoothen the transient response of an IBR already installed in a grid by modifying its set point without accessing its internal parameters in a model-free approach. Coordinated SPAACE (CSPAACE) further enhances SPAACE performance by incorporating communication links to exchange tracking error values between IBRs; however, this creates openings for cyberattacks. This work adds a detection and mitigation algorithm for both denial of service (DoS) and false data injection (FDI) attacks on the communication channels. Long short-term memory (LSTM) detects anomalies in the inputs received from other inverters, and bidirectional LSTM (BiLSTM) mitigates the adverse effect of attacks on the voltage and frequency stability of a microgrid. A hybrid co-simulation platform is developed using a computer running PSCAD/EMTDC software, a network switch, and two Raspberry Pi computers, where the cyberattacks are conducted on the network switch using one of the Pis. The testbed is used to study the effectiveness of the proposed detection and mitigation method under DoS and FDI attacks and various grid transients.

INDEX TERMS Cyberattack, deep learning, detection, denial of service (DoS), false data injection (FDI), inverter-based resources (IBR), LSTM, BiLSTM, machine learning, model-free, microgrid, mitigation.

I. INTRODUCTION

As more photovoltaic (PV) plants and wind turbines are installed in the grid, the number of inverter-based resources (IBR) is increasing. IBR integration introduces new challenges in the control of the power system due to the low inertia and fast transient response of IBR-heavy grids compared to grids with only synchronous machine generators [1]. Research has developed new IBR control techniques to overcome these challenges [2]. Many PV and wind IBRs have been added to the grid in the past few years, increasing the need for methods to enhance the control of installed IBRs without accessing their internal control parameters.

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Wang^{ID}.

Set point automatic adjustment with correction enabled (SPAACE)-based techniques, a category of model-free control methods [3], [4], can make an IBR's transient response smoother by modifying the input set point of the inverter without requiring access to its internal control parameters [5], [6]. Reference [7] proposes a load frequency controller for an interconnected power system and uses SPAACE to improve the controller performance. SPAACE can incorporate communication to further improve performance. Reference [8] proposes coordinated SPAACE (CSPAACE), a two-level SPAACE algorithm for a microgrid that uses error signals communicated between inverters to improve net dynamic response. CSPAACE relies on communication links between devices [8], exposing the microgrid to cyberattacks.

Cyberattacks on the power system are a real issue, and attackers can compromise communications in a power system to launch cyberattacks intended to cause voltage and frequency instability [9]. The 2015 cyberattack on the Ukrainian power system affected 225,000 Ukrainian customers, and attackers also used firmware patching attacks, which overwrite the firmware of substation devices, to disable remote operation [10]. The U.S. Department of Energy (DOE) reported 117 disturbances due to vandalism and cyber events in 2023, six of which were caused by cyberattacks. A cyberattack in March 2023 disrupted monitoring at the control center in King County, Washington for about half an hour [11]. In 2020, attackers integrated their malicious software with SolarWinds' Orion software using a supply chain attack. The attackers accessed the local network of customers who used the updated software, showing that attackers can even infiltrate a local area network disconnected from the Internet [12].

Machine learning (ML) and deep learning (DL) algorithms are used for detecting false data injection (FDI) and denial of service (DoS) cyberattacks and mitigating their adverse effects. These algorithms use data extracted from the system without the need for detailed system models. A convolutional neural network (CNN)-based algorithm is developed to detect FDI in the sensor and communication link data of a battery storage system in [13], with the detection time ranging from 20 to 50 seconds. Reference [14] uses CNN to detect DoS cyberattacks in a cyber-physical system on a timescale of several minutes or more. However, these methods may not be suitable for situations requiring shorter detection times to mitigate transients caused by cyberattacks. A hybrid graph convolutional LSTM model and CNN techniques are used to detect stealthy attacks on network-connected devices in a smart grid in [15], but the method requires that all the devices share the same communication network, which is not always the case. Reference [16] combines LSTM and model predictive control to detect, classify, and mitigate cyberattacks in power electronics-dominated grids, but this approach requires a partial system model, limiting its application scope. Other methods monitor the system from a central location, risking a single point of failure. In [17], long short-term memory (LSTM) and a rule-based approach are combined to detect cyberattacks in a microgrid using a central manager communicating with distributed devices. A remedial action-based method mitigates the adverse effects of FDI on smart grids using LSTM, a deep recurrent neural network (DRNN), and a central operator monitoring the grid in [9]. Some methods do not address how to prevent false positive detection of transient events, such as load changes, as cyberattacks. Reference [18] develops an RNN-based approach to FDI detection and mitigation in a microgrid, and [19] develops a method based on CNN and LSTM that uses micro-PMU data to detect cyberattacks, but the impact of grid transients on these algorithms is not discussed.

Communication is now an inseparable part of the power system, sending electrical measurements for grid monitoring and commands for remote control of the power system to enable rapid grid restoration. As a result, the power system and its communication system need to be co-simulated to study the cybersecurity of the power system. Software-based co-simulation is easy to use and does not require hardware; however, it cannot easily model all the details necessary for accurate simulation [20]. On the other hand, co-simulating a real-time simulator, such as OPAL-RT or RTDS, with communication hardware gives accurate results; however, it is expensive mainly due to the cost of the real-time simulators [21]. Using a hybrid testbed, where power simulation tools are co-simulated with the hardware communication devices, increases the accuracy of the simulation and reduces the implementation cost [22]. This work develops and uses a hybrid co-simulation platform for cybersecurity study.

Using communicated error signals, CSPAAACE improves the dynamic response of a power system by adjusting the real and reactive set points of inverters on a timescale of milliseconds. However, this makes CSPAAACE susceptible to cyberattacks as attackers can manipulate these set points to cause voltage and frequency instability [23]. Therefore, this work proposes a cyber-resilient CSPAAACE technique by adding a detection and mitigation algorithm. The effectiveness of the proposed method is tested in a 100% inverter-based microgrid with 8 inverter-based resources (IBR). The contributions of this paper are as follows:

- The adverse effects of both a successful FDI and DoS attack on the CSPAAACE algorithm employed in grid-following (GFL) inverters are studied, where GFL inverters are further separated into two isolated local area networks to enhance resiliency against cyberattacks.
- Using local measurements and a model-free approach, an LSTM-based method detects DoS and various FDI attacks, such as step, ramp, and random attacks, and a bidirectional LSTM (BiLSTM)-based method mitigates and prevents voltage and frequency violation in real-time in a fully inverter-based microgrid.
- The LSTM-based method does not falsely detect grid transients, such as short circuit faults, set point changes, and load changes, as cyberattacks.
- A hybrid co-simulation platform is developed using a computer running PSCAD/EMTDC software, a network switch, and two Raspberry Pi computers, where cyberattacks are conducted on the network switch using man-in-the-middle attacks.

Section II discusses the foundations of inverter control and CSPAAACE. Section III discusses the details of the detection algorithm, Section IV discusses the mitigation algorithm, and Section V discusses the test system implementation. Section VI is the performance evaluation and Section VII concludes the paper.

TABLE 1. Abbreviations and their meanings.

Abbreviation	Meaning
IBR	Inverter-based resource
PV	Photovoltaic
SPAACE	Set point automatic adjustment with correction enabled
CSPAACE	Coordinated set point automatic adjustment with correction enabled
GFL	Grid-following
GFM	Grid-forming
PMU	Phasor measurement unit
RTU	Remote terminal unit
DOE	Department of Energy
FDI	False data injection
DoS	Denial of service
ML	Machine learning
DL	Deep learning
CNN	Convolutional neural network
RNN	Recurrent neural network
DRNN	Deep recurrent neural network
LSTM	Long short-term memory
BiLSTM	Bidirectional long short-term memory
SGD	Stochastic gradient descent
RMSprop	Root mean square propagation
BCE	Binary cross entropy
MAE	Mean absolute error
MSE	Mean squared error
MSLE	Mean squared logarithmic error
FLOP	Floating point operation

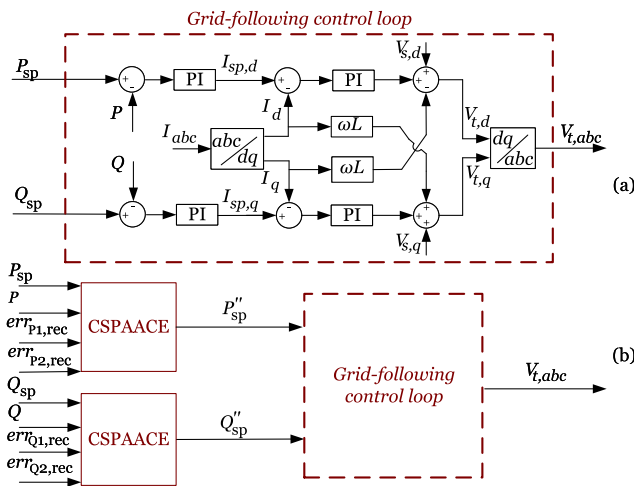


FIGURE 1. GFL inverter control diagram: (a) without CSPAACE and (b) with CSPAACE.

II. INVERTER CONTROL BASICS

This section discusses details of GFL control without CSPAACE, GFL control with CSPAACE, and grid-forming (GFM) control for an inverter.

A. GRID-FOLLOWING (GFL) CONTROL MODE

This section discusses conventional and CSPAACE-based GFL control.

1) CONVENTIONAL GFL CONTROL

In the GFL mode of operation, the inverter’s control inputs are its real and reactive power set points, and the inverter tries to

match its output real and reactive power to the set point values using PI controllers. Fig. 1(a) shows the control diagram of a GFL inverter connected to the grid through an RL filter [24]. The filter inductance is L , ω is the angular frequency, I is the inverter output current, V is the terminal voltage, P and Q are the inverter output real and reactive power, and P_{sp} and Q_{sp} are the reference real and reactive power. The voltage and current measurements are converted from the abc -frame to the dq -frame to calculate the output voltage. Finally, the calculated voltage is converted back from the dq -frame to the abc -frame. A phase-locked loop (PLL) is used to estimate the voltage angle of the grid and is used in abc -frame to dq -frame conversion.

Algorithm 1 GFL Control With CSPAACE Pseudocode

- 1: $err_P(t_k) = P_{sp} - P(t_k)$
- 2: $err_Q(t_k) = Q_{sp} - Q(t_k)$
- 3: $err_P(t_k - T_{pred}) = P_{sp} - P(t_k - T_{pred})$
- 4: $err_Q(t_k - T_{pred}) = Q_{sp} - Q(t_k - T_{pred})$
- 5: $err_{P,pred}(t_k + T_{pred}) = 2 err_P(t_k) - err_P(t_k - T_{pred})$
- 6: $err_{Q,pred}(t_k + T_{pred}) = 2 err_Q(t_k) - err_Q(t_k - T_{pred})$
- 7: $P'_{sp}(t_k + T_{pred}) = P_{sp} + m err_{P,pred}(t_k + T_{pred})$
- 8: $Q'_{sp}(t_k + T_{pred}) = Q_{sp} + m err_{Q,pred}(t_k + T_{pred})$
- 9: $P''_{sp}(t_k + T_{pred}) = P'_{sp} + m (err_{P1,rec}(t_k + T_{pred}) + err_{P2,rec}(t_k + T_{pred}))$
- 10: $Q''_{sp}(t_k + T_{pred}) = Q'_{sp} + m (err_{Q1,rec}(t_k + T_{pred}) + err_{Q2,rec}(t_k + T_{pred}))$

FIGURE 2. CSPAACE pseudocode for the P and Q channel of a GFL inverter controller.

2) COORDINATED SPAACE (CSPAACE) BASICS

CSPAACE is a method to make the dynamic response of a controlled system smoother [8] by incorporating communication into SPAACE. SPAACE calculates the modified set points to make the response smoother using the tracking error $err(t_k)$:

$$err(t_k) = x_{sp} - x(t_k), \quad (1)$$

where x_{sp} is the set point, $x(t_k)$ is the system response variable, and t_k is the current time step. SPAACE uses the predicted error for the following time step, $err_{pred}(t_k + T_{pred})$, which is linearly extrapolated using the error value at the current and previous time steps:

$$err_{pred}(t_k + T_{pred}) = 2 err(t_k) - err(t_k - T_{pred}), \quad (2)$$

where T_{pred} is the error prediction horizon. Moreover, SPAACE modifies the set point to $x'_{sp}(t_k + T_{pred})$ based on the predicted error:

$$x'_{sp}(t_k + T_{pred}) = x_{sp} + m err_{pred}(t_k + T_{pred}), \quad (3)$$

where the scaling factor m is a constant. CSPAACE uses error signals received from other units to enhance SPAACE performance. In this work, CSPAACE uses only the error signals from the two neighboring units of an IBR to calculate

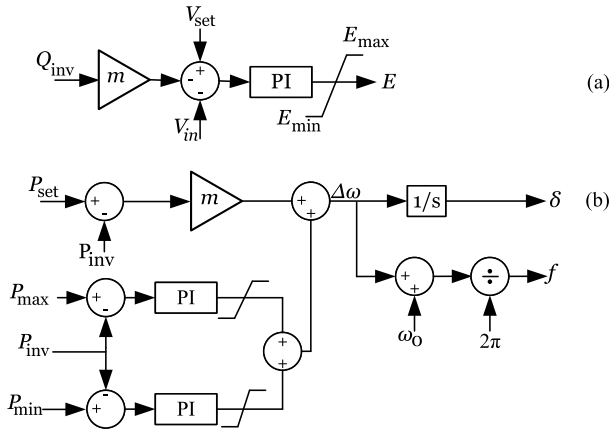


FIGURE 3. Droop control for GFM mode: (a) $Q-V$, (b) $P-f$ [24].

its modified set point $x''_{sp}(t_k + T_{pred})$:

$$x''_{sp}(t_k + T_{pred}) = x'_{sp}(t_k + T_{pred}) + m (err_{1,rec}(t_k + T_{pred}) + err_{2,rec}(t_k + T_{pred})), \quad (4)$$

where $err_{1,rec}(t_k + T_{pred})$ and $err_{2,rec}(t_k + T_{pred})$ are the predicted errors received from the first and the second neighboring units.

3) CSPAAACE-BASED GFL CONTROL

Fig. 1(b) shows the block diagram of GFL control with CSPAAACE, receiving input errors err from two neighboring units for both the real and reactive power set points. Moreover, Fig. 2 shows the pseudocode implementation of CSPAAACE for the real power set point P_{sp} and reactive power set point Q_{sp} . The modified real power set point $P''_{sp}(t_k + T_{pred})$ and reactive power set point $Q''_{sp}(t_k + T_{pred})$ calculated by CSPAAACE are sent to the GFL control loop.

B. GRID-FORMING (GFM) CONTROL MODE

In the GFM control mode, the inverter control inputs are the reference values of the voltage and frequency, and the real and reactive output power of the inverter are adjusted to maintain the grid voltage and frequency using droop control. Droop control modifies the real and reactive power output when the voltage magnitude or frequency deviates from its nominal values, as shown in Fig. 3. In $Q-V$ droop control, the reactive power of the inverter is adjusted based on the voltage deviation, as shown in Fig. 3(a). In $P-f$ droop control, the real power of the inverter is adjusted based on the frequency deviation, as shown in Fig. 3(b). It is essential to operate a certain number of inverters in the GFM mode to maintain the voltage and frequency of an isolated fully inverter-based microgrid since there is no voltage and frequency support from the bulk grid.

III. DETECTION METHOD

This section discusses the basics of LSTM and BiLSTM and the details of the proposed detection method, training dataset creation, and hyperparameter tuning.

A. LONG SHORT-TERM MEMORY (LSTM)

LSTM is a type of recurrent neural network (RNN) designed to learn patterns in sequential data [25]. LSTMs are better at learning long-term patterns than traditional RNNs, and they avoid the vanishing gradient problem [19]. An LSTM network predicts time series values of an output variable using the given input time series. The network uses a hidden layer of cells to process the input vectors. Fig. 4(a) shows an LSTM cell structure that modifies the cell state in a controlled manner. C_{t-1} is the previous cell state; h_{t-1} is the previous output, or hidden state; and x_t is the current input, passing through gate units. Each gate unit has weight values, W_f , W_i , W_c , and W_o , and bias values, b_f , b_i , b_c , and b_o . The outputs of the gates produce a new cell state C_t and hidden state h_t in each time step. Through this recurrent process, the LSTM remembers long-term information.

Each gate uses an activation function to transform its input. Equation (5) shows a sigmoid activation function converting the gate inputs to weights between 0 and 1. A gate uses these weights to determine how much of a value is allowed to pass through the gate.

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (5)$$

The forget gate in (6) selects how much of the previous state C_{t-1} is kept based on C_{t-1} , h_{t-1} , and x_t values.

$$f_t = \sigma(W_f [C_{t-1}, h_{t-1}, x_t] + b_f). \quad (6)$$

Additionally, activation functions calculate the cell and hidden states. In (7), a hyperbolic tangent activation function transforms h_{t-1} and x_t to calculate \tilde{C}_t , the set of potential new cell state values.

$$\tilde{C}_t = \tanh(W_c [h_{t-1}, x_t] + b_c), \quad (7)$$

The input gate i_t in (8) weights the new values based on C_{t-1} , h_{t-1} , and x_t .

$$i_t = \sigma(W_i [C_{t-1}, h_{t-1}, x_t] + b_i), \quad (8)$$

The updated cell state in (9) is a combination of C_{t-1} , multiplied by f_t , and new candidate values \tilde{C}_t , multiplied by i_t .

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t. \quad (9)$$

In (10), the output gate calculates weights for C_t based on C_t , h_{t-1} , and x_t .

$$o_t = \sigma(W_o [C_t, h_{t-1}, x_t] + b_o). \quad (10)$$

Lastly, the cell output h_t is determined in (11) using o_t and C_t .

$$h_t = o_t \tanh(C_t). \quad (11)$$

The outputs of all time steps in one layer are input to the next layer. This is repeated until the last layer, which produces the final output vector prediction.

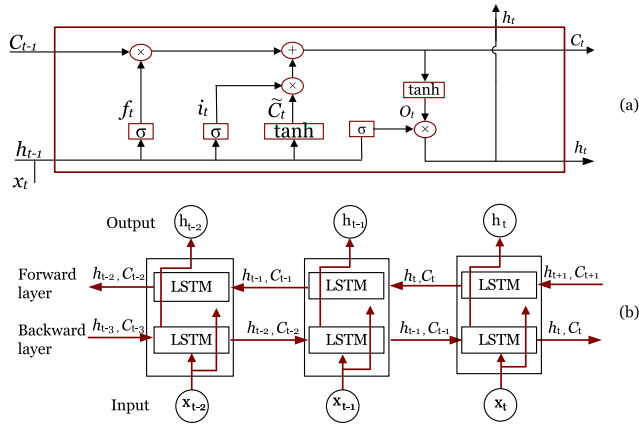


FIGURE 4. (a) The inside of an LSTM unit, and (b) the data exchange in the time domain in a BiLSTM unit.

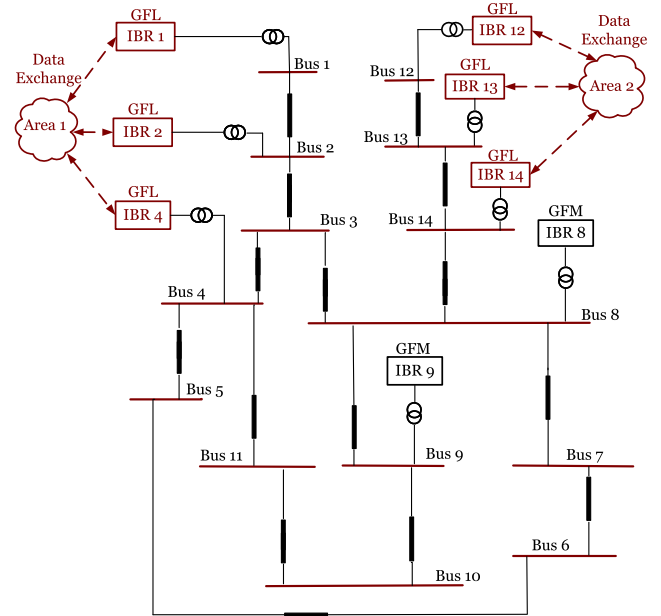
B. BIDIRECTIONAL LSTM (BiLSTM)

BiLSTM is a variant of LSTM that adds a second hidden layer in the opposite direction of the first. This allows the BiLSTM to be trained in both the forward and backward directions at once. The first layer processes the input sequence normally. The second layer, which does not interact with the first layer, processes the same sequence in reverse as shown in Fig. 4(b). The outputs of the two layers are then merged to produce a final output [26]. BiLSTM outperforms LSTM for many tasks involving complex sequence prediction as its architecture allows it to account for future characteristics of time series data [26].

C. PROPOSED DETECTION METHOD

In the proposed method, each IBR has its own cyberattack detection algorithm since the IBRs have varied characteristics. Each detection algorithm uses an LSTM model trained to detect when the error inputs of the IBR are falsified in an FDI attack, and it also detects a DoS attack when timeout occurs due to an idle connection. The detection models are trained using Python 3.8 and Keras 2.12.0 on a Core i7 computer with 16 GB RAM. For a given IBR, the IBR’s real power output, reactive power output, grid RMS voltage, output current, and the two neighboring units’ real and reactive power error values are the eight inputs of its detection algorithm. These features are sampled at 1 ms per time step and passed to the algorithm in windows of 3 time steps. At each time step, the LSTM algorithm uses the given inputs to predict either no attack 0 or an attack 1 on the error signals. The LSTM algorithm cannot be trained on all possible grid transients and operation points. Moreover, its accuracy cannot reach beyond a certain level without the model overfitting and performing worse on unseen data. For these reasons, even a well-trained model predicts a few false positives during grid transients. To handle this issue, the detection algorithm is not activated unless the LSTM output is 1 for 6 consecutive time steps, indicating that the IBR is under attack. The detection model for each IBR consists of two LSTM hidden layers followed

by a dense output layer. The hidden layers have 8 neurons each, use the tanh activation function to compute their cell and hidden states, and use the sigmoid activation function to compute the outputs of their input, forget, and output gates. A sigmoid function on the output layer produces the final prediction of 0 or 1. The training process uses the Adam optimizer, a learning rate of 1×10^{-3} , a batch size of 16, and the binary cross entropy (BCE) loss function.



IBRs. Moreover, FDI attacks can be launched on the error output of an IBR, causing all other IBRs in its network to receive its falsified signal, or on the error input of an IBR, causing that IBR alone to receive a falsified signal.

In a step attack, err_{in} is set to v_{step} after the attack is launched at t_{attack} , as shown in the following:

$$err_{in} = \begin{cases} err_{non-attack} & t < t_{attack} \\ v_{step} & t \geq t_{attack}. \end{cases} \quad (12)$$

In a ramp attack, err_{in} increases linearly to v_{ramp} over t_d , as shown in the following:

$$err_{in} = \begin{cases} err_{non-attack} & t < t_{attack} \\ v_{ramp} \frac{t - t_{attack}}{t_d} & t_{attack} \leq t \leq t_{attack} + t_d \\ v_{ramp} & t > t_{attack} + t_d. \end{cases} \quad (13)$$

For FDI attack scenarios in the dataset, v_{step} and v_{ramp} range from 0.05 to 0.3 and from -0.05 to -0.3 in steps of 0.05. The ramp duration t_d is set to 200, 500, 750, or 1000 ms.

These scenarios can train the model to detect other FDI attacks not included in the dataset, such as a random attack, in which err_{in} is set to a stochastic signal. The attack signal follows a Bernoulli distribution, as in [28], and on each time step after t_{attack} , err_{in} becomes either zero or a predefined value v_{random} :

$$err_{in} = \begin{cases} v_{random} & \text{with probability } p \\ 0 & \text{with probability } 1 - p. \end{cases} \quad (14)$$

Attackers can also launch DoS attacks to prevent communication among IBRs, causing them to continue using the last set of error values received. If these values are severe, the results of the attack are similar to those of a step or ramp FDI attack. Thus, the FDI attack cases can train the model to detect certain types of DoS attacks before the connection timeout even though the dataset contains no DoS scenarios. In this work, each IBR also has a timeout mechanism to indicate a DoS attack if a communication link has been idle for 200 ms, the default minimum retransmission timeout for TCP in most operating systems [29]. Unlike the retransmission timer, which times out if a transmission is not acknowledged [30], the DoS attack timer times out if no data is received; however, the two use the same timeout value because they are similarly affected by the network's communication rate and packet loss.

80% of the scenarios in the dataset are grid transients, including load changes, set point changes, and faults, and 20% of the scenarios are FDI attacks. The dataset is divided into training and validation data in a 70/30 split, with 169,000 samples of training data and 73,000 samples of validation data. Across different IBRs, the attack cases within the training set are varied to fit the IBR for which the model is being trained. For example, a step attack on IBR 1 in IBR 1's version of the dataset is an identical step attack on IBR 12 in IBR 12's version of the dataset.

E. HYPERPARAMETER TUNING

To determine the optimal hyperparameters for the detection algorithm, the hyperparameter values of the model are varied and the model performance compared across multiple tests. Table 2 shows the results of these tests for IBR 1, with the model's accuracy on the training data, accuracy on the validation data, and total training time given for each. To ensure accurate comparison among hyperparameters, only one hyperparameter is varied at a time, and the others are fixed at set values: 2 LSTM hidden layers, the tanh and sigmoid activation functions, a 3-step window size, 8 neurons per hidden layer, a training batch size of 32, and the Adam optimizer with a 1×10^{-3} learning rate. All the models are trained for 400 epochs and evaluated at the end of the training period. The same hyperparameters are used for the rest of the IBRs since they use the same controller parameters.

The hyperparameter tuning results are selected based on validation accuracy and training time. From the cases comparing the performance of the detection model as an LSTM and BiLSTM for varying numbers of hidden layers, a two-layer LSTM is selected as its model has the highest accuracy on the validation data. In the window size tests, an input window size of 10 provides the highest validation accuracy, but a window size of 3 is selected instead to reduce memory requirements and training time. From the tests for the number of neurons per layer, a size of 8 neurons per layer is selected as its model achieves the highest validation accuracy. In the batch size tests, a smaller batch size tends to increase validation accuracy but also increases training time, so an intermediate batch size of 16 is selected. The Adam optimizer and a learning rate of 1×10^{-3} are selected as they provide a high validation accuracy. Table 3 shows the activation function tests, which examine the performance of the model as the pair of activation functions used in both of its hidden layers is varied. The combination of tanh for the hidden and cell states and sigmoid for the input, output, and forget gates is selected as its model has the highest validation accuracy across the tests. Table 4 compares the model performance across different combinations of input features. Because the full combination of P , Q , V , I , and four err signals provides the highest validation accuracy, all eight features are used as input in the final model.

After the final model configuration and hyperparameter values are determined, detection models are trained for all six inverters. A copy of the detection model being trained is saved at each epoch and evaluated on the training and validation data. Additionally, the validation data are separated into grid transient cases and attack cases, and the model is evaluated on both sets of cases at each epoch. Figure 6 shows the performance of the IBR 1 detection model across all four metrics. The validation accuracy for attack cases does not reach above 98% before epoch number 180; thus, the number of epochs for the final model is selected to be greater than 180 to ensure that detection works properly. Moreover, a model at a training epoch with high validation accuracy for both grid transients and attacks is selected to minimize

TABLE 2. Detection model hyperparameter tuning.

Parameter		Detection		
Name	Value	Training Accuracy	Validation Accuracy	Training Time (min)
# of LSTM Hidden Layers	1	99.90%	92.23%	54.57
	2	99.74%	98.86%	96.27
	3	99.71%	98.78%	136.21
	4	99.58%	96.75%	167.11
# of BiLSTM Hidden Layers	1	99.89%	92.67%	54.78
	2	99.76%	97.68%	82.68
Window Size	1	99.23%	98.04%	78.01
	2	99.59%	98.49%	88.01
	3	99.73%	98.95%	96.16
	4	99.77%	97.90%	103.73
	5	99.78%	99.20%	111.57
	10	99.84%	99.38%	150.17
# of Neurons Per Layer	1	95.09%	96.47%	82.58
	2	96.53%	94.89%	85.36
	4	99.38%	97.49%	86.27
	8	99.69%	99.21%	73.77
	16	99.83%	97.09%	87.07
	Batch Size	2	99.59%	99.35%
4		99.78%	97.13%	512.12
8		99.79%	98.97%	244.05
16		99.76%	99.13%	133.12
32		99.73%	98.57%	79.29
64		99.59%	97.47%	59.89
Optimizer	Adam	99.69%	99.21%	73.77
	SGD	99.39%	98.72%	73.57
	RMSprop	98.62%	96.61%	72.59
Learning Rate	1×10^{-4}	99.23%	98.57%	79.98
	1×10^{-3}	99.52%	98.90%	79.04
	1×10^{-2}	99.71%	98.04%	79.52

TABLE 3. Detection model activation function tuning.

Activation Function		Detection		
Cell and Hidden States	Input, Forget, and Output Gates	Training Accuracy	Validation Accuracy	Training Time (min)
tanh	Sigmoid	99.59%	99.31%	80.38
Relu	Sigmoid	99.63%	98.80%	59.43
tanh	Softplus	99.76%	98.34%	63.09
tanh	Relu	99.57%	98.03%	75.98
Softplus	Sigmoid	99.64%	99.15%	77.52
Sigmoid	Softplus	99.54%	99.08%	99.68
Sigmoid	Sigmoid	99.41%	97.67%	57.30

the number of false predictions. To avoid overfitting, the number of epochs for the final IBR 1 model is set to an intermediate value of 296 since the accuracy changes little afterward. Table 5 shows the training epoch selected for the final model of each IBR and the performance of that model on its dataset.

IV. MITIGATION METHOD

This section discusses the proposed mitigation method for both DoS and FDI attacks, the creation of the training dataset, and model hyperparameter tuning.

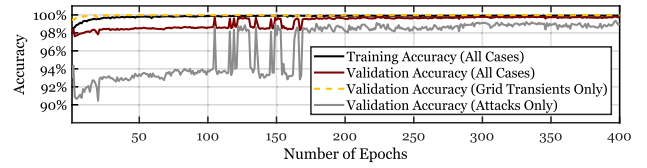


FIGURE 6. Detection model performance at each training epoch for IBR 1.

TABLE 4. Detection model input feature selection.

Input Feature					Detection		
P	Q	V	I	err_{in} (x4)	Training Accuracy	Validation Accuracy	Training Time (min)
✓	✓	✓	✓	✓	99.65%	99.39%	80.38
✓	✓	✓	✓	✗	90.29%	78.71%	75.11
✓	✓	✓	✗	✓	99.74%	97.56%	88.48
✓	✓	✗	✓	✓	99.67%	93.29%	88.52
✓	✗	✓	✓	✓	99.70%	96.73%	87.89
✗	✓	✓	✓	✓	99.72%	99.13%	87.93
✓	✓	✗	✗	✓	99.55%	96.53%	74.88
✗	✗	✓	✓	✓	99.74%	96.43%	75.04
✓	✓	✗	✗	✓	99.55%	96.53%	74.88
✗	✗	✓	✓	✓	99.74%	96.43%	75.04

TABLE 5. Final LSTM detection model selection.

IBR	Selected Epoch	Training Accuracy (All Cases)	Validation Accuracy		
			All Cases	Grid Transients Only	FDI Attacks Only
1	296	99.91%	99.80%	99.98%	99.16%
2	178	99.90%	99.56%	99.94%	98.22%
4	286	99.92%	98.12%	99.79%	92.17%
12	249	99.92%	99.81%	99.98%	99.22%
13	185	99.99%	99.12%	99.92%	96.25%
14	270	99.92%	99.31%	99.98%	96.97%

A. PROPOSED MITIGATION METHOD

Similar to the detection, each IBR has its own cyberattack mitigation algorithm. The mitigation algorithm is a single BiLSTM hidden layer and an output layer, and it uses the same input features as the detection algorithm. Data is sampled at 1 ms per time step, and the BiLSTM algorithm processes input data in windows of 3 time steps. At each time step, the BiLSTM algorithm predicts four error channel values, for the real and reactive power of the two neighboring IBRs, and corrects the falsified error channel values by replacing them with the predicted values after attack detection is activated. For a DoS attack, where no error input is available from the other IBRs, the predicted error values are used instead. The BiLSTM hidden layer consists of 16 neurons and uses the sigmoid activation function on its cell and hidden states and the softplus activation function on its input, output, and forget gates. The outputs of the forward and backward LSTMs of the hidden layer are merged through summation. The mitigation model training process uses the Adam optimizer with a learning rate of 1×10^{-4} , a batch size of 2, and the mean squared error (MSE) loss function.

B. TRAINING DATASET FOR MITIGATION

The mitigation training dataset consists of simulated grid transient scenarios from the CIGRE 14-bus grid. These scenarios include load change, set point change, and fault cases with the same range of values as those used in the detection training dataset, discussed in Section III-D. Since FDI cases cause voltage and frequency instability, they are excluded from the mitigation training dataset so that the mitigation algorithm only predicts error inputs that keep the voltage and frequency stable. The mitigation model for each IBR is trained by comparing the model's error input predictions to the actual received error values in the various cases. To validate the models on unseen data, the dataset is split into 82,000 samples of training data and 39,000 samples of validation data in a 70/30 ratio.

C. HYPERPARAMETER TUNING

The performance of the mitigation model for IBR 1 is compared across multiple tests with varying hyperparameter values, as shown in Table 6. Across all tests, only one hyperparameter is tuned at a time and the others are kept fixed. The fixed values are a single BiLSTM hidden layer, 8 neurons in the hidden layer, the tanh activation functions on the cell and hidden states of the BiLSTM, the softplus activation function on the the forget, input, and output gates, the sum merge mode for combining the outputs of the forward and reverse layers of the BiLSTM, a window size of 3, a training batch size of 2, the Adam optimizer with a 1×10^{-4} learning rate, and the MSE loss function. The models are all trained for 150 epochs and evaluated at the final epoch. As in the detection training, the same mitigation hyperparameters are used for all IBRs.

Model performance across hyperparameter values is judged by validation MSE and training time. A single BiLSTM is selected for the hidden layer because it achieves the lowest validation MSE. For the same reason, 16 neurons per hidden layer and a window size of 3 are selected for the final model. A training batch size of 2 is also selected for providing the lowest validation MSE; a training batch size of 1 provides lower training MSE but causes severe overfitting and worse validation performance. The MSE loss function and the Adam optimizer with a learning rate of 1×10^{-4} are selected for the training process as they allow the model to achieve lower validation MSE. Various modes of combining the forward and reverse BiLSTM outputs, including concatenation, summation, averaging, and multiplication, are tested and the sum merge mode is selected for having the lowest validation MSE. Table 7 shows the combinations of activation functions tested for the mitigation model. The combination of a sigmoid function on the cell and hidden states and the softplus function on the input, output, and forget gates is selected because it provides the lowest MSE for both the training and validation data.

Fig. 7 shows the performance of the final IBR 1 mitigation model at each training epoch. Although the model has its lowest validation MSE below epoch 30, it has not fully

learned the training data at this point, so epoch 46 is selected for the final model instead. This epoch number gives a low MSE for both training and validation, and the model is not overtrained at this point. Similar reasoning is used to select the final models for the other five IBRs, shown in Table 8.

TABLE 6. Mitigation model hyperparameter tuning.

Parameter		Mitigation		
Name	Value	Training MSE	Validation MSE	Training Time (min)
# of LSTM Hidden Layers	1	4.32×10^{-4}	9.22×10^{-4}	92.47
	2	4.40×10^{-4}	9.44×10^{-4}	158.43
	3	4.97×10^{-4}	1.28×10^{-3}	195.53
	4	4.91×10^{-4}	1.27×10^{-3}	251.46
# of BiLSTM Hidden Layers	1	4.96×10^{-4}	9.07×10^{-4}	154.04
	2	4.16×10^{-4}	1.18×10^{-3}	210.03
Window Size	1	4.42×10^{-4}	9.79×10^{-4}	92.18
	2	4.03×10^{-4}	1.05×10^{-3}	95.68
	3	4.96×10^{-4}	9.07×10^{-4}	154.04
	4	4.23×10^{-4}	1.11×10^{-3}	157.64
	5	4.32×10^{-4}	1.50×10^{-3}	155.99
	10	4.49×10^{-4}	1.05×10^{-3}	202.17
# of Neurons Per Layer	1	1.13×10^{-3}	1.70×10^{-3}	154.99
	2	8.97×10^{-4}	1.73×10^{-3}	152.85
	4	5.55×10^{-4}	1.23×10^{-3}	154.91
	8	5.05×10^{-4}	1.13×10^{-3}	154.64
	16	4.65×10^{-4}	1.04×10^{-3}	157.39
Batch Size	1	9.19×10^{-5}	2.43×10^{-3}	142.84
	2	4.32×10^{-4}	9.22×10^{-4}	92.47
	4	8.76×10^{-4}	1.38×10^{-3}	47.61
	8	1.16×10^{-3}	1.73×10^{-3}	28.32
	16	1.33×10^{-3}	1.87×10^{-3}	27.32
Optimizer	Adam	4.12×10^{-4}	8.15×10^{-4}	133.55
	SGD	4.15×10^{-4}	1.02×10^{-3}	129.41
	RMSprop	8.51×10^{-4}	1.93×10^{-3}	133.77
Learning Rate	1×10^{-5}	5.71×10^{-4}	1.18×10^{-3}	154.18
	1×10^{-4}	4.83×10^{-4}	1.06×10^{-3}	155.03
	1×10^{-3}	5.83×10^{-4}	1.09×10^{-3}	153.44
Loss Function	MSE	5.10×10^{-4}	1.32×10^{-3}	157.4
	MAE	1.39×10^{-3}	1.68×10^{-3}	157.1
	MSLE	1.55×10^{-3}	1.98×10^{-3}	157.1
BiLSTM Merge Mode	Concatenate	4.27×10^{-4}	9.70×10^{-4}	160.96
	Sum	4.22×10^{-4}	7.50×10^{-4}	172.38
	Average	4.02×10^{-4}	1.27×10^{-3}	161.54
	Multiply	5.77×10^{-4}	1.66×10^{-3}	160.15

TABLE 7. Mitigation model activation function tuning.

Activation Function		Mitigation		
Cell and Hidden States	Input, Forget, and Output Gates	Training MSE	Validation MSE	Training Time (min)
tanh	Sigmoid	4.51×10^{-4}	1.38×10^{-3}	139.60
Relu	Sigmoid	4.72×10^{-4}	1.18×10^{-3}	151.45
tanh	Softplus	5.01×10^{-4}	1.19×10^{-3}	155.84
tanh	Relu	4.39×10^{-4}	1.15×10^{-3}	128.32
Softplus	Sigmoid	4.17×10^{-4}	8.55×10^{-4}	101.08
Sigmoid	Softplus	3.99×10^{-4}	6.90×10^{-4}	99.76
Sigmoid	Sigmoid	5.53×10^{-4}	1.33×10^{-3}	88.12

V. TEST SYSTEM

This section discusses the implementation of the employed test system and details of the data exchange between the power system and the communication layer.

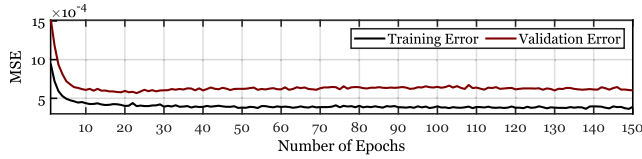


FIGURE 7. Mitigation model performance at each training epoch for IBR 1.

TABLE 8. Final mitigation model selection.

IBR	Selected Epoch	Training MSE	Validation MSE
1	46	3.90×10^{-4}	6.02×10^{-4}
2	93	1.87×10^{-4}	2.50×10^{-4}
4	59	4.53×10^{-4}	6.78×10^{-4}
12	74	2.88×10^{-5}	2.37×10^{-5}
13	76	6.14×10^{-5}	4.09×10^{-5}
14	96	4.67×10^{-5}	3.19×10^{-5}

A. CIGRE 14-BUS NORTH AMERICAN SYSTEM

The CIGRE 14-bus North American distribution system is selected for the power system, and it is modified by adding eight IBRs, where six of them are in GFL mode with CSPAAACE and the others are in GFM mode, as shown in Fig. 5. The IBR number represents which bus the IBR is connected to; for instance, IBR 4 is connected to bus 4. IBR 1, 2, and 4 exchange their real and reactive power error values every 1 ms using an isolated local area network, area 1. IBR 12, 13, and 14 exchange their real and reactive power error values at the same rate using the isolated local area 2. PSCAD/EMTDC is used for the power system modeling. The base values of the power and voltage are, respectively, 10 MW and 12.47 kV. Each IBR has an output voltage of 480 V and is connected to a medium voltage bus via a three-phase 480 V to 12.47 kV transformer. Table. 9 shows the controller parameters: proportional and the integrator gains of the PI controller for the d - and q -axes voltage and current, the maximum and minimum d - and q -axes current, the minimum and maximum real and reactive power, the droop gain for the $Q - V$ and $P - f$ droop control, and the GFM voltage and frequency reference. The real and reactive power set points are selected to minimize the grid loss. The real power set points of IBRs 1, 2, 4, 12, 13, and 14 are 3, 2, 2, 3, 1, and 1.5 MW, respectively. The reactive power set points of IBRs 1, 2, 4, 12, 13, and 14 are 1, 1, 1, 1, 0.5, and 1 MVar, respectively.

B. COMPARISON OF SPACE WITH CSPAAACE

SPACE makes the transient response of an IBR smoother. CSPAAACE, which augments SPACE with communication links, further improves the transient response by reducing rise time [8]. This is illustrated in the following examples.

TABLE 9. IBR controller parameters.

Grid-Following		Grid-Forming	
Parameter	Value	Parameter	Value
PI gain K_p for I_d	1.5	PI gain K_p for V_d	0.5
PI gain K_i for I_d	0.003	PI gain K_i for V_d	0.003
PI gain K_p for I_q	1.5	PI gain K_p for V_q	0.5
PI gain K_i for I_q	0.003	PI gain K_i for V_q	0.003
$I_{dq,max}$	1.2 pu	Droop gain $Q - V$	0.05
$I_{dq,min}$	-1.2 pu	Droop gain $P - f$	0.05
P_{max}	1 pu	f_{ref}	60 Hz
P_{min}	0 pu	V_{ref}	1 pu
Q_{max}	0.6 pu	-	-
Q_{min}	-0.6 pu	-	-

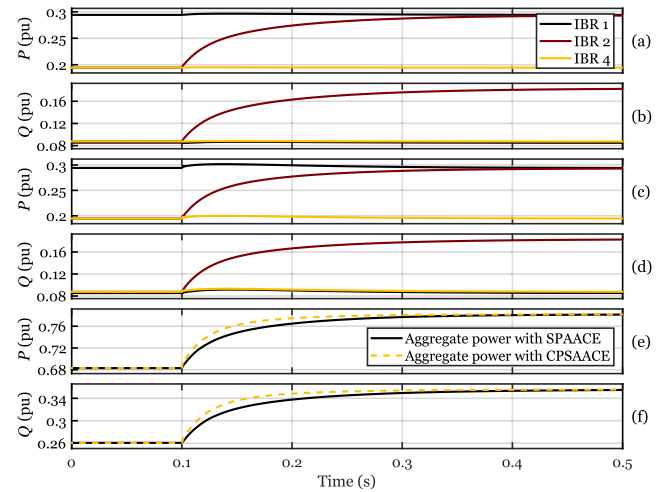


FIGURE 8. Simulation results for IBRs 1, 2, and 4 output powers during a set point change in IBR 2: (a) real power with SPACE, (b) reactive power with SPACE, (c) real power with CSPAAACE, (d) reactive power with CSPAAACE, (e) aggregate real power, and (f) aggregate reactive power.

1) REAL AND REACTIVE POWER SET POINT CHANGE

Fig. 8 shows the simulation results for IBRs 1, 2, and 4 for a 1 MW and 1 MVar set point increase in IBR 2 at $t = 100$ ms. Figs. 8(a) and (b) show the real and reactive power outputs of IBRs 1, 2, and 4 with SPACE. Figs. 8(c) and (d) show the real and reactive power outputs of IBRs 1, 2, and 4 with CSPAAACE. From Fig. 8(e), the aggregate real and reactive power outputs of the three IBRs with SPACE achieve 90% of their total increase in 141 ms and 152 ms rise times, respectively. The aggregated real and reactive power with CSPAAACE rise in 80 ms and 81 ms, respectively. Thus, CSPAAACE decreases the real power rise time by 43% and the reactive power rise time by 47% in this scenario.

2) THREE-PHASE BOLTED FAULT

Fig. 9 shows the output voltage of IBR 1 with SPACE and CSPAAACE during a bolted three phase-to-ground fault at bus 5 that starts at $t = 100$ ms and clears at $t = 200$ ms. With SPACE, the voltage drops from 0.978 pu to 0.424 pu during the fault; with CSPAAACE, it drops to 0.474 pu, a 12%

improvement. After the fault clears, the voltage recovers to 0.90 pu in 38 ms with SPAACE and in 18 ms with CSPAAACE, a 1.9 times faster rise time. CSPAAACE outperforms SPAACE by reducing the voltage drop during the short circuit fault and the voltage recovery time after the fault clearance.

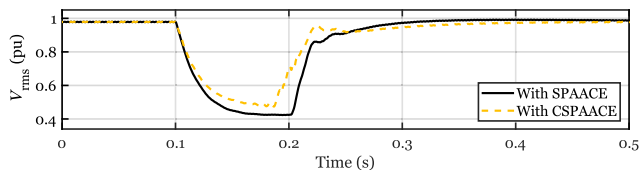


FIGURE 9. Simulation results for IBR 1 output voltage during a bolted three-phase fault at bus 5.

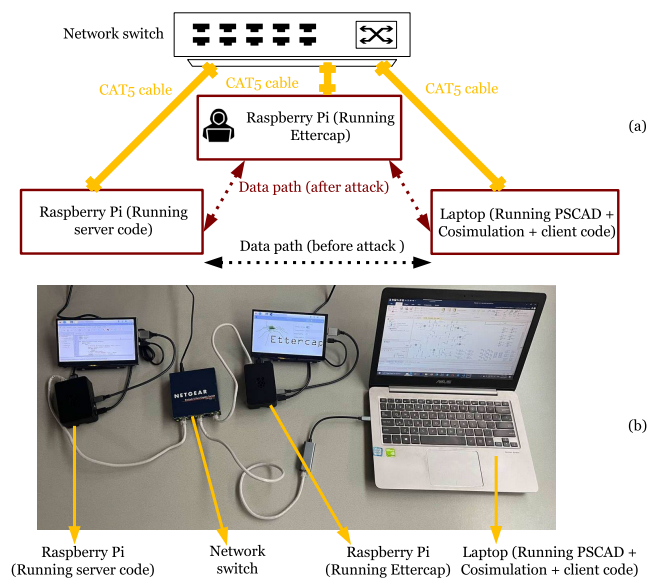


FIGURE 10. Hybrid co-simulation cyber testbed setup: (a) the schematic and data exchange and (b) the picture of the setup.

C. HYBRID CO-SIMULATION IMPLEMENTATION

Fig. 10 shows the hybrid co-simulation platform implemented in this work, including a computer running PSCAD/EMTDC software, a network switch, and two Raspberry Pi computers. PSCAD/EMTDC simulates the power system layer, data is exchanged through the network switch, and cyberattacks are launched on the network switch using the malicious Raspberry Pi.

Fig. 11 shows how data is exchanged in the cyber layer and how the detection and mitigation methods are implemented within the communication system. Remote terminal units (RTU) are used to exchange data, send electrical measurements, and receive commands and data from other units. Fig. 11 shows that each IBR has its own RTU to communicate with other units. Electrical measurements, including bus measurements P , V , I , and Q and the CSPAAACE real and reactive power error values, are sent from PSCAD/EMTDC to the co-simulation Python code using the inbuilt co-simulation block in PSCAD/EMTDC. The co-simulation Python code is integrated within the TCP/IP client code to enable data

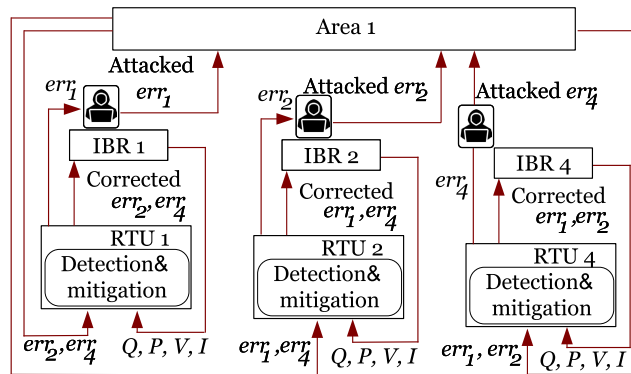


FIGURE 11. Data exchange between the IBRs for area 1.

TABLE 10. Computational complexity of detection and mitigation models.

Model	Number of Parameters	FLOPs	Inference Time
Detection	1.1×10^3	1.0×10^3	210 μ s
Mitigation	1.1×10^3	2.2×10^3	280 μ s

exchange through the network switch, with the Raspberry Pi running TCP/IP server code. The co-simulation code sends the calculated real and reactive power error values to the other Raspberry Pi using TCP/IP communication through the network switch. The Raspberry Pi running the server code represents both area 1 and area 2, enabling data exchange between IBRs in the same area. The malicious Raspberry Pi runs Ettercap software to conduct man-in-the-middle attacks, which are used for packet sniffing, packet modification, and DoS attacks.

VI. PERFORMANCE EVALUATION

This work studies cyberattacks on the GFL units with CSPAAACE. IBRs are targets of DoS attacks and step and ramp FDI attacks to test the effectiveness of the proposed detection and mitigation algorithm. In some scenarios, multiple IBRs are under the same type of attack or different types of attacks, leading to more severe voltage and frequency disturbances. The detection algorithm is further tested under various grid transients, such as short circuit faults, set point changes, and load changes, to ensure it does not mistakenly detect grid transients as cyberattacks.

Table 10 summarizes the computational complexity of the detection and mitigation models in terms of three metrics: the number of parameters in the model, the number of floating point operations (FLOP) per prediction, and the average inference time required for the model to make a prediction on a Core i7 computer. When measuring the inference time, the trained models are first optimized using TensorFlow Lite 2.12.0, which quantizes some of the model weights to 8-bit integers, reducing memory usage and execution time [31]. With this optimization, both the detection and mitigation models make their predictions in less than 300 μ s. This is within the 1 ms sampling period, enabling real-time operation.

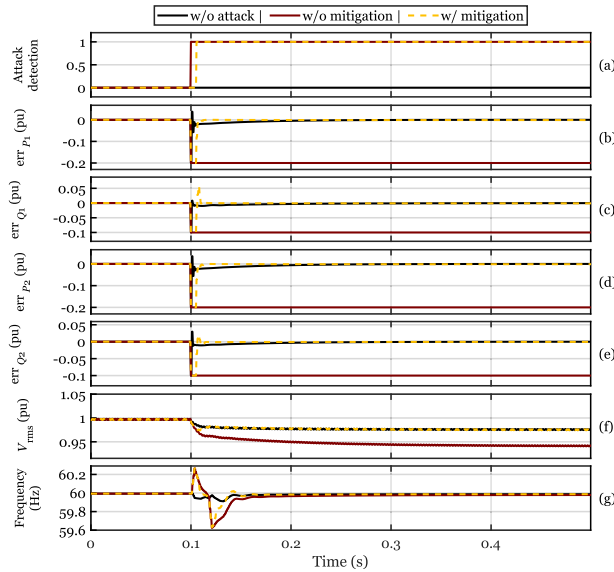


FIGURE 12. IBR 4 simulation results for a DoS attack during set point changes in IBRs 1, 2, and 4: (a) attack detection signal, (b) received err_{P1} (c) received err_{Q1} , (d) received err_{P2} , (e) received err_{Q2} , (f) voltage output, and (g) frequency.

A. DENIAL OF SERVICE (DOS) ATTACK

Fig. 12 shows the simulation results for IBR 4 in the case of a DoS attack. In this case, attackers continuously monitor packets sent in the area 1 local network. When they detect a sudden change in the error values due to a grid transient, they launch a DoS attack to disrupt all communication in area 1, preventing the IBRs from receiving new error values. In this case, the DoS attack is launched right after a -1 MW and -0.5 MVar set point change in IBRs 1, 2, and 4 at $t = 0.1$ s. IBR 4 detects this attack 6 ms after its start, as shown in Fig. 12(a). Figs. 12(b)–(e) show the IBR 1 and IBR 2 real and reactive power errors for a set point change with no attack, with a DoS attack, and with the proposed detection and mitigation. In an unmitigated attack, IBRs 1, 2, and 4 continue to use the last set of error values received before the attack. With the proposed method, the mitigation algorithm predicts new error inputs for IBRs 1, 2, and 4 after the attack detection, restoring them to their pre-attack values. Fig. 12(f) shows the voltage output of IBR 4, which decreases from 0.996 pu to 0.940 pu in an unmitigated attack. With the proposed mitigation method, it decreases to a steady state value 0.973 pu, the same as the set point change with no attack. The grid frequency, in Fig. 12(g), varies from 59.91 to 60.01 Hz for a non-attacked set point change, 59.62 to 60.27 Hz for a successful attack, and 59.63 to 60.27 Hz with detection and mitigation. In this case, the proposed method detects and mitigates the DoS attack earlier than the timeout mechanism, which flags the idle communication channels 200 ms after the attack start.

B. FALSE DATA INJECTION (FDI) ATTACKS

Step, ramp, and random FDI attacks are studied on single and multiple IBRs.

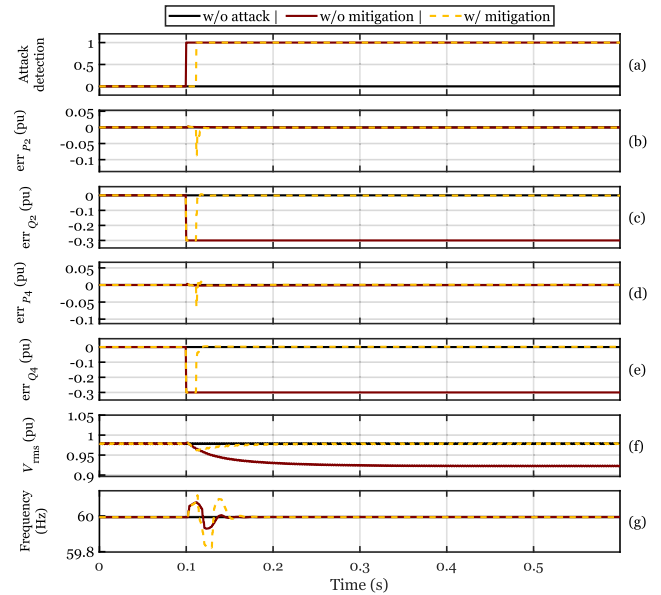


FIGURE 13. IBR 1 simulation results for an FDI step attack on the reactive power error inputs of IBR 1: (a) attack detection signal, (b) received err_{P2} (c) received err_{Q2} , (d) received err_{P4} , (e) received err_{Q4} , (f) voltage output, and (g) frequency.

1) STEP ATTACK ON ERROR INPUT OF IBR 1

Fig. 13 shows the simulation results for a -0.3 pu step attack on both reactive power error input channels of IBR 1 at $t = 0.1$ s. The detection model detects the attack 12 ms after the start of the attack, as shown in Fig. 13(a). Figs. 13(b) and (c) show the real and reactive power error received from IBR 2 and Figs. 13(d) and (e) show the real and reactive power error received from IBR 4 for non-attacked cases, attacked cases without detection and mitigation, and using the proposed method. After IBR 1 detects the attack, the mitigation method restores the reactive power error signals to their pre-attack values. Fig. 13(f) shows the IBR 1 output voltage drops from 0.978 pu to 0.922 pu in a successful attack. With detection and mitigation, however, the voltage drops to 0.961 pu and returns to its pre-attack value within 60 ms of the attack detection. Fig. 13(g) shows the frequency of the grid, varying from 59.93 to 60.08 Hz in an unmitigated attack and 59.82 to 60.12 Hz in a mitigated attack.

2) STEP ATTACKS ON ERROR OUTPUTS OF IBRS 1, 2, AND 4

Fig. 14 shows the simulation results for IBR 4 for a -0.15 pu step attack on the real and reactive power error output channels of IBRs 1, 2, and 4 at $t = 100$ ms. As Fig. 14(a) shows, IBR 4 detects the attack and turns on mitigation 6 ms after the attack start. The mitigation algorithm then restores the real and reactive power error signals from IBR 2 and the real and reactive power error signals from IBR 3 to their pre-attack values, as shown in Figs. 14(b)–(e). The voltage output of IBR 4, shown in Fig. 14(f), drops from 0.996 pu to 0.958 pu in a successful attack. With detection and mitigation, it drops to 0.986 and returns to the pre-attack value 11 ms after the attack detection. As shown in Fig. 14(g), the grid frequency

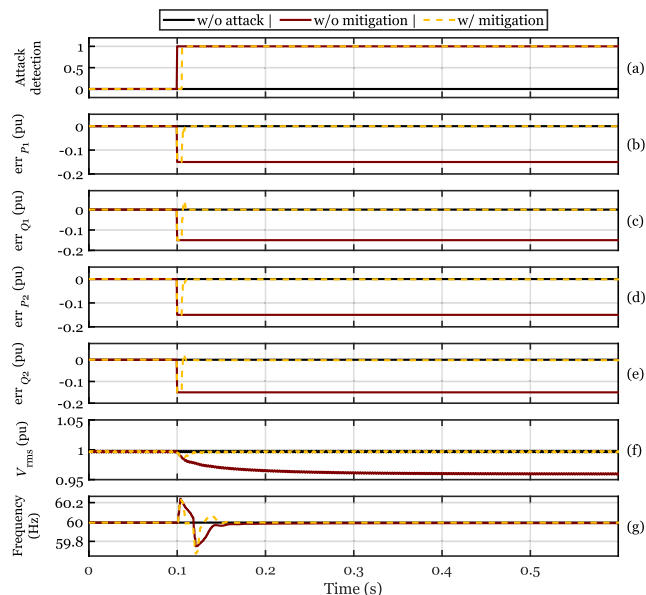


FIGURE 14. IBR 4 simulation results for FDI step attacks on the real and reactive power error outputs of IBRs 1, 2, and 4: (a) attack detection signal, (b) received err_{P1} (c) received err_{Q1} , (d) received err_{P2} , (e) received err_{Q2} , (f) voltage output, and (g) frequency.

varies from 59.75 to 60.24 Hz in an unmitigated attack and 59.67 to 60.24 Hz using the proposed mitigation method.

3) RAMP ATTACKS ON ERROR OUTPUTS OF IBRS 12, 13, AND 14

Fig. 15 shows the simulation results for IBR 14 for a 0.1 pu ramp attack on both the real and reactive power outputs of IBRs 12, 13, and 14 from $t = 0.1$ s to $t = 0.3$ s. Fig. 15(a) shows IBR 14's attack detection signal, which turns on 8 ms after the attack start. Figs. 15(b)–(e) show the error input signals, returning to their pre-attack values using the proposed mitigation method. Fig. 15(f) shows that the IBR 14 output voltage ramps up from 0.993 pu to 1.035 pu in an unmitigated attack. In the case with mitigation, the voltage does not change significantly. Fig. 15(g) shows the grid frequency, which changes slightly.

4) STEP AND RAMP ATTACKS ON ERROR OUTPUTS OF IBRS 12 AND 14

Fig. 16 shows the simulation results for IBR 13 for a hybrid attack: a 0.2 pu ramp attack on the real power error outputs of IBRs 12 and 14 from $t = 100$ ms to $t = 600$ ms and a 0.2 pu step attack on the reactive power error outputs of IBRs 12 and 14 at $t = 100$ ms. As shown in Fig. 16(a), IBR 13 detects the attack 16 ms after its start and turns on mitigation. This restores the real and reactive power errors from IBR 12 and the real and reactive power errors from IBR 14 to their pre-attack values, as shown in Figs. 16(b)–(e). Fig. 16(f) shows the IBR 13 output voltage, which rises from 0.977 pu to 1.040 pu with no mitigation; with mitigation, it increases to 0.990 pu and returns to its pre-attack value 116 ms after the attack detection. Fig. 16(g) shows that the grid frequency

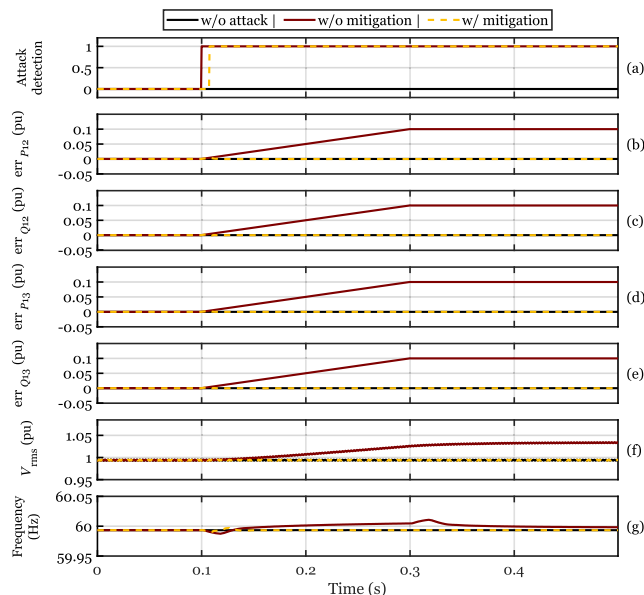


FIGURE 15. IBR 14 simulation results for FDI ramp attacks on the real and reactive power error outputs of IBRs 12, 13, and 14: (a) attack detection signal, (b) received err_{P12} (c) received err_{Q12} , (d) received err_{P13} , (e) received err_{Q13} , (f) voltage output, and (g) frequency.

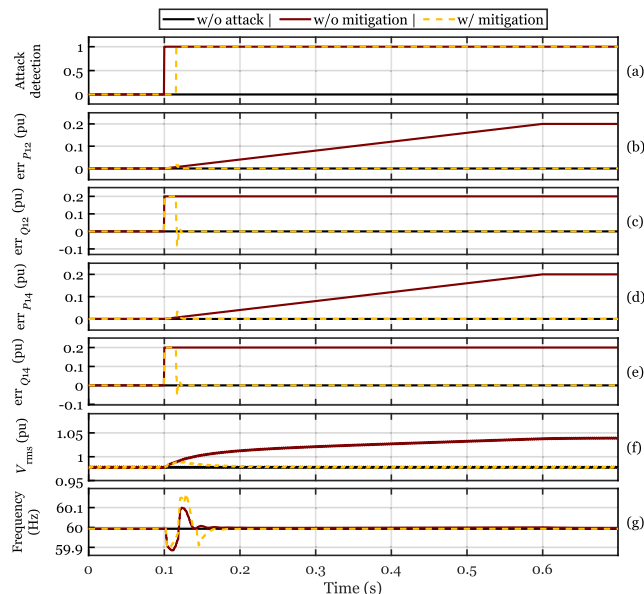


FIGURE 16. IBR 13 simulation results for FDI ramp attacks on the real power and FDI step attacks on the reactive power error outputs of IBRs 12 and 14: (a) attack detection signal, (b) received err_{P12} (c) received err_{Q12} , (d) received err_{P14} , (e) received err_{Q14} , (f) voltage output, and (g) frequency.

varies from 59.88 to 60.10 Hz in an unmitigated attack and from 59.89 to 60.17 Hz using the proposed method.

5) STEP AND RAMP ATTACKS ON ERROR OUTPUTS OF IBRS 1, 4, 12, AND 14

Fig. 17 shows the simulation results for IBR 2 for a hybrid attack on IBRs 1, 4, 12, and 14, consisting of a -0.1 pu step attack on the real power error outputs of these inverters at $t = 0.1$ s and a -0.1 pu ramp attack on the reactive power error

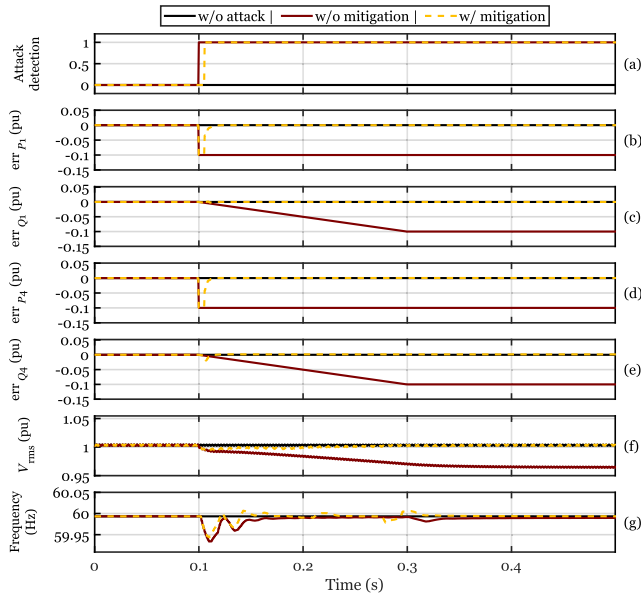


FIGURE 17. IBR 2 simulation results for FDI step attacks on the real power and FDI ramp attacks on the reactive power error outputs of IBRs 1, 4, 12, and 14: (a) attack detection signal, (b) received err_{p1} (c) received err_{q1} , (d) received err_{p4} , (e) received err_{q4} , (f) voltage output, and (g) frequency.

outputs from $t = 0.1$ s to $t = 0.3$ s. IBR 2 detects the attack 6 ms after its start, as shown in Fig. 17(a). Figs. 17(b)–(e) show the four error signals from IBRs 1 and 4, all of which return to their pre-attack values at the start of mitigation. In a successful attack, the voltage output decreases from 1.002 pu to 0.963 pu; in a mitigated attack, however, it drops to 0.995 pu and fully returns to its pre-attack value 105 ms after the detection, as shown in Fig. 17(f). The grid frequency, shown in Fig. 17(g), varies from 59.93 Hz to 59.99 Hz at the start of a successful attack. Using the proposed mitigation method, it varies from 59.94 to 60.00 Hz after the attack start.

6) RANDOM ATTACKS ON ERROR OUTPUTS OF IBRS 1, 2, AND 4

Fig. 18 shows the simulation results for IBR 1 for a random attack on the real and reactive power error outputs of IBRs 1, 2, and 4 at $t = 100$ ms. At each time step, the error signals have a probability $p = 0.5$ of being 0.3 pu. IBR 1 detects the attack and activates mitigation 120 ms after the attack start, as Fig. 18(a) shows. This restores the error signals from IBRs 2 and 4, shown in Figs. 18(b)–(e), to their pre-attack values. The output voltage, shown in Fig. 18(f), reaches 1.12 pu in a successful attack, but with mitigation, it reaches 1.09 pu and returns to its pre-attack value 183 ms after the detection. Fig. 18(g) shows the grid frequency, which varies from 58.88 to 61.47 Hz in an unmitigated attack and from 59.55 to 61.15 Hz in a mitigated attack. In this case, the proposed method detects and mitigates a random attack even though the detection models are not trained on this attack type.

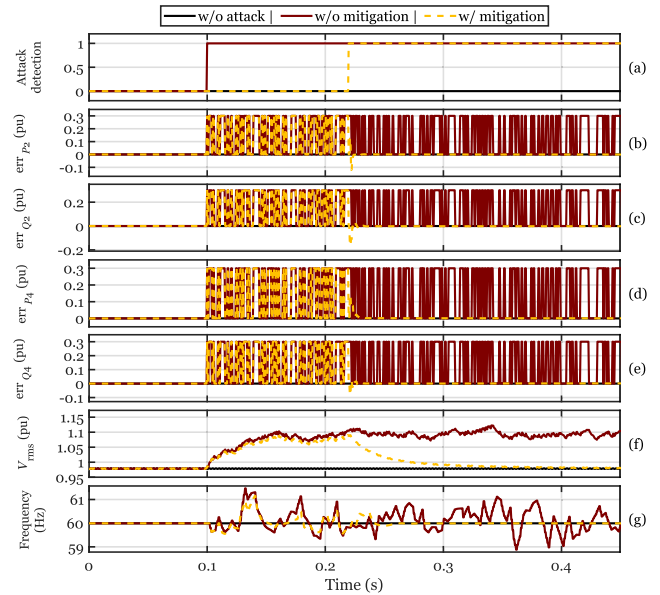


FIGURE 18. IBR 1 simulation results for FDI random attacks on the real and reactive power error outputs of IBRs 1, 2, and 4: (a) attack detection signal, (b) received err_{p2} (c) received err_{q2} , (d) received err_{p4} , (e) received err_{q4} , (f) voltage output, and (g) frequency.

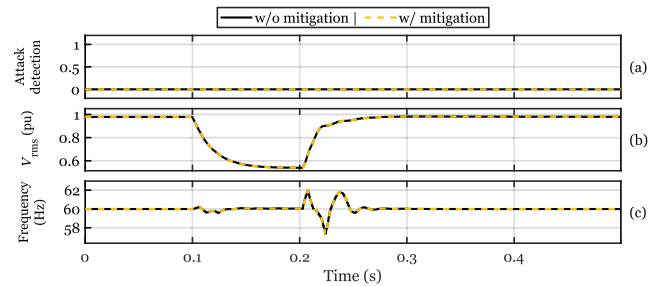


FIGURE 19. IBR 1 simulation results for a three-phase short circuit fault at bus 5: (a) attack detection signal, (b) voltage output, and (c) frequency.

C. DETECTION ALGORITHM PERFORMANCE UNDER GRID TRANSIENTS

This section evaluates the performance of the detection algorithm under three-phase short circuit faults, set point changes, and load changes.

1) THREE-PHASE SHORT CIRCUIT FAULT

Fig. 19 shows the simulation results for IBR 1 in the case of a 1 Ω three phase-to-ground fault at bus 5 starting at $t = 0.1$ s and clearing at $t = 0.2$ s. As shown in Fig 19(a), the IBR 1 detection algorithm does not activate through the duration of the fault. Therefore, both with and without the proposed method, IBR 1's output voltage drops from 0.978 pu to 0.539 pu during the fault and returns to its pre-fault value 80 ms after the fault clearance, as shown in Fig 19(b). The grid frequency Fig 19(c) varies from 57.35 to 61.97 Hz after the fault clearance and is also identical for the two cases.

2) SET POINT CHANGE

Fig. 20 shows the simulation results for IBR 13 for a 1.5 MW and 1.5 MVar set point increase in IBRs 2 and 13. The

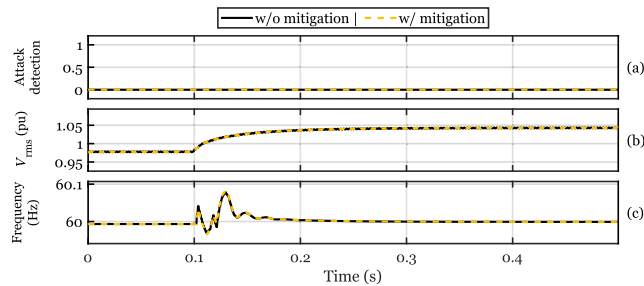


FIGURE 20. IBR 13 simulation results for a set point change in IBRs 12 and 13: (a) attack detection signal, (b) voltage output, and (c) frequency.

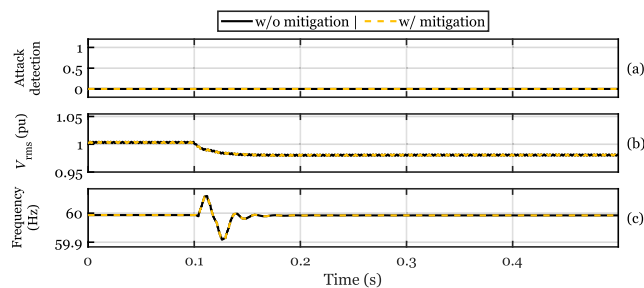


FIGURE 21. IBR 2 simulation results for a load change at bus 1: (a) attack detection signal, (b) voltage output, and (c) frequency.

IBR 13 detection algorithm does not detect this transient as an attack, as Fig 20(a) shows. Thus, the IBR response is identical both with and without the proposed method. The IBR 13 voltage output rises from 0.977 pu to 1.044 pu in both cases, as shown in Fig 20(b). The grid frequency, shown in Fig 20(c), varies from 59.97 to 60.08 Hz during the set point change in both cases.

3) LOAD CHANGE

Fig. 21 shows the simulation results for IBR 2 for an 8 MW and 6 MVar load increase at bus 1. When using the proposed method, attack detection does not activate, as shown in Fig. 21(a). The IBR response is unaffected by the proposed method. This is shown by the output voltage of IBR 2 in Fig. 21(b), which decreases from 1.002 pu to 0.979 pu when the proposed method is used as well as when it is not used. Fig. 21(c) shows the grid frequency, which varies from 59.91 to 60.06 Hz during the load change in both cases.

VII. CONCLUSION

This work proposes a cyber-resilient CSPAAACE algorithm, a model-free control method, by integrating the detection and mitigation of DoS and different FDI attacks, such as ramp, step, and random attacks, on the augmented CIGRE 14-bus North American 100% inverter-based microgrid with eight IBRs. A hybrid co-simulation testbed is created using a computer running PSCAD/EMTDC, a network switch, and two Raspberry Pi computers, where communication is implemented using TCP/IP and cyberattacks are conducted on network hardware using one of the Raspberry Pis.

The effectiveness of the detection and mitigation is tested for multiple scenarios, and the detection method does not mistakenly detect other grid transients, such as short circuit faults, load changes, and set point changes, as cyberattacks. Future work includes using physics-informed ML algorithms for cyberattack detection and mitigation.

ACKNOWLEDGMENT

The authors acknowledge Dr. A. Mohammadhassani's contributions for providing a PSCAD/EMTDC model for an IBR with CSPAAACE capabilities. Also, the authors acknowledge K. Angel's contributions for using the Ettercap program on a Raspberry Pi to conduct man-in-the-middle attacks. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy, the U.S. Department of Defense, or the United States Government.

REFERENCES

- [1] Z. A. Obaid, L. M. Cipcigan, L. Abraham, and M. T. Muhssin, "Frequency control of future power systems: Reviewing and evaluating challenges and new control methods," *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 1, pp. 9–25, Jan. 2019.
- [2] M. Hosseinzadehtaher, A. Zare, A. Khan, M. F. Umar, S. D'Silva, and M. B. Shadmand, "AI-based technique to enhance transient response and resiliency of power electronic dominated grids via grid-following inverters," *IEEE Trans. Ind. Electron.*, vol. 71, no. 3, pp. 2614–2625, Apr. 2023.
- [3] P. G. Ipoum-Ngome, D. L. Mon-Nzongo, R. C. C. Flesch, J. Song-Manguelle, M. Wang, and T. Jin, "Model-free predictive current control for multilevel voltage source inverters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9984–9997, Oct. 2021.
- [4] A. Jabbarnejad, S. Vaez-Zadeh, P. Jamallo, and J. Rodriguez, "Low-complexity model-free combined control of grid-connected converters under normal and abnormal grid conditions," *IEEE Trans. Energy Convers.*, vol. 38, no. 4, pp. 2409–2419, Jun. 2023.
- [5] A. Mehrizi-Sani and R. Iravani, "Online set point modulation to enhance microgrid dynamic response: Theoretical foundation," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2167–2174, Nov. 2012.
- [6] M. Beikbabaei, B. Alexander, A. Venkataramanan, and A. Mehrizi-Sani, "Memory-based set point modulation for improved transient response of distributed energy resources," in *Proc. IEEE Ind. Electron. Soc. Conf. (IECON)*, Chicago, IL, USA, Nov. 2024, pp. 1–6.
- [7] A. Ketabi and M. H. Fini, "An adaptive set-point modulation technique to enhance the performance of load frequency controllers in a multi-area power system," *J. Electr. Syst. Inf. Technol.*, vol. 2, no. 3, pp. 391–405, Dec. 2015.
- [8] M. Syed, A. Mehrizi-Sani, M. Robowska, E. Guillo-Sansano, D. Wang, and G. Burt, "Dynamically robust coordinated set point tracking of distributed DERs at point of common coupling," *Int. J. Electr. Power Energy Syst.*, vol. 143, Dec. 2022, Art. no. 108481.
- [9] E. Naderi and A. Asrari, "A deep learning framework to identify remedial action schemes against false data injection cyberattacks targeting smart power systems," *IEEE Trans. Ind. Informat.*, vol. 20, no. 2, pp. 1208–1219, Feb. 2024.
- [10] DRAGOS Group Rep. (Jun. 2024). *CRASHOVERRIDE Analysis of the Threat to Electric Grid Operations*. [Online]. Available: <https://www.dragos.com/resources/>
- [11] U.S. Dept. Energy (DOE). (Jun. 2024). *Electric Disturbance Events Annual Summaries*. [Online]. Available: https://www.oe.netl.doe.gov/OE417_annual_summary.aspx
- [12] DRAGOS Group Rep. (Jun. 2024). *ICS Cybersecurity Year in Review 2020*. [Online]. Available: <https://www.dragos.com/resources/>
- [13] H.-J. Lee, K.-T. Kim, J.-H. Park, G. Bere, J. J. Ochoa, and T. Kim, "Convolutional neural network-based false battery data detection and classification for battery energy storage systems," *IEEE Trans. Energy Convers.*, vol. 36, no. 4, pp. 3108–3117, Dec. 2021.

- [14] B. Hussain, Q. Du, B. Sun, and Z. Han, "Deep learning-based DDoS-attack detection for cyber-physical system over 5G network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 860–870, Feb. 2021.
- [15] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sep. 2023.
- [16] M. Baker, A. Y. Fard, H. Althuwaini, and M. B. Shadmand, "Real-time AI-based anomaly detection and classification in power electronics dominated grids," *IEEE J. Emerg. Sel. Topics Ind. Electron.*, vol. 4, no. 2, pp. 549–559, Apr. 2023.
- [17] M. Karanfil, D. E. Rebbah, M. Debbabi, M. Kassouf, M. Ghafouri, E. S. Youssef, and A. Hanna, "Detection of microgrid cyberattacks using network and system management," *IEEE Trans. Smart Grid*, vol. 14, no. 3, pp. 2390–2405, May 2023.
- [18] M. Beikbabaei and A. Mehrizi-Sani, "Real-time simulation of a resilient control center for inverter-based microgrids," in *Proc. IEEE Ind. Electron. Soc. Conf. (IECON)*, Chicago, IL, USA, Nov. 2024, pp. 1–6.
- [19] Q. Li, F. Li, J. Zhang, J. Ye, W. Song, and A. Mantooth, "Data-driven cyberattack detection for photovoltaic (PV) systems through analyzing micro-PMU data," in *Proc. IEEE Energy Convers. Congr. Expo. (ECCE)*, Detroit, MI, USA, Oct. 2020, pp. 431–436.
- [20] M. Beikbabaei, A. Venkataramanan, and A. Mehrizi-Sani, "EMT-based co-simulation of power system and communication networks," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Glasgow, U.K., Oct. 2023, pp. 1–6.
- [21] M. M. S. Khan, J. A. Giraldo, and M. Parvania, "Attack detection in power distribution systems using a cyber-physical real-time reference model," *IEEE Trans. Smart Grid*, vol. 13, no. 2, pp. 1490–1499, Mar. 2022.
- [22] H. Lin and B. Shrestha, "Cyber-physical testbed: Case study to evaluate anti-reconnaissance approaches on power grids' cyber-physical infrastructures," in *Proc. Learn. Authoritative Secur. Exp. Results (LASER) Workshop*, Sep. 2022, pp. 1–12.
- [23] A. Khan, M. B. Shadmand, and S. K. Mazumder, "Intrusion detection system for multilayer-controlled power electronics-dominated grid," *IEEE Access*, vol. 10, pp. 98329–98347, 2022.
- [24] W. Du, F. K. Tuffner, K. P. Schneider, R. H. Lasseter, J. Xie, Z. Chen, and B. Bhattarai, "Modeling of grid-forming and grid-following inverters for dynamic simulation of large-scale distribution systems," *IEEE Trans. Power Del.*, vol. 36, no. 4, pp. 2035–2045, Aug. 2021.
- [25] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8852–8859, Sep. 2020.
- [26] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6645–6649.
- [27] *Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources*, document CIGRE Task Force C6.04, CIGRE Work. Group, 2014.
- [28] A. Basit, M. Tufail, M. Rehan, W. Ahmed, A. Radwan, and I. Ahmed, "Event-based secure filtering under two-channel stochastic attacks and switching topologies over wireless sensor networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 4, pp. 3704–3717, Jul. 2024.
- [29] S. Zou, J. Huang, J. Wang, and T. He, "Flow-aware adaptive pacing to mitigate TCP incast in data center networks," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 134–147, Feb. 2021.
- [30] J. Zhou, Z. Li, Q. Wu, P. Steenkiste, S. Uhlig, J. Li, and G. Xie, "TCP stalls at the server side: Measurement and mitigation," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 272–287, Feb. 2019.
- [31] TensorFlow Lite. (Aug. 2024). *Post-Training Quantization*. [Online]. Available: https://www.tensorflow.org/lite/performance/post_training_quantization



MILAD BEIKBABAEI (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Tehran, Iran, in 2020 and 2022, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Virginia Tech, USA. His research interests include cybersecurity, power systems, renewable integration, microgrid control and protection, and 5G applications for power systems.



CAROLINE LARSEN (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from Virginia Tech, USA, in 2024, where she is currently pursuing the M.Sc. degree in electrical engineering. Her research interests include cybersecurity for power systems, control of power systems and power electronics, and renewable integration.



ALI MEHRIZI-SANI (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 2011. From 2007 to 2011, he was a Connaught Scholar with the University of Toronto. He is currently a Professor with Virginia Tech, Blacksburg, VA, USA. He previously was an Associate Professor with Washington State University, Pullman, WA, USA, and a Visiting Professor with Graz University of Technology (TU Graz), Graz, Austria. His research interests include power system applications of power electronics and integration of renewable energy resources. He was a recipient of the 2018 IEEE PES Outstanding Young Engineer Award, the 2018 ASEE PNW Outstanding Teaching Award, the 2017 IEEE Mac E. Van Valkenburg Early Career Teaching Award, the 2017 WSU EECS Early Career Excellence in Research, the 2016 WSU VCEA Reid Miller Excellence in Teaching Award, the 2011 NSERC Post-Doctoral Fellowship, and the 2007 Dennis Woodford prize. He is a Senior Editor of *IEEE TRANSACTIONS ON ENERGY CONVERSION* and an Editor of *IEEE POWER ENGINEERING LETTERS* and *IET Generation, Transmission and Distribution*.

• • •