Data-driven Detection of Stealth Cyber-attacks in DC Microgrids

Takiddin, Abdulrahman; Rath, Suman; Ismail, Muhammad; Sahoo, Subham

Published in:
I E E E Systems Journal

DOI (link to publication from Publisher):
10.1109/JSYST.2022.3183140

Publication date:
2022

Document Version
Accepted author manuscript, peer reviewed version

Data-driven Detection of Stealth Cyber-attacks in DC Microgrids

Abdulrahman Takiddin, *Graduate Student Member, IEEE*, Suman Rath, Muhammad Ismail, *Senior Member, IEEE* and Subham Sahoo, *Member, IEEE*

***Abstract*—Cyber-physical systems like microgrids contain numerous attack surfaces in communication links, sensors, and actuators forms. Manipulating the communication links and sensors is done to inject anomalous data that can be transmitted through the cyber-layer along with the original data stream. The presence of malicious, anomalous data packets in the cyber-layer of a DC microgrid can create hindrances in fulfilling the control objectives, leading to voltage instability and affecting load dispatch patterns. Hence, detecting anomalous data is essential for the restoration of system stability. This paper answers two important research questions: Which data-driven detection scheme offers the best detection performance against stealth cyber-attacks in DC microgrids? What is the detection performance improvement when fusing two features (i.e., current and voltage data) for training compared with using a single feature (i.e., current)? Our investigations revealed that (i) adopting an unsupervised deep recurrent autoencoder anomaly detection scheme in DC microgrids offers superior detection performance compared with other benchmarks. The autoencoder is trained on benign data generated from a multi-source DC microgrid model. (ii) Fusing current and voltage data for training offers a 14.7% improvement. The efficacy of the results is verified using experimental data collected from a DC microgrid testbed when subjected to stealth cyber-attacks.**

***Index Terms*—DC microgrids, anomaly detection, LSTM-autoencoder, cybersecurity.**

NOMENCLATURE

$\bar{\mathbf{V}}$	Vector notation of average voltage estimate
$\mathbf{I}^{pu}$	Vector notation of per-unit output current of all the agents
$\mathbf{L}$	Laplacian matrix
$\mathbf{W}$	Row-stochastic matrix representing the distribution of attack elements in the microgrid
c	Steady-state reference value
$H_1(s), H_2(s)$	Secondary layer PI controllers
$I_{(.)}$	Current readings
$IV_{(.)}$	Current and voltage readings
K	Number of agents
M_k	Set of neighbours of the k^{th} agent
V_{ref}, I_{ref}	Global reference voltage and current quantities for each agent
$\boldsymbol{H}$	Encoder
$\boldsymbol{R}$	Decoder
$\boldsymbol{x}$	Training row
$\boldsymbol{X}_{\text{TR}}$	Training Set

A. Takiddin is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA (email: abdulrahman.takiddin@tamu.edu).

S. Rath is with the Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA (e-mail: srath@nevada.unr.edu)

M. Ismail is with the Department of Computer Science, Tennessee Tech University, Cookeville, TN 38505, USA (email: mismail@tntech.edu).

S. Sahoo is with the Department of Energy, Aalborg University, 9220 Aalborg, Denmark (e-mail: sssa@energy.aau.dk).

I. INTRODUCTION

DC microgrids facilitate hassle-free integration of renewable energy sources [1], helping to achieve lower levels of Carbon-emission through decreased dependence on fossil fuels (e.g., coal) for power generation [2], [3]. The ability to function autonomously provides immunity to such systems against potential impacts of external faults [4]. The main control challenges faced by DC microgrids during autonomous operation are regulation of voltage and load current sharing among the distributed generators (DGs). These objectives are achieved through the use of secondary controllers coupled with communication networks to aid real-time data exchange. Such networks may have a centralized or distributed topology. However, distributed secondary control is more reliable as it is not affected due to single-point-failures [5].

The use of information and communication technology (ICT) to achieve control objectives exposes the microgrid to manipulative cyber-attacks [6]. These attacks can target the communication infrastructure [7], sensor measurements [8], and/or controllers [9]. Malicious manipulation of any of these attack surfaces may generate anomalous data. In this context, the term *anomalous data* refers to the abnormal elements present in a stream of data that do not exhibit the expected behavioral patterns. Though faults can also be the source of such anomalies [10], [11], fault-based anomalies are less sophisticated, unlike attack-based anomalies that can be specially modeled and injected through stealth attacks to inflict the desired level of damage. Such abnormal elements may propagate through the network to achieve specific objectives like voltage instability or disruptions in optimal load sharing arrangements among DGs. The following paragraphs depict some of the detection techniques proposed recently.

A. Related Works

[10] used parametric time-frequency logic to detect cyber-attack and fault-based anomalies in DC microgrids. The proposed detector extracts time-frequency information from training datasets (consisting of anomalous data) and uses the same to identify abnormal elements (present along with the normal inputs) during the testing phase. In [12], an attack detector was presented that can compare groups of elements on

the basis of whether they satisfy certain invariants. Detection of discrepancies implies the presence of false data. A signal temporal logic-based anomaly detection strategy has been presented in [13]. State estimation-based anomaly detection techniques have been proposed in [14]–[16]. However, well-crafted stealthy cyber-attacks can easily fool state observers [17]–[19]. Also, state estimation methods also require prior knowledge about the physical structure of the system. Physics-informed anomaly detection techniques have been proposed in [20], [21], which are particularly focused on distinguishing between large signal disturbances, such as grid/sensor faults and cyber-attacks.

Detection strategies that employ data-driven machine learning-based tools generally do not require information about the physical architecture of the system. Machine learning-based techniques perform anomaly detection by comparing live/captured data from the cyber-physical system with predicted values generated on the basis of reference datasets available for their training. Such techniques can be broadly categorized into four types: supervised learning, unsupervised learning, reinforcement learning [22], and semi-supervised learning-based approaches [23]. The main difference between the four categories lies in the type of reference datasets used during their training phase. Unlike the other three, supervised learning models can only be trained using labeled datasets that may or may not be accessible to researchers. [24] suggested the use of multi-class support vector machines (SVMs) for anomaly detection in microgrids. SVMs are examples of supervised learning models. In [25], a deep learning-based anomaly detection technique has been proposed to identify sensor-level cyber-attacks in DC microgrids. The authors in [26] have used an improved feedforward neural network-based approach to detect anomalies (generated as a consequence of sensor-level data integrity attacks) in microgrids. However, the authors have only considered anomaly detection in the advanced metering infrastructure and ignored other potential vulnerabilities (e.g., DG-level sensors).

Unfortunately, the aforementioned works require the availability of labeled data to train the detector. The availability of such data is not always true, especially for the zero-day cyber-attack data (attacks that have not been detected before). Also, capturing important features from the data is necessary to achieve high detection performance.

B. Contributions

In order to fill the gap in the literature, this paper answers two important research questions:

- Which data-driven detection scheme offers the best performance against stealth cyber-attacks in DC microgrids?
- Is adopting a single feature (i.e., current) sufficient for training the detector, or will fusing two features (i.e., current and voltage data) improve the results, and what would the detection improvement level be?

It turns out that the characteristics of an ideal detector for this application is to present (i) an unsupervised anomaly detection that needs to be trained using only benign data while being able to detect malicious data during the testing phase. Such an ability is possible via learning high quality features from the input (normal) data during the training phase. This enables the detector to effectively find and mark malicious data elements that do not exhibit the identified features. The detector should have (ii) a deep structure to perceive the complex patterns within the data. (iii) a recurrent mechanism to capture the time-series temporal correlations. (iii) feature fusion that incorporates current and voltage data to further improve the detection, as this enables the detector to capture distinct representations from both features. To achieve this, we carry out the following contributions.

- We utilize a long short-term memory stacked autoencoder (LSTM-SAE) as a deep recurrent unsupervised anomaly detector to identify abnormal data elements in autonomous DC microgrids. This detector is trained using datasets obtained during normal operation of a K-DG DC microgrid model with distributed network topology.
- We compare the performance of the proposed LSTM-SAE to benchmark detectors including unsupervised auto-regressive integrated moving average (ARIMA) model, one-class support vector machine (SVM), and feedforward stacked autoencoder (F-SAE) that are trained on the benign behavior. We also examine the use of supervised two-class SVM, feedforward, convolutional neural network (CNN), and LSTM classifiers trained and tested on both classes. Sequential gird-search hyperparameter optimization is carried out to enhance the results.
- We conduct multiple experiments. In the first one, using current datasets, the stacked and recurrent structure of the LSTM-SAE model provides an improvement of up to 18.3% in detection rate (DR), 12.7% in false alarm (FA), and 31% in highest difference (HD) compared to the benchmark detectors. The second experiment fuses current and voltage datasets such that the decision whether the sample is benign or malicious is based on two data sources. Doing so provided a further improvement of up to 4.7% in DR, 11.5% in FA, and 14.7% in HD. The accuracy of the results is verified further using a dataset obtained from an experimental DC microgrid testbed. The results are consistent when validated, the detection performance varies by around $\pm 0.4\%$ in most cases.

The rest of the paper is structured as follows. Section II describes cyber-physical preliminaries of microgrids. Section III discusses the used datasets. Section IV presents the details about the cyber-attacks detectors. Section V discusses the experimental results. Section VI concludes the paper.

II. Cyber-Physical Preliminaries of Microgrids

This paper considers an autonomously operating DC microgrid system with K sources. The architecture of the microgrid is shown in Fig. 1. Each of the sources (interfaced using DC/DC buck converters for regulated power conversion) are connected to one another via tie-lines. These elements collectively represent the microgrid physical layer. Operation of the power electronic converters occurs in voltage controlled mode. Proper voltage regulation and current sharing are achieved using a cooperative secondary control framework where a local

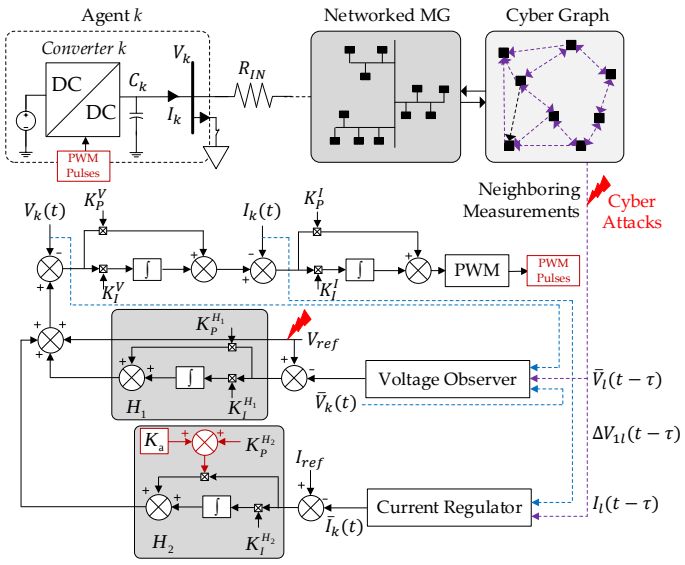


Fig. 1. Control structure of a networked DC microgrid with many agents operating with a distributed cyber graph under the presence of cyber-attacks.

controller is associated with each of the DGs [27]. All the local controllers are connected through a distributed communication network, which requires each controller to share information only with its neighboring controller(s).

The cyber layer can be considered as a graph (consisting of multiple nodes and edges), where each node represents an agent and each edge represents a communication link that connects two agents. Elements of the network compose an adjacency matrix, $\mathbf{A} = [a_{kj}] \in \mathbb{R}^{N\times N}$, where the communication weights may be expressed as $a_{kj} > 0$, if $(\psi_k, \psi_j) \in \mathbf{E}$ ($\mathbf{E}$ denotes an edge which connects ψ_k i.e., the local node and ψ_j i.e., the neighboring node). Else, $a_{kj} = 0$. The matrix for inbound cyber information can be represented as $\mathbf{Z}_{\text{in}} = \texttt{diag}\{\sum_{k\in K} a_{kj}\}$. The Laplacian matrix $\mathbf{L}$ is said to be *balanced*, if $\mathbf{A}$ and $\mathbf{Z}_{\text{in}}$ are equal (since, $\mathbf{L} = \mathbf{Z}_{\text{in}} - \mathbf{A}$).

Each of the controller units can be represented as an *agent* in the cyber layer, sending and receiving a group of measurements:

$$\mathbf{x} = \{\bar{\mathbf{V}}, \mathbf{I}^{\text{pu}}\} \tag{1}$$

with their respective neighboring agents to attain average voltage regulation and proportionate current sharing. Considering preliminaries of the communication graph, control input of the local secondary controller (associated with each DG) can be stated as:

$$u_k(t) = \sum_{j\in M_k} \underbrace{a_{kj}(x_j(t) - x_k(t))}_{e_k(t)} \tag{2}$$

where, $u_k = \{u_k^V, u_k^I\}$, $e_k = \{e_k^V, e_k^I\}$ (according to the elements present in x). Additionally, M_k is the set of neighbors of agent k. To clarify the error formulation in (11), we can simplify it using:

$$e_k^V(t) = a_{kj}(\bar{V}_j(t) - \bar{V}_k(t)) \tag{3}$$

$$e_k^I(t) = a_{kj}(I_j^{pu}(t) - I_k^{pu}(t)) \tag{4}$$

A similar extrapolation can be done to represent u_k.

TABLE I
STEALTH ATTACKS IN DC MICROGRIDS IN [29] AND [31]

Affected Counterparts	Modeling
Voltage [29]	$\mathbf{W}x_{attack}^V = 0$
Current [31]	$\mathbf{W}x_{attack}^I = 0$

Remark I: *According to the cooperative synchronization law [28], consensus will be achieved by all agents (who participate in distributed control) using* $\dot{\mathbf{x}}(t) = -\mathbf{L}\mathbf{x}(t)$ *to finally converge to* $\lim_{t\to\infty} x_k(t) = c,\ \forall\ k \in K$.

Using (2), the local control inputs necessary to achieve the control targets (average voltage regulation and proportionate sharing of load current) can be acquired from the secondary controller by using the voltage correction terms as mentioned below (for k^{th} agent) [29]:

Average Voltage Regulation:

$$\Delta V_{1_k} = H_1(s)(V_{ref} - \bar{V}_k) \tag{5}$$

Proportionate Current Sharing:

$$\Delta V_{2_k} = H_2(s)(I_{ref} - u_k^I) \tag{6}$$

where, $\bar{V}_k = V_k + \int_0^\tau \sum_{j\in M_k} u_k^V d\tau$. For proportionate current sharing, $I_{ref} = 0$. Correction terms acquired in (5) and (6) can be added to the global reference voltage for achievement of local voltage references (for the k^{th} agent) using:

$$V_{ref}^k = V_{ref} + \Delta V_{1_k} + \Delta V_{2_k}. \tag{7}$$

The target objectives mentioned in (3) and (4) are achieved by using (7) as the local reference voltage (for the k^{th} agent).

As per the distributed consensus algorithm for a heavily connected digraph (in the DC microgrid) [30], the system objectives [using (1)-(7)] shall converge to:

$$\lim_{t\to\infty} \bar{V}_k(t) = V_{ref},\ \lim_{t\to\infty} u_k^I(t) = 0\ \ \forall\ k\ \in\ K. \tag{8}$$

As shown by the red symbols in Fig. 1, malicious attackers may try to corrupt the cyber-layer in several ways (e.g., false data injection, denial-of-service, etc.) to disturb the achievement of the objectives mentioned in (8). In case of a stealth attack, the attack vector penetrates deep in the control layer by deceitfully hiding from the system operator. The ability to access multiple nodes allows such vectors to create disturbances that can be continued over an elongated stretch of time and enables them to forcefully cause generation outages. This may ultimately result in system shutdown. Hence, identifying the compromised node(s) is essential to prevent malware propagation (reducing chances of further destabilization).

Such attacks can perform coordinated manipulation to fool the system observer via the following additions in (1):

$$\mathbf{u}^a(t) = \mathbf{L}\mathbf{x}(t) + \mathbf{W}\mathbf{x}_{attack} \tag{9}$$

where $\mathbf{u}^a, \mathbf{x}$, and $\mathbf{x}_{attack}$ denote the vector representation of the attacked control input $u_k^a = \{u_k^{Va}, u_k^{Ia}\}$, the states $x_k = \{\bar{V}_k, I_k^{pu}\}$, and the attack elements $x_{attack_k} = [x_{attack_k}^V, x_{attack_k}^I]^T$, respectively. It should be noted that $\mathbf{x}_{attack}$ could be a step, sawtooth, sinusoidal, or an unbounded

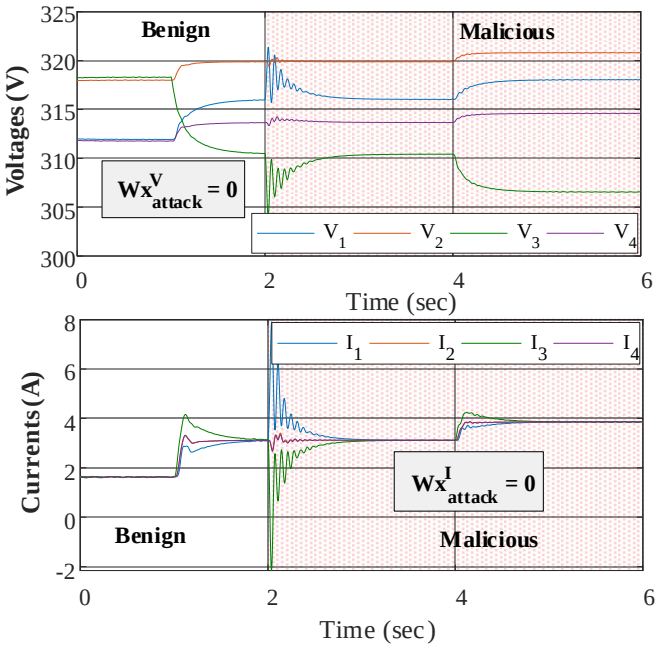


Fig. 2. Local voltage and current for each DG. Attack is initiated at t=2s.

signal. Further, $\mathbf{W} = [w_{kj}]$ depicts a row-stochastic matrix with its elements expressed by:

$$w_{kj} = \begin{cases} -\frac{1}{M_k+1}, & j \in M_k \\ 1 + \sum_{j \epsilon M_k} w_{kj}, & j = k \\ 0, j \notin M_k, j \neq k \end{cases} \tag{10}$$

The diagonal entries denote the placement of attack elements in locally measured $\mathbf{x}$. Moreover, the non-zero entries in off-diagonal elements in $\mathbf{W}$ represent the communicated measurements. Using (9), we formalize that an undetectable attack can be maintained if and only if the sum of the change in state produced by the attack and the zero input evolution of the state induced by the attack belong to the system's weakly unobservable subspace. Although $\mathbf{W}\mathbf{x}_{attack}$ will always be equal to zero from a system level perspective, the change identified across an agent is suppressed by the opposite shift in the remaining agents, without contributing any significant dynamics into the system.

III. Data Preparation

An autonomous DC microgrid model (as shown in Fig. 1) with distributed secondary control architecture is designed in the MATLAB/Simulink environment. The system consists of $K = 4$ DGs connected to each other via tie lines. The simulated parameters are found in the Appendix. The datasets are generated using this virtual test system. DG-level current and voltage measurements are observed and recorded. Benign values represent system parameters during normal operation. Malicious values are obtained by modifying certain measurements to model a cyber-attack (as per the stealth attack modeling strategy mentioned in [32]). The current and voltage measurement blocks are used to sense the local current and voltage for each DG. This data is then saved for each DG, where they are cooperating to achieve a common objective in (8). The experiments are verified further using experimental data from a DC microgrid testbed described in Section V.D.2.

A. Benign Data

To obtain the benign dataset, the simulation model is run without injecting any bias in voltage and current measurements. Thus, the system is allowed to operate normally without any manipulations. As shown in Fig. 2, the current and voltage data plotted before t = 2 sec are benign as it does not contain any bias/attack elements.

B. Malicious Data

To obtain the malicious data, the attack vector (shown in Table I) is injected into current and voltage measurements using (6). Fig. 2 shows local voltage and current for each DG when subjected to voltage and current attacks after t = 2 sec. Despite the presence of these attacks, the objectives mentioned in (5) are achieved, which makes them *stealthy* in nature. As a result, it is difficult to identify the compromised elements accurately in microgrids, which mandates automated efforts.

For each class, there is an equal number of current and voltage samples of 5.6 million readings each. For the anomaly detectors, we split the benign readings into a disjoint train $\boldsymbol{X}_{\text{TR}}$ and test sets using a $2:1$ ratio, whereas we concatenate the malicious readings with the benign test set to build the final test set $\boldsymbol{X}_{\text{TST}}$. For the supervised detectors, we concatenate both readings from both classes and split them into disjoint train $\boldsymbol{X}_{\text{TR}}$ and test $\boldsymbol{X}_{\text{TST}}$ sets using the ratio of $2:1$.

IV. Anomaly Detection

This section first discusses common machine learning-based solutions adopted to detect anomalies along with their limitations. Then, it investigates the adoption of an autoencoder-based detection and how it can overcome the limitations.

A. Benchmark Detectors

This subsection discusses several machine learning-based cyber-attacks detectors. For a comprehensive comparative analysis, we examined detectors with various characteristics including shallow/deep structure, static/recurrent mechanism, and supervised/unsupervised detection mechanism to determine which sets of characteristics lead to the best detection performance. Specifically, we investigated the use of ARIMA, one-class SVM, and F-SAE as anomaly detectors. Then, we examine the use of a two-class SVM, feedforward neural network, CNN, and LSTM classifiers as supervised detectors.

1) Anomaly Detectors: ARIMA is considered as a shallow dynamic anomaly detector trained in order to predict future patterns using minimum prediction mean square error (MSE). Then, during testing, it detects abnormal patterns whenever the MSE exceeds a certain threshold [33]. The one-class SVM is also a shallow static anomaly detector that is trained only on benign data, which is then tested on both benign and malicious samples. The F-SAE is a static deep detector that learns the behavioral patterns of benign samples throughout the reconstruction process and detects malicious samples based on their deviation from the benign ones [34].

2) Supervised Detectors: The two-class SVM is a classifier that is trained on both, benign and malicious samples, which is then tested on both types of samples [35] to make a decision using a decision boundary. The feedforward [36] model is a static deep detector that learns the behavior of samples in a singular direction using stacked hidden layers. The CNN model is a deep detector that performs convolutions on the time-series data to extract relevant features. The LSTM model is a deep recurrent neural network (RNN) type where information flows in recurrent cycles to hold previous knowledge.

There are three main limitations with such models. First, shallow architectures are not capable of capturing the complex patterns and temporal correlations present in the time-series datasets. Second, static detectors do not capture well the time-series nature of the data. Third, the detection of the supervised detectors is limited to seen attacks that are part of the training set, and hence, they are vulnerable to unseen (zero-day) attacks that are not part of the training set. Such factors negatively affect the performance of these detectors. Next, we present a deep dynamic anomaly detector that detects unseen attacks due to its unsupervised learning nature.

B. Autoencoder-based Anomaly Detection

This subsection investigates the use of autoencoders for anomaly detection due to two key features. Firstly, autoencoders may be stacked into several hidden layers, and hence, we can develop a deep structure that is capable of extracting more representative and relevant features from our datasets. Secondly, autoencoders can be equipped with a sequence-to-sequence (seq2seq) structure, and hence, they have the ability to better capture the time-series nature of our datasets. Both of these features help improve the overall detection performance, and to improve it further, a sequential grid hyperparameter optimization is carried out.

Autoencoders are types of anomaly detectors [34] that operate by learning the behavioral patterns of a (normal) class. The learned behavioral patterns of that class are then used to identify abnormal deviations from those learned patterns. Herein, we use this deviation for anomaly detection. Using anomaly detectors, specifically autoencoders, is an effective approach that aids in detecting anomalies using the reconstruction error during the reconstruction process of the data. Using SAEs, the dimensionality of the data is reduced during the encoding step and the data is reconstructed during the decoding step, where the reconstruction error represents the differences among the initial and reconstructed data. SAEs are trained on benign samples where the parameters of the encoder and decoder are optimized to have minimized reconstruction errors. Let $\boldsymbol{x}$ denote the rows of the training dataset $\boldsymbol{X}_{\text{TR}}$, $\boldsymbol{H} = f_\Theta(\boldsymbol{x})$ for the encoder, and $\boldsymbol{R} = g_\Theta(\boldsymbol{x})$ for the decoder, and Θ denote the SAE parameters where

$$\min_{\Theta} C(\boldsymbol{x}, g_\Theta(f_\Theta(\boldsymbol{x}))), \qquad \boldsymbol{x} \in \boldsymbol{X}_{\text{TR}}. \tag{11}$$

$C(\boldsymbol{x}, g_\Theta(f_\Theta(\boldsymbol{x})))$ represents the cost function (i.e. the MSE), which is responsible for penalizing $g_\Theta(f_\Theta(\boldsymbol{x}))$ due to its deviation from $\boldsymbol{x}$. Using the cost function (11), benign data will have a smaller reconstruction error compared to malicious data (anomalies). To detect an anomaly, the reconstruction error has to exceed a specific threshold value.

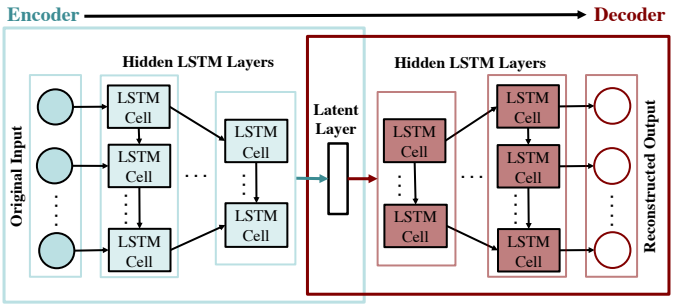


Fig. 3. Illustration of the LSTM-based stacked autoencoder architecture.

Herein, we adopt an RNN-based autoencoder, namely, LSTM for two reasons. First, it can enhance the detection performance due to its capability of capturing complex patterns and the temporal correlation in the time-series data. Second, it can overcome the vanishing gradient problem while learning temporal correlation over long intervals. Fig. 3 presents the structure of the deep LSTM-based stacked autoencoder (LSTM-SAE). The LSTM-SAE model comprises two LSTM-based RNNs; deep LSTM encoder and decoder [37], [38] where ($\boldsymbol{x} \in \boldsymbol{X}_{\text{TR}}$) denotes the LSTM encoder's input, where it encodes the time-series vector in a hidden state. This represents identifying an alternative representation of the time-series data that is more compact into the latent layer [39]. Within the encoder, after the input layer, there are L and N_l hidden LSTM layers and cells, respectively, in each LSTM layer. Within the decoder, the LSTM encoder's output is carried out as the LSTM decoder's input, which is responsible for reconstructing the initial time-series data. During training, the LSTM-SAE aims to minimize the MSE of the input-output reconstruction.

An LSTM cell presents a state c_t at a time instant t and produces a hidden state h_t as an output. The access to such a cell is controlled by input $i_{\text{E},t}$, forget $f_{\text{E},t}$, and $o_{\text{E},t}$ output gates in the encoder and additional input $i_{\text{D},t}$, forget $f_{\text{D},t}$, and output $o_{\text{D},t}$ gates. A data sample $\boldsymbol{x}_t$ at time t as well as the previous hidden states of the LSTM cells within the same layer ($h_{\text{E},t-1}$ in the encoder and $h_{\text{D},t-1}$ in the decoder) are the LSTM cell's external inputs. The cell state ($c_{\text{E},t-1}$ in the encoder and $c_{\text{D},t-1}$ in the decoder) is the LSTM cell's internal inputs. To activate the gates, the aforementioned external and internal inputs as well as the activation functions and bias are initiated. The encoder's last timestep presents the h' and c' states that are fed as the starting hidden and cell states in the decoder. Algorithm 1 shows the overall operation mechanism of the LSTM-SAE. Specifically, lines 9 - 13 and 18 - 22 present the calculation of $i_{\text{E/D},t}$, $f_{\text{E/D},t}$, and $o_{\text{E/D},t}$. The learnable weight matrices and bias vectors are denoted by $\boldsymbol{W}^l_{(\cdot)}$, $\boldsymbol{U}^l_{(\cdot)}$, $\boldsymbol{V}^l_{(\cdot)}$, and $\boldsymbol{b}^l_{(\cdot)}$. Solving (11) results in obtaining the optimal learnable parameters.

After training on $\boldsymbol{X}_{\text{TR}}$, the testing is applied on $\boldsymbol{X}_{\text{TST}}$. The cost function measures the MSE among the initial and reconstructed data, whenever it is smaller than a specific threshold,

Algorithm 1: Training of LSTM-SAE

Input Data: $\boldsymbol{X}_{\text{TR}}$
Initialization: $\boldsymbol{U}^l_{(\cdot)}$, $\boldsymbol{W}^l_{(\cdot)}$, $\boldsymbol{V}^l_{(\cdot)}$, and $\boldsymbol{b}^l_{(\cdot)}$ $\forall l$
while *not converged* **do**
  for *each* x **do**
    Feed Forward
    Encoder:
    for *each* $(l = 1, \ldots, L/2)$ **do**
      for *each timestep* t **do**
        $\boldsymbol{i}^l_{\text{E},t} = \varphi(\boldsymbol{W}^l_i \boldsymbol{x}^l_t + \boldsymbol{U}^l_i \boldsymbol{h}^l_{\text{E},t-1} + \boldsymbol{V}^l_i \boldsymbol{c}^l_{\text{E},t-1} + \boldsymbol{b}^l_i)$,
        $\boldsymbol{f}^l_{\text{E},t} = \varphi(\boldsymbol{W}^l_f \boldsymbol{x}^l_t + \boldsymbol{U}^l_f \boldsymbol{h}^l_{\text{E},t-1} + \boldsymbol{V}^l_f \boldsymbol{c}^l_{\text{E},t-1} + \boldsymbol{b}^l_f)$,
        $\boldsymbol{c}^l_{\text{E},t} = f^l_{\text{E},t} c^l_{\text{E},t-1} + i^l_{\text{E},t} \tanh(\boldsymbol{W}^l_c \boldsymbol{x}^l_t + \boldsymbol{U}^l_c \boldsymbol{h}^l_{\text{E},t-1} + \boldsymbol{b}^l_c)$,
        $\boldsymbol{o}^l_{\text{E},t} = \varphi(\boldsymbol{W}^l_o \boldsymbol{x}^l_t + \boldsymbol{U}^l_o \boldsymbol{h}^l_{\text{E},t-1} + \boldsymbol{V}^l_o \boldsymbol{c}^l_{\text{E},t} + \boldsymbol{b}^l_o)$,
        $\boldsymbol{h}^l_{\text{E},t} = \boldsymbol{o}^l_{\text{E},t} \tanh(\boldsymbol{c}^l_{\text{E},t})$,
      end
      $\boldsymbol{h}'^l = \boldsymbol{h}^l_{\text{E},t}$,
      $\boldsymbol{c}'^l = \boldsymbol{c}^l_{\text{E},t}$.
    end
    Decoder:
    At initial timestep, the decoder hidden and cell states = h'^l and c'^l.
    Encoder output is passed as decoder input $\breve{\boldsymbol{x}}$
    for *each hidden layer* $l = L/2 + 1, \ldots, L$ **do**
      for *each timestep* t **do**
        $\boldsymbol{i}^l_{\text{D},t} = \varphi(\boldsymbol{W}^l_i \breve{\boldsymbol{x}}^l_t + \boldsymbol{U}^l_i \boldsymbol{h}^l_{\text{D},t-1} + \boldsymbol{V}^l_i \boldsymbol{c}^l_{\text{D},t-1} + \boldsymbol{b}^l_i)$,
        $\boldsymbol{f}^l_{\text{D},t} = \varphi(\boldsymbol{W}^l_f \breve{\boldsymbol{x}}^l_t + \boldsymbol{U}^l_f \boldsymbol{h}^l_{\text{D},t-1} + \boldsymbol{V}^l_f \boldsymbol{c}^l_{\text{D},t-1} + \boldsymbol{b}^l_f)$,
        $\boldsymbol{c}^l_{\text{D},t} = f^l_{\text{D},t} c^l_{\text{D},t-1} + i^l_{\text{D},t} \tanh(\boldsymbol{W}^l_c \boldsymbol{x}^l_t + \boldsymbol{U}^l_c \boldsymbol{h}^l_{\text{D},t-1} + \boldsymbol{b}^l_c)$,
        $\boldsymbol{o}^l_{\text{D},t} = \varphi(\boldsymbol{W}^l_o \breve{\boldsymbol{x}}^l_t + \boldsymbol{U}^l_o \boldsymbol{h}^l_{\text{D},t-1} + \boldsymbol{V}^l_o \boldsymbol{c}^l_{\text{D},t} + \boldsymbol{b}^l_o)$,
        $\boldsymbol{h}^l_{\text{D},t} = \boldsymbol{o}^l_{\text{D},t} \tanh(\boldsymbol{c}^l_{\text{D},t})$,
      end
    end
    Back propagation: Compute:
    $\nabla_{\boldsymbol{W}^l_{(\cdot)}} C$, $\nabla_{\boldsymbol{U}^l_{(\cdot)}} C$, $\nabla_{\boldsymbol{V}^l_{(\cdot)}} C$, and $\nabla_{\boldsymbol{b}^l_{(\cdot)}} C$
  end
  update of bias and weight:
  $\boldsymbol{W}^l_{(\cdot)} = \boldsymbol{W}^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{\boldsymbol{W}^l_{(\cdot)}} C$
  $\boldsymbol{U}^l_{(\cdot)} = \boldsymbol{U}^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{\boldsymbol{U}^l_{(\cdot)}} C$
  $\boldsymbol{V}^l_{(\cdot)} = \boldsymbol{V}^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{\boldsymbol{V}^l_{(\cdot)}} C$
  $\boldsymbol{b}^l_{(\cdot)} = \boldsymbol{b}^l_{(\cdot)} - \frac{\eta}{K} \sum_x \nabla_{\boldsymbol{b}^l_{(\cdot)}} C$
end
Output: Optimal $\boldsymbol{U}^l_{(\cdot)}$, $\boldsymbol{W}^l_{(\cdot)}$, $\boldsymbol{V}^l_{(\cdot)}$, and $\boldsymbol{b}^l_{(\cdot)}$ $\forall l$.

the sample is given the label $y = 0$ (benign), otherwise, the sample is assigned the label $y = 1$ (malicious). The same model is utilized throughout the different experiments. We generate current and voltage readings throughout four equal subsets $\{I_1, I_2, I_3, I_4\}$ and $\{V_1, V_2, V_3, V_4\}$, respectively. The first experiment employs current data as an input (single feature) with binary labels; benign and malicious. The second experiment employs two features; current and voltage readings. Fusing the current and voltage datasets results in $\{IV_1, IV_2, IV_3, IV_4\}$ with binary labels; benign and malicious. Such a fusion method is applied where the model considers both the current and voltage readings during each timestep in an iterative process. This way, the reconstruction error comes from both readings in order to determine whether the sample is benign or malicious, which enhances the detection performance. For all experiments, we run the detectors on each subset and report the performance separately.

C. Performance Evaluation of the Detectors

We report three performance metrics to assess the detection performance. A true positive (TP) sample is a malicious one and detected as malicious. Similarly, a true negative (TN) sample is a benign one and detected as benign. In contrast, a false positive (FP) sample is a benign one, but detected as malicious and a false negative (FN) sample is a malicious one, but identified as benign. The reported performance metrics include detection rate (DR = TP/(TP+FN)), which specifies the amount of malicious samples that are detected as malicious, false alarm (FA = FP/(TN+FP)) that gives the amount of benign samples detected as malicious, and highest difference (HD = DR - FA) that subtracts FA from DR.

D. Threshold Values

To get the performance metrics' scores, we generate a confusion matrix by comparing $\boldsymbol{Y}_{\text{CAL}}$ to $\boldsymbol{Y}_{\text{TST}}$. Determining $\boldsymbol{Y}_{\text{CAL}}$ is done using a threshold that is compared to the reconstruction error. We determine this threshold according to the median of the interquartile range (IQR) of the receiver operating characteristic (ROC) curve. Scores that are smaller than that threshold value denote benign samples, whereas scores that are larger than that value represent malicious samples.

E. Hyperparameter Optimization

The selection of the ideal hyperparameter values for the detectors helps enhance detection performance. L denotes the ideal number of LSTM layers, which is the same in both, the encoder and decoder. N_l denotes the ideal number of neurons within the LSTM layers. O, D, A_{H}, and A_{O} denote the optimal optimizer, dropout rate, hidden activation function, and output activation function, respectively.

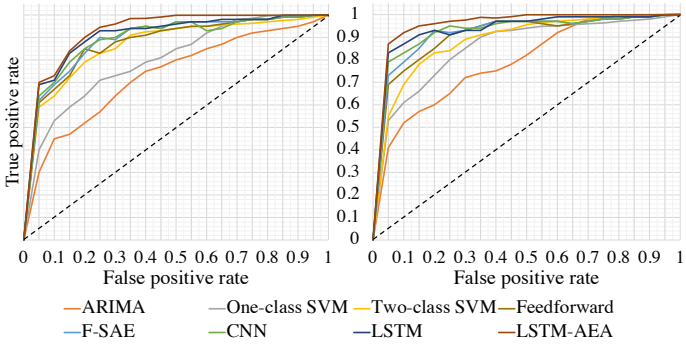
Algorithm 2 shows that the conducted hyperparameter optimization is done using four sequential steps. Since the amount of hyperparameters that we are optimizing is large, an exhaustive grid search might be associated with higher computational complexity. Therefore, we implement a grid search that is sequential instead. To select the hyperparameters, cross-validation is conducted over $\boldsymbol{X}_{\text{TR}}$. P^* denotes the hyperparameter ultimate settings that lead to improving DR against our validation set, where the given setting of hyperparameters results in a specific model (MD).

V. Simulation Results

Herein, we discuss the performance of the benchmark as well as the LSTM-SAE models when detecting anomalies. The results are reported for both of the conducted experiments as mentioned in Section IV.B.

Algorithm 2: Hyperparameter Optimization

Initialization: Optimizer = SGD, dropout rate = 0, hidden activation = Relu, output activation = Softmax
Output: The optimized hyperparameters
Input: $\boldsymbol{X}_{\text{TR}}$
for $L \in \mathcal{L}$ **do**
  for $N_l \in \mathcal{N}$ **do**
    Algorithms 1 is applied with L, N_l, and the remaining initial hyperparameters;
    DR is recorded;
  end
end
The optimal L^* and N_l^* and the remaining initial hyperparameters present MD1
for $O \in \mathcal{O}$ **do**
  Algorithm 1 is applied with MD1's hyperparameters and o;
  DR is recorded;
end
L^*, N_l^* and O^* and the remaining initial hyperparameters present MD2
for $D \in \mathcal{D}$ **do**
  Algorithms 1 is applied with MD2's hyperparameters and D;
  DR and FA;
end
L^*, N_l^*, O^*, and D^* and the remaining initial hyperparameters present MD3
for $A_{\text{H}} \in \mathcal{A}_{\text{H}}$ **do**
  for $A_{\text{O}} \in \mathcal{A}_{\text{O}}$ **do**
    Algorithms 1 is applied with MD3's hyperparameters and A_{H} and A_{O};
    DR and FA;
  end
end
L^*, N_l^*, O^*, D^*, A_{H}^*, and A_{O}^* define the optimal parameters.



(a) Using current datasets (b) Using current and voltage datasets

Fig. 4. ROC curves of the investigated detectors.

A. Computational Complexity

Training the examined detectors is conducted offline on an NVIDIA GeForce RTX 2070 hardware accelerator using Keras API. The offline training of benchmark detectors takes 1 hour and the LSTM-SAE takes 1.5 hours. The online testing requires 1.6 seconds to report a decision on a single reading.

B. Threshold Values

For the investigated anomaly detectors, the ROC curves illustrated in Fig. 4 are utilized to specify the detectors' threshold values to separate benign from malicious samples. Dividing the curve into three quartiles and obtaining the IQR's median lead to the subsequent threshold values: 0.54, 0.45, and 0.59 for the ARIMA-based, one-class SVM, and LSTM-SAE-based detectors, respectively in the first experiment (using current data). In the second experiment (using current and voltage data), the threshold values are: 0.51, 0.43, 0.52, and 0.55 for the ARIMA-based, one-class SVM, F-SAE, and LSTM-SAE detectors, respectively. The ROC curve for the two-class SVM is also plotted in Fig 4 for comparisons.

C. Hyperparameter Optimization

The selection of the ultimate hyperparameter values of the LSTM-SAE model is from: $\mathcal{L} = \{2, 3, 4, 5, 6\}$ for the number of layers, $\mathcal{N} = \{200, 300, 400, 500\}$ for the number of neurons, $\mathcal{O} = \{\text{SGD, Adam, Adamax}\}$ for the optimizer, $\mathcal{D} = \{0, 0.2, 0.4\}$ for the dropout rate, $\mathcal{A}_{\text{H}} = \{\text{Relu, Sigmoid, Tanh}\}$ for the hidden activation function, $\mathcal{A}_{\text{O}} = \{\text{Softmax, Sigmoid}\}$ for the output activation function.

For both of the experiments, the ideal hyperparameter combination of the LSTM-SAE detector turns out to be as follows. The optimal number of LSTM layers is four, where the optimal number of neurons in the two encoder layers is $(500, 300)$ with the inverse order $(300, 500)$ in the decoder's side. The optimal optimizer and dropout rate are Adam and 0.2, respectively. Sigmoid is the optimal choice for both, the hidden and output activation functions. In the ARIMA-based detector, the differencing and moving average values are 1 and 0, respectively. For the SVM detectors, scale and sigmoid are the ideal kernel and gamma, respectively. The optimal feedforward parameters are 6 layers with 300 neurons, Adamax optimizer, 0.2 dropout rate, and Sigmoid hidden and output activation function. The F-SAE model has the same amount of layers and neurons as the LSTM-SAE with an SGD optimizer, 0.4 dropout rate, and Sigmoid and Softmax for the hidden and output activation functions, respectively. The LSTM-model has 6 layers with 500 cells, Adam optimizer, no dropout rate, weight constraint of 5, ReLU and Softmax hidden and output activation function, respectively, as the ideal parameters.

D. Performance Evaluation

This subsection discusses the detection performance of the examined detectors using the simulated data discussed in Section III. We also use experimental data to validate the performance results.

1) Simulated Data: Table II presents the results of the first experiment, which reports the performance of the developed detectors using only the four current datasets as well as their average performance. The average performance of the LSTM-SAE-based detector shows that it significantly outperforms the rest of the detectors. Specifically, the LSTM-SAE-based detector outperforms the benchmark detectors by $3.5-18.3\%$, $2.6-12.7\%$, and $6.1-31\%$ in DR, FA, and HD, respectively. Table III summarizes the results of the second experiment, which reports the performance of the examined detectors using the four current and voltage datasets. According to the

TABLE II
PERFORMANCE USING SIMULATED CURRENT DATASETS

Model	Metric	Simulated dataset				Avg
		I_1	I_2	I_3	I_4	
ARIMA	DR	74.2	73.4	72.2	72.3	73.0
	FA	30.4	30.2	31.4	32.1	31.0
	HD	43.8	43.2	40.8	40.2	42.0
One-class SVM	DR	79.7	78.5	77.7	77.3	78.3
	FA	27.9	28.4	28.9	28.9	28.5
	HD	51.8	50.1	48.8	48.4	49.8
Two-class SVM	DR	84.2	83.7	81.9	82.3	83.0
	FA	22.9	22.4	24.2	23.5	23.3
	HD	61.3	61.3	57.7	58.8	59.8
Feedforward	DR	85.5	85.3	85.2	85.4	85.4
	FA	22.4	22.7	21.7	22.9	22.4
	HD	63.1	62.6	63.5	62.5	62.9
F-SAE	DR	86.5	87.2	87.2	87.5	87.1
	FA	22.1	22.2	21.4	21.3	21.8
	HD	64.4	65.0	65.8	66.2	65.4
CNN	DR	87.3	87.7	87.1	87.5	87.4
	FA	20.9	21.5	20.5	22.1	21.3
	HD	66.4	66.2	66.6	65.4	66.2
LSTM	DR	87.4	88.6	88.1	87.0	87.8
	FA	20.7	21.1	21.1	20.8	20.9
	HD	66.7	67.5	67.0	66.2	66.9
LSTM-SAE	DR	90.1	91.4	91.5	92.1	91.3
	FA	18.5	17.1	19.5	18.2	18.3
	HD	71.6	74.3	72.0	73.9	73.0

TABLE III
PERFORMANCE USING SIMULATED CURRENT AND VOLTAGE DATASETS

Model	Metric	Simulated dataset				Avg
		IV_1	IV_2	IV_3	IV_4	
ARIMA	DR	78.2	77.1	76.8	78.4	77.6
	FA	20.5	22.1	20.9	20.2	20.9
	HD	57.7	55.0	55.9	58.2	56.7
One-class SVM	DR	83.1	83.4	82.2	83.4	83.0
	FA	18.3	18.0	19.2	19.5	18.8
	HD	64.8	65.4	63.0	63.9	64.3
Two-class SVM	DR	84.2	88.4	82.1	88.4	85.8
	FA	16.1	16.7	16.2	16.3	16.3
	HD	68.1	71.7	65.9	72.1	69.5
Feedforward	DR	89.2	90.6	89.3	90.1	89.8
	FA	13.1	12.9	13.5	12.6	13.0
	HD	76.1	77.7	75.8	77.5	76.8
F-SAE	DR	89.7	90.4	90.5	90.6	90.3
	FA	11.1	11.4	11.3	12.0	11.5
	HD	78.6	79.0	79.2	78.6	78.9
CNN	DR	90.4	90.6	89.9	90.9	90.5
	FA	9.5	10.1	11.2	11.0	10.5
	HD	80.9	80.5	78.7	79.9	80.0
LSTM	DR	90.4	90.9	91.7	90.6	90.9
	FA	9.4	10.1	10.5	9.5	9.9
	HD	81.0	80.8	81.2	81.1	81.0
LSTM-SAE	DR	94.1	93.6	94.4	94.0	94.0
	FA	6.6	7.8	5.4	7.2	6.8
	HD	87.5	85.8	89.0	86.8	87.3

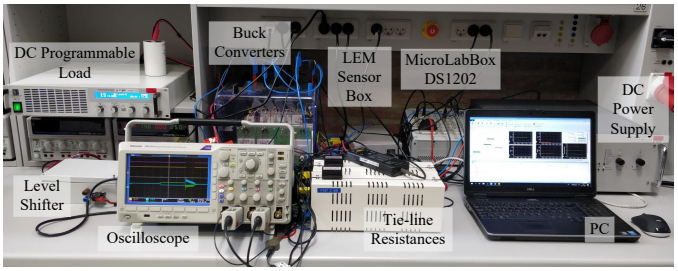


Fig. 5. Experimental setup of a cooperative DC microgrid comprising of N = 2 agents controlled by dSPACE MicroLabBox DS1202 supplying power to the programmable constant power load.

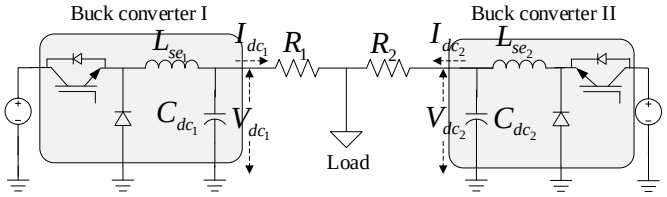


Fig. 6. Single line diagram of the experimental setup shown in Fig. 5.

simulation results, the LSTM-SAE-based detector also outperforms the rest of the benchmark detectors by $3.1-16.4\%$, $3.1-14.1\%$, and $6.3-30.6\%$ in DR, FA, and HD, respectively. The superior performance of the LSTM-SAE-based detector is due to its deep structure, which gives it the ability to better capture the complex patterns of the data. Also, its recurrent architecture allows it to apprehend the temporal correlations within the time-series data. Moreover, given its unsupervised anomaly training nature, the detection is done on totally unseen data, which means that it can detect zero-day attacks.

Fusing the voltage and current data helps in improving the detection performance of the detectors. Specifically, the average HD of the detectors has improved by $9.7-14.8\%$. This is due to the fact that utilizing the obtained reconstruction error from both the current and voltage data helps in increasing the models' certainty regarding the decision on whether a sample is benign or malicious. Conducting such a data fusion method provided an improvement of up to 4.6% in DR, 11.5% in FA, and 14.7% in HD.

2) Validation on Experimental Data: As illustrated in Fig. 5, the multi-labeled dataset is obtained from a DC microgrid experimental testbed that is operating at a voltage reference $V_{dc_{ref}}$ of 48 V with N = 2 DC/DC buck converters that are tied radially to a programmable load (voltage-dependent mode). Each converter is controlled using the control structure in Fig. 1 by dSPACE MicroLabBox DS1202 (target), with control commands from the ControlDesk in the PC (host). A single line diagram of the experimental setup is shown in Fig. 6. The control strategy is operated under the presence and absence of stealth cyber-attacks throughout the local and neighboring

TABLE IV
PERFORMANCE USING EXPERIMENTAL CURRENT DATA

Model	Metric	Exp dataset		Avg *(difference)*
		I_1	I_2	
ARIMA	DR	73.1	72.3	72.7 *(-0.3)*
	FA	32.1	31.6	31.9 *(+0.9)*
	HD	41.0	40.7	40.9 *(-1.1)*
One-class SVM	DR	79.2	77.5	78.4 *(+0.1)*
	FA	28.1	29.4	28.8 *(+0.3)*
	HD	51.1	48.1	49.6 *(-0.2)*
Two-class SVM	DR	83.1	81.8	82.5 *(-0.5)*
	FA	22.4	24.6	23.5 *(+0.2)*
	HD	60.7	57.2	59.0 *(-0.8)*
Feedforward	DR	85.3	86.2	85.8 *(+0.4)*
	FA	23.1	22.5	22.8 *(+0.4)*
	HD	62.2	63.7	63.0 *(+0.1)*
F-SAE	DR	86.4	87.5	87.0 *(-0.1)*
	FA	22.0	21.2	21.6 *(-0.2)*
	HD	64.4	66.3	65.4 *(0.0)*
CNN	DR	87.4	86.7	87.1 *(-0.3)*
	FA	21.0	20.7	20.9 *(-0.4)*
	HD	66.4	66.0	66.2 *(0.0)*
LSTM	DR	88.3	87.6	88.0 *(+0.2)*
	FA	20.6	20.1	20.4 *(-0.5)*
	HD	67.7	67.5	67.6 *(+0.7)*
LSTM-SAE	DR	91.1	91.8	91.5 *(+0.2)*
	FA	18.9	18.2	18.6 *(+0.3)*
	HD	72.2	73.6	72.9 *(-0.1)*

TABLE V
PERFORMANCE USING EXPERIMENTAL CURRENT AND VOLTAGE DATA

Model	Metric	Exp dataset		Avg *(difference)*
		IV_1	IV_2	
ARIMA	**DR**	78.8	77.6	78.2 *(+0.6)*
	FA	21.1	20.5	20.8 *(-0.1)*
	HD	57.7	57.1	57.4 *(+0.6)*
One-class SVM	**DR**	82.2	81.7	82 *(-1.0)*
	FA	18.3	18.7	18.5 *(-0.3)*
	HD	63.9	63.0	63.5 *(-0.8)*
Two-class SVM	**DR**	87.6	85.4	86.5 *(+0.7)*
	FA	15.9	16.7	16.3 *(0.0)*
	HD	71.7	68.7	70.2 *(+0.7)*
Feedforward	**DR**	88.9	89.7	89.3 *(-0.3)*
	FA	13.1	12.5	12.8 *(-0.2)*
	HD	75.8	77.2	76.5 *(-0.1)*
SAE	**DR**	89.0	90.7	89.9 *(-0.4)*
	FA	12.4	11.0	11.7 *(+0.2)*
	HD	76.6	79.7	78.2 *(-0.6)*
CNN	**DR**	90.4	91.1	90.8 *(+0.3)*
	FA	10.5	10.7	10.6 *(+0.2)*
	HD	79.9	80.4	80.2 *(+0.1)*
LSTM	**DR**	90.8	90.9	90.9 *(0.0)*
	FA	9.7	9.4	9.6 *(-0.3)*
	HD	81.1	81.5	81.3 *(+0.3)*
LSTM-SAE	**DR**	94.3	94.1	94.2 *(+0.2)*
	FA	6.3	6.9	6.6 *(-0.2)*
	HD	88.0	87.2	87.6 *(+0.3)*

measurements. The parameters of the experimental testbed are given in Appendix. The results shown in Tables IV and V verify the correctness of our conducted simulations. $\{I_1, I_2\}$ and $\{IV_1, IV_2\}$ denote the current and voltage readings from the two converters, respectively. Running the investigated detection schemes on the testbed offers consistent performance that varies only by around $\pm 0.4\%$ compared to the detection performance using the simulated data.

VI. CONCLUSION

This paper answered two important research questions regarding data-driven-based approaches for stealth cyber-attack detection in DC microgrids. Our extensive experiments provide the following conclusions: (1) Adopting an LSTM-based stacked autoencoder offers superior detection performance compared to benchmark machine learning-based detectors due to its deep recurrent structure. Such characteristics help in discovering the complex patterns and temporal correlations of the time-series dataset. Also, the LSTM-SAE model can detect unseen attacks since it is an unsupervised anomaly detector that is trained only on benign data. Utilizing only current data for training, the LSTM-SAE model offered an improvement of up to 18.3% in DR, 12.7% in FA, and 31% in HD compared to benchmark detectors. (2) Performing feature fusion that incorporates current and voltage data for training improved the detection performance further by up to 4.7% in DR, 11.5% in FA, and 14.7% in HD as it enables the detector to capture distinct representations from both features. Running the investigated detection schemes on a real testbed offered consistent performance that varies only by $\pm 0.4\%$ compared to the detection performance using the simulated data.

APPENDIX

Simulation Parameters

The test model is composed of four DGs (rated for 6 kW each). The line parameter R_{kl} is attached from the k^{th} agent to the l^{th} agent where each agent has identical controller gains.
Plant: R_{12} = 1.8 Ω, R_{14} = 1.3 Ω, R_{23} = 2.3 Ω, R_{43} = 2.1 Ω
Converter: L_k= 3 mH, C_k= 250 μF, I_{min} = 0 A, I_{max} = 18 A, V_{min} = 270 V, V_{max} = 360 V.
Controller: $V_{dc_{ref}}$= 315 V, $I_{dc_{ref}}$ = 0, $K_P^{H_1}$ = 3, $K_I^{H_1}$ = 0.01, $K_P^{H_2}$ = 4.5, $K_I^{H_2}$ = 0.32, G_{VP} = 2.8, G_{VI} = 12.8, G_{CP} = 0.56, G_{CI} = 21.8, V_{in} = 270 V.

Experimental Testbed Parameters

The system is composed of two sources with 600 W equally rated converters, and for each converter, the controller gains are consistent.
Plant: R_1 = 0.9 Ω, R_2 = 1.2 Ω
Converter: L_{se_i}= 3 mH, C_{dc_i}= 100 μF
Controller: $V_{dc_{ref}}$= 48 V, $I_{dc_{ref}}$ = 0, $K_P^{H_1}$ = 1.92, $K_I^{H_1}$ = 15, $K_P^{H_2}$ = 4.5, $K_I^{H_2}$ = 0.08.

REFERENCES

[1] F. Al-Ismail, "DC microgrid planning, operation, and control: A comprehensive review," *IEEE Access*, vol. 9, pp. 36 154–36 172, 2021.

[2] M. M. Rahman and A. Mallick, "Measurement of the carbon footprint for bangladesh's electricity generation in 2009-15," in *Emerging Tech. in Computing, Comm. and Electronics (ETCCE)*, 2020, pp. 1–6.

[3] C. Marpaung, A. Soebagio, and R. Shrestha, "The role of carbon capture and storage and renewable energy for CO2 mitigation in the indonesian power sector," in *Inter. Power Eng. Conf. (IPEC)*, 2007, pp. 779–783.

[4] S. Rath, D. Pal, P. S. Sharma, and B. K. Panigrahi, "A cyber-secure distributed control architecture for autonomous ac microgrid," *IEEE Systems Journal*, pp. 1–12, 2020.

[5] T. Qian *et al.*, "Event-triggered updating method in centralized and distributed secondary controls for islanded microgrid restoration," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1387–1395, 2019.

[6] S. Sahoo, J. C.-H. Peng, S. Mishra, and T. Dragičević, "Distributed screening of hijacking attacks in DC microgrids," *IEEE Transactions on Power Electronics*, vol. 35, no. 7, pp. 7574–7582, 2019.

[7] F. Ahmadloo and F. R. Salmasi, "A cyber-attack on communication link in distributed systems and detection scheme based on h-infinity filtering," in *IEEE Inter. Conf. on Industrial Tech. (ICIT)*, 2017, pp. 698–703.

[8] S. Mazumder *et al.*, "A review of current research trends in power-electronic innovations in cyber–physical systems," *IEEE Jour. Emerging and Selct. Tpcs. in Power Electronics*, vol. 9, no. 5, pp. 5146–5163, 2021.

[9] S. Sahoo, T. Dragičević, and F. Blaabjerg, "Cyber security in control of grid-tied power electronic converters—challenges and vulnerabilities," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, no. 5, pp. 5326–5340, 2021.

[10] O. A. Beg, L. V. Nguyen, T. T. Johnson, and A. Davoudi, "Cyber-physical anomaly detection in microgrids using time-frequency logic formalism," *IEEE Access*, vol. 9, pp. 20 012–20 021, 2021.

[11] R. Moghaddass and J. Wang, "A hierarchical framework for smart grid anomaly detection using large-scale smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5820–5830, 2017.

[12] O. A. Beg, T. T. Johnson, and A. Davoudi, "Detection of false-data injection attacks in cyber-physical DC microgrids," *IEEE Transactions on industrial informatics*, vol. 13, no. 5, pp. 2693–2703, 2017.

[13] O. A. Beg, L. V. Nguyen, T. T. Johnson, and A. Davoudi, "Signal temporal logic-based attack detection in DC microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3585–3595, 2018.

[14] T. Vu *et al.*, "Distributed optimal dynamic state estimation for cyber intrusion detection in networked DC microgrids," in *Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, 2019, pp. 4050–4055.

[15] N. Muralidhar *et al.*, "illiad: Intelligent invariant and anomaly detection in cyber-physical systems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 3, pp. 1–20, 2018.

[16] G. Anagnostou *et al.*, "Observer-based anomaly detection of synchronous generators for power systems monitoring," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4228–4237, 2018.

[17] P. Cheng, Z. Yang, J. Chen, Y. Qi, and L. Shi, "An event-based stealthy attack on remote state estimation," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4348–4355, 2019.

[18] S. Paudel, P. Smith, and T. Zseby, "Stealthy attacks on smart grid pmu state estimation," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, pp. 1–10.

[19] E.-N. S. Youssef and F. Labeau, "False data injection attacks against state estimation in smart grids: Challenges and opportunities," in *IEEE Canadian Conf. on Electrical & Comp. Eng. (CCECE)*, 2018, pp. 1–5.

[20] K. Bhatnagar, S. Sahoo, F. Iov, and F. Blaabjerg, "Physics guided data-driven characterization of anomalies in power electronic systems," in *6th IEEE Workshop on the Electronic Grid (eGRID)*, 2021, pp. 01–06.

[21] K. Gupta, S. Sahoo, R. Mohanty, B. K. Panigrahi, and F. Blaabjerg, "Decentralized anomaly characterization certificates in cyber-physical power electronics based power systems," in *IEEE 22nd Workshop on Control and Model. of Power Electronics (COMPEL)*, 2021, pp. 1–6.

[22] H. o. Rouzbahani, "Anomaly detection in cyber-physical systems using machine learning," in *Handbook of big data privacy*. Springer, 2020, pp. 219–235.

[23] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on AI and ML*, vol. 3, no. 1, pp. 1–130, 2009.

[24] A. A. Khan, O. A. Beg, M. Alamaniotis, and S. Ahmed, "Intelligent anomaly identification in cyber-physical inverter-based systems," *Electric Power Systems Research*, vol. 193, p. 107024, 2021.

[25] H. Cui *et al.*, "Cyber attack detection process in sensor of DC microgrids under electric vehicle based on hilbert-huang transform and deep learning," *IEEE Sensors Journal*, 2020.

[26] A. Kavousi, W. Su, and T. Jin, "A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids," *IEEE Trans. on Industrial Informatics*, vol. 17, no. 1, pp. 650–658, 2020.

[27] V. Nasirian, S. Moayedi, A. Davoudi, and F. L. Lewis, "Distributed cooperative control of DC microgrids," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2288–2303, 2015.

[28] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[29] S. Sahoo, S. Mishra, J. C. Peng, and T. Dragičević, "A stealth cyber-attack detection strategy for DC microgrids," *IEEE Transactions on Power Electronics*, vol. 34, no. 8, pp. 8162–8174, 2019.

[30] S. Sahoo and S. Mishra, "A distributed finite-time secondary average voltage regulation and current sharing controller for DC microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 282–292, 2019.

[31] S. Sahoo *et al.*, "On detection of false data in cooperative DC microgrids—a discordant element approach," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6562–6571, 2020.

[32] S. Sahoo, S. Mishra, J. C.-H. Peng, and T. Dragičević, "A stealth cyber-attack detection strategy for DC microgrids," *IEEE Transactions on Power Electronics*, vol. 34, no. 8, pp. 8162–8174, 2018.

[33] V. Krishna, R. Iyer, and W. Sanders, "ARIMA-Based modeling and validation of consumption readings in power grids," in *Critical Information Infrastructures Security*. Springer, 2016, pp. 199–210.

[34] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Systems Journal*, pp. 1–12, Jan. 2022.

[35] P. Jokar, N. Arianpoo, and V. C. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2016.

[36] Z. Zhang *et al.*, "Delay-tolerant predictive power compensation control for photovoltaic voltage regulation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4545–4554, 2021.

[37] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Variational auto-encoder-based detection of electricity stealth cyber-attacks in AMI networks," in *European Signal Processing Conf. (EUSIPCO)*. Amsterdam, Netherlands, 18–21 Jan. 2021, pp. 1590–1594.

[38] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based detection of electricity stealth cyberattacks in AMI networks," in *Intern. Symposium on Signals, Circuits and Systems (ISSCS)*. Iasi, Romania, 15–16 Jul. 2021, pp. 1–6.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.