Detecting a Stealthy Attack in Distributed Control for Microgrids using Machine Learning Algorithms

Mingxiao Ma, Abdelkader Lahmadi, Isabelle Chrisment

Detecting a Stealthy Attack in Distributed Control for Microgrids using Machine Learning Algorithms

Mingxiao Ma, Abdelkader Lahmadi, Isabelle Chrisment
Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
Email: {mingxiao.ma, abdelkader.lahmadi, isabelle.chrisment}@loria.fr

***Abstract*—With the increasing penetration of inverter-based distributed generators (DG) into low-voltage distribution microgrid systems, it is of great importance to guarantee their safe and reliable operations. These systems leverage communication networks to implement a distributed and cooperative control structure. However, the detection of stealthy attacks with a large impact and weak detection signals on such distributed control systems is rarely studied. In this paper, we address the problem of detecting a stealthy attack, named MaR, on the communication network of a microgrid while an attacker modifies the voltage measurement with the reference values. We collect datasets from a hardware platform modeled after a simplified microgrid and running the MaR attack performed with a Man-in-the-Middle (MitM) technique. We use the collected datasets to compare different attack detection algorithms based on multiple categories of machine learning algorithms. Our results show that the Random Forest algorithm outperforms the others to detect suspicious packets modified by a MitM attacker with an accuracy close to 97%.**

***Index Terms*—Microgrid, Security, Attack detection, Machine learning, CPS systems.**

I. Introduction

Communication networks are playing an increasingly significant role in guaranteeing reliable and safe operations of the complex electrical power gird coordination [1]. However, the extensive use of communication networks also introduces cybersecurity vulnerabilities to the power systems, especially with the growing adaption of distributed generators (DGs), where more data is collected, transmitted, and processed through communication networks.

Cyber attacks and their related detection issues have not been extensively explored in the power distribution network of a power grid [2–4]. Existing work mainly concentrates on deception attacks against the physical layer of smart grid systems. A typical deception attack is the false data injection (FDI), which has been substantially studied in the literature. As described in the distributed sparse attacks introduced in [5], the FDI attack injects false data into the local measurements of the phasor measurement units (PMUs) in a hierarchically structured network. The detection of an FDI attack is usually based on state vector estimation (SVE), where the detector gets the system state estimated from the observed measurements and then computes the residual between the observation and the estimation. If the residual is greater than the given threshold, an FDI attack is detected [5, 6]. However, it is challenging to accurately recover state vectors for SVE in networks with sparse Jacobian measurement matrix [5]. Sparse reconstruction techniques can be an option to solve this problem, however the sparsity of the state vectors restricts its performance [5]. Moreover, there is a possibility that the injected false data vectors could reside in the column space of the sparse matrix and satisfy certain sparsity conditions, then the FDI attacks, known as unobservable or stealthy attacks, cannot be detected [6, 7].

Compared to the above traditional attack detection methods based on threshold comparison, statistical learning algorithms are able to detect not only observable but also unobservable attacks [8]. The attack detection problem can be modeled with machine learning techniques for statistical classification of measurements. Specifically, the attack detection is a typical two-classes (normal and abnormal) classification problem. Supervised learning methods could be employed to design attack detectors at the communication network level [9]. In [8], experimental investigations prove that machine learning algorithms outperform SVE-based methods in detection attacks for smart grid systems. Besides, in the low-voltage distribution microgrid case, there is no globally centralized estimation system to dynamically measure and estimate the system states, so traditional SVE based detection methods cannot be applied directly.

In this work, we study attack detection for microgrids with a distributed and cooperative control system. The control system has a hierarchical structure including quadratic droop control as primary control and distributed tracking control as secondary control. Distributed and cooperative control improves the scalability and reliability of microgrids because of its better support for a more substantial penetration of local loads and DGs than traditional centralized control structure as considered in [1] and [4]. Based on this distributed and cooperative control model, a stealthy cyberattack named measurement as reference attack (MaR) that targets the communication links connecting each DG is considered. MaR attack can cause serious impact like system voltage fluctuation and reference synchronization errors, which is detailed in our previous work [10]. We build an experimental microgrid platform with Raspberry Pi and Arduino devices to realize a distributed and cooperative control system. We implement the MaR stealthy attack using a Man-in-the-Middle (MitM) technique and generate real datasets to assess the attack detection mechanism. We design a simple threshold comparison based attack detection method and also apply five groups of supervised machine learning algorithms (Bayesian Algorithms, Decision Tree Algorithms, Instance-based Algorithms, Artificial Neural Network Algorithms, Ensemble Algorithms) to detect the stealthy attack. We compare these different methods to identify the best performing detection scheme. We find that Random Forest outperforms the others to detect suspicious packets modified by a MitM attacker. To the best of our knowledge, this is the first work studying machine learning based detection methods for detecting attack targeting distributed and cooperative controlled microgrid systems.

The rest of the paper is organized as follows. Section II presents a distributed cooperative control model of microgrid and its associated threat model. Section III describes the experiments that we carried to generate datasets from an experimental microgrid platform. Section IV evaluate the performance of both threshold based and machine learning based attack detection methods. Finally, we conclude the paper in Section V.

II. Problem Formulation

In this section, we describe a distributed and cooperative control system of a microgrid and its threat model using the measurement-as-reference (MaR) attack that we have designed in [10].

A. Distributed cooperative control system

In our previous work [10], we propose a distributed cooperative control scheme for microgrid voltage dynamics. As shown in Fig. 1, the microgrid is composed of interconnected DG units. Each DG unit consists of lumped inverter-based distributed energy resources (DER) and loads. We assume that there are N DG units in the considered microgrid and the DG Unit_1 is directly connected to the main grid.

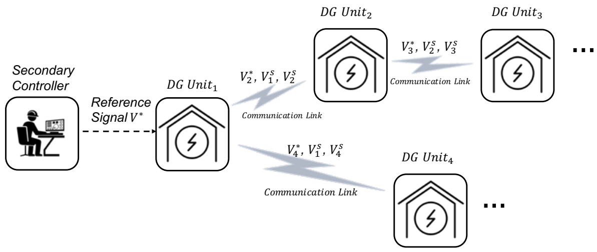


Fig. 1: A distributed and cooperative microgrid control system consisting of both primary and secondary controls. Interconnected DG Units can exchange data with neighboring DG units through sparse communication networks to transmit reference values and measurement values (denoted by the superscript s) [10].

1) Communication network model: At the communication level of the microgrid, each DG unit is considered as one node of the communication graph and each communication link is represented as an edge. The communication network from Fig. 1 has a line topology. A generic microgrid network topology can be denoted by a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A_G)$, with a nonempty N-node finite set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, a edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and an adjacency matrix A_G. The set $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ represents the neighbor set of node i. a_{ij} represents the weight for the edge connecting from node j to node i. Note $a_{ij} = 0$ if no data transfers from node j to node i. For a N-node digraph, the adjacency matrix $A_G = [a_{ij}] \in \mathbb{R}^{N\times N}$.

Denote $D = diag\,\{d_i\} \in \mathbb{R}^{N\times N}$ as the diagonal in-degree matrix of digraph $\mathcal{G}$, where $d_i = \sum\limits_{j\in\mathcal{N}_i} a_{ij}$ is the sum of weights from neighbors of node i. $L = D - A_G$ is defined as the Laplacian matrix of digraph $\mathcal{G}$. Note that the row sums of L are all zero because D and A_G have equal row sums.

2) Primary and secondary controllers: Inverter-based DG units can be controlled by a droop controller [1]. By synchronizing with its neighboring nodes and reading from local meters, the droop controller of each DG unit obtains reference values and voltage magnitude measurements. Denote V^* as the reference value obtained from the secondary controller and V_j as the voltage magnitude of the j-th DG unit. Note each inverter-based DG unit can be modeled as a single integrator. Applying the quadratic droop control developed in [11] to calculate the voltage control output, we have the following primary control law describing the voltage dynamics of each DG:

$$\tau_i \dot{V}_i(t) = -\kappa_i V_i^c(t)(V_i^c(t) - V_i^{c*}(t)) - Q_i^c(t), \tag{1}$$

where $\tau_i > 0$ is the time-constant of the DG inverter, $\kappa_i > 0$ is the droop control gain, $V_i^c(t)$ is the received voltage measurement, $Q_i^c(t)$ is the reactive power injection, and $V_i^{c*}(t)$ is the received reference value obtained from the secondary controller.

The aim of the secondary control is the generation and synchronization of the reference signal V^*. The microgrid system depicted in Fig. 1 can be viewed as a multi-agent system, so we develop the secondary control by applying the distributed cooperative control of multi-agent systems [12].

In the communication digraph $\mathcal{G}$, only the *leader* node gets access to the reference instruction V^* from the secondary controller by a weighted *pinning gain* g_i. The pinning matrix $G = diag\,\{g_i\} \in \mathbb{R}^{N\times N}$ denotes all the pinning gains of the digraph.

A *tracking problem* from the cooperative control theory is applied to synchronize the reference values of DGs. The leader node determines the consensus of all DG units synchronizing to. Denote $V_{ref} = [V_1^* \ldots V_N^*]^\top$ as the reference matrix where V_i^* represents the reference setpoint for DG unit i. At the steady state, the DGs reference synchronization holds as follows:

$$\dot{V}_{ref} = -(L + G)(V_{ref} - \mathbf{1} \cdot V^*) \tag{2}$$

Since $L + G$ have positive real eigenvalues, the desired tracking performance of equation (2) can be guaranteed when the system dynamics get steady. The reference setpoint of DG unit i can be written as:

$$\dot{V}_i^* = \sum_{j\in\mathcal{N}_i} a_{ij}(V_j^* - V_i^*) + g_i(V^* - V_i^*) \tag{3}$$

B. Threat model

Without loss of generality, the attacker is assumed to target the connection between DG unit i and unit $i+1$. Since the reference V^*_{i+1} and the measurement of the previous node V^s_i have very close values and similar dynamics, the attacker could maliciously replace V^*_{i+1} with V^s_i. This attack is *naturally stealthy*, and it can cause a severe impact on the voltage control stability as described in our previous work [10].

Measurement as reference (MaR) attack. By intercepting the network connection between DG unit i and unit $i+1$, an attacker compromises the exchanged data, and maliciously substitutes the reference signal $V^*_{i+1}(t)$ for the voltage measurement $V^s_i(t)$, written as

$$V^*_{i+1}(t) = V^s_i(t), \tag{4}$$

Additionally, the voltage dynamics of DG unit $i+1$ under MaR attack can be written as

$$\tau_i \dot{V}_{i+1}(t) = -\kappa_{i+1} V^c_{i+1}(t)(V^c_{i+1}(t) - V^s_i(t)) - Q^c_{i+1}(t) \tag{5}$$

Multiple techniques could be used to realize this attack by compromising the DG units or their communication links. In this work we consider that the attacker employs a Man-in-the-Middle (MitM) technique to compromise the communication link between two DGs and replace the reference value with the voltage measurement in a transparent way by acting as a proxy which make the attack more difficult to detect. To facilitate the analysis of the MaR attack, the *Jacobian linearization* of (1) is computed at stable state $(\bar{V}, \bar{V}^{c*})$ where $\dot{V}_i(t) = 0$. $x(t) = V(t) - \bar{V}$ and $u(t) = V^{c*}(t) - \bar{V}^{c*}$ respectively represent the voltage and reference deviations.

The analogous linearized system under MaR attack with a MitM technique is described by

$$\begin{aligned} \dot{x}(t-\Delta t) &= Ax(t-\Delta t) + \tau^{-1}_{i+1}\kappa_{i+1}e_{i+1}u(t-\Delta t) \\ y_j(t-\Delta t) &= e_j^\top x(t-\Delta t) \\ u(t-\Delta t) &= e_i^\top x(t-\Delta t), \end{aligned} \tag{6}$$

where $A = -[\tau]^{-1}([\kappa] + W)$, $[\tau] = diag\left\{[\tau_1 \dots \tau_N]^\top\right\}$, $[\kappa] = diag\left\{[\kappa_1 \dots \kappa_N]^\top\right\}$ and W is computed from $Q^c_i(t)$ (as detailed in [4]), $e_i \in \mathbb{R}^N$ is the i-th column of the N-dimensional identity matrix, and Δt is the time delay introduced due to the MitM attack. We can quantify the voltage fluctuation caused by MaR attack by computing the deviation of $y_j(t)$.

III. Datasets collection methodology

In this section, we detail our hardware platform [13] to emulate a microgrid system with the distributed and cooperative control system depicted in Section II. In the built hardware platform, we further implement a MitM attack to fulfill the MaR attack described in Section II. We also collect real datasets from the platform by running the attack and study its detection algorithms.

A. Methodology

The main objective of our experimental platform is to create a network connecting all DG units to transmit a voltage instruction provided by a secondary controller. The DGs are organized in a tree topology where the root DG gets the instructions from the secondary controller and forwards them to its connected DGs alongside its own voltage measurements. This communication process acts between the nodes until the transmitted information reaches the last node of the tree. An intruder connects to the same network and runs a MitM attack between two nodes. It aims to stealthily modify the packets which go from the sender to the receiver to replace the reference value with the measured voltage value.

As shown in Fig. 2, each DG is composed of one Raspberry Pi and one Arduino board acting as the primary controller, two motors A and B acting as a voltage generator, and one lightbulb acting as an electrical load. The Raspberry Pi devices are connected through a WiFi network and play the role of the microgrid's primary controllers. Each of them sends instructions to the Arduino board through a USB port to control the voltage generator by using a PI control algorithm. The DC motor B is connected to DC motor A and generates voltage to turn on the light bulb. The generated voltage will power the light bulb, and its fluctuation affects the level of brightness. We consider that the communications between DGs are not encrypted and using the TCP protocol.

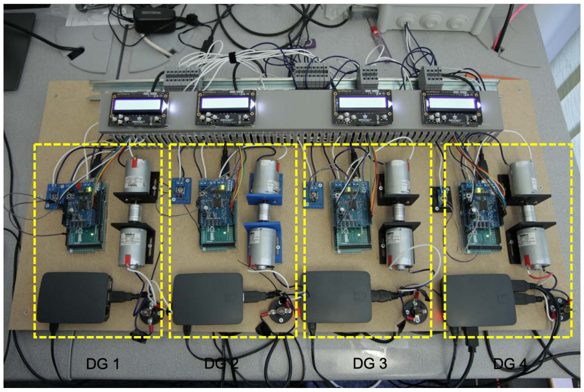


Fig. 2: The microgrid platform consisting of 4 DG units, where each DG unit is composed of one Raspberry Pi, one Arduino board, two motors and one light bulb to demonstrate the effect of voltage instability on the level of brightness [13].

In order to give instructions from one DG to another, we implement a server and a client on each of them. The server obtains the instructions whereas the client gives them to the next node in the tree. The client firstly connects to the server on a specific port. It gets the current voltage measurement from the local Arduino board through the serial port. Then a new packet is created and sent to the server of the next node.

B. Attack implementation

For the system described in Fig. 1 and its experimental setup depicted in Fig. 2, we implement our MaR attack that affects the voltage stability and reference signal synchronization as detailed in our previous study [10].

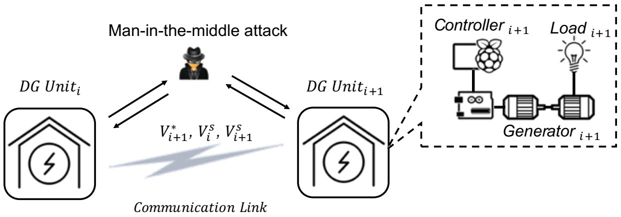


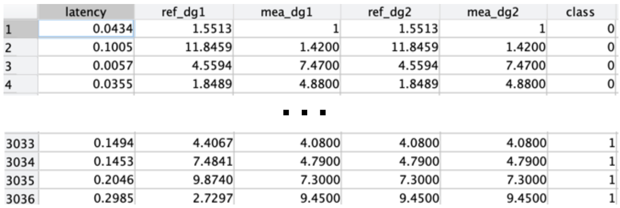
Fig. 3: MaR attack implementation using the MitM technique.

In this scenario, we demonstrate the MaR attack when the attacker uses a MitM technique to introduce itself between two communicating DGs neighbors. As shown in Fig. 3, the attacker introduces itself into a conversation between two DG units and acts as relay/proxy, impersonates both nodes, and gets access to the communication packets that the two nodes are sending to each other. To inject a malicious instruction into the network between two DGs, the attacker intercepts a data packet, modifies it by replacing the reference value with the voltage measurement and sends it to its destination.

C. Dataset description

We run the platform under two different scenarios: normal operations without attack and under the MaR attack using the MitM technique. For the two scenarios, we collect the exchanged packets in the link between the DG_1 to DG_2 to study the attack detection methods.

The linearized MaR attack model in Eq. (6) describes that the attack can not only manipulate the reference and voltage measurement values but also introduce a certain amount of time delay, which helps us to select the features for statistical machine learning. We use *tcpdump* tool to capture the packets transmitted on the network and dissect them to extract five features according to the model in Eq. (6): the packet latency, the respective reference and measurement values of DG_1 to DG_2. We compute the latency as the time difference between sending and receiving timestamps of the same packet from DG_1 to DG_2. We label each sample as normal (label 0) when running the platform without attacks and abnormal (label 1) when the platform is running the MaR attack. A sample of the collected dataset is shown in Fig. 4.



	latency	ref_dg1	mea_dg1	ref_dg2	mea_dg2	class
1	0.0434	1.5513	1	1.5513	1	0
2	0.1005	11.8459	1.4200	11.8459	1.4200	0
3	0.0057	4.5594	7.4700	4.5594	7.4700	0
4	0.0355	1.8489	4.8800	1.8489	4.8800	0
			...			
3033	0.1494	4.4067	4.0800	4.0800	4.0800	1
3034	0.1453	7.4841	4.7900	4.7900	4.7900	1
3035	0.2046	9.8740	7.3000	7.3000	7.3000	1
3036	0.2985	2.7297	9.4500	9.4500	9.4500	1

Fig. 4: A sample of captured data with 5 features and a class label: packet $latency$, reference values ref_dg1 and ref_dg2, measurement values mea_dg1 and mea_dg2, and $class = 0$ for normal operations without attack and $class = 1$ for operations under attack.

We collected 3036 samples in total, among which, 1991 samples are labeled as ”normal,” and 1045 samples are labeled as ”attack”.

IV. Attack Detection

In this section, we study the detection problem of the MaR attack modeled in Section II. We use datasets obtained from the physical platform described in Section III to compare traditional and supervised machine learning algorithms for the MaR attack detection.

A. Performance metrics

In microgrid systems, we require the attack detection algorithms to be capable to predict the label of a sample as normal or abnormal with good performance to avoid false alarms. Therefore, in our study and comparison of the different algorithms, we measure four metrics named the true positives (tp), the false positives (fp), the true negatives (tn), and the false negatives (fn).

In addition, learning capabilities of the algorithms are measured by the classical metrics which are: Precision $Prec = tp/(tp+fp)$, Recall $Rec = tp/(tp+fn)$, and Accuracy $Acc = (tp+tn)/(tp+tn+fp+fn)$.

The precision metric describes the performance in predicting positive samples, while the recall metric represents the capability to identify all the positive samples. Finally, the accuracy metric measures the total classification performance. For example, if $Prec = 1$, no normal samples are misclassified as abnormal, while if $Rec = 1$, no abnormal samples are misclassified as normal. If $Acc = 1$, each sample classified as abnormal is truly an attacked sample, and each sample classified as normal is truly a secure measurement. We also employ receiver operating characteristic (ROC) curves to visually display the classification performance of each studied algorithm by plotting the true positive rate (TPR) against the false positive rate (FPR), where $TPR = TP/(TP+FN)$, and $FPR = FP/(FP+TN)$. For each ROC curve, we identify its area under the curve (AUC) where the best classifiers have the largest values.

B. Detection using threshold comparison

Threshold comparison based methods applied to SVE cannot be deployed directly in the microgrid system described in Section II, because there is no globally centralized estimation system in the distribution network of power systems. However, we can still utilize the same threshold comparison idea to design a straightforward detection algorithm of our deliberate attack. Since the voltage measurement value V_i and reference value V_i^* at each DG follow the equation 1, we compute the residue $r = |V_i - V_i^*|$ and compare r with a certain threshold δ, where $\delta > 0$. If $r \geq \delta$, we consider that the system is under attack.

As shown in Fig. 5, we observe that when the threshold value δ is low, the attack detection achieves high recall and low accuracy. When we increase δ, the recall decreases rapidly and becomes close to 0 when $\delta = 1.2V$, and the accuracy

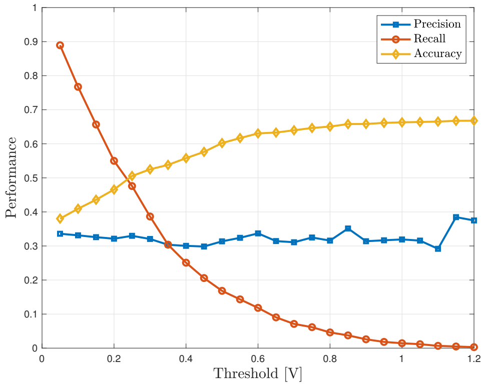


Fig. 5: Performance metrics of the threshold comparison method.

gets higher gradually. Overall, the precision of this detection algorithm is very low (lower than 0.4). Fig. 6 depicts the ROC curve of this detection method where the true positive rate and the false positive rate are computed under different thresholds. We observe that the performance of a classifier based on this method is quite bad and it is not better than random.

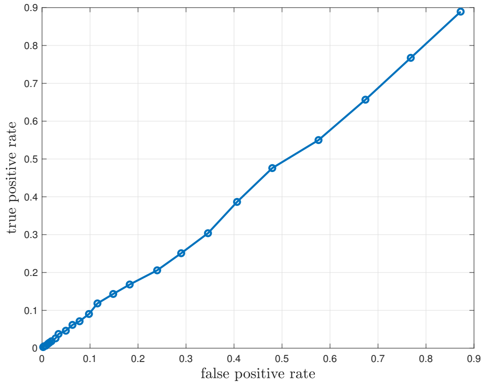


Fig. 6: ROC curve of a classifier based on threshold comparison applied to detect the MaR attack.

C. Detection using supervised machine learning algorithms

The attack detection problem can be considered as a typical supervised machine learning classification problem that returns the corresponding labels of communication packets exchanged between two DGs. According to the system described in equations (6) and (3), we choose five features to be used by the ML algorithms as input: the delay $latency$, the two voltage measurement values mea_dg1 and mea_dg2at DG_i and DG_{i+1} and the two reference values ref_dg1 and ref_dg2 at DG_i and DG_{i+1}, as shown in Fig. 4.

In order to comprehensively study how statistic learning algorithms perform in our attack detection problem, we should cover most of the popular individual supervised learning methods. Moreover, in statistical learning theory, it is usually difficult to find one rule or one feature set which outperforms other rules or features of individual classification methods, so we also need to study ensemble learning, which combines a collection of classifiers to solve this problem. Therefore, we adopt seven different machine learning algorithms [14] grouped into five categories:

- Bayesian Algorithms: Naive Bayes classifier assumes that all the feature values are independent with each other in terms of the class variable.
- Decision Tree Algorithms: Decision Tree is a flowchart-like structure where each internal node denotes a test on a data attribute, each branch denotes the outcome of the test, and each leaf node denotes a class prediction.
- Instance-based Algorithms: k-Nearest Neighbor (KNN) finds k samples in training data that are closest to the test sample and assigns the most frequent label among these samples to the new sample. Support-Vector Machine (SVM) constructs a hyperplane or set of hyperplanes in a high-dimensional space for classification.
- Artificial Neural Network Algorithms: Multilayer Perceptrons (MLP) optimizes the weights for the activation function of neurons organized in a network architecture.
- Ensemble Algorithms: AdaBoost combines ”weak classifiers” into a single ”strong classifier”. Random Forest is a meta estimator using a number of decision tree classifiers on various sub-samples of the dataset and averaging them to improve its predictive accuracy.

Table I provides the optimal parameters that we identified for the classification of our datasets using the scikit-learn implementation of these machine learning algorithms.

TABLE I: Parameters of the adopted machine learning algorithms.

Detection algorithm	Parameters
Ada Boost	DecisionTree estimator with max_depth=20, number of estimators=30, learning rate=0.1
Decision Tree	max_depth=20
KNN	number of neighbors=10, metric=’cosine’
Multi Layer Perceptron	number of neurons=100, L2 penalty=1, max_iter=1000
Naive Bayes	default parameters
Random Forest	number of trees=30, max_depth=20, max_features=3
SVC Linear	kernel=’linear’, regularization parameter=1.0
SVC polynomial	kernel=’poly’, degree=2, gamma=’auto’, regularization parameter=1.0
SVC RBF	kernel=’rbf’, gamma=’auto’, regularization parameter=1.0

The ROC curves of the selected machine learning algorithm are shown in Fig. 7. We observe that Random Forest has an $AUC = 0.99$, which is more significant than any other algorithms. It means that it has a 99% chance that the model will be capable of distinguishing between positive class and negative class.

We see more details of the performance of the different machine learning algorithms compared to the threshold method in Table II. We find that Naive Bayes has a relatively worse performance in terms of three metrics, which shows the features we choose are not independent with each other.

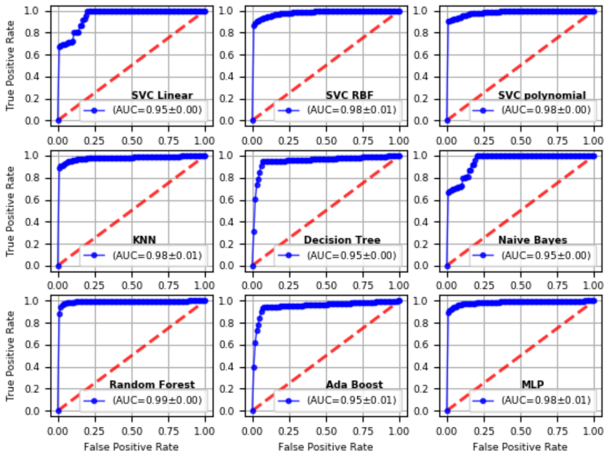


Fig. 7: ROC curves of the different machine learning algorithms.

SVM has the best performance in terms of precision; however, Random Forest has the best performance in terms of recall and accuracy. As a result, we find that a detection method based on Random Forest has the best performance.

TABLE II: Performance of the different attack detection algorithms.

Detection algorithm	Precision	Recall	Accuracy
Threshold Comparison ($\delta = 0.3$)	0.39	0.32	0.53
Ada Boost	0.93	0.94	0.96
Decision Tree	0.93	0.94	0.95
KNN	0.99	0.88	0.96
Multi Layer Perceptron	0.97	0.87	0.95
Naive Bayes	0.79	0.72	0.83
Random Forest	0.96	0.95	0.97
SVC Linear	0.74	0.85	0.8
SVC polynomial	1.00	0.85	0.95
SVC RBF	1.00	0.82	0.94

V. Conclusion and Future Work

In this paper, we study the attack detection of the measurement as reference attack targeting communication links of microgrid systems with a distributed cooperative control structure. We collect datasets from an experimental microgrid platform running this attack. We study both the classic threshold comparison detection method and various supervised machine learning algorithms to compare their performance. We find that Random Forest based method has the best performance to detect such attacks. As future work, we will design and implement faults in our experimental platform to verify the performance of machine learning and even deep learning algorithms to identify attacks, normal and faulty behaviors.

Acknowledgment

This work is supported by fundings from the French Government grant FUI 23 PACLIDO (Protocoles et Algorithmes Cryptographiques Légers pour l'Internet Des Objets).

References

[1] A. Teixeira, K. Paridari, H. Sandberg, and K. Johansson, "Voltage control for interconnected microgrids under adversarial actions," in *IEEE ETFA conference*, 2015.

[2] A. Giacomoni, S. M. Amin, and B. Wollenberg, "A control and communications architecture for a secure and reconfigurable power distribution system: An analysis and case study," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1678–1684, 2011.

[3] D. Kundur, X. Feng, S. Mashayekh, S. Liu, T. Zourntos, and K. Butler-Purry, "Towards modelling the impact of cyber attacks on a smart grid," *International Journal of Security and Networks*, vol. 6, no. 1, pp. 2–13, 2011.

[4] M. Ma, A. Teixeira, J. van den Berg, and P. Palensky, "Voltage control in distributed generation under measurement falsification attacks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8379–8384, 2017.

[5] M. Ozay, I. Esnaola, F. Vural, S. Kulkarni, and H. Poor, "Sparse attack construction and state estimation in the smart grid: Centralized and distributed models," *IEEE JSAC*, vol. 31, no. 7, pp. 1306–1318, 2013.

[6] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[7] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM TISSEC*, vol. 14, no. 1, p. 13, 2011.

[8] M. Ozay, I. Esnaola, F. Vural, S. Kulkarni, and H. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 8, pp. 1773–1786, 2015.

[9] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE symposium on security and privacy*, pp. 305–316.

[10] M. Ma and A. Lahmadi, "On the impact of synchronization attacks on distributed and cooperative control in microgrid systems," in *2018 IEEE SmartGridComm.*

[11] J. Simpson-Porco, F. Dörfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.

[12] A. Bidram, F. Lewis, and A. Davoudi, "Distributed control systems for small-scale power networks: Using multiagent cooperative control theory," *IEEE Control Systems*, vol. 34, no. 6, pp. 56–77, 2014.

[13] M. Ma, A. Lahmadi, and I. Chrisment, "Demonstration of synchronization attacks on distributed and cooperative control in microgrids," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM).*

[14] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.