IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 4, 2019, accepted January 17, 2020, date of publication January 23, 2020, date of current version January 31, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2968934*

# Extremely Randomized Trees-Based Scheme for Stealthy Cyber-Attack Detection in Smart Grid Networks

**MARIO R. CAMANA[1], SAEED AHMED[1,2], CARLA E. GARCIA[1], AND INSOO KOO[1]**

[1]School of Electrical and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea
[2]Department of Electrical Engineering, Mirpur University of Science and Technology (MUST), Mirpur 10250, Pakistan

Corresponding author: Insoo Koo (iskoo@ulsan.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) through the Korean Government through the Ministry of Science and ICT (MSIT) under Grant NRF-2018R1A2B6001714.

**ABSTRACT** Smart grids have become susceptible to cyber-attacks, being one of the most diversified cyber–physical systems. Measurements collected by the supervisory control and data acquisition system can be compromised by a smart hacker, who can cheat a bad-data detector during state estimation by injecting biased values into the sensor-collected measurements. This may result in false control decisions, compromising the security of the smart grid, and leading to financial losses, power network disruptions, or a combination of both. To overcome these problems, we propose a novel approach to cyber-attacks detection, based on an extremely randomized trees algorithm and kernel principal component analysis for dimensionality reduction. A performance evaluation of the proposed scheme is done by using the standard IEEE 57-bus and 118-bus systems. Numerical results show that the proposed scheme outperforms state-of-art approaches while improving the accuracy in detection of stealth cyber-attacks in smart-grid measurements.

**INDEX TERMS** Machine learning, KPCA, extra-trees, cyber-attacks, cyber-security.

## I. INTRODUCTION

The notion of smart grids (SGs) is realized by modern computing and bi-directional communications systems being combined with the typical electrical power grid. Due to increased dependence on communications technologies, the SG's susceptibility to cyber-attacks has escalated. Conventionally, the measurement data are collected from the electric power grid by a supervisory control and data acquisition (SCADA) system, which is composed of remote terminal units (RTUs) and communications networks. The RTUs consist of sensors to collect the data, and actuators to execute the control commands initiated by the energy management system (EMS) in the power control center (PCC). The suitability and fitness of the collected data are exceedingly substantial in order to initiate precise and accurate control decisions. Hence, the legacy systems employ a device called a bad-data detector (BDD) to analyze the reliability of the sensor-collected measurement data. However, the newly discovered stealthy cyber-attack (SCA) [1] is considered capable of dodging the legacy BDD. A smart hacker can intelligently craft the attack vector to inject biased values into the sensor-collected measurement data [1]. Such a malicious attack cannot be identified by the legacy BDD, and can result in fiscal loss, a fractional interruption in power system operations, or a combination of financial loss and disruption [2], [3]. Due to the detrimental effects of such malicious activities on the secure and reliable operations of SGs, there exists the necessity to study attack-detection measures.

The meter measurements collected by the PCC from every subsystem of the smart grid are the bus voltages, branch reactive power flows, and bus real and reactive power injections [1]. Then, the state variables are estimated at the PCC via the state estimation process based on the meter measurement. In this paper, we refer to state estimation-measurement features (SE-MF) as the meter measurements collected from the components of the SG, which are used to estimate the state variables.

Many sensors are utilized to collect parameters from the electric power transmission network due to its lengthened geographic span. Machine learning (ML)-based approaches are becoming popular among researchers, because these

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Agostino Ardagna.

techniques exploit the sensor measurements directly to detect an SCA without needing detailed mathematical models of the power grid. Moreover, ML-based detection utilizes historical measurements to perceive the data patterns of the normal (unattacked) system, and no prior information is needed about the subsystems of the power network.

In the literature, several approaches that do not use ML algorithms have been introduced for SCA-attack detection [4], [5]. However, the use of ML-based schemes is becoming popular among researchers owing to their efficacy in classification problems involving complex data distributions. Fadlullah *et al.* [6] proposed a Gaussian process regression scheme to forecast malicious attacks, and it is based on probabilistic distributions to predict abnormal operational behavior. Zhang *et al.* [7] presented a distributed system based on the support vector machine (SVM) to detect cyber-attacks, in which the authors considered the deployment of intelligent modules at different levels of the SG collaborating amongst themselves. Hink *et al.* [8] investigated ML-based approaches to differentiate between normal operations, cyber-attack events, and natural disruptions in SGs, where the authors compared ML algorithms such as random forest, naïve Bayes, the SVM, and AdaBoost. Hao *et al.* [9] studied random and targeted attacks in SGs, where they proposed robust principal component analysis (PCA) for the detection of false data–injection attacks. Ozay *et al.* [10] provided a comparison study of several ML algorithms, such as the SVM, AdaBoost, and perceptron, for attack detection in SGs, in which the ML-based approaches achieved a higher performance, compared with state vector estimation methods. The aforementioned approaches did not use dimension reduction (DR) methods or feature selection (FS) techniques to tackle the dimensionality issue, which becomes more important as the power system size increases, since the computational complexity is higher as the number of features increases.

Esmalifalak *et al.* [11] proposed reducing the dimensionality of data by using a PCA-based technique, and then, an SVM is trained over labeled data to detect stealthy false-data injection. Ahmed *et al.* [12] presented a genetic algorithm (GA) as an FS scheme to deal with the dimensionality issue, and an SVM-based algorithm, using a Gaussian kernel, to detect covert cyber-deception attacks in SGs. A study for unlabeled historical SE-MF data was presented in [13], where the authors tackled the dimensionality issue with a PCA-based technique and introduced the unsupervised isolation forest (iForest) algorithm to detect covert data-integrity assaults in SGs. However, PCA is only suitable for linear dimensionality reduction, and complex data structures cannot be well projected in a linear subspace [14]. Furthermore, FS methods are based on removing insignificant features and selecting only the most important ones based on the accuracy of the classifier or the value of a problem-dependent function. But in SGs, a feature of the SE-MF dataset represents information about the buses in the power system. Hence, using historical data, the feature selection procedure may remove non-attacked features from the SE-MF dataset while considering them insignificant. Nonetheless, during the deployment of the model, a malicious user can attack a different RTU that was not affected in the historical data, making it difficult to detect the cyber-attack, since the related feature could have been removed for the FS algorithm.

Unlike previous studies, in this paper, we tackle the dimensionality issue with a Kernel PCA (KPCA)-based technique, which is a generalization of PCA for nonlinear dimensionality reduction. KPCA is a DR method that does not remove any features, and instead, projects the features into a new lower-dimensional space, where a classifier algorithm can easily recognize and separate normal and attacked samples. Thus, the computational cost at the PCC is reduced, since we decrease the number of features, where a fast and efficient classification algorithm is needed to keep the computational cost of the entire process as low as possible. Based on this objective, we propose an extremely randomized trees (Extra-Trees)-based approach to detect SCA attacks in SGs. The Extra-Trees algorithm is an ensemble method characterized by being computationally efficient and providing high accuracy [15], where the strength of the randomization helps to achieve a greater reduction in the variance, compared with other ensemble methods like random forest or AdaBoost. Furthermore, to study a more realistic dataset, this paper is the first one to investigate the impact of noisy labels in an SE-MF dataset in SGs. Thus, the main contributions of this paper are summarized as follows.

- We study the SCA attack on SE-MF datasets, and how a BDD is not able to detect such attacks in conventional power systems.
- We propose KPCA as a nonlinear dimensionality reduction method to handle the increasing computational complexity in big power systems. KPCA represents the dataset in a lower-dimensional space while preserving most of the original information. Furthermore, the Extra-Trees algorithm is used to detect the presence of SCA attacks in the SE-MF dataset. Thus, the projected features of KPCA are used as input to the Extra-Trees algorithm to classify normal and attacked samples.
- We investigate a realistic scenario, where a percentage of the labels in the SE-MF dataset are corrupted. In particular, we study the classification problem with noisy labels, where a percentage of the true labels in the training dataset are independently flipped.
- The performance of the proposed scheme is compared with several approaches in the literature, such as SVM and iForest, by using standard IEEE 57-bus and 118-bus test systems. The numerical results show that the proposed scheme achieves the highest accuracy and the lowest computational time among the compared ML-based approaches. In addition, we compare the efficiency of the proposed KPCA technique for DR with state-of-art methods such as PCA, fast independent component analysis (ICA), neighborhood components analysis (NCA), binary particle swarm optimization (PSO), and the GA.

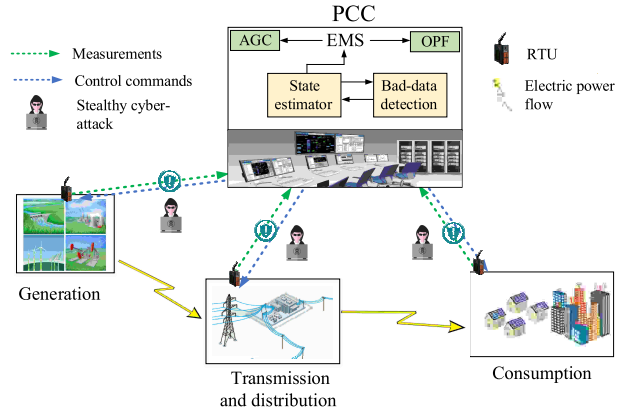



**FIGURE 1. Stealthy cyber attack in a smart grid communications network.**

The paper is structured as follows. State estimation, traditional bad-data detection, and the nature of an SCA attack in SG networks are described in Section II. In Section III, we present the proposed scheme for SCA detection based on KPCA and Extra-Trees. The simulation results are provided in Section IV. Finally, conclusions are made in Section V.

## II. SYSTEM MODEL

### A. ELECTRIC POWER NETWORK

Several electric generators are connected to a large number of consumers across a wide topographical expanse in the power transmission system. Several paths and lines, generally deployed in a redundant manner, aim to guarantee the supply of power to any consumer from the generating source, taking into account the expenses and frugality of the transmission path. Figure 1 illustrates the communications network used to interconnect the devices and the power system to the power control center, where the objective is to provide efficient monitoring and control of the power system.

### B. STATE ESTIMATION

Sensors and actuators are installed in different entities of the electric power grid. The measurements are gathered via SCADA systems in the PCC, where the power system states (composed of bus voltage angles and magnitudes) are estimated by utilizing the sensor-collected data. The estimation of the state variables, $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_n]^T$, are carried out considering the meter measurements, $\mathbf{z} = [z_1, z_2, \ldots, z_m]^T$, of the power system, where $n$ and $m$ are positive integers, and $\gamma_i, z_j \in \mathbb{R}$ with $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The process to obtain the state variables at the PCC is called the state estimation. The state estimation process is based on the meter measurement and uses a power flow model, which is composed of a set of equations to represent the energy flow on each transmission line of the SG. The alternating current (AC) power flow model is composed of nonlinear equations considering real and reactive power. The state variables are related to the measurements in the AC power flow model as follows:

$$\mathbf{z} = h(\boldsymbol{\gamma}) + \mathbf{e}, \tag{1}$$

where $h(\boldsymbol{\gamma})$ represents the non-linear relation between state $\boldsymbol{\gamma}$ and measurement $\mathbf{z}$, and $\mathbf{e} = [e_1, e_2, \ldots, e_m]^T$ is the Gaussian measurement noise vector where elements have standard deviation $\sigma$. However, the AC power flow model can be computationally expensive and there is no guarantee to converge to a solution [1]. Therefore, it is a common approach in the literature [10], [11], [13], [16] to approximate the AC power flow model using a linearized power flow model called the direct current (DC) power flow model. Then, the non-linear model in (1) can be reformulated as an (DC) power flow model as follows:

$$\mathbf{z} = \mathbf{H}\boldsymbol{\gamma} + \mathbf{e}, \tag{2}$$

where $\mathbf{H}$ refers to the Jacobian matrix, which can be approximated as follows [17], [18]:

$$\mathbf{H} = \left.\frac{\partial h(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}\right|_{\gamma=0}. \tag{3}$$

$\mathbf{H}$ consists of impedance and topological measurements. To calculate the estimation of $\boldsymbol{\gamma}$, which is represented by $\hat{\boldsymbol{\gamma}}$ and defines the best fit for the meter measurements, the statistical criteria weighted least squares (WLS) [19] is generally employed. Hence, by using the WLS criterion and assuming that the sensor error follows a normal distribution with zero mean, the estimated state is given as follows:

$$\hat{\boldsymbol{\gamma}} = (\mathbf{H}^T \boldsymbol{\Omega} \boldsymbol{H})^{-1} \mathbf{H}^T \boldsymbol{\Omega} \boldsymbol{z} = \mathbf{G}\mathbf{z}, \tag{4}$$

where the matrix $\mathbf{G} = (\mathbf{H}^T \boldsymbol{\Omega} \boldsymbol{H})^{-1} \mathbf{H}^T \boldsymbol{\Omega}$ with a diagonal matrix $\boldsymbol{\Omega}$, whose elements are given by

$$\boldsymbol{\Omega} = \begin{bmatrix} \sigma_1^{-2} & & \\ & \ddots & \\ & & \sigma_m^{-2} \end{bmatrix} \tag{5}$$

### C. CONVENTIONAL BAD-DATA DETECTION

The measurements collected by the sensors can be compromised and damaged due to several potential reasons, such as medium noise in wireless communications, erroneous meters, and malicious user behavior. Traditional power systems use a residual-based detector for a BDD to detect any corruption in the measurements of the sensors [11]. Let $\mathbf{r}$ be the residual, calculated as the difference between observed sensor measurements $\mathbf{z}$ and estimated measurements $\hat{\mathbf{z}}$ at the PCC, and it is defined as follows:

$$\mathbf{r} = \mathbf{z} - \hat{\mathbf{z}} = \mathbf{z} - \mathbf{H}\hat{\boldsymbol{\gamma}}. \tag{6}$$

The BDD in the PCC detects the presence of corrupted measurements by evaluating the $L_2 - norm \left\| \mathbf{z} - \mathbf{H}\hat{\boldsymbol{\gamma}} \right\|$ and

comparing it with a predefined threshold, $\tau$ [1]. Then, the current sensor measurement is considered not attacked if the following condition is satisfied:

$$\max_i |r_i| < \tau, \tag{7}$$

where $r_i$ is the $i-th$ component of the residual vector $\mathbf{r}$, and $\tau$ is a predefined threshold. Otherwise, the measurement is considered to be corrupted, and an alarm is executed.

### D. STEALTHY CYBER-ATTACK: BASIC PRINCIPLE

Equipped with knowledge of the topology of the power system, a hacker might insert corrupted data into meter measurements $\mathbf{z}$ by designing an assault vector, $\mathbf{a} = [a_1, a_2, \ldots, a_m]^T$, to dodge the BDD [20]. Let $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$, be the observed measurement that may contain corrupted data due to an SCA. The attack-space is large for the attacker, who selects any of the non-zero arbitrary elements in attack vector $\mathbf{a}$. Thus, an attack on the $i-th$ measurement, $z_i$, is created by altering it with a fake measurement, $z_i + a_i$, which means that element $a_i$ of the attack vector has a non-zero value.

The BDD calculates the $L_2 - norm$ of residual vector $\mathbf{r}$ to identify the existence of corrupted or erroneous measurements. Nevertheless, if the assailant crafts attack vector $\mathbf{a}$ by using $\mathbf{a} = \mathbf{H}\boldsymbol{\alpha}$, where the vector $\boldsymbol{\alpha}$ has length $n$ with non-zero elements, the observed measurement vector, $\mathbf{z}_a$ (holding the attack vector) can evade conventional detection. Note that the vector $\boldsymbol{\alpha}$ has the same dimension as the state variables $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \ldots, \gamma_n]^T$.

To explain the reason, let $\hat{\boldsymbol{\gamma}}_a$ represent the estimate of state variables when we have attacked meter measurements $\mathbf{z}_a$, i.e.,

$$\hat{\boldsymbol{\gamma}}_a = \mathbf{G}\mathbf{z} + \mathbf{G}\mathbf{a} = \hat{\boldsymbol{\gamma}} + \mathbf{G}\mathbf{H}\boldsymbol{\alpha} = \hat{\boldsymbol{\gamma}} + \boldsymbol{\alpha}. \tag{8}$$

Next, we calculate the $L_2 - norm$ of the attacked measurements residual, $\mathbf{r}_a$, as follows:

$$\begin{aligned} \|\mathbf{r}_a\|_2 &= \left\|\mathbf{z}_a - \mathbf{H}\hat{\boldsymbol{\gamma}}_a\right\|_2 \\ &= \left\|(\mathbf{z}+\mathbf{a}) - \mathbf{H}(\hat{\boldsymbol{\gamma}} + \boldsymbol{\alpha})\right\| \\ &= \left\|\left(\mathbf{z} - \mathbf{H}\hat{\boldsymbol{\gamma}}\right) + (\mathbf{a} - \mathbf{H}\boldsymbol{\alpha})\right\|_2 = \left\|\left((\mathbf{z} - \mathbf{H}\hat{\boldsymbol{\gamma}})\right)\right\|_2 \\ &= \|\mathbf{r}\|_2. \end{aligned} \tag{9}$$

Then, we can see that the attacked measurement residual is the same as that without altered data. Therefore, if the original measurement, $\mathbf{z}$, can pass the BDD, then $\mathbf{z}_a$ will be able to bypass the BDD. One requirement for SCA is to know the topology of the power system, which is available at the PCC of the power companies [1]. Although the access to the PCC can be difficult, we consider the case that the hacker can obtain the information of the topology of the power system, which is a common assumption in several other studies of the literature [11], [12], [13], [16].

### E. STEALTHY CYBER-ATTACK MODEL

Generally, SCA attacks are separated into two categories: 1) the load change attack, and 2) the load redistribution attack [20]-[22]. The objective of the hacker is to obtain a specific change in the state variables of the power system. Then, we can create a vector by injecting an attack in a set of state variables. Next, using $\mathbf{a} = \mathbf{H}\boldsymbol{\alpha}$, the hacker is able to corrupt the observed measurements $\mathbf{z}_a = \mathbf{z} + \mathbf{a}$, which will not be detected by the BDD. In this paper, we focus on real-time detection of the SCA in sensor-collected measurements. Consequently, we create a generalized attack, assuming that the assailant has sufficient information about the power network topology.

In the SCA attack, to obtain a specific change in the state variables of the power system, the attacker adds a fake value to the sensor-collected measurements, changing the real power insertion and real power flows. For example, with the target being to modify state variable $\gamma_2$ by injecting an attack of 6%, we can create an attack vector, $\boldsymbol{\alpha}$, using the following equation:

$$\boldsymbol{\alpha} = [0, 0.06\gamma_2, 0, \ldots, 0]. \tag{10}$$

Then, we obtain the attack vector $\mathbf{a} = \mathbf{H}\boldsymbol{\alpha}$, whose non-zero elements represent the $i$-th meter measurements to be attacked. The aforementioned general procedure has been widely used in the literature [11], [12], [13], [20], [16] for false data injection attacks or SCA in SGs.

Employing state vector $\hat{\boldsymbol{\gamma}}_a = \hat{\boldsymbol{\gamma}} + \boldsymbol{\alpha}$ and power flow equations, the measurements corrupted due to the SCA are given as follows:

$$\mathbf{z}_a = \mathbf{H}\boldsymbol{\gamma}_a + \mathbf{e}. \tag{11}$$

## III. PROPOSED SCHEME FOR SCA ASSAULT DETECTION

In this section, a fast and accurate ML-based scheme is proposed to detect SCA attacks. The features used by the proposed approach are the meter measurements of the SG, and each sample of the SE-MF dataset corresponds to the meter measurements at a specific time. For example, $\mathbf{z}_{t_0} = \left[z_{1_{t_0}}, z_{2_{t_0}}, \ldots, z_{m_{t_0}}\right]^T$ is the first sample of the SE-MF dataset and corresponds to the meter measurements collected at the PCC at time $t_0$ and the last sample of the dataset $\mathbf{z}_{t_N} = \left[z_{1_{t_N}}, z_{2_{t_N}}, \ldots, z_{m_{t_N}}\right]^T$ corresponds to the meter measurements collected at the PCC at time $t_N$. The main objective is to detect the presence or absence of an SCA in a sample at the PCC. Then, we can use an ML-based algorithm with two labels corresponding to attack and non-attack, respectively.

In addition, the evaluation method is based on k-fold cross-validation which splits the whole SE-MF dataset into $K$ roughly equal parts and performs $K$ independent evaluation instances of the proposed model. In the first evaluation instance of the k-fold cross-validation, the first part is used as testing data and the rest is used as training data which guarantees unseen data to evaluate the model. In the second instance, the second part serves as testing data while the rest is being used for training, where the process continues until reach $K$ evaluation instances. A flowchart for the model

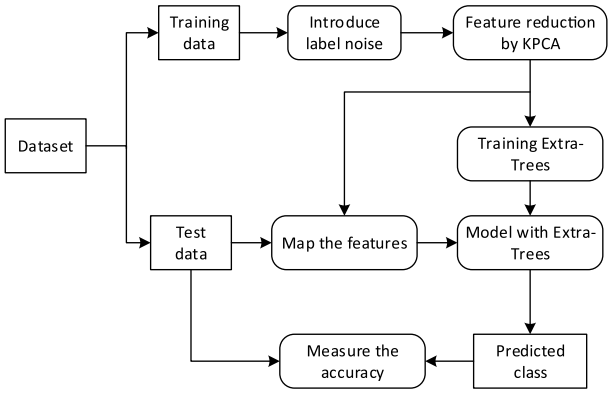



**FIGURE 2.** Evaluation methodology based on k-fold cross-validation.

evaluation of the proposed scheme for the classification of assaulted and normal measures is illustrated in Figure 2, where we show one instance of k-fold cross-validation.

The first step in the proposed scheme is employing KPCA to tackle the dimensionality issue, and then, we utilize the Extra-Trees algorithm as an attack detection approach. In order to study a more realistic scenario, we consider an attack detection problem where the labels of the samples are randomly corrupted. In particular, a percentage of the true labels of the training data are independently flipped, which is called label noise [23]. However, there are testing data with noise-free labels, which are used to evaluate the performance of the proposed approach.

### A. KERNEL PRINCIPAL COMPONENTS

A common approach for dimensionality reduction is principal component analysis [13], [11], [24]. PCA is a linear technique to project the data to a low-dimensional space with little loss of information, where the new features can achieve the largest variance [14]. However, PCA does not consider nonlinearities inherent in data with complicated structures. To cope with this problem, Kernel PCA is presented as a nonlinear generalization of PCA. In general, KPCA is composed of two steps: (1) mapping the training data from the original space into a high-dimensional feature space, and (2) performing traditional PCA in the feature space [25], [26]. To address the computational cost of high-dimensional mapping, KPCA uses kernel methods.

We consider a training dataset, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^D$ with $i = 1, .., N$, which can be projected to an $M$-dimensional feature space, with $M >> D$, through nonlinear transformation $\phi(\mathbf{x})$. Therefore, each point of the training dataset is projected to a point, $\phi(\mathbf{x}_i)$, in the extended feature space.

First, we assume that the features in the $M$-dimensional space have a zero mean, as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i) = 0. \tag{12}$$

Then, we compute the covariance matrix, with dimension $M \times M$ of the projected features, as

$$\mathbf{C} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\,\phi(\mathbf{x}_i)^T. \tag{13}$$

The eigenvalues, $\lambda_k$, and eigenvectors, $\mathbf{v}_k$, of the covariance matrix are computed by solving the eigenvalue problem:

$$\lambda_k \mathbf{v}_k = \mathbf{C}\mathbf{v}_k, \quad k = 1, \ldots, M. \tag{14}$$

Based on (13) and (14), we obtain

$$\lambda_k \mathbf{v}_k = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\,\phi(\mathbf{x}_i)^T\mathbf{v}_k. \tag{15}$$

The eigenvector can be rewritten as a linear combination of the projected training data points, as follows:

$$\mathbf{v}_k = \sum_{i=1}^{N}\alpha_{ki}\phi(\mathbf{x}_i). \tag{16}$$

Substituting (16) into (15), we have

$$\lambda_k \sum_{i=1}^{N}\alpha_{ki}\phi(\mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\,\phi(\mathbf{x}_i)^T\sum_{j=1}^{N}\alpha_{kj}\phi(\mathbf{x}_j). \tag{17}$$

Then, the kernel trick is applied by defining the kernel function, $\kappa(\mathbf{x}, \mathbf{y})$, and the kernel matrix, $\mathbf{K}$, as follows:

$$\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j). \tag{18}$$

Now by substituting (18) into (17), and multiplying both sides by $\phi(\mathbf{x}_k)^T$, we have

$$\lambda_k \sum_{i=1}^{N}\alpha_{ki}\kappa(\mathbf{x}_k, \mathbf{x}_i) = \frac{1}{N}\sum_{i=1}^{N}\kappa(\mathbf{x}_k, \mathbf{x}_i)\sum_{j=1}^{N}\alpha_{kj}\kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{19}$$

In matrix notation, (19) can be represented as

$$\lambda_k N\mathbf{K}\alpha_k = \mathbf{K}^2\boldsymbol{\alpha}_k, \tag{20}$$

where $\boldsymbol{\alpha}_k$ is defined as $\boldsymbol{\alpha}_k = [a_{k1}, a_{k2}, \ldots, a_{kN}]^T$ and can be found by solving the following:

$$\lambda_k N\boldsymbol{\alpha}_k = \mathbf{K}\boldsymbol{\alpha}_k. \tag{21}$$

The centered kernel matrix $\tilde{\mathbf{K}}$ substitutes for kernel matrix $\mathbf{K}$ when the projected training dataset does not have a zero mean. The centered kernel matrix is given by

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{U}\mathbf{K} - \mathbf{K}\mathbf{U} + \mathbf{U}\mathbf{K}\mathbf{U}, \tag{22}$$

where $\mathbf{U}$ is the $N \times N$ matrix with each element equal to $1/N$ [14], [26].

For any point $\mathbf{x}$, the resulting kernel principal components can be represented as the following set of features:

$$y_k(\mathbf{x}) = \sum_{i=1}^{N}\alpha_{ki}\kappa(\mathbf{x}, \mathbf{x}_i), \quad k = 1, \ldots, M \tag{23}$$

Finally, we can limit the number of principal components to $P \leq M$ for dimensionality reduction. Hence, the new features for point $\mathbf{x}$ are $\mathbf{x}_{new} = [y_1, \ldots, y_P]^T$.

In this paper, we use the Gaussian kernel, defined as follows:

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right), \tag{24}$$

where $\sigma$ is a free parameter.

Note that the features in the SE-MF dataset are the meter measurements $\mathbf{z} = [z_1, z_2, \ldots, z_m]^T$ of the power system, which are reduced to $P$ features by using KPCA. Then, the input dataset of the Extra-Trees algorithm is composed of the transformed $P$ features of KPCA. The steps in KPCA for dimensionality reduction are summarized in Table 1.

**TABLE 1. Kernel PCA for dimensionality reduction.**

| | |
|---|---|
| 1: | **inputs:** Training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where the sample $\mathbf{x}_i = \{f_1, f_2, ..., f_D\}$ is a $D$-dimensional vector, the new number of features $P$, and the value of the parameter $\sigma$ in the Gaussian kernel. |
| 2: | Select the Gaussian kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / 2\sigma^2\right)$. |
| 3: | Based on the training dataset $\mathbf{X}$, construct the kernel matrix $\mathbf{K}$ from (18). |
| 4: | Compute the centered kernel matrix $\tilde{\mathbf{K}}$ using (22). |
| 5: | Find vectors $\alpha_k$ of the centered kernel matrix by replacing $\mathbf{K}$ with $\tilde{\mathbf{K}}$ in (21). |
| 6: | **output:** For any data point (training or testing), compute the $P$ principal components $[y_1(\mathbf{x}), \ldots, y_P(\mathbf{x})]^T$ (new set of features) using (23). |

### B. EXTREMELY RANDOMIZED TREES

Tree-based ensemble methods are popular approaches for supervised classification and regression problems [27], [28], [29]. The robustness of ensemble methods relies on the capacity to combine the prediction of several models, which results in better performance compared to what could be obtained from a single model. The best performance of tree-based ensemble methods is achieved when the base learners are independent of one another, which can be achieved by using very different training algorithms for every decision tree, or by randomization [30]. Randomization when growing trees entails greater tree diversity, and helps to reduce the correlation, i.e. making the decision trees more independent. However, an ensemble method can result in a substantial increase in computational cost, since it needs to train several individual classifiers, and its computational requirements can grow exponentially when it deals with a large dataset. Therefore, we focus on the Extra-Trees algorithm [15], which works similar to, but much faster than, random forest [31].

Extra-Trees consist of a large number of individual decision trees, where the whole training dataset is used to grow each decision tree. A decision tree is composed of a root node, child nodes, and leaf nodes, as illustrated in Figure 3. Starting at the root node, the Extra-Trees algorithm essentially chooses a split rule based on a random subset of features and a partially random cut point. This process is repeated in each child node until reaching a leaf node. The Extra-Trees algorithm consists of three fundamental parameters: the number of decision trees in the ensemble ($M$), the number of features to select randomly ($K$), and the minimum number of instances needed to split a node ($n_{\min}$ ).

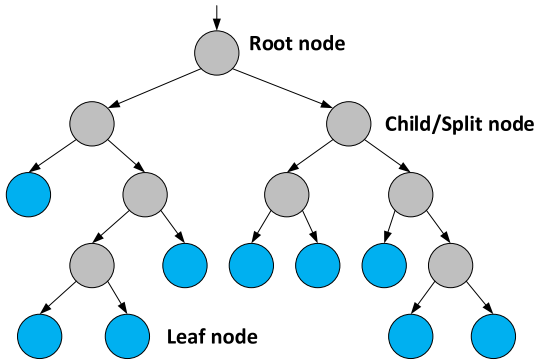



**FIGURE 3. Illustration of a decision tree.**

**TABLE 2. Extra-trees splitting algorithm.**

| | |
|---|---|
| 1: | **inputs:** Training subset $\mathbf{S}_p = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{Q_p}\}$, where the sample $\mathbf{s}_i = \{f_1, f_2, ..., f_D\}$ is a $D$-dimensional vector, the number of attributes to select randomly, $K$, and the minimum number of samples required to split a node, $n_{\min}$. |
| 2: | **If** $Q_p < n_{\min}$ or all observations within the node have an identical label.<br>Stop splitting and define the node as a leaf node. |
| 3: | **Else**<br>Select a random subgroup of $K$ features $\{f_1, f_2, ..., f_K\}$ among the original $D$ features. |
| 4: | **For** each feature $k$ in the subgroup **do**:<br>Find $f_k^{\max}$ and $f_k^{\min}$ as the maximal and minimal values of the feature $k$ in subset $\mathbf{S}_p$.<br>Obtain a random cut-point, $f_k^c$, uniformly in the range $\left[f_k^{\min}, f_k^{\max}\right]$.<br>Set $\left[f_k < f_k^c\right]$ as a candidate split<br>**End for** |
| 5: | Select a split $[f_* < f_*^c]$ such that $Score(f_*^c) = \min_{k=1,...,K} Score(f_k^c)$ |
| 6: | **Output:** best split $[f_* < f_*^c]$ at the child node $p$. |

Formally, given a training dataset, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where the sample $\mathbf{x}_i = \{f_1, f_2, \ldots, f_D\}$ is a $D$-dimensional vector with $f_j$ as the feature and $j \in \{1, 2, \ldots, D\}$, Extra-Trees generates $M$ independent decision trees. In each decision tree, $\mathbf{S}_p$ denotes the subset of training dataset $\mathbf{X}$ at child node $p$. Then, at each node $p$, the Extra-Trees algorithm selects the best split based on $\mathbf{S}_p$ and a random subgroup of features by following the algorithm described in Table 2.

In detail, subset $\mathbf{S}_p$ at child node $p$ is divided into two sets: $\mathbf{S}_p^{right}$ containing those samples satisfying the condition of the split rule, and $\mathbf{S}_p^{left}$ containing the rest of the training samples. In order to select the best split, we use Gini impurity [30] as a score function, i.e. the candidate split with the lowest value for Gini impurity is chosen as the best split rule, which is kept constant while the tree is growing. The process is repeated in each child node until it achieves a minimum number of samples required to split ($n_{\min}$), or when all the samples in

subset $\mathbf{S}_p$ have an identical label. Finally, each leaf node is represented by the label of the samples in subset $\mathbf{S}_p$.

In the testing phase, a test sample passes to each of the decision trees and across each child node, where the best splits are used to forward the test sample to the left or right child node until reaching a leaf node. The class for the test sample for any decision tree is defined by the leaf node where the test sample arrives, and the total prediction of the Extra-Trees algorithm is defined as the majority of votes by the $M$ decision trees. Note that in the proposed scheme, the inputs for Extra-Trees are the transformed features of KPCA, i.e. the number of features defined by $D$ in the Extra-Trees algorithm matches the value of the number of principal components, $P$, selected in the KPCA technique.

The Extra-Trees algorithm is able to reduce the variance and bias more strongly than other randomization schemes, like those used in random forest. The variance is created by the excessive sensitivity of the model to small fluctuations in the training dataset (high variance can cause overfitting), and it is reduced due to the explicit randomization in the selection of the subset of features and the choice of the cut-point. On the other hand, the bias, which can be measured as the ability to correctly generalize unseen data (high bias can cause underfitting), is minimized because the full original training dataset is used to learn each decision tree [15].

When a SG is composed of separated subregions located in different geographical areas, large-scale SG, it is necessary to take into account important factors such as bandwidth, latency and computational capabilities. In [32], the authors study a wide-area SG represented by a set of local control centers connected to a global control center, where each local control center is responsible for a set of meters of the SG. Therefore, a hacker can block or manipulate the smart meters and control centers, and also can attack the communication channels. Furthermore, in a wide-area SG, collecting and processing a huge amount of data in a centralized setup is susceptible to node failure and has limitations such as energy and bandwidth [32]. Then, we present two schemes to apply the proposed approach in a large-scale SG: centralized setup and distributed setup.

In a centralized setup, all the meter measurements are processed at the PCC and the procedure for the attack detection follows the description of Section III. In this case, we consider that the communication network infrastructure of the SG is robust and has enough resources to transmit the meter measurements to the PCC. We refer to the communication network infrastructure presented in [33]. In this architecture, the generation, transmission, and distribution systems are interconnected through substations and transmission lines, while the communication toward the operation center is composed of wide area network (WAN), local area network (LAN) and field area network (FAN). The collected measurements are transmitted to the control center through remote terminal units (RTU) in the SCADA system, while in the distribution system, the advanced metering infrastructure (AMI) provides real-time (RT) communication to millions of smart meters [33]. The WAN can be supported by different communication technologies such as WiMax, Cellular, fiber optic, etc [34], [35]. The requirements of a WAN in SG according to [35] are a data rate higher than 10Mbps and coverage distance up to 100km. In addition, optical communication is commonly used for the transmission between the transmission/distribution substations and the PCC since it provides high data rates and low latency. For instance, using wavelength-division multiplexing (WDM) the maximum theoretical data rate is 40Gbps with coverage up to 100km [35]. Therefore, the PCC is capable to receive all the information from the meter measurements to perform the state estimation and the attack detection procedure.

In addition, the proposed scheme based in Extra-Trees with KPCA does not require high computational capabilities to detect an SCA. In particular, the simulation time to verify if a sample composed of meter measurements was attacked or not, in an SG using an IEEE 118-bus system, is around 3 ms. Furthermore, in a real power grid, the measurements are taken in an interval of a few minutes, e.g. 15-minutes [36], which can provide enough time to overcome possible delays in the communication network. In addition, even though the centralized approach becomes difficult to implement for a wide-area SG where each sub-area communicates to the global center with a wireless communication infrastructure [34], forthcoming 5G technology for the development of smart grids can provide critical and timely services for the centralized approach.

In a distributed setup, the local centers have the task of collecting and processing the meter measurements in their respective subregions, while maintaining a communication link with their neighboring local control centers and the global center. In [32], the authors proposed a system model for a distributed setup in SGs, where we can obtain a set of state variables that depend on the measurements collected at each local center. For instance, we refer to the example in [32]. Given a system composed of two subregions with a state vector $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \gamma_3, \gamma_4]^T$ and a particular measurement matrix $\mathbf{H}$, the local state vector for the first local center is $\boldsymbol{\gamma}^1 = [\gamma_1, \gamma_2, \gamma_3]^T$ and for the second local center is $\boldsymbol{\gamma}^2 = [\gamma_2, \gamma_3, \gamma_4]^T$ at a specific time, where the meter measurements are $\mathbf{z}^1 = [z_1, z_2]^T$ and $\mathbf{z}^2 = [z_3, z_4, z_5]^T$ for the first and second local center, respectively. Note that local centers can share one or more state variables, where meter measurements from neighbor local centers can provide information about the same state variable. Please refer to [32], [34] for a detailed description of a distributed system in SGs.

To describe an extension of the proposed approach to distributed systems, we assume that exist $L$ subregions, where the meter sends their measurements to only one local center $l$ with $l = 1, \ldots L$. The measurement vector collected at the $l$-th local center is denoted as $\mathbf{z}^l$ and the state vector

is defined as $\boldsymbol{\gamma}^l$. Therefore, we can deploy an instance of the proposed KPCA-ExtraTreess algorithm in each $l$-th local center to detect an SCA over one possible state estimate $\boldsymbol{\gamma}^l$. The features could be the meter measurements collected at the local center $\mathbf{z}^l$ and those reported by their neighboring local centers (in a distributed SG system, pre-processed meter measurements are shared between neighboring local centers to perform the state estimation, where the transmission of local statistics and control signals between local centers and the global center is assumed to be instantaneous according to the reference [32]). Then, the results obtained from the algorithm in each local center are transmitted to the global center to make a decision.

## IV. SIMULATION RESULTS

In this section, we numerically evaluate the performance of our proposed approach to detecting SCA attacks in state estimation. We used standard IEEE 118-bus and 57-bus test systems, where the simulation results were averaged over 10 experiments in each system, performing five-fold cross-validation in each experiment. The simulation of the power network (in particular, the Jacobian matrix) was developed with the Matpower 6.0 toolbox [37]. The DC power flow analysis was used to approximate the state vector and measurement dataset from the AC power flow model. In the proposed model, state variable vector $\boldsymbol{\gamma}$ in the $T$-bus system is composed of $(T-1)$ bus voltage phase angles, and the meter measurement vector is composed of branch-active power flows and a set of active power injections into the buses. In order to carry out a more realistic power grid scenario, we used stochastic loads following a uniform load distribution [11] in the range of $[0.9 \times L_0 - 1.1 \times L_0]$, where $L_0$ is the base load. The attack was generated based on the attack model in [20], where the malicious user has total information on the topology of the power grid, and is able to access several sensors. We considered the degree of damage (defined as the difference between the real value and the attacked value to the state variables) as being randomly selected from between 4% to 6%. Furthermore, we studied an attack detection problem where the labels of the samples were randomly corrupted (training samples), and there existed unseen data with noise-free labels (testing samples), as illustrated in Figure 2.

The proposed scheme was compared with several ML-based schemes discussed in the literature, such as AdaBoost [38], random forest [31], and multilayer perceptron (MLP), where we used one hidden layer, and set the number of hidden neurons at 2/3 of the size of the features, following the recommendation in [39]. Furthermore, we evaluated some approaches with dimensionality reduction methods proposed in the literature for SCA attack detection, such as PCA with an SVM using a Gaussian kernel [11], a GA with an SVM using a Gaussian kernel [12], and PCA with iForest [13].

To evaluate the performance of the proposed scheme, we selected three metrics: accuracy, receiver operating characteristic (ROC) curves, and the value of the area under the ROC curve (ROC AUC).

- The accuracy is the ratio of correct detections, which is evaluated as follows:

$$Accuracy = \frac{TP + TN}{Total\ samples}, \tag{25}$$

  where $TP$ is the number of true positives, i.e. the samples that are detected as attacked and that are, in fact, attacks, and $TN$ is the number of true negatives, i.e. the samples that are classified as normal and that are, in fact, normal.
- The ROC curve illustrates the true positive rate ($TPR$) versus the false positive rate ($FPR$). $TPR$ represents the ratio of attacked samples that are correctly detected by the algorithm, and $FPR$ represents the ratio of normal instances that are incorrectly classified as an attack. The expressions for $TPR$ and $FPR$ are as follows:

$$TPR = \frac{TP}{TP + FN}, \tag{26}$$

  and

$$FPR = \frac{FP}{TN + FP}, \tag{27}$$

  where $FN$ is the number of false negatives, i.e. attack samples incorrectly classified as normal, and $FP$ is the number of false positives, i.e. normal samples incorrectly detected as an attack.
- The ROC AUC is the area enclosed by the ROC curve, and provides a single scalar value representing the ROC performance. The ROC AUC measures how good the algorithm is at distinguishing between an attack and a non-attack. In this way, a perfect classifier has ROC AUC $= 1$, and a totally random classifier has ROC AUC $= 0.5$.

Table 3 shows a comparison of the KPCA technique for several numbers of principal components using the standard IEEE 118-bus and 57-bus systems with 1000 training samples. We can see that the accuracy and the AUC value are similar among the six principal components, where we can see that the accuracy reached by using two components is the highest, which also allows decreasing the number of features, thus, reducing the computational complexity. Hence in the experiments, the proposed scheme uses two principal components for the kernel PCA technique, with $\sigma = 0.1$ for the IEEE 118-bus system, and $\sigma = 0.2$ for the IEEE 57-bus system. Therefore, the features are reduced from 489 in the 118-bus system, and from 216 in the 57-bus IEEE system, to two features through the KPCA algorithm. Then, the new training dataset is composed of two features per sample, and the Extra-Trees algorithm uses the number of decision trees in the ensemble as $M = 50$, and the number of features to select randomly as $K = 2$. We also compared the proposed scheme with the Extra-Trees algorithm without KPCA, using $M = 100$ and the default values from [15], to demonstrate the benefits of including dimensionality reduction in the proposed approach.

**TABLE 3.** Performance comparison of the number of principal components from the KPCA technique in the IEEE 118-bus and 57-bus systems.

| No. of PCs | Accuracy 118-bus | AUC value 118-bus | Accuracy 57-bus | AUC value 57-bus |
|---|---|---|---|---|
| 2 | 98.35% | 0.97796 | 98.20% | 0.97548 |
| 3 | 98.34% | 0.97792 | 98.19% | 0.97537 |
| 4 | 98.29% | 0.97751 | 98.14% | 0.97451 |
| 5 | 98.21% | 0.97679 | 98.13% | 0.97428 |
| 6 | 97.72% | 0.96871 | 98.09% | 0.97380 |

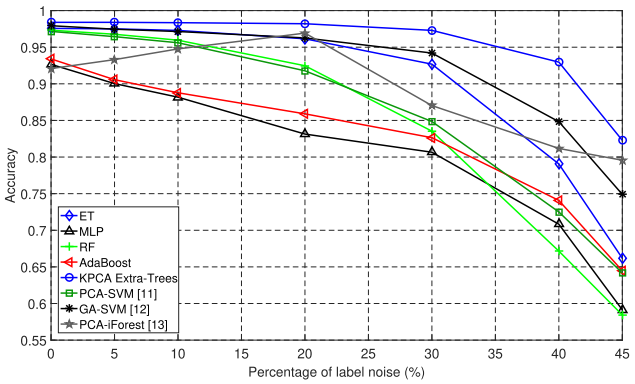



**FIGURE 4.** Accuracy vs. percentage of label noise of the proposed KPCA Extra-Trees scheme compared with state-of-art ML approaches.

First, we studied the impact of the percentage of label noise in the standard IEEE 118-bus system, where we used a dataset with 1000 training samples. Figure 4 illustrates the accuracy in the prediction versus the percentage of label noise introduced into the training data. We observe that as we increase the percentage of label noise, the accuracy decreases in all the schemes except in PCA–iForest, since the algorithms are trained with noisy labels that affect the ability of the algorithm to generalize for unseen data. Based on [15], the values of $n_{\min}$ in the Extra-Trees algorithm depend on the level of noisy labels in the training dataset, where a slight increase in the value of $n_{\min}$ is able to deal with an increment in the percentage of label noise. In this way, we select and reduce the value of $n_{\min}$ from $n_{\min} = 10$ for 0% to 20% of the label noise until $n_{\min} = 60$ for 40% to 45% of the label noise. Furthermore, the KPCA technique does not use the labels to perform dimension reduction, which is a significant factor for dealing with noisy labels, as we notice when we compare the proposed KPCA Extra-Trees scheme versus Extra-Trees without DR. With iForest, we see that the accuracy increases as we introduce more label noise up to 20%; after that, the accuracy starts to decrease. This behavior is because iForest is an unsupervised algorithm, i.e. it does not take into account the labels, and in the literature is trained with just normal samples; however, in this experiment, we observe that a small number of attack samples in the training phase can improve the overall accuracy, which was validated in the experiments in [40]. Furthermore, we can

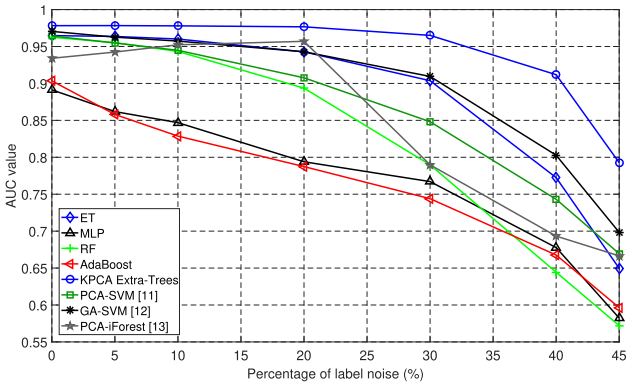



**FIGURE 5.** ROC AUC vs. percentage of label noise of the proposed KPCA extra-trees scheme compared with state-of-art ML approaches.

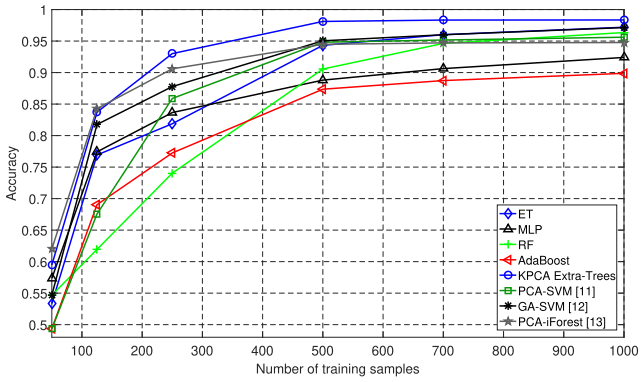
see that the proposed KPCA Extra-Trees scheme achieves the better accuracy among the compared approaches for all the percentages of label noise following the GA–SVM scheme from [12].

Figure 5 illustrates the ROC AUC value versus the percentage of label noise introduced in the training data. We observe behavior similar to Figure 4, where the best value for ROC AUC is reached by the proposed KPCA Extra-Trees scheme, with GA–SVM as the second-best approach. In order to study a hard realistic scenario, in the following simulations, we use 10% as the percentage of label noise.

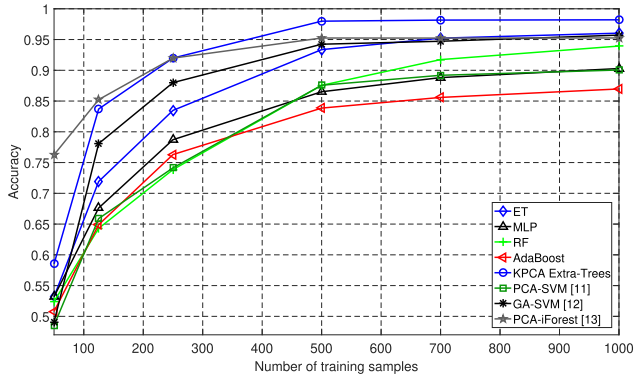
Figure 6 presents the accuracy in prediction versus the number of training samples in the IEEE 118-bus and 57-bus systems. We observe that the proposed KPCA Extra-Trees algorithm detects the SCA attack reliably from 500 training samples, and a further increase in the training samples does not provide a significant improvement in accuracy. As a result, we see that the proposed KPCA Extra-Trees scheme outperforms other approaches, and the best accuracy is reached with a lower number of training samples. Besides, GA–SVM [12] and PCA–iForest [13] present good accuracy in SCA attack detection, where PCA–iForest exhibits one of the best accuracies of all the ML schemes when we deal with a low number of training points (i.e., less than 250 training samples).

Figure 7 shows the ROC AUC value versus the number of training samples in the IEEE 118-bus and 57-bus systems. Similar to the results obtained in Figure 6, the highest AUC value in the proposed KPCA Extra-Trees scheme is reached with 500 training samples, KPCA Extra-Trees being the scheme with the highest AUC value among all the considered ML approaches. Note that the PCA–iForest [13] scheme can be considered the second-best approach from the point of view of the ROC AUC metric, outperforming other approaches, such as GA–SVM [12] and PCA–SVM [11].

Figure 8 presents the computational time used for the considered approaches during the training phase versus the number of training samples. We performed the simulations

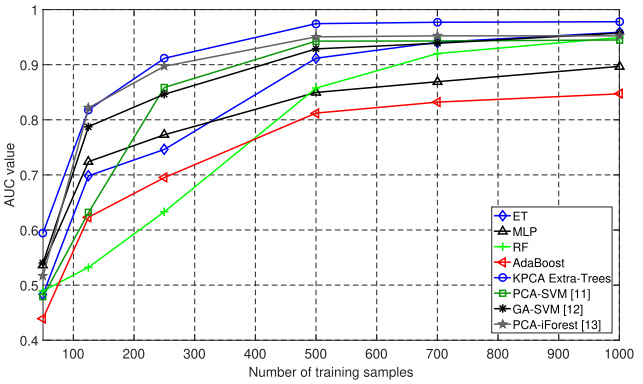



IEEE 118-bus system

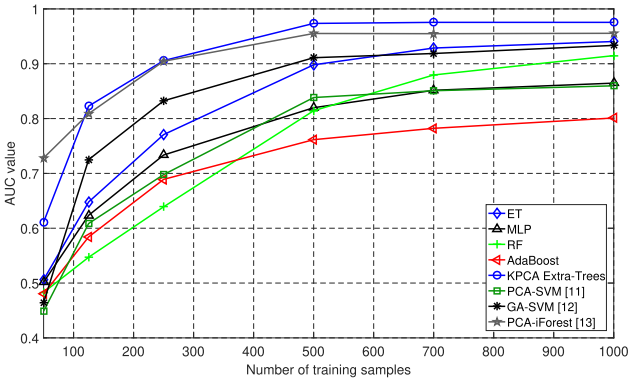



IEEE 57-bus system

**FIGURE 6.** Accuracy vs. number of training samples of the proposed KPCA extra-trees scheme compared with state-of-art ML approaches.




IEEE 118-bus system




IEEE 57-bus system

**FIGURE 7.** ROC AUC vs. number of training samples of the proposed KPCA extra-trees scheme compared with state-of-art ML approaches.

on an Intel Core i7-6700K CPU with 16 GB of main memory. We observed that the proposed KPCA Extra-Trees scheme can achieve the lowest computational time among all the algorithms with a high amount of training samples. These results are because of the dimensionality reduction performed by KPCA, which allows reducing the features from 489 in the IEEE 118-bus and from 216 in the IEEE 57-bus system to two features. Furthermore, the Extra-Trees splitting algorithm described in Table 2 allows us to significantly reduce the training time of the Extra-Trees algorithm.

Figure 9 shows the ROC curves in the standard IEEE 118-bus and 57-bus systems, where we used 1000 training samples. We observe that the highest area under the curve is achieved by the proposed KPCA Extra-Trees scheme, with a detection accuracy nearing 1 for both test buses. Note that the ROC AUC parameter, studied in Figure 7, validates the results illustrated in the ROC curves.

Finally, we compared the proposed KPCA method for dimensionality reduction with other methods, such as PCA [13], fast independent component analysis (ICA) [41], locally linear embedding (LLE) [42], Neighborhood Components Analysis (NCA) [43], binary PSO (BPSO) [44] and the GA [12]. The simulations were carried out using Extra-Trees as

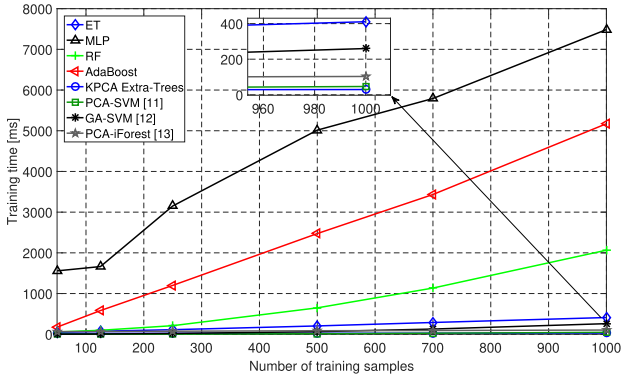
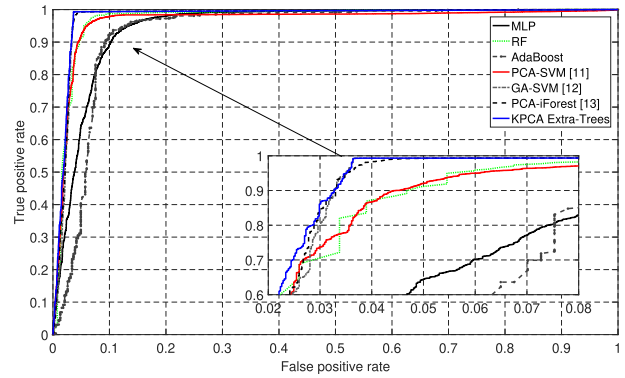
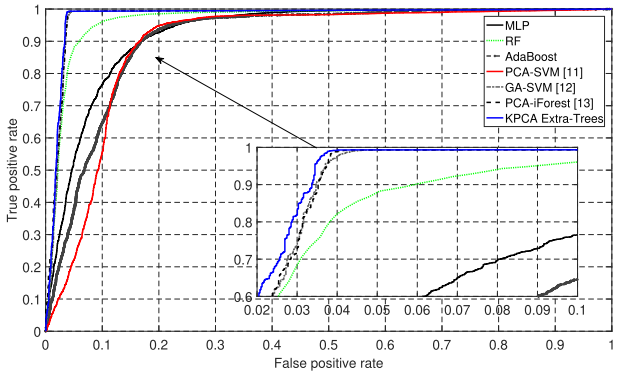



**FIGURE 8.** Computational training time vs. number of training samples of the proposed KPCA extra-trees scheme compared with state-of-art ML approaches.

the algorithm for SCA attack detection with 1000 training samples in the standard IEEE 118-bus system. In the dimensionality reduction methods of KPCA, PCA, and Fast ICA, we used two components, i.e. the new training dataset was composed of two features per sample. For the LLE and NCA algorithms, we achieved the best accuracy by using 20 features. Furthermore, the feature selection algorithms used for the comparison, i.e. BPSO and GA, follow the approach




IEEE 118-bus system




IEEE 57-bus system

**FIGURE 9.** ROC curve of the proposed KPCA extra-trees scheme compared with state-of-art ML approaches.

**TABLE 4.** Performance comparison between KPCA and other feature selection algorithms in the IEEE 118-bus and 57-bus systems.

| Algorithm | Accuracy | AUC value | Time of alg. [sec] | Time Extra-Trees [sec] |
|---|---|---|---|---|
| **None** | 97.20% | 0.9587 | — | 0.4106 |
| **KPCA** | 98.35% | 0.9779 | 0.0681 | 0.0245 |
| **PCA** | 97.82% | 0.9702 | 0.0159 | 0.0247 |
| **Fast ICA** | 97.89% | 0.9704 | 0.1806 | 0.0246 |
| **LLE** | 71.88% | 0.5507 | 0.3905 | 0.0455 |
| **NCA** | 95.15% | 0.9462 | 6.5712 | 0.0916 |
| **BPSO** | 96.27% | 0.9433 | 24.646 | 0.1218 |
| **GA** | 97.62% | 0.9652 | 19.529 | 0.1033 |

in [12] by using the filter-based FS mechanism to be independent of the classifier algorithm. In the experiments, the BPSO algorithm used five particles and 25 iterations, selecting on average 253 features among the original 489 features, and the GA algorithm used a population of 10 and 20 generations, selecting on average 251 features. Table 4 shows the accuracy and ROC AUC value of the proposed KPCA, compared with other dimension-reduction and feature-selection algorithms, where Time of alg. refers to the computational time to train the dimension-reduction/feature-selection algorithm, and Time Extra-Trees is the computational time to train the Extra-Trees algorithm using the output features obtained by the considered algorithms. We observe that the proposed KPCA algorithm for dimensionality reduction achieves the best accuracy and ROC AUC value, with a low value in computational time. Furthermore, it is observed that by using KPCA we can reduce the training time for the Extra-Trees algorithm by about 15 times.

## V. CONCLUSION

In this paper, we propose a DR-based ML scheme for the detection of SCA attacks in SG networks. In order to solve the computational complexity created by a high-dimensional space in large-sized power systems, we apply the KPCA technique to transform the data into a lower-dimensional space. The data transformed by KPCA become the input for the Extra-Trees algorithm, which is a fast and efficient algorithm to detect SCA attacks. We selected the standard IEEE 118-bus and 57-bus systems for the performance evaluation, taking into account the metrics of accuracy, ROC curve, and ROC AUC value. The dataset is composed of historical active power flow measurements and active power injections into the buses, which were collected at the PCC of the power network. The proposed scheme was compared with several ML-based approaches described in the literature. The numerical results validate that the proposed KPCA Extra-Trees–based detection approach outperforms the state-of-art ML-based schemes in terms of accuracy and ROC AUC value. A more realistic scenario was evaluated by considering training data corrupted with noisy labels, where the proposed scheme provides robust performance against noisy labels, mainly because KPCA does not take the labels into account when performing dimension reduction, and the Extra-Trees algorithm can deal with a noisy label by tuning the parameter of the minimum number of instances required in a child node to perform the split. In addition, we compared the computational complexity of all considered ML-based schemes, and found that the proposed scheme has the lowest computational time, which means that the proposed scheme can provide fast and accurate detection of SCA attacks in SGs.

## REFERENCES

[1] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *TISSECACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–33, May 2011.

[2] K. Khanna, B. K. Panigrahi, and A. Joshi, "Data integrity attack in smart grid: Optimised attack to gain momentary economic profit," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 16, pp. 4032–4039, Dec. 2016.

[3] X. Liu and Z. Li, "Local load redistribution attacks in power systems with incomplete network information," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1665–1676, Jul. 2014.

[4] B. Li, R. Lu, W. Wang, and K.-K. R. Choo, "Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system," *J. Parallel Distrib. Comput.*, vol. 103, pp. 32–41, May 2017.

[5] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, "Realtime detection of false data injection in smart grid networks: An adaptive CUSUM method and analysis," *IEEE Syst. J.*, vol. 10, no. 2, pp. 532–543, Jun. 2016.

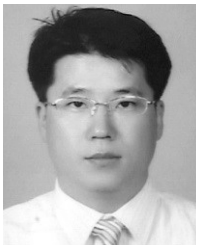
[6] Z. Fadlullah, M. Fouda, N. Kato, X. Shen, and Y. Nozaki, "An early warning system against malicious activities for smart grid communications," *IEEE Netw.*, vol. 25, no. 5, pp. 50–55, Sep. 2011.

[7] Y. Zhang, L. Wang, W. Sun, R. C. G. Ii, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.
[8] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, pp. 1–8.
[9] J. Hao, R. J. Piechocki, D. Kaleshi, W. H. Chin, and Z. Fan, "Sparse malicious false data injection attacks and defense mechanisms in smart grids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1–12, Oct. 2015.
[10] M. Ozay, I. Esnaola, F. T. Y. Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1773–1786, Aug. 2016.
[11] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1644–1652, Sep. 2017.
[12] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning," *IEEE Access*, vol. 6, pp. 27518–27529, May 2018.
[13] S. Ahmed, Y. Lee, H. Seung-Ho, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2765–2777, Mar. 2019.
[14] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," Aug. 2014. *arXiv:1207.3538v3*. [Online]. Available: https://arxiv.org/abs/1207.3538
[15] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, Mar. 2006.
[16] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 198–207, Feb. 2017.
[17] A. Abur and A. G. Expósito, *Power System State Estimation: Theory and Implementation*. Boca Raton, FL, USA: CRC Press, 2004.
[18] J. Casazza and F. Delea, *Understanding Electric Power Systems: An Overview of the Technology and the Marketplace*. Hoboken, NJ, USA: Wiley, 2011.
[19] A. Abdallah and X. Shen, *Security and Privacy in Smart Grid* (Springer-Briefs in Electrical and Computer Engineering). Cham, Switzerland: Springer, 2018.
[20] S. K. Singh, K. Khanna, R. Bose, B. K. Panigrahi, and A. Joshi, "Joint-transformation-based detection of false data injection attacks in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 89–97, Jan. 2018.
[21] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 382–390, Jun. 2011.
[22] S. Bi and Y. J. Zhang, "Using covert topological information for defense against malicious attacks on DC state estimation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 7, pp. 1471–1485, Jul. 2014.
[23] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.
[24] C. Jing and J. Hou, "SVM and PCA based fault classification approaches for complicated industrial process," *Neurocomputing*, vol. 167, pp. 636–642, Nov. 2015.
[25] S. W. Choi, C. Lee, J.-M. Lee, J. H. Park, and I.-B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemometrics Intell. Lab. Syst.*, vol. 75, no. 1, pp. 55–67, Jan. 2005.
[26] R. T. Samuel and Y. Cao, "Nonlinear process fault detection and identification using kernel PCA and kernel density estimation," *Syst. Sci. Control Eng.*, vol. 4, no. 1, pp. 165–174, Jan. 2016.
[27] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.
[28] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, Feb. 2018.
[29] C. E. G. Moreta, M. R. C. Acosta, and I. Koo, "Prediction of digital terrestrial television coverage using machine learning regression," *IEEE Trans. Broadcast.*, vol. 65, no. 4, pp. 702–712, Dec. 2019.
[30] A. Geron, *Hands-On Machine Learning With Scikit-Learn and TensorFlow*, 1st ed. Sebastopol, CA, USA: O'Reilly, Mar. 2017.
[31] L. Breiman, "Random forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[32] M. N. Kurt, Y. Yilmaz, and X. Wang, "Distributed quickest detection of cyber-attacks in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2015–2030, Aug. 2018.
[33] H. He and J. Yan, "Cyber-physical attacks and defences in the smart grid: A survey," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 13–27, Dec. 2016.
[34] S. Li, Y. Yilmaz, and X. Wang, "Quickest detection of false data injection attack in wide-area smart grids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2725–2735, Nov. 2015.
[35] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," *Comput. Netw.*, vol. 67, pp. 74–88, Jul. 2014.
[36] J. Jiang and Y. Qian, "Defense mechanisms against data injection attacks in smart grid networks," *IEEE Commun. Mag.*, vol. 55, no. 10, pp. 76–82, Oct. 2017.
[37] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
[38] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, Jan. 2009.
[39] S. Karsoliya, "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture," *Int. J. Eng. Trends Technol.*, vol. 3, no. 6, pp. 714–717, Dec. 2012.
[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation sforest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
[41] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.
[42] S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[43] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 513–520.
[44] S. M. Vieira, L. F. Mendonça, G. J. Farinha, and J. M. C. Sousa, "Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients," *Appl. Soft Comput.*, vol. 13, no. 8, pp. 3494–3504, Aug. 2013.

**MARIO R. CAMANA** received the B.E. degree in electronics and telecommunications engineering from Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016. He is currently pursuing the degree with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. His research interests include machine learning, optimizations, and MIMO communications.

**SAEED AHMED** received the B.E. and M.E. degrees in electrical engineering from the University of AJ&K, Pakistan, in 2005 and 2010, respectively, and the Ph.D. degree from the University of Ulsan, South Korea, in 2019. He served as a Transmission Engineer with telecom industry for eight years. He has a vast experience in planning, surveying, and deploying microwave and optical fiber-based core and access PDH/SDH/SONET/DWDM networks. He joined the Mirpur University of Science and Technology (MUST), Mirpur, Pakistan, as an Assistant Professor, in 2012. His research interests include energy-efficient resource allocation in cognitive radios, smart grid (SG) communication technologies, smart grid cyber security, and the Internet of Things (IoT).

**CARLA E. GARCIA** received the B.E. degree in electronics and telecommunications engineering from Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016. She is currently pursuing the degree with the School of Electrical Engineering, University of Ulsan, Ulsan, South Korea. Her main research interests include machine learning, MIMO communications, NOMA, and optimizations.



**INSOO KOO** received the B.E. degree from Konkuk University, Seoul, South Korea, in 1996, and the M.Sc. and Ph.D. degrees from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 1998 and 2002, respectively. From 2002 to 2004, he was a Research Professor with the Ultrafast Fiber-Optic Networks Research Center, GIST. In 2003, he was a Visiting Scholar with the Royal Institute of Science and Technology, Stockholm, Sweden. In 2005, he joined the University of Ulsan, Ulsan, South Korea, where he is currently a Full Professor. His current research interests include spectrum sensing issues for CRNs, channel and power allocation for cognitive radios (CRs) and military networks, SWIPT MIMO issues for CRs, MAC, and routing protocol design for UW-ASNs, and relay selection issues in CCRNs.