Sustainable Energy, Grids and Networks 34 (2023) 101017

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan

# Ensemble models for circuit topology estimation, fault detection and classification in distribution systems

Aswathy Rajendra Kurup [a,*], Adam Summers [b], Ali Bidram [a], Matthew J. Reno [b], Manel Martínez-Ramón [a]

[a] Department of Electrical Engineering, The University of New Mexico, NM, USA
[b] Sandia National Laboratories, NM, USA

ARTICLE INFO

*Article history:*
Received 6 July 2022
Received in revised form 23 November 2022
Accepted 2 February 2023
Available online 8 February 2023

*Keywords:*
Distribution systems
Adaptive protection
Topology estimation
Fault detection
Convolutional neural networks
Support vector machines

ABSTRACT

This paper presents a methodology for simultaneous fault detection, classification, and topology estimation for adaptive protection of distribution systems. The methodology estimates the probability of the occurrence of each one of these events by using a hybrid structure that combines three sub-systems, a convolutional neural network for topology estimation, a fault detection based on predictive residual analysis, and a standard support vector machine with probabilistic output for fault classification. The input to all these sub-systems is the local voltage and current measurements. A convolutional neural network uses these local measurements in the form of sequential data to extract features and estimate the topology conditions. The fault detector is constructed with a Bayesian stage (a multitask Gaussian process) that computes a predictive distribution (assumed to be Gaussian) of the residuals using the input. Since the distribution is known, these residuals can be transformed into a Standard distribution, whose values are then introduced into a one-class support vector machine. The structure allows using a one-class support vector machine without parameter cross-validation, so the fault detector is fully unsupervised. Finally, a support vector machine uses the input to perform the classification of the fault types. All three sub-systems can work in a parallel setup for both performance and computation efficiency. We test all three sub-systems included in the structure on a modified IEEE123 bus system, and we compare and evaluate the results with standard approaches.



## 1. Introduction

The protection system is a critical component of power grids used to detect and isolate faults. A protection system should ascertain sensitivity and selectivity requirements. Sensitivity is defined as the ability to detect and isolate faults fast enough to avoid their spread. Selectivity is the ability to minimize the number of customers experiencing power outages during the isolation process [1]. The protection system plays a critical role in improving the power system reliability and resilience and avoiding major outages with possible cascading effects [2–4]. More recently, adaptive protection has been introduced as a promising solution to effectively modify protection responses in real-time based on the prevailing system conditions.

Adaptive protection is defined as a protection system that can modify its protection actions in real-time according to the latest power system changes [2,5–9]. One of the power system changes that can significantly impact the fault current values is a circuit configuration change. Conventionally, adaptive protection systems rely heavily on the communication infrastructure to monitor system condition changes [10]. However, the performance of conventional adaptive protection systems is impaired when the communication network is down because of cyber-attacks and communication link failures. In this paper, a communication-free adaptive protection solution is proposed. The communication-free adaptive protection devices can replace conventional protection relays. The proposed protection devices can estimate the prevailing circuit configuration and detect different types of faults accordingly. For this, the data needed for the adaptive protection operation is collected locally in each relay. This local data, consisting of measures of the voltage and current is then used by a machine learning meta-structure to estimate the grid topology, the fault, and fault type probabilities given the current measures.

Knowing the configuration of a circuit is of particular importance for the adaptive protection of distribution systems to adaptively take protection actions depending on the topology of the circuit [11,12]. In a protection system, fast fault detection is of particular significance to quickly remove faults before endangering human life and damage to power system equipment [13].

* Corresponding author.
*E-mail address:* aswathyrk@unm.edu (A. Rajendra Kurup).

https://doi.org/10.1016/j.segan.2023.101017
2352-4677/

The estimation of the grid configuration or topology can be complicated by the fact that there are situations where the switches and circuit breakers lose their communications capabilities, and in these cases, only local observations are available. In these cases, topology estimation can be implemented by the use of the physical model of the grid [14,15], or just using data observed from the grid. These data consist of the electrical magnitudes of synchrophasors, smart meters, or measurement equipment in a given location of the grid [16].

In a modern distribution system, with the possibility of different configurations, the post-fault measurements from the protection devices are highly impacted by the prevailing circuit condition. The conventional protection schemes with fixed settings are well-tuned for one circuit configuration. Therefore, upon the change of the circuit configuration, the protection device may not be able to satisfy the reliability and selectivity requirement of a protection system.

To tackle this challenge, this paper proposes the use of machine learning to effectively account for the different circuit configurations on the actions taken by the protection device. The proposed innovations of the present work are the following:

- A one-dimensional convolutional neural network (CNN) is utilized that processes a window of the electrical measurements taken at the circuit relays to estimate the circuit configuration, and whose output is a probabilistic estimation of the configuration given the observed input.
- Simultaneously with the configuration estimation, automatic fault detection is performed. The novelty consists of the use of a structure that, with the use of a Gaussian Process (GP) regressor that estimates the instantaneous values of the current from the voltages, computes the posterior probability error covariance (which is assumed to be Gaussian), whitens it and then, with the help of a one-class SVM (OC-SVM) [17] structure, the probability of fault is estimated. The fault detection is fully unsupervised and it works by computing the residuals of the observation predictions [18]. A set of fault detectors are trained, each one for each of the configurations, which allows obtaining fault probabilities conditional to each configuration.
- The configuration and fault probabilities are combined with the fault type probability, by training a set of fault classifiers, each one for each configuration. The corresponding conditional fault-type probabilities are combined with the rest of the probability estimations in order to obtain marginal probabilities. This combination of probabilistic estimations (soft decisions) is optimal in a Bayesian sense and it makes the structure more robust than a structure based merely on binary (or hard) decisions.
- The proposed fault detection approach is based entirely on local measurements and thought for these cases where communications are not possible or not consistent, thus graph modeling of the grid is not possible. This communication-free, setting-less approach can replace the conventional protection relays in power grids.

This work is a continuation of the work presented in [19] where we introduced a preliminary structure for configuration classification and fault classification. The novelties with respect to that paper consist of a novel, unsupervised, fully automatic fault detection procedure that estimates fault probabilities rather than offering a binary detection of these events. Also, the configuration classification and fault classification are here probabilistic. Indeed, this structure combines the probabilities corresponding to each event to optimize the detection.

The presented results show good error probabilities in configuration and error classification and the probability of false alarms in fault detection. It should be noted that the possibility of modeling a distribution system with different operating conditions (e.g., circuit configurations) offers a data-driven approach based on machine learning. Indeed, when an accurate model of the grid is available and a large quantity of data corresponding to the different possible configurations of the grid can be produced, then supervised machine learning algorithms can be effectively trained to detect such configurations. In this case, different grid faults can be also simulated and machine learning algorithms can be trained to detect and classify these faults. This work is a continuation of the work presented in [19].

The rest of the paper is organized as follows: the next section presents the structure and methodology, including the three used subsystems. The experimental section includes the generation of data, the training of all substructures, and the results and discussion of the experiments. We finish the paper with the conclusion section.

## 2. Related work

### 2.1. Topology estimation

The recent researches on topology estimation mostly use State estimation (SE) based techniques. In such approaches, the system state variables are estimated across a network using physical laws and measured quantities. Such approaches helped with network forensics, mainly for spotting modeling errors or bad measurements. A weighted least squares (WLS) approach has been utilized in [20] for obtaining the status of circuit breakers. The probabilistic model was further improved using additional data associated with active and reactive power in circuit breakers. Later, a generalized state estimation (GSE) was proposed in [21] which combines the two basic power system monitoring modules, i.e., state estimation and topology processing. Further, [22] introduces SE in a distribution network and proposes a bayesian approach using linear least square estimation. The concept of protection schemes based on the dynamic state estimation (DSE) was introduced in [23]. Though they were inspired by differential protection that utilizes limited data, the proposed approach requires all the measurements in the protection zone. Additionally, they also use all physical laws in the dynamic protection zone for estimation and the system ignores external faults and identifies only internal faults. Later on, there were DSE-based protection techniques such as [24] where support vector machines (SVM) were utilized to classify line faults. The transmission line models and the measurement data are utilized by DSE for estimating the states of the line. The training speed of the approach is increased using Lasso regression. [25] studies topology estimation in a distribution system based on time-series measurements of voltage. This approach uses a lookup table and certain thresholds to obtain the best result on topology detection. The work in [26] presents a topology estimation methodology based on sparse Markov Random fields. In [27], a regression structure adjusted by second-order methods are used to estimate the structure from smart meter measurements.

### 2.2. Fault detection and classification

There are several possible approaches for fault detection in the related literature, that have been studied extensively. Among them, the unsupervised anomaly detection methods are the most suitable ones in the present application, because they do not need labeled training data and they are robust to the fact that there is a huge imbalance between normal and faulty data. Non-supervised novelty detection methods can be taxonomized as density-based,

bagging, deep learning, and linear methods. Density-based methods are methods that construct a model of the data based on a clustering, from which a direct measure of the likelihood of the data can be estimated. For example, [28] computes data histograms to determine the level of likelihood of the data, and [29] uses distance measures between data to determine the level of [*sic*] "outlierness" of a sample. The works in [30,31] uses the well-known k-nearest neighbor (KNN), which can be seen as a simplification of a Gaussian mixture model (GMM) when this GMM is trained with an Expectation–Maximization (EM) method [32]. The KNN can be viewed as a mixture of Gaussians with an identity as a covariance matrix where the posterior probabilities of the model are either 0 or 1. Thus, in a way, the KNN provides an indirect measure of the likelihood of the samples. Bagging models are based on the construction and combination of different outlier detectors with techniques based on bagging or subsampling of the data, with the aim of reducing the uncertainty of the detection (see, e.g. [33,34]).

Deep learning methods for outlier detection using non-supervised methods that construct a reduced dimensionality distribution of the data. For example, [35] uses the well-known autoencoder structure [36]. In [37], autoencoder ensembles are introduced together with adaptive sampling to construct the base models of the ensemble. A different technique, based on generative adversarial networks [38] is used in [39]. The above-mentioned algorithms are particularly well suited for cases where the number of data available for training is very high. But in situations where the number of samples is low, or when the use of reduced sets is desired in order to reduce further the computational burden, linear algorithms are a possible choice when these algorithms can be modified to endow them with nonlinear properties through the kernel trick (see e.g. [40]) as, for example, [41] or [42].

A linear algorithm that is specifically designed to be constructed together with the kernel trick is the well-known OC-SVM [17]. The SVM training criterion is particularly well suited for cases where the available dataset is small or when a low computational burden is required. Also, the number of parameters to be adjusted by cross-validation is small, and the cross-validation can be skipped in this application as will be seen further. This is thus the choice for the present application. A fairly equivalent algorithm also based on SVM is the Support Vector Data Description (SVDD) method presented in [43].

Most of the works applying fault detection in power systems utilized traditional feature extraction methods and ML algorithms for performing the detection or classification. In [24], the difference between measured and estimated values, called residual values was used to learn more about the fault type. Additionally, there were several ML-based approaches for identifying fault locations and fault types in power systems. In [44], a Discrete wavelet transform (DWT) is used to obtain the transient information from voltage measurements which are further passed through an SVM classifier for determining the fault sections. Later, another wavelet transform (WT) and SVM-based approach [45] was proposed to extract features from three-phase fault current using WT and perform statistical measures on them to be passed to SVM. In [46], the authors propose an approach for fault diagnosis in a distribution network using a novel optimization algorithm called perturbed particle optimization (PPSO) to support SVM to improve its performance.

## 3. Methodology

Our proposed approach creates an ensemble model for the joint circuit topology estimation, fault detection, and fault classification. The ensemble model for the joint circuit topology estimation and fault detection classifiers consists of two main blocks that operate in parallel (Fig. 1) and whose outputs are combined later. The first block is a CNN trained with a window of voltage and current samples obtained from one of the relays. The network is trained to detect the configuration, and its output is an estimation of the probability of each configuration. Thus, the CNN output can be written as the probability $p(C_j|\mathbf{x})$ where $C_j$ represents configuration $j$ and $\mathbf{x}$ is the input sample.

Four fault detectors are trained to detect faults from observations $\mathbf{x}$ of each one of the configurations. The corresponding structure can be seen in Fig. 1. During the test phase, all fault detectors are given all data. Therefore, each data is tested for fault detection assuming all possible hypotheses over the possible configurations. These detectors provide an estimation of the fault probability given the sample, under each one of the possible hypotheses over configurations $C_j$. These probabilities are denoted as $p(F|C_j, \mathbf{x})$, where $F$ represents the event of a fault. Finally, a set of blocks are trained to classify over faults. These blocks are trained with faulty data only, and a different classifier is trained for each one of the possible configurations. Again, during the test phase, the data is introduced into the classifiers regardless of the configuration and the fault event. Therefore, the outputs are the fault classification under the hypothesis that a fault has happened and under the hypothesis of each one of the possible configurations. Each classifier outputs probabilities $p(Fc_i|C_j, F, \mathbf{x})$, where $Fc_i$ represents fault class $i$. These probabilities are read as the probability that the fault is class $i$ given the topology configuration of class $j$, given a fault has happened, and given observation $\mathbf{x}$. In Fig. 1 three possible fault classes are considered, so $1 \leq i \leq 3$.

Once the estimations of probabilities $p(C_j|\mathbf{x})$, $p(F|C_j, \mathbf{x})$ and $p(Fc_i|C_j, F, \mathbf{x})$ are computed in parallel for a given input $\mathbf{x}$, the estimations of the fault probabilities and the fault classes are computed as

$$p(F|\mathbf{x}) = \sum_j p(C_j|\mathbf{x}) p(F|C_j, \mathbf{x}) \tag{1}$$

and

$$p(Fc_i|\mathbf{x}) = \sum_j p(C_j|\mathbf{x}) p(Fc_i|C_j, F, \mathbf{x}) p(F|C_j, \mathbf{x}) \tag{2}$$

This structure is justified by the necessity to minimize the probability of error in the decision by fully exploiting the capabilities of the machine learning substructures, and also being aware of their limitations. On the one hand, any ML procedure has a probability of error. Therefore, in a system like the one presented here, it is risky to fully trust the decision of the CNN regarding the present configuration and then choose a single fault detector and a single fault classifier. Instead, we use the scores that the CNN provides for a given output. These scores are normalized (i.e., they add to one), so they have properties of probability mass function. In particular, these scores are modeled as posterior probabilities or, in other words, probabilities of each one of the configurations given the observation $\mathbf{x}$. These scores are used to weigh the decision of each one of the fault detectors and classifiers. In the case that the CNN provides a clear classification or a classification where one of the scores is close to 1 and the rest are close to zero, the system will automatically select one of the classifiers and one of the fault detectors. But in case of an unclear classification, where two or more scores have a significant value, instead of choosing the classification corresponding to the highest score, a more parsimonious solution consists of using all classifiers corresponding to these scores. In the case where the highest score does not match the actual configuration, the corresponding fault detector and classifier may not work properly, thus providing unclear classifications (i.e. with low scores). But one of the fault classifiers and fault detectors will match the actual configuration,

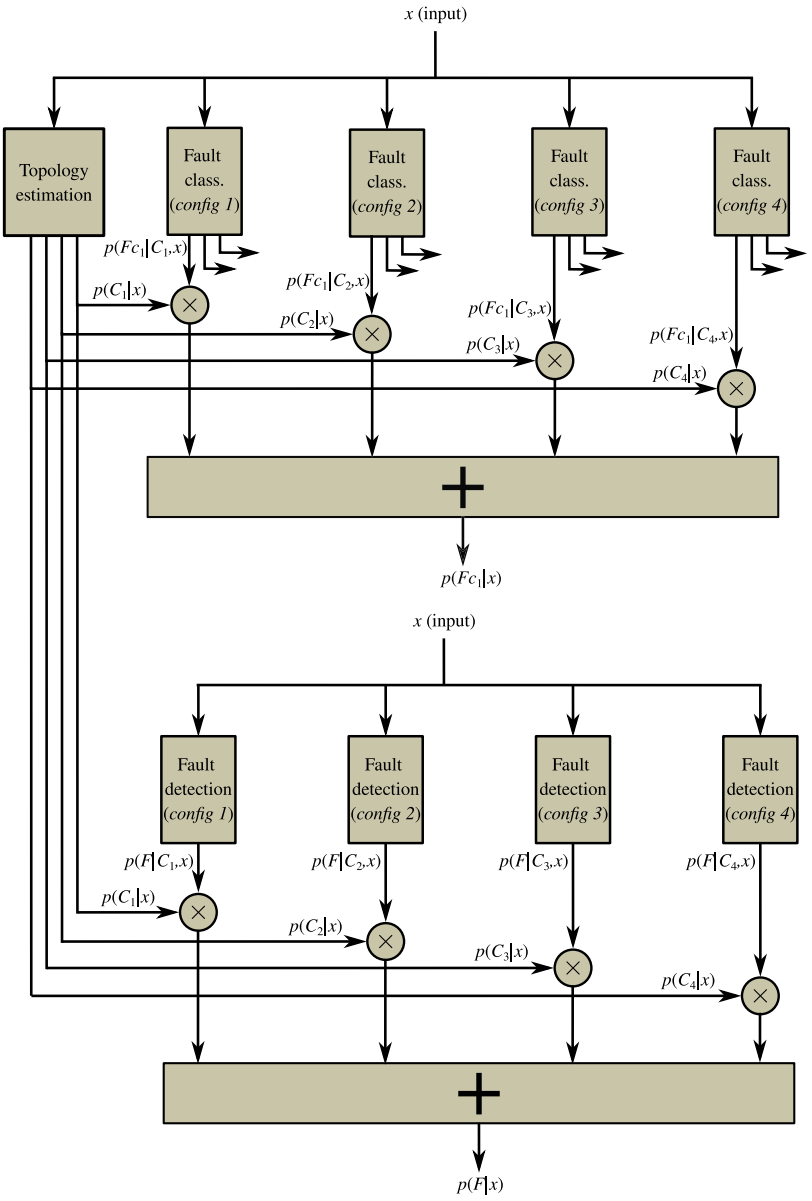


**Fig. 1.** Proposed structure for configuration classification, fault detection, and fault classification probability estimations. The topology estimation block estimates the configuration. The fault classifiers determine the fault class probability given the configuration. The output corresponding to $p(Fc_1|C_j, F, \mathbf{x})$ for each one of the classifiers is connected to the corresponding class probabilities $p(C_j|\boldsymbol{x})$ and then added together to compute the marginal fault class probability. The unconnected arrows of the classifiers represent the conditional probabilities for $Fc_2$ and $FC_3$, with whom the same operation is implemented to compute the corresponding fault class marginal probabilities, but the corresponding sub-blocks have been omitted for brevity.

and they will provide a good score for one of the classes, and then the system will select these ones, thus reducing the probability of error.

On the other hand, instead of just providing a classification, the system outputs a probability estimation. This can be advantageous in a decision-making process. If the probability of fault or the probability of fault class for a given class is very high, one can make a decision safely. In cases where the probability of fault is close to 0.5 or the probabilities of a fault class are all similar, this means that the algorithm is not able to provide a safe criterion to make a decision, and the user may decide not to use the result.

The use of different classifiers and fault detectors for different configurations is justified by the reduction of the complexity of the structure, which usually translates into better performance. Indeed, during the training phase, every subsystem (except the CNN) is trained with data of a single configuration, which produces data distributions simpler than the distribution of the whole data. This helps to improve the performance of the machines. The drawback of this is a more complex structure that needs a separate detection of the class. CNNs are best known for capturing spatial information in images or temporal information in time series data. Hence, we use CNN for this task due to its well-known capabilities in solving such complex problems in temporal data, as long as the quantity of data is sufficient.

The CNN and the fault classifiers are supervised, thus they are trained with simulated data which has been previously labeled. The training of the fault classifiers would not be possible if the data was real, because usually faults have a low likelihood, hence collecting enough faulty data and labeling it is an impossible task. Nevertheless, we choose a nonsupervised fault detector, which is designed by taking into account the characteristics of a real fault. In other words, the fault detector simply estimates the likelihood of the observed events, and it classifies a sample as a fault when its likelihood is low. A supervised method would also

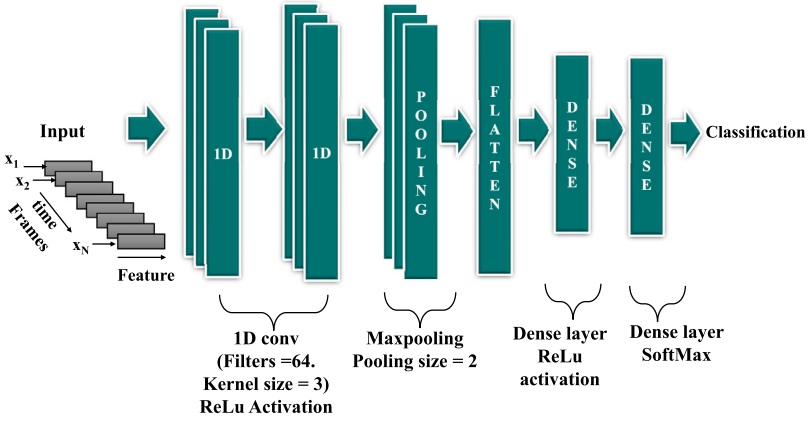


**Fig. 2.** Structure of the CNN used for the classification of configurations. The first and second layers perform 1D convolution with a kernel of length 3 which is followed by ReLU activation. A single max-pooling layer subsamples the outputs, which are converted into a vector (flattened). The final stage consists of two dense layers. The first one has ReLU activations and the second one has SoftMax activations so the output of the network has properties of probability mass function.

be possible in this context, but a non-supervised one is better because it is able to distinguish novelties that are not present in the simulation, which may otherwise be unnoticed. Also, the nonsupervised structure will be more robust to the different levels of noise that may appear in real data, which are not present in the simulated one.

### 3.1. Configuration classification with CNN

The name of convolutional neural network is given to this class of deep learning structures because they use the convolution operation as a means to extract features of the data [47]. The standard CNN procedure consists of two basic sections. The first one applies the convolutions, and the second one is a standard neural network that produces the desired estimation operation. The first block usually contains a layer that performs a convolution between the input and a set of convolution kernels, which are usually matrices of trainable parameters. The outputs of the convolutions are then subsampled. The subsampling operation divides the result of the convolutions into small blocks and keeps the sample with the maximum value in an operation called MaxPool, which keeps the dominant data and discards the rest [48]. The outputs of the MaxPool operations are then passed through a nonlinear function. This function is usually a logistic function $\phi(z) = \frac{1}{1+exp(-z)}$ or, more commonly, a rectified linear unit (ReLU) $\phi(z) = \max(0, z)$. This output may be processed again with the same procedure, which results in new convolutions of reduced dimension (due to the max pooling). The outputs of the last convolutional layer are considered the extracted features of the input pattern, and they are then concatenated in a vector and passed through a dense neural network of one or more layers, that produces the estimation.

The intuitive principle of the CNN is that the model learns the representation of the data through the feature learning process [49] consisting of these convolutions, which add certain invariant properties to the extracted features. The CNN used here processes time series, so the convolutions are performed in one dimension, and the convolutional kernels are simply short sequences of trainable parameters.

The structure of CNN used for this application can be seen in Fig. 2. The CNN used here receives a multivariate time series input. So, the input data at time instant $t$ is in a 2-D format of $N$ x $N_{feat}$ consisting of $N$ time steps and $N_{feat}$ features. The time steps correspond to a set of consecutive samples in the form of a data frame. Therefore, at each time instant $t$ and for each one of the features, a window of $N$ samples (the present sample at time $t$ plus past $N-1$ samples shown in Fig. 2 as $x_1, x_2, \ldots, x_N$ with $x_i$ having $N_{feat}$ features) is used as an input to each one of the 1-D convolutions of the CNN.

The data is first processed with two convolutional layers, each one having 64 kernels of length 3. After that we apply a dropout layer with a dropout rate defined as 0.5, which corresponds to dropout probability, followed by maxpooling of dimension 2 (this is, from every two sequential samples at the output of the convolutions, we keep the one with the maximum amplitude and we discard the other one). This output is then flattened and passed onto a dense neural network with two layers. The first dense layer with 100 outputs has ReLu activation and the final layer, with $K$ outputs have a softmax activation function

$$\phi_k(z_k) = \frac{\exp(-z_k)}{\sum_j \exp(-z_j)} \tag{3}$$

with $1 \leq k \leq K$ which generates the probability of the sample belonging to the given class. In order to provide accurate posterior probability estimates, the neural network must be trained with a backpropagation algorithm with cross-entropy objective function. This criterion maximizes the log posterior probability of the class given the input (see e.g. [47]), and it provides reasonable probability estimates, as it will be seen in the experiments. An accurate posterior probability with the use of SVMs is more difficult given that the SVM criterion does not optimize a posterior probability.

### 3.2. Fault detection

In this section, we introduce a machine learning metastructure to detect the faults based on an indirect measure of the likelihood of the observation. Since the faults are assumed to have a low probability, we consider that an event with a low probability will be a fault candidate.

One of the advantages of the SVM approaches is that they are model-free, so the user does not need to make assumptions about the distribution of the data. Also, their generalization capabilities make them trainable with low samples [50,51], which is an advantage with respect to mixtures of experts, which usually need more data. Both OC-SVM and SVDD are designed to automatically separate normal data from novel data using maximum margin criterion. The OC-SVM adapts a strategy that assumes that almost all data can be separated from the origin of coordinates using a hyperplane. Novel data, or anomalies, are those samples whose likelihood is low and, therefore, they will be separated from the normal data, which is assumed, in this linear approach, to form a cluster. These are the samples left in the space between the

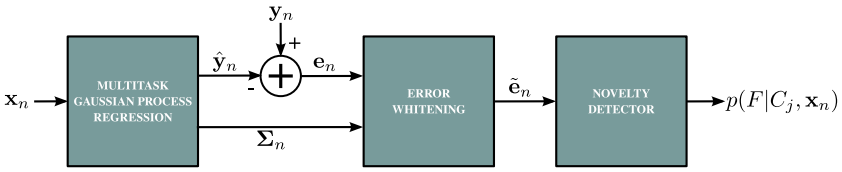


**Fig. 3.** A fault detector is constructed and trained for each one of the possible configurations. The input data $\mathbf{x}_n$ at instant $n$ is used as a predictor for a sample $\mathbf{y}_n$. The prediction is done with an MTGP, which outputs the mean prediction for that sample, plus the corresponding predictive covariance matrix. Then, the prediction error is computed and then whitened with the covariance matrix. The whitened sample is introduced in an unsupervised SVM novelty (i.e., fault) detector. The soft output of the SVM is the estimation of the fault probability.

origin and the separating hyperplane. In order to keep the normal data the closest possible to the plane and the novelties as far away as possible, the distance of the hyperplane to the origin is maximized. Since this linear algorithm is very restrictive, the data is first transformed into a higher dimension Hilbert space $\mathcal{H}$ through a nonlinear mapping $\varphi(\cdot)$. The space is endowed with a kernel dot product $k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle$, where, by virtue of the Mercer's theorem [52,53], function $k(\cdot, \cdot)$ only needs to be positive definite in order to be a dot product into a Hilbert space. If the separating hyperplane is defined as $\mathbf{w}^\top \varphi(\mathbf{x}) + \rho = 0$, the optimization criterion can be written as

$$\begin{aligned} &\text{Minimize } \|\mathbf{w}\|^2 + \frac{1}{N\nu} \sum_{n=1}^{N} \xi_n - \rho \\ &\text{subject to } \begin{cases} \mathbf{w}^\top \varphi(\mathbf{x}_n) + \rho \geq -\xi_n \\ \xi_n \geq 0 \end{cases} \end{aligned} \tag{4}$$

where $\xi_n$ are slack variables or losses that are positive if sample $\varphi(\mathbf{x})$ is between the plane and the origin, and zero otherwise. This constrained problem may be solved by Lagrange optimization by multiplying each constraint $\mathbf{w}^\top \varphi(\mathbf{x}_n) + \rho \geq -\xi_n$ by a Lagrange multiplier $\alpha_n$, which leads to the dual problem

$$\begin{aligned} &\text{Minimize } \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \\ &\text{subject to } \begin{cases} 0 \leq \alpha_n \leq \frac{1}{N\nu} \\ \sum_{n=1} \alpha_i = 1 \end{cases} \end{aligned}$$

$\mathbf{K}$ is the matrix of the kernel dot product between data with entries $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. This dual can be solved by quadratic programming in an efficient way. It can be seen from the dual constraints that $\nu$ is an upper bound of the number of novelties present in the training data. After the optimization, detector can be written as $f(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{k}(\mathbf{x}_i, \mathbf{x}) + \rho$, where $\mathbf{x}_i$ is the set of training data.

Notice also that this algorithm is unsupervised. During the test phase, we must only evaluate whether the test sample is at one or other side of the plane in order to decide if it is a novelty. There are two free parameters of this algorithm that need to be adjusted, which are the value of $\nu$ and the parameters of the used kernel function. A very popular kernel function is the square exponential $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$. Thus, the second parameter to validate is $\gamma$. This precludes the use of this strategy as unsupervised since a set of labeled data is needed to validate this parameter.

We propose to avoid a strict validation of this parameter by using a preprocessing of the input data that has the effect of standardizing it, with the assumption that the error has a Gaussian distribution with zero mean and some covariance matrix. If the error can be standardized (i.e. to have zero mean and identity covariance matrix), the kernel parameter can be easily guessed (Fig. 3).

The idea consists of predicting one part of the present observation, say $\mathbf{y}_n$ with another part of the present observation $\mathbf{x}_n$. For example, the current at instant $n$ is to be predicted from the voltage. If the observation is normal, with the use of an adequate predictor, the current will be predicted with low error. But if the sample is a novelty (i.e., a fault), the prediction error is expected to be much higher, as the likelihood of the sample is low, and similar samples will not be present in the training sample (or their presence will be negligible), so the prediction scheme will not have information about it in order to predict it. We use a multitask Gaussian process algorithm in order to produce the prediction since this algorithm produces a posterior probability distribution of the sample, where a Gaussian assumption is taken with respect to the prediction error, which can be further whitened with the use of the covariance matrix of the predictive error distribution.

A prediction can be constructed for vector $\mathbf{x}_n \in \mathbb{R}^D$ using a nonlinear multitask prediction model formulated by means of a kernel function, which has the form

$$\hat{\mathbf{y}}_n = \mathbf{C} \otimes \mathbf{k}(\mathbf{x}_n) \boldsymbol{\alpha} \tag{5}$$

where $\mathbf{k}(\mathbf{x}_n)$ is a vector whose elements $k(\mathbf{x}_i, \mathbf{x})$ are the kernel dot products between the $N$ training samples and the test sample. Usually, the kernel dot product is defined as

$$k(\mathbf{x}, \mathbf{x}') = \sigma_1^2 k_f(\mathbf{x}, \mathbf{x}') + \sigma_2^2$$

where $k_f(\mathbf{x}, \mathbf{x}')$ is a positive definite function and $\sigma_1^2$ and $\sigma_2^2$ are two positive scalars necessary to determine the covariance and the mean of the predictions. These parameters must be validated in a Gaussian Process framework. Vector $\boldsymbol{\alpha} \in \mathbb{R}^N$ contains parameters $\alpha_{i,j}$ that needs to be optimized, and $\mathbf{C} \in \mathbb{R}^{D \times D}$ is a matrix containing the covariance terms between elements of $\mathbf{x}$, which has to be validated, and $\otimes$ represents the Kronecker product. Matrix $\mathbf{C} \otimes \mathbf{k}(\mathbf{x}_n)$ has thus dimensions $D \times N$.

In order to optimize the estimator, it is assumed that the training estimation error $e_n = \mathbf{y}_n - \hat{\mathbf{y}}_n$ has a Gaussian distribution with covariance matrix $\boldsymbol{\Sigma}_n$. Given the kernel dot product, it is straightforward to prove that the set of dual variables $\alpha_i$ are modeled as latent random variables whose prior distribution is a multivariate Gaussian with covariance matrix $\mathbf{K}^{-1}$. With this in mind, a predictive posterior distribution can be constructed for the test samples $\mathbf{y}_n$, which is a Gaussian with mean and covariance

$$\begin{aligned} &\hat{\mathbf{y}}_n = [\mathbf{C} \otimes \mathbf{k}(\mathbf{x}_n)]^\top \tilde{\boldsymbol{\alpha}}, \\ &\hat{\boldsymbol{\Sigma}}_n = \boldsymbol{\Gamma} \otimes k(\mathbf{x}_n, \mathbf{x}_n) - [\boldsymbol{\Gamma} \otimes \mathbf{k}(\mathbf{x}_n)]^\top \left(\tilde{\mathbf{K}} + \boldsymbol{\Sigma}_p \mathbf{I}\right)^{-1} \boldsymbol{\Gamma} \otimes \mathbf{k}(\mathbf{x}_n), \end{aligned} \tag{6}$$

Then, the prediction algorithm outputs a mean of the prediction and a covariance matrix of this prediction. With this, a prediction error $\mathbf{e}_n = \hat{\mathbf{y}}_n - \mathbf{y}_n$ is computed. The whitening process simply consists of transforming the error into a random variable whose covariance matrix is an identity. Indeed, if error $\mathbf{e}_n$ is drawn from a Gaussian distribution $\mathcal{N}(\mathbf{e}_n | 0, \boldsymbol{\Sigma}_n)$ of zero mean and covariance $\boldsymbol{\Sigma}_n$, then its negative log-likelihood is proportional to

$$-\log \mathcal{N}(\mathbf{e}_n | 0, \boldsymbol{\Sigma}_n)^{-1} \propto \mathbf{e}_n^\top \boldsymbol{\Sigma}_n \mathbf{e}_n \tag{7}$$

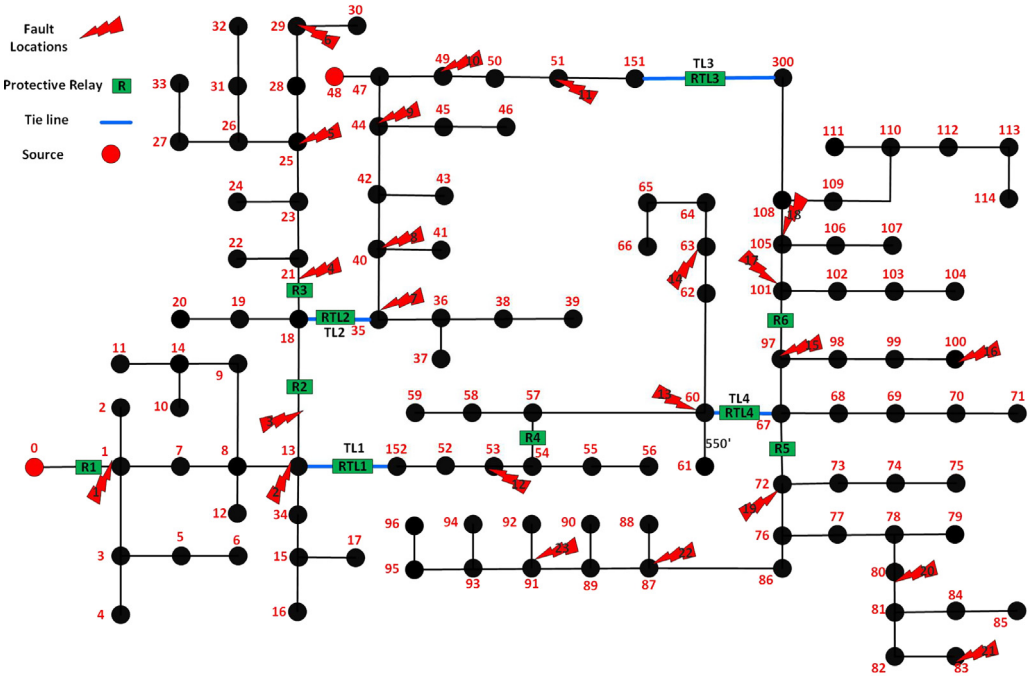


**Fig. 4.** Modified IEEE 123 bus distribution system showing the fault locations, the four tie-lines, and relay locations, indicated as R. Relay tie-lines are shown as RTL. Three types of faults(C-G, A-B, and A-B-C-G) were introduced at 23 different locations.

The covariance admits a decomposition in eigenvectors $\mathbf{Q}$ and positive eigenvalues included in the diagonal matrix $\boldsymbol{\Lambda}$, with the form $\boldsymbol{\Sigma}_n = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$. Therefore, the previous expression can be written as

$$\mathbf{e}_n^\top \boldsymbol{\Sigma}_n^{-1}\mathbf{e}_n = \mathbf{e}_n^\top \mathbf{Q}\boldsymbol{\Lambda}^{-1}\mathbf{Q}^\top\mathbf{e} = \mathbf{e}_n^\top \mathbf{Q}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{I}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^\top\mathbf{e} \tag{8}$$

From Eq. (8), it is clear that vector $\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^\top\mathbf{e}$ is a Gaussian random variable with identity covariance matrix, and therefore the whitening process can be written as

$$\tilde{\mathbf{e}}_n = \boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^\top\mathbf{e} \tag{9}$$

as represented in Fig. 3.

This output is normalized and isotropic, so in principle, a simple threshold could be used to detect anomalies. The use of the above-described OC-SVM is justified precisely to avoid the need for a heuristic to adjust this threshold. Instead, we let the SVM construct a multidimensional threshold by applying the OC-SVM criterion.

### 3.3. Fault classification

In order to classify the faults, we use a standard classifier. Since several classes of faults are considered (labeled as CG (Phase C to ground), AB (line to line) or ABCG (three phases to ground) in Section 4.1) classifiers are multiclass. The fault classification section contains a set of classifiers intended to estimate the probability of a fault class given the observation and the hypothesis over each one of the topologies. Notice that the system does not assume that a fault has been detected, so the system detects the classes independently of the decision of the fault detection.

The estimation needs the use of a classifier with a soft output with the properties of a probability density estimation. In this case, this simply means that the output is a real number between 0 and 1. In principle, the most suitable method for this would be a Gaussian process classifier, which is trained with a Bayesian approach and that makes assumptions about the distribution of the labels [54]. Nevertheless, the training uses a square matrix with dimensions equal to the number of training samples. While the previous section uses a GP regression, that system uses less number of input samples as it is explained in the experiment section. Given the quantity of data used during the training, this leads to classifiers with a very high computational burden both in training and in tests. A reasonable approach that is sparse in nature is an SVM classifier with soft output, which is given probabilistic properties by simply adding a sigmoidal output to it.

## 4. Experiments

### 4.1. Grid simulation

Fig. 4 shows the modified IEEE 123 test note feeder bus system [55]. It is known for its convergence stability and capability to model unbalanced lines. It has a nominal voltage of 4.16 kV.

For this experiment, the IEEE 123 bus system was simulated in MATLAB/Simulink 2019b. The simulation model of the IEEE 123 node system in Simulink adopts a variable load profile applied to the model. Similar to all power system simulations, the load and generation are varying through time (customers have different numbers of lights on during the day vs night, and use different amounts of air conditioning in summer vs. winter). The IEEE 123 node uses a simplified model of this where all customer types (residential vs. commercial) follow the same time-series variation load profile, and generation follows a PV generation profile. Faults are then applied at different points in time (hour and day) to create the range of fault sample data.

Four different configurations were simulated with three simulation incidents per day for each configuration. Simulations during each day start at 9 AM, 12 PM, and 6 PM. Each simulation instance starts with the system's normal condition (no-fault), then a fault is applied (the duration of the fault is randomly selected), and finally, the fault is removed and the system operates under post-fault condition. The parameters of the IEEE 123 node are extracted from the https://cmte.ieee.org/pes-testfeeders/resources/ [56]. The model was simulated with the

**Table 1**
Four tie-line based circuit topology configurations.

| Configuration | RTL1 | RTL2 | RTL3 | RTL4 |
|---|---|---|---|---|
| 1 | Closed | **OPEN** | Closed | Closed |
| 2 | Closed | Closed | **OPEN** | Closed |
| 3 | Closed | Closed | Closed | **OPEN** |
| 4 | **OPEN** | Closed | Closed | Closed |

Runge–Kutta discrete solver at a time-step of 50 μs. The collected data included the sequential components of voltage and current measured at the protection device locations. The simulated fault locations are shown in Fig. 4. For each protection device, the datasets were concatenated for all configuration scenarios into one single file with a length of 5.6 million samples. The system was modified to only include the relays in Fig. 4.

It is to note that the circuit reconfiguration through tie lines is a common approach in modern distribution systems to accommodate load transfer when a portion of the system is experiencing an overloading condition. Under this scenario, to avoid damaging the feeders, power is transferred to a feeder with a higher power-carrying capacity. The IEEE 123 system represents a reduced-order model of an actual distribution circuit. To accommodate different circuit configurations, four tie lines were added. However, out of the four tie lines, only one of them can be open. This is a common practice in utilities to ensure that the system stays radial and no part of the system is islanded. The number of circuit configurations depends on the distribution system architecture and the number of tie lines. Depending on the system, having two, three, or four tie lines in a distribution network is a very standard practice in electric power utilities. For example, in [57], the authors have considered four tie lines in their study system.

The test circuit can be operated in four different configurations (Table 1), changed by the status of tie-lines RTL1 to RTL4. Only one of the four tie-line is in the open position for each circuit configuration. The circuit contains 10 protection devices, including the four tie-lines.

Three types of faults were simulated in all configurations, as described in [58], that were applied to the different locations indicated in Fig. 4. The faults are classified as:

1. CG fault: Phase C to ground fault. It is a single line-to-ground (SLG) fault caused by one conductor touching the neutral conductor or contact with vegetation.
2. AB fault: A phase connected to the B phase creates a short circuit (LL). It is line to line type fault where current arcs between the two lines.
3. ABCG fault: Three phases A, B, and C are connected together as well as to the ground. This is a three-phase to-ground fault (3LG).

The data generated consisted of 5.6M samples for each configuration and for each one of the 10 relay locations. 26% of the data contained one of the different faults introduced in the 23 locations specified in Fig. 4. 80% of the faults were of type CG, 15% was AB, and 5% of the faults were ABCG. These distributions are based on the fact that single-line-to-ground faults (e.g., CG) are the most common faults in power grids and their probability is around 80%. Line-to-line faults (e.g., AB) are less probable to occur compared to single-line-to-ground faults (with around 15% probability). Three-line-to-ground faults are the least probable faults in power grids with a probability of 5%. The fault type probability distribution is extracted from [59]. The fault impedances randomly ranged from 0.1 to 5 $\Omega$ and was generated using uniform distribution. The data collected at each relay consisted of the three-phase RMS voltages and currents.

### 4.2. Performance of the CNN-based configuration classifier

Training of the CNN was done using 50% of the total data from the first half of the year (2.8 million samples) and part of this training data was used for validation. At each time instant $t$ and for each one of the features, a window of 100 samples (the present sample plus 99 past ones) is used as an input to each one of the 1-D convolutions of the CNN. The training was done in batches of 32 windows of 100 samples for 10 epochs. The rest of the 50% of the data from the other half of the year was used for testing. The entire data consisting of faults was passed through CNN and the classification results were obtained. The model used Adam optimizer [60] during training that was initialized to minimize the categorical cross-entropy loss, hence aiming for higher accuracy during validation. The Adam optimizer was initialized with a learning rate of $\eta = 0.001$, the exponential decay rate for the first moment (the mean) $\beta_1 = 0.9$, the exponential decay rate for the second moment (the uncentric variance) $\beta_2 = 0.999$, and the small constant for numerical stability $\epsilon = 1e^{-7}$. The final dense layer had four softmax outputs for each of the configurations. An SVM model for circuit topology estimation was trained for comparison purposes. The hyper-parameter tuning was done using the Exhaustive Grid Search method. This search exhaustively generates the optimum values from a grid of a given range of parameters. Here two grids were investigated: one using a linear kernel and the C values were chosen from [1, 10, 100], and the other using a radial basis function (RBF) kernel with width parameter $\gamma$. The values of the SVM parameter C were chosen from [1, 10, 100] and the $\gamma$ values were [0.001, 0.01, 0.1]. The optimum parameter was found to be a linear kernel with C=100. The criterion for selecting the optimum parameter was to maximize the validation accuracy. The SVM model was trained on 50% of the data from the first half of the year. A section of the training data was used for validation. The bandwidth of the data is very narrow. In particular, the input data is constant for long periods of time. Then for training, the data can be subsampled with no loss of information. Additionally, SVM does not extensively require huge number of samples for training. The data was downsampled by a factor of 300 according to its bandwidth. Therefore, a total of 9500 samples were used for the training of the SVM. The rest of the data comprising of 2.8 million samples were used for testing.

Fig. 5 shows the comparison of the CNN and the SVM in the topology classification for all relays, as presented in [19]. The CNN achieved a classification rate very close to 100% for all relays, while the SVM was not able to achieve good performance in many relays, particularly in relays R2 and R3. The horizontal axis of the left pane in Fig. 6 represents the calibration histogram of the predicted class probabilities of the CNN, this is, the predicted probabilities of a given class in intervals of 0.1. The vertical axis represents the fraction of samples that actually belong to that class (measured probability), averaged for all classes. The graphic shows that the predicted probability and the actual one are accurate, except for the range between 0.4 and 0.5, where the actual probability is much lower than the actual. Since the predicted probability is less than 0.5, this actually produced a probability of error much lower than the predicted. This is due to the fact that the experiment actually produced only a few samples with patterns similar to the ones that produce a predicted around 0.5, which results in a poor response in that area.

Fig. 6 shows, in logarithmic units, the normalized histograms of the probability of configuration $P(C_j|x)$, when the actual topology corresponds to configuration 1. It can be seen that in most cases the probability of configuration 1 is high, but in many cases, the estimation is too conservative. Indeed, errors can only be observed when the probabilities are between 0.3 and 0.5. This is, when the configuration probability is high, the probability of error is low, so the decision can be trusted, as desired in this kind of classifier.

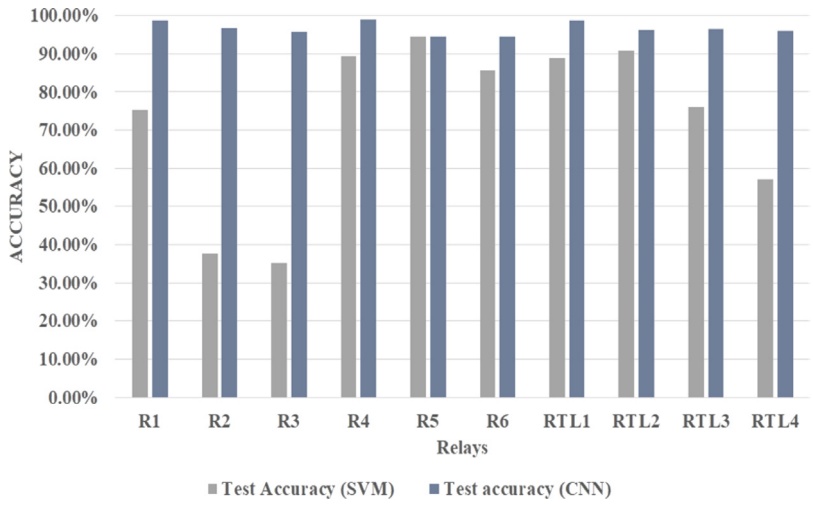


**Fig. 5.** Comparison of test accuracy for topology estimation using SVM and CNN [19].

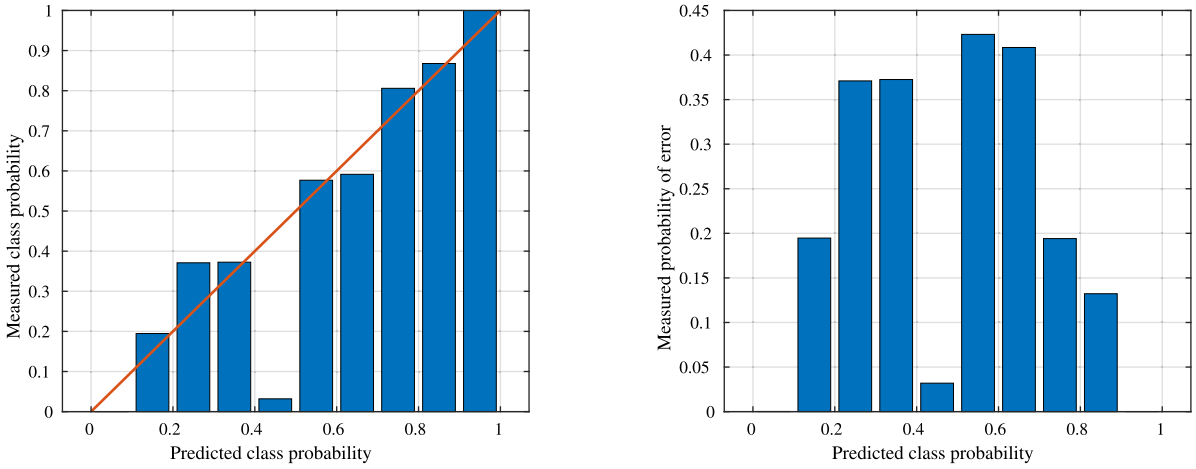


**Fig. 6.** Left: Histogram showing the measured probability of a configuration class as a function of the predicted probability of that class given an observed voltage and current. Right: Histogram of the probability of error in the prediction of a class as a function of the predicted probability of that class.

### *4.3. Performance of the SVM for fault classification*

In the case of fault classification, two different SVM classifiers were trained: a four-class SVM that classifies the input data into normal data (NF) and data with three fault types (CG, AB, and ABCG) and a three-class SVM to classify only the faults. The training procedure was exactly the same as the one described in Section 4.2 for the topology classification SVM. The three-class SVM is designed to be used in combination with the fault detector described in Section 3.3.

Fig. 7 shows the confusion matrices of both systems. It can be seen that the four-class classifier has reasonable performance, It is able to detect non-faulty samples with a 99.63% of detection rate, and it has a similar performance in the classification of faults. If a three-class classifier is used for the classification of faulty samples only, the classification performance is slightly increased. Fig. 8 shows a more detailed perspective, as presented in [19]. Here, four different classifiers are trained with data from each one of the configurations. In these classifiers, specialized in each configuration, we measured their error rate for each one of the relays. It can be seen that when a 3-class classifier for faulty samples only is used, the performance of the classifier increases due to the reduced complexity of this classifier with respect to the four class one. The probabilities obtained with this classifier, nevertheless, are not accurate, since the outputs are saturated towards one or zero. This effect is due to the fact that the multiclass classifier is not trained with a probabilistic criterion, but using the maximum margin criterion. Therefore, a calibration histogram is not provided.

When a classifier is trained with data corresponding to each one of the configurations, the configuration classifier described above is needed to select the fault classifier corresponding to the current topology. Besides when a 3-class fault classifier only is used, a fault detector is needed to determine whether the fault has happened. We test the three systems together in the next subsection.

### *4.4. Combined performance of the three subsystems*

The fault detector consists of two stages that are trained in a non-supervised way, that is, all data needed to train the structure consists of available observations of current and voltage, where the information about the presence or absence of a fault was not provided. Nevertheless, for the structure to function properly, it is assumed that a fault is a rare event, thus having a very low likelihood. Therefore, both structures were trained with normal data only, which assumes that in a real situation the presence of faults is low enough for them to be ignored by the GP section. This section is trained in order to estimate the three values of

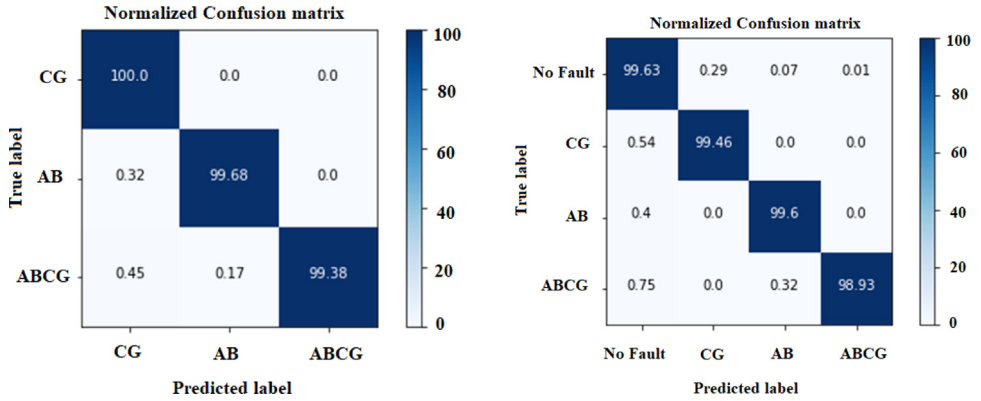


**Fig. 7.** Confusion matrix showing the percentage of detection for *Left:* fault classification trained with normal data along with faulted data (using a four class SVM) and *right*: with faults only data (using a three-class SVM) averaged for all relays and configurations.

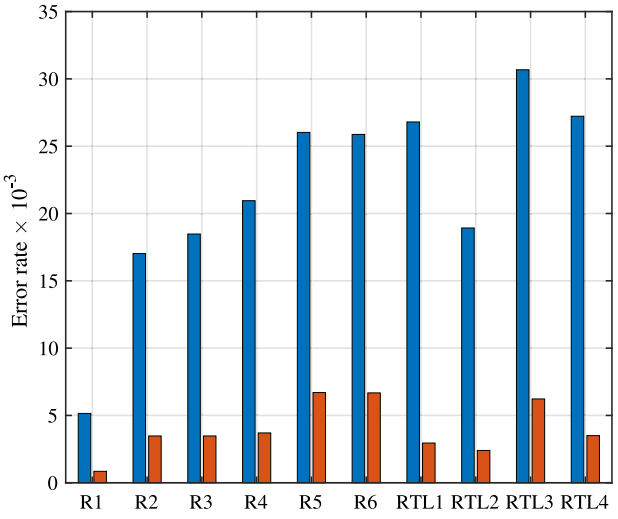


**Fig. 8.** Test error rate comparison for fault classifier with normal data (four class SVM, in blue) and with faults only (three class SVM, in red) for all the relays averaged for all configurations [19]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the current from the three values of the voltage. The GP produces predictive means of the prediction and the predictive covariance (for the case of MTGP) or the predictive variance (for the case of independent GPs). This is used to normalize the error, assumed to be Gaussian into a unitary zero mean distribution. The normalized errors are used to train the OC-SVM that detects the errors. Thus, during the training, no faults are expected but, if they are present, they will be detected by the OC-SVM.

The GPs are linear and their hyperparameters are optimized through the maximization of the training error likelihood. The OC-SVM uses square exponential kernels, with a width hyperparameter $\sigma^2$. The OC-SVM has a hyperparameter $\nu$. Since in a real situation there are not faults available for cross-validation, the values are chosen heuristically. Since the input to the OC-SVM are unitary Gaussians (identity covariance or variance equal to 1), the kernel parameter is simply fixed to $\sigma^2 = 1$. Parameter $\nu$ is arbitrarily fixed to 0.1 This parameter fixes a lower bound on the fraction of the support vectors, which will be in this case equal to 10% of the training data. The fault detector is trained with $10^5$ consecutive samples and then tested with $10^5$ consecutive samples, corresponding to about 5 days and 18 h.

**Table 2**
Final fault classification error rates ($\times 10^{-3}$) of the proposed combined metastructure in Eq. (2).

| | R1 | R2 | R3 | R4 | R5 | R6 | RTL1 | RTL2 | RTL3 | RTL4 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_e$ | 6.2 | 65.5 | 84.1 | 39.9 | 10.6 | 36.0 | 25.9 | 45.5 | 26.1 | 24.5 |

**Table 3**
Probabilities of false alarm $p_{fa}$ and of no detection $1-p_d$ ($\times 10^{-3}$) of the proposed structure.

| | R1 | R2 | R3 | R4 | R5 | R6 | RTL1 | RTL2 | RTL3 | RTL4 |
|---|---|---|---|---|---|---|---|---|---|---|
| $p_{fa}$ | 7.9 | 4.1 | 4.6 | 6.4 | 8.8 | 7.5 | 7.5 | 7.1 | 7.7 | 7.0 |
| $1-p_d$ | 1.3 | 76.4 | 176 | 1.9 | 1.3 | 1.8 | 1.8 | 1.7 | 1.6 | 1.5 |

The three subsystems are tested by implementing (2), where each subsystem provides a probabilistic estimate of the configuration, fault class, and fault event.

Table 2 shows the fault classification error rates for all relays and all configurations of the combined metastructure when the fault detector has been tested with $5 \times 10^4$ consecutive samples, corresponding to about 3 days after the training. The minimum error was of $6.2 \times 10^{-3}$ in relay R1, while the worse cases were reported in relays R2 and R3, with values of $65.5 \times 10^{-3}$ and $84.1 \times 10^{-3}$. These are the relays where the configuration is classified with the highest probability.

The fault detection loss or $1-p_d$, where $p_d$ is the probability of detection, and the probability of false alarm $p_{fa}$ for the metastructure, obtained by the implementation of Eq. (1). The probability of false alarm is less than $10^{-2}$ in all cases, while the probability of loss is less than $2 \times 10^{-3}$ in all cases except for relays R2 and R3, which are the most difficult ones, as it has been seen while testing the configuration classification. This decreased performance has here the consequence of a lower probability of detection (see Table 3).

Fig. 9 (left) shows the fault probability calibration histogram or the measured probability of fault given the input sample as a function of the predicted probability of fault given that sample. The histogram has been constructed by grouping the events that produced probabilities in the ten intervals $p$ and $p = 0.1$ for $0 \leq p \leq 1$. The probability of fault of the events in all these groups is represented in the histogram, which is normalized between zero and 1. Similarly, the right graph of the figure shows the error probability relative to each one of these groups. The probability estimation is less accurate than the class probabilities of the CNN because the optimization of the SVM does not contain a probabilistic criterion. Nevertheless, the prediction is reasonable, except in the interval between 0.4 and 0.5, with an estimation below the actual value, which increases the error rate above 0.5.

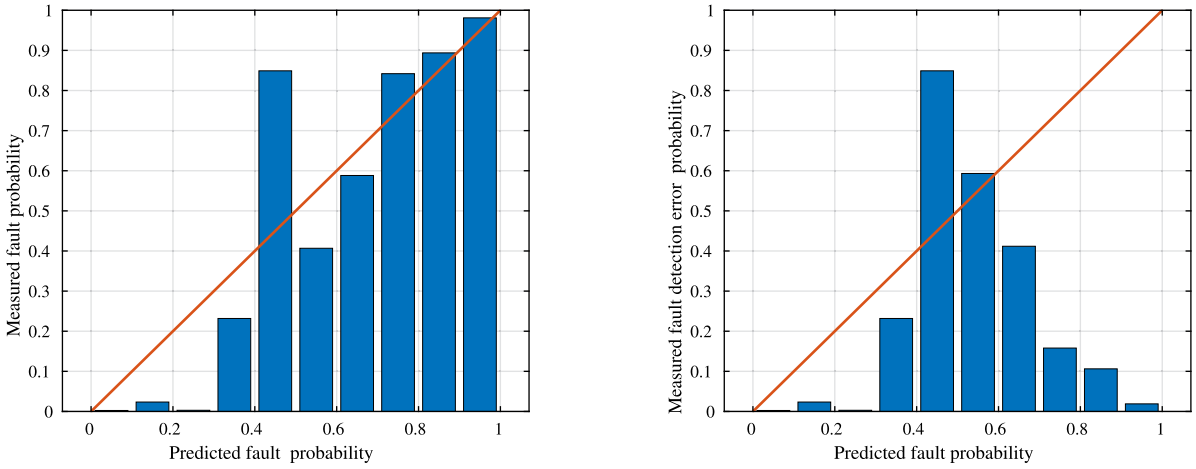


**Fig. 9.** Left: Histogram of the measured probability of fault as a function of the predicted probability of fault. Right: the measured probability of error in fault detection as a function of the predicted probability of fault.

This phenomenon is an accuracy due to the lack of events in this interval, and it produces an effect analogous to the one in the calibration histogram of the CNN (Fig. 6).

Fig. 10 shows the performance of the fault detector as the length of the sample sequence increases. The experiment is an example of the fact that the fault classifier must be periodically retrained (or retrained online). It corresponds to a single training followed by a test with a sequence of $10^5$ samples. The test is repeated for data with the four possible configurations, and the results are averaged. The graphs represent the accumulated probability of loss ($1 - Pd$) (left) and the accumulated probability of false alarm ($P_{fa}$). Therefore, there is a peak when a loss or a false alarm is recorded, and then, as time increases, the probabilities decrease until a loss or a false alarm is registered again. Due to the non-stationary nature of the data (as explained in Section 4.1), its probability distribution changes with time. There is a moment (around sample number $6 \times 10^4$) where a single fault produces a burst of detection errors in two of the relays that significantly increases the probability of loss. A similar effect occurs for 7 of the relays in the probability of false alarm. There are two more such events later that increase the probabilities again. One such event is illustrated in the left pane of Fig. 11. The probability of loss is all the time higher for relays R2 and R3. Here it can be seen that, conversely, the corresponding probability of a false alarm is lower, which suggests that a manual readjustment of the threshold (automatically adjusted by the OC-SVM) could improve the performance in relays R2 and R3.

Regarding to the relatively high probability of false alarms (1 out of 100 cases), these false alarms are mostly due to the transient produced by the simulated IEEE 123 bus system at the output of the fault detector when the fault is removed. An example of this behavior can be seen in Fig. 11, which shows a transient while a fault of type 1 is disabled, and the fault detection data is taken from relay RTL3 in topology configuration 1. If these transients can be ignored, since they happen once a fault has been already detected, then the probability of a false alarm decreases dramatically, as it can be seen in Table 4 and Fig. 12. While the minimum $p_{fa}$reported in the graph is zero, actually these measures simply mean that zero false alarms were reported during the corresponding intervals. The table shows the probability of a false alarm for a sequence of $5 \times 10^4$ samples, which means that the measured false alarm rate is less than $2 \times 10^{-5}$.

**Table 4**
Probabilities of false alarm $p_{fa}$ and of no detection $1-p_d$ ($\times 10^{-3}$) of the proposed structure when the transients at the end of the fault period are ignored.

| | R1 | R2 | R3 | R4 | R5 | R6 |
|---|---|---|---|---|---|---|
| $p_{fa}$ | $> 0.002$ | $> 0.002$ | 0.32 | $> 0.002$ | $> 0.002$ | $> 0.002$ |
| $1 - p_d$ | 1.3 | 76.4 | 176.5 | 1.9 | 1.3 | 1.8 |
| | RTL1 | RTL2 | RTL3 | RTL4 | | |
| $p_{fa}$ | 0.01 | $> 0.002$ | $> 0.002$ | $> 0.002$ | | |
| $1 - p_d$ | 1.8 | 1.7 | 1.6 | 1.5 | | |

### 4.5. Computational complexity of each subsystem

Each of the subsystems was trained individually using the system with RAM memory of 16 GB and NVIDIA GeForce GTX 1060 GDDR5 GPU with graphic RAM memory of 6 GB. CNN-based topology estimation system was implemented using Keras library in Tensorflow. The total training time for this algorithm was around 2.5 h for the data using 2.8 million samples. The test time per sample is around 0.4 ms. As for the SVM classifier for the Fault classification, lesser training samples resulted in smaller training duration and the test time was in the order of microseconds. The implementation of this classifier was done using the Scikit-learn library in python. The fault detection subsystem made use of two different modules, i.e., the Multi-task Gaussian process and OC-SVM. The MTGP implementation was implemented using the GPyTorch library in PyTorch and the OC-SVM module was implemented using the Scikit-learn library in python. The overall training time for the entire module was in the range of 7–10 min and the test time was computed to be in the order of $10^-5$ seconds. The complete implementation and code files can be accessed through link https://github.com/Fviramontes8/LAMP to GitHub repository.

## 5. Conclusion

This work presents a meta-structure that uses different machine learning methods to simultaneously estimate the topology of the grid and detect and classify faults. In our approach, a CNN is in charge of estimating the probability that a given configuration is present in the network, and, independently, a set of classifiers, by assuming that there is a fault in the bus, classifies it for any possible configuration of the bus, by providing conditional probabilities of the type of fault. A third subsystem uses a predictive

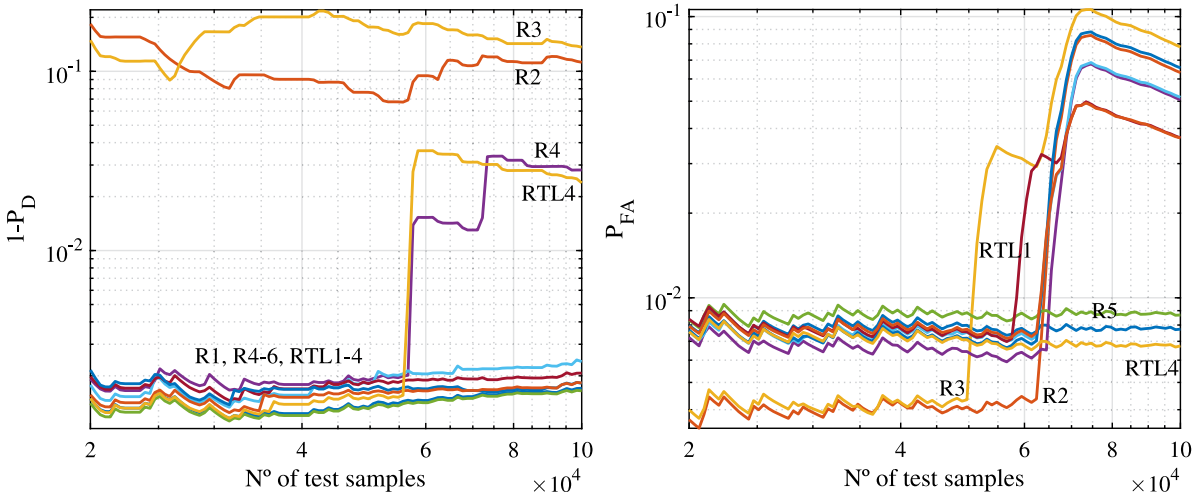


**Fig. 10.** Left: Probability of loss $(1 - P_d)$ for all relays as a function of the length of the sequence of test samples. Right: Probability of false alarm for all relays. When the number of samples approaches $6 \times 10^4$, corresponding to about 3 days, the performance of the detector decreases.

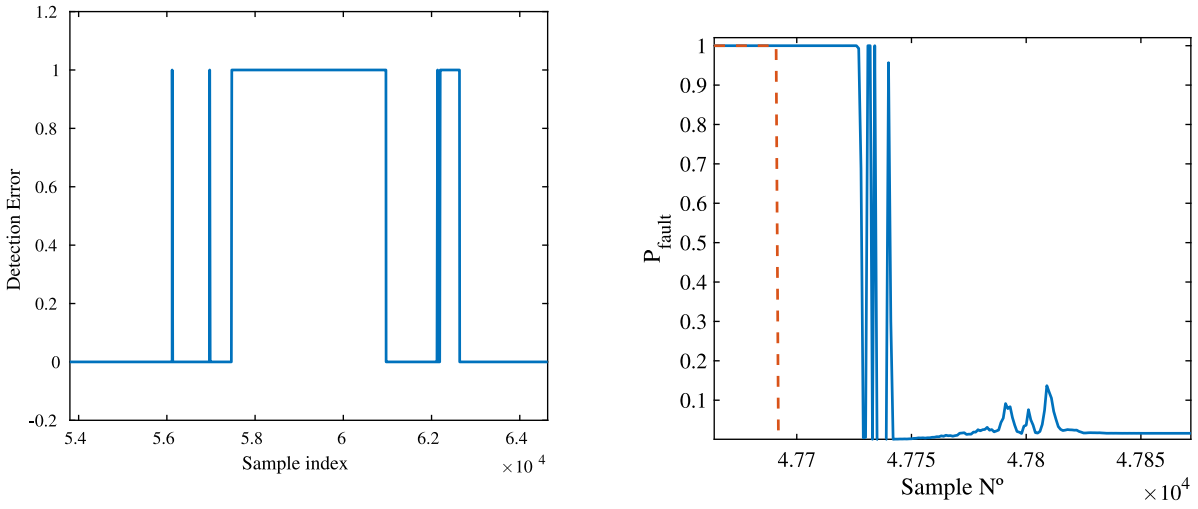


**Fig. 11.** Left: Burst of errors produced in Relay 4 after testing the fault detectors for about $6 \times 10^4$ samples. Right: Example of the transient that appears at the fault detector output when the fault is removed from the system. The observation corresponds to relay RTL3, the configuration is 1 and the fault is type 1. The dashed line is 1 during the fault event and zero otherwise.

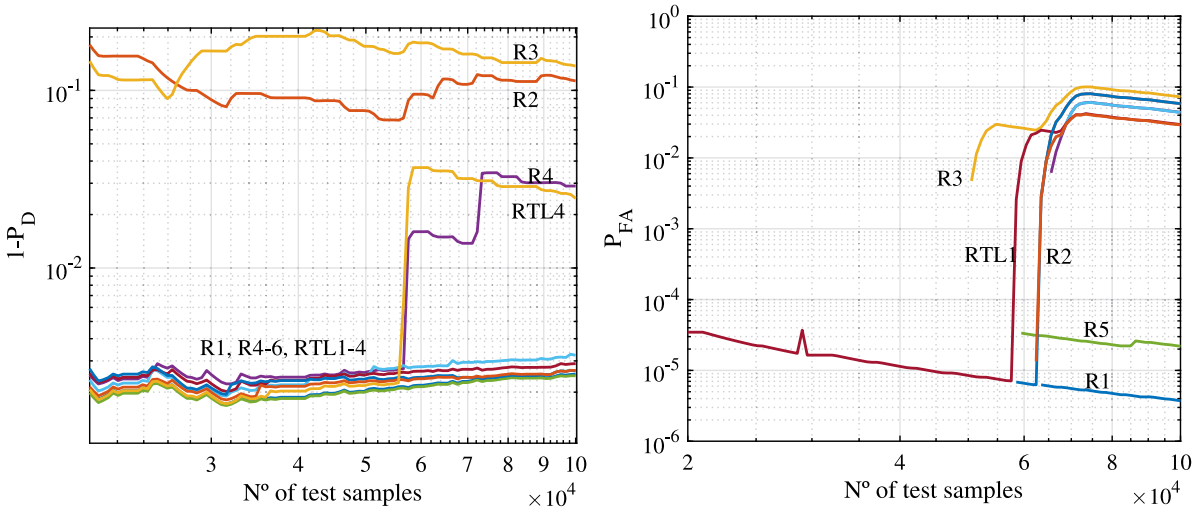


**Fig. 12.** Probability of loss and probability of false alarm in fault detection when the transient after the error is not accounted for. While the probability of loss is about the same as in Fig. 10, the probability of false alarm is dramatically lower when the number of test samples is less than $5 \times 10^4$.

approach to indirectly determine the likelihood of a sample and then estimates the probability of that a fault is present, conditional to each configuration. The combination of the three outputs provides the marginal probability of fault class, probability of fault, and probability of configuration. The experimental data was simulated in Simulink® using the modified IEEE 123 bus. The data contained four different circuit configurations and we simulated three different faults in this data, at different positions of the bus. This is a continuation of work [19], where we presented a non-probabilistic methodology for topology estimation and fault classification. In this continuation, we use a probabilistic approach, introduce the fault detection methodology and provide the formulation for the fusion of all subsystems. The results show a high probability of fault classification and detection and a very low probability of false alarms. In our future work, we will assess the sensitivity of this type of structure against the signal-to-noise and interference ratio. We will also test different non-supervised structures for fault detection based on deep learning. Finally, the sensitivity of the methodology against non-stationary signals and adaptive approaches need to be tackled.

## CRediT authorship contribution statement

**Aswathy Rajendra Kurup:** Conceptualization, Methodology, Software, Visualization. **Adam Summers:** Data curation, Writing – original draft. **Ali Bidram:** Supervision, Investigation, Validation, Funding acquisition. **Matthew J. Reno:** Writing – reviewing & editing. **Manel Martínez-Ramón:** Conceptualization, Writing – original draft, Supervision.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

I have shared link to the sample data/code in the manuscript.

## Acknowledgments

This material is based upon work supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, United States and the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number 36533. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly-owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Partly supported by National Science Foundation (NSF) EPSCoR, United States grant number OIA-1757207 and the King Felipe VI endowed Chair of the UNM, United States. The authors would like to thank the UNM CARC, supported in part by NFS, for providing the high-performance computing and large-scale storage resources used in this work

## References

[1] P.M. Anderson, Power System Protection, McGraw-Hill New York, 1999.
[2] H.F. Albinali, A. Meliopoulos, Resilient protection system through centralized substation protection, IEEE Trans. Power Deliv. 33 (3) (2018) 1418–1427.
[3] J. Liu, J. Duan, H. Lu, Y. Sun, Fault location method based on EEMD and traveling-wave speed characteristics for HVDC transmission lines, J. Power Energy Eng. 3 (4) (2015) 106–113.
[4] Y. Seyedi, H. Karimi, Coordinated protection and control based on synchrophasor data processing in smart distribution networks, IEEE Trans. Power Syst. 33 (1) (2018) 634–645.
[5] D. Sikeridis, A. Bidram, M. Devetsikiotis, M.J. Reno, A blockchain-based mechanism for secure data exchange in smart grid protection systems, in: 2020 IEEE 17th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2020, pp. 1–6.
[6] J. Seuss, M.J. Reno, R.J. Broderick, S. Grijalva, Determining the impact of steady-state PV fault current injections on distribution protection, Sandia National Laboratories, SAND2017-4955, 2017.
[7] M.N. Alam, Adaptive protection coordination scheme using numerical directional overcurrent relays, IEEE Trans. Ind. Inform. 15 (1) (2018) 64–73.
[8] L. Che, M.E. Khodayar, M. Shahidehpour, Adaptive protection system for microgrids: Protection practices of a functional microgrid system, IEEE Electrif. Mag. 2 (1) (2014) 66–80.
[9] R. Moxley, F. Becker, Adaptive protection—What does it mean and what can it do? in: 2018 71st Annual Conference for Protective Relay Engineers (CPRE), IEEE, 2018, pp. 1–4.
[10] S. Venkata, M.J. Reno, W. Bower, S. Manson, J. Reilly, G.W. Sey Jr., Microgrid protection: Advancing the state of the art, Sandia National Laboratories, SAND2019-3167, Albuquerque, 2019.
[11] B. Poudel, D.R. Garcia, A. Bidram, M.J. Reno, A. Summers, Circuit topology estimation in an adaptive protection system, in: North American Power Symposium, 2021.
[12] D.R. Garcia, B. Poudel, A. Bidram, M.J. Reno, Substation-level circuit topology estimation using machine learning, in: IEEE Innovative Smart Grid Technologies, ISGT, 2022.
[13] I. Hwang, S. Kim, Y. Kim, C.E. Seah, A survey of fault detection, isolation, and reconfiguration methods, IEEE Trans. Control Syst. Technol. 18 (3) (2010) 636–653.
[14] A. Monticelli, Electric power system state estimation, Proc. IEEE 88 (2) (2000) 262–282.
[15] A. Abur, D. Shirmohammadi, C. Cheng, Estimation of switch statuses for radial power distribution systems, in: Proceedings of ISCAS'95-International Symposium on Circuits and Systems, Vol. 2, IEEE, 1995, pp. 1127–1130.
[16] C. Francis, V. Rao, R.D. Trevizan, M.J. Reno, Topology identification of power distribution systems using time series of voltage measurements, in: 2021 IEEE Power and Energy Conference At Illinois, PECI, IEEE, 2021, pp. 1–7.
[17] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.
[18] I. Hwang, S. Kim, Y. Kim, C.E. Seah, A survey of fault detection, isolation, and reconfiguration methods, IEEE Trans. Control Syst. Technol. 18 (3) (2010) 636–653.
[19] A.R. Kurup, M. Martínez-Ramón, A. Summers, A. Bidram, M.J. Reno, Deep learning based circuit topology estimation and fault classification in distribution systems, in: 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), 2021, pp. 01–05.
[20] G.N. Korres, P.J. Katsikas, Identification of circuit breaker statuses in WLS state estimator, IEEE Trans. Power Syst. 17 (3) (2002) 818–825.
[21] V. Kekatos, G.B. Giannakis, Joint power system state estimation and breaker status identification, in: 2012 North American Power Symposium, NAPS, IEEE, 2012, pp. 1–6.
[22] R. Dobbe, W. van Westering, S. Liu, D. Arnold, D. Callaway, C. Tomlin, Forecasting-based state estimation for three-phase distribution systems with limited sensing, 2018, arXiv preprint arXiv:1806.06024.
[23] J. Xie, A.P.S. Meliopoulos, B. Xie, Transmission line fault classification based on dynamic state estimation and support vector machine, in: 2018 North American Power Symposium, NAPS, 2018, pp. 1–5.
[24] A.S. Zamzam, X. Fu, N.D. Sidiropoulos, Data-driven learning-based optimization for distribution system state estimation, IEEE Trans. Power Syst. 34 (6) (2019) 4796–4805.
[25] G. Cavraro, R. Arghandeh, K. Poolla, A. Von Meier, Data-driven approach for distribution network topology detection, in: 2015 IEEE Power & Energy Society General Meeting, IEEE, 2015, pp. 1–5.
[26] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, L. Schenato, Identification of power distribution network topology via voltage correlation analysis, in: 52nd IEEE Conference on Decision and Control, 2013, pp. 1659–1664, http://dx.doi.org/10.1109/CDC.2013.6760120.
[27] J. Zhang, Y. Wang, Y. Weng, N. Zhang, Topology identification and line parameter estimation for non-pmu distribution network: A numerical method, IEEE Trans. Smart Grid 11 (5) (2020) 4440–4453.

[28] M. Goldstein, A. Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, in: KI-2012: Poster and Demo Track, Citeseer, 2012, pp. 59–63.
[29] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.
[30] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 427–438.
[31] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2002, pp. 15–27.
[32] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[33] A. Lazarevic, V. Kumar, Feature bagging for outlier detection, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 157–166.
[34] C.C. Aggarwal, S. Sathe, Theoretical foundations and algorithms for outlier ensembles, Acm Sigkdd Explor. Newslett. 17 (1) (2015) 24–47.
[35] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, 2014, pp. 4–11.
[36] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[37] J. Chen, S. Sathe, C. Aggarwal, D. Turaga, Outlier detection with autoencoder ensembles, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 90–98.
[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[39] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, X. He, Generative adversarial active learning for unsupervised outlier detection, IEEE Trans. Knowl. Data Eng. 32 (8) (2019) 1517–1528.
[40] J. Shawe-Taylor, N. Cristianini, et al., Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[41] J. Hardin, D.M. Rocke, Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator, Comput. Statist. Data Anal. 44 (4) (2004) 625–638.
[42] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, Tech. Rep., Dept. of Electrical and Computer Engineering, Miami University, Coral Gables, FL, 2003.
[43] D.M. Tax, R.P. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[44] H. Livani, C.Y. Evrenosoglu, A machine learning and wavelet-based fault location method for hybrid transmission lines, IEEE Trans. Smart Grid 5 (1) (2013) 51–59.
[45] M. Shafiullah, M. Ijaz, M. Abido, Z. Al-Hamouz, Optimized support vector machine & wavelet transform for distribution grid fault location, in: 2017 11th IEEE International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), IEEE, 2017, pp. 77–82.
[46] H.T. Thom, C. Ming-Yuan, V.Q. Tuan, A novel perturbed particle swarm optimization-based support vector machine for fault diagnosis in power distribution systems, Turk. J. Electr. Eng. Comput. Sci. 26 (1) (2018) 518–529.
[47] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, (2) MIT press Cambridge, 2016.
[48] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[49] Y. LeCun, Y. Bengio, et al., Convolutional networks for images, speech, and time series, in: The Handbook of Brain Theory and Neural Networks, Vol. 3361, (10) 1995, p. 1995.
[50] J.L. Rojo-Álvarez, M. Martínez-Ramón, M. de Prado-Cumplido, A. Artés-Rodríguez, A.R. Figueiras-Vidal, Support vector method for robust ARMA system identification, IEEE Trans. Signal Process. 52 (1) (2004) 155–164.
[51] S. Salcedo-Sanz, J.L. Rojo-Álvarez, M. Martínez-Ramón, G. Camps-Valls, Support vector machines in engineering: an overview, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 4 (3) (2014) 234–267.
[52] J. Mercer, Functions of positive and negative type, and their connection the theory of integral equations, Philos. Trans. R. Soc. Lond. Ser. A 209 (441–458) (1909) 415–446.
[53] M.A. Aizerman, E.M. Braverman, L.I. Rozonoer, Theoretical foundations of potential function method in pattern recognition, Autom. Remote Control 25 (6) (1964) 917–936.
[54] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, MIT press Cambridge, MA, 2006.
[55] W.H. Kersting, Radial distribution test feeders, IEEE Trans. Power Syst. 6 (3) (1991) 975–985.
[56] K.P. Schneider, B.A. Mather, B.C. Pal, C.-W. Ten, G.J. Shirek, H. Zhu, J.C. Fuller, J.L.R. Pereira, L.F. Ochoa, L.R. de Araujo, R.C. Dugan, S. Matthias, S. Paudyal, T.E. McDermott, W. Kersting, Analytic considerations and design basis for the IEEE distribution test feeders, IEEE Trans. Power Syst. 33 (3) (2018) 3181–3188.
[57] Z. Wang, K. Wang, J. Xu, Topology identification for distribution network driven by diverse measurement data, in: 2021 3rd International Conference on Electrical Engineering and Control Technologies, CEECT, IEEE, 2021, pp. 268–273.
[58] C.B. Jones, A. Summers, M.J. Reno, Machine learning embedded in distribution network relays to classify and locate faults, in: 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, ISGT, IEEE, 2021, pp. 1–5.
[59] T.A. Short, Electric Power Distribution Handbook, CRC Press, 2003.
[60] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.