Neural Computing and Applications (2024) 36:1741–1759
https://doi.org/10.1007/s00521-023-09155-y

ORIGINAL ARTICLE

Fault distance estimation for transmission lines with dynamic regressor selection

Leandro A. Ensina[1] · Luiz E. S. de Oliveira[1] · Rafael M. O. Cruz[2] · George D. C. Cavalcanti[3]

Received: 22 August 2023 / Accepted: 20 October 2023 / Published online: 15 November 2023

Abstract
The transmission line is one of the most crucial electric power system components, demanding special attention since they are subject to failures that can cause disruptions in energy supply. In this scenario, the fault location emerges as a fundamental task, providing an approximate position where the failure occurred in the line. This paper presents a method for fault location using a novel dynamic regressor selection (DRS) framework, in which we introduce embedded normalization, incorporating the data scaling process inside the framework. The DRS aims to select the most accurate models from a pool of regressors to predict the fault distance. Moreover, since there is a lack of a public dataset, this paper presents and makes a new fault dataset available to the scientific community with several features extracted from current and voltage signals of representative failure events. In our experiments, we demonstrate the importance of this embedded normalization as well as the significance in the variation of critical hyperparameters of the DRS strategy, such as the distance metric used to define the region of competence and the criterion to select the best regressors from the pool of predictors. This work also presents the definition of the oracle concept in DRS, which represents the ideal predictor selection scheme. The results demonstrate the effectiveness of the proposed method with a mean error of 0.7086 km ± 0.0068 km, representing 0.1712% ± 0.0016% of the transmission line extension (414 km).

Keywords Ensemble · Dynamic selection · Fault analysis · Power transmission

1 Introduction

Transmission lines are an essential part of the electric power system, enabling the efficient transport of large amounts of electricity over long distances. However, transmission lines are subject to faults that can cause disruptions in the flow of electricity and potentially damage the equipment. A fault refers to a disruption or disturbance in the normal flow of electricity in the components of a power system, such as the increase in the current flow to one or more phases [70]. Short-circuit faults can be divided into asymmetrical and symmetrical. Asymmetrical failures comprise line-to-ground (LG) and line-to-line faults, with the involvement or not of the ground (LL and LLG), while symmetrical ones include three-phase failures (LLL) [55]. Several factors can result in a transmission line fault, such as insulation failures, short circuits caused by birds, trees, or fire under the lines, in addition to weather conditions like rain, snow, wind, and lightning [2, 21].

In this scenario, it is possible to use voltage and current signals collected by equipment like digital fault recorders or protective relays for failure diagnosis. In particular, fault location is a critical aspect of transmission line maintenance and repair, as it enables power companies to quickly and accurately identify the location of a fault and dispatch a repair staff to fix the issue. The purpose of the fault

✉ Leandro A. Ensina
leandro.ensina@ufpr.br

Luiz E. S. de Oliveira
luiz.oliveira@ufpr.br

Rafael M. O. Cruz
rafael.menelau-cruz@etsmtl.ca

George D. C. Cavalcanti
gdcc@cin.ufpe.br

[1] Department of Informatics (DInf), Federal University of Paraná (UFPR), Curitiba, PR, Brazil

[2] École de Technologie Supérieure (ÉTS), University of Quebec, Montreal, QC, Canada

[3] Centre of Informatics (CIn), Federal University of Pernambuco (UFPE), Recife, PE, Brazil

location is to estimate the approximate position on the transmission line where the failure occurred.

The literature shows different strategies for fault analysis in transmission lines, which can be classified into three groups [32, 45, 69]: traveling waves, impedance-based, and Artificial Intelligence (AI). However, conventional approaches (traveling waves and impedance-based) demonstrate some notable disadvantages [32, 44, 69]. The traveling wave technique is complex, requiring a high sampling rate and high computational cost [27, 60, 63], while the impedance-based approach can be affected by the variation of the fault parameters, mainly for faults with high resistances [27, 40].

In turn, AI-based methods require a significant amount of data for training, representing its main drawback [24, 55]. Consequently, a crucial problem in the literature is the lack of public datasets for transmission line failures. Generally, each author simulates their fault events to evaluate their methods, but these data are not publicly available. The UFPAFaults database provided by Morais et al. [47] is an exception. However, this dataset does not allow fault location, limiting only to the fault type classification.

In this work, we make a new fault database available to the scientific community with several features extracted from voltage and current signals. This dataset contains a total of 168,000 fault events, with voltage and current data from a single terminal of the considered transmission line: 500 kV, 414 km, 60 Hz. Therefore, it enables researchers to evaluate their approaches for fault classification and location.

Nowadays, methods based on AI have gained more attention, mainly due to their high performance and adaptability to different fault conditions and parameters [11, 24, 60]. Mishra and Ray [45] and Mukherjee et al. [51] presented comprehensive reviews about fault location, reporting that most related works use individual (monolithic) algorithms, especially support vector regression (SVR) and neural networks.

However, several studies have demonstrated the advantages and performance superiority of ensemble models over individual ones for classification and regression tasks for different areas [15, 42]. The use of different regressors, for example, can result in different predictions due to the specific properties of each predictor and, when considered together, can result in more accurate and precise performances [38, 65].

Ensemble learning typically has three phases [15, 42]: **Generation phase**—responsible for training a pool of base models. The ensemble is homogeneous when the same induction algorithm is used for all models. Otherwise, it is named heterogeneous; **Selection phase**—responsible for selecting the best predictors in the ensemble, which can be static or dynamic. In the former, the same subset of models is used to predict the label for all test samples. In the latter, a single model or a subset of predictors from the pool is selected on the fly for each unknown example; **Combination phase**—responsible for combining the predictions when the last phase returns more than one model.

In this scenario, a growing topic in the ensemble approach is dynamic regressor selection (DRS) [48, 57], in which a subset of base models is selected from the pool based on a particular rule for each new test pattern to be predicted. In dynamic selection techniques, it is expected that different models are experts in a specific region of the feature space, a.k.a. Region of Competence (RoC) [16].

The literature has already demonstrated that DRS algorithms can benefit from different competence measures to assess the performances of the regressors [49]. However, other aspects should be explored as well since they can contribute to a performance gain. In this context, we assume that DRS algorithms can also be benefited from the use of distinct distance measures and criteria to select the best models based on their performance in the RoC. This hypothesis is supported by some works which have already shown for the classification problem that the RoC refinement reverberates in the improvement of the prediction system [13, 14, 67].

In response, we propose in this work a new method for fault location based on a novel DRS framework with three main characteristics: variation of the distance metric used to define the RoC; variation of the criterion to select the best regressors from the ensemble; and the incorporation of the data scaling process inside the DRS technique (embedded normalization). An important process included in our DRS framework and must be highlighted is the embedded normalization, which scales[1] the test pattern and the validation set together individually for each new example to define the RoC. The intuition for this online process is related to the fact that distance-based algorithms predominantly used to determine the RoC, such as the k-Nearest Neighbors (kNN), are sensitive to the scale of the data, which can affect the examples returned to compose this region. In other words, features with discrepant scales will dominate the others and may lead to an imperfect RoC. At least at this moment, tuning hyperparameters and evaluating multiple configurations automatically by the DRS framework itself is not part of the method and should be performed manually by the researchers.

Thus, before executing the normalization process, we added the test pattern inside the original validation set and then conducted the normalization of this set, being removed from the validation set right after the data scaling.

[1] In this paper, we assume the terms scaling and normalization as equivalents.

This process is carried out individually for each new example and allows us to obtain a RoC with samples more similar to the test pattern and, consequently, achieve better results of the DRS to estimate the fault position. To the best of our knowledge, this is the first work that applies the DRS strategy for fault location.

It is crucial to emphasize that the normalization process plays an essential role in adequate data such that every feature varies within the same range. Amorim et al. [3] presented a comprehensive set of experiments demonstrating the importance of normalization for classification performance considering several algorithms. In the present study, we also carried out some experiments demonstrating its impact on the performance regarding the regression task (fault location), specifically assessing the normalization in our DRS technique for different scenarios.

In order to determine a reference to how good can be the performance of a DRS method, we introduce the concept of oracle in DRS, widely used for the classification task [15]. The oracle is an abstract strategy that always selects the model that predicts the correct label for the given query sample if such a model exists [15, 39]. In other words, it represents the best possible performance that the ensemble can attain, representing the ideal predictor selection scheme. In addition to the oracle of the whole ensemble, we also assessed the oracle only among the selected models by our DRS scheme.

The contributions of this paper are: (1) the development and the availability of a new dataset with several features of fault events for a transmission line; (2) a fault location method using a novel DRS framework, which introduces an embedded normalization; (3) the evaluation of the performance of our technique compared to different ensembles algorithms, including a recent DRS technique proposed in the literature; and (4) the definition of the oracle concept in DRS.

This paper is organized as follows. Section 2 shows some related work. Section 3 describes the proposed database. Section 4 presents an overview of DRS techniques. Section 5 describes the proposed DRS framework. Section 6 shows the experimental protocol. Section 7 presents and discusses our results. Section 8 concludes this work.

2 Related work

Several methods can be found in the literature for fault location in transmission lines using oscillography data, i.e., three-phase current and/or voltage signals measured for one or two ends of the line, primarily through AI algorithms to estimate the failure distance. In the remainder of the section, we describe a brief review of the methods found in the literature that apply AI algorithms in fault location, aiming to present the main ideas used in the state of the art. A complete review of this topic can be found in the following papers: [1, 24, 45, 51, 55, 60]. Table 1 summarizes some information about the works described in this section, especially concerning the transmission line evaluated by the respective methods and the regression scheme, composed of single or multiple algorithms divided into modules, used to predict the fault location. These details about the works are complemented in Table 2, presented in the next section.

Ray et al. [54] proposed a hybrid fault location method using one post-fault cycle for both current and voltage waveforms, combining Discrete Wavelet Transform (DWT) and Wavelet Packet Transform (WPT). Next, the authors extract features from the decomposed signals that include energy, standard deviation, arithmetic mean, kurtosis, skewness, and entropy. Finally, the feature set was normalized between the range of [0, 1] to then feed a neural network. The method was tested on a 400 kV, 50 Hz, 300 km transmission line, with a sampling rate of 30 kHz.

Ekici et al. [19] used the WPT to decompose the faulty current and voltage signals, considering a half cycle of pre-fault and a half cycle of post-fault. Next, the WPT coefficients were used to calculate the energy and entropy features for each signal separately. These features were then used as input to a neural network. The authors used a 380 kV, 360 km transmission line with a sampling rate of 10 kHz.

Fei et al. [23] proposed a scheme based on the SVR algorithm using only the post-fault voltage signal from a single end of the system. A data window containing a 1/8 cycle post-fault signal is submitted to a low-pass filter to eliminate the noise and then used as the feature vector to the SVR. The method was evaluated on a 400 kV, 50 Hz, 300 km transmission line, with a sampling rate of 20 kHz.

Swetapadma and Yadav [63] showed a method that uses a Decision Tree (DT) to estimate the fault distance. The feature extraction was performed considering a signal processing technique for one pre-fault cycle and one post-fault cycle for both current and voltage waveforms, calculating the standard deviation from the processed signals to obtain the Singular Value Decomposition (SVD) of both signals. For this purpose, the authors assessed two techniques for signal processing: Discrete Fourier Transform (DFT) and DWT. The results demonstrated that the DT-DWT combination performed slightly better.

It is important to mention that the method demands four fault location modules, one for each class of fault type (LG, LL, LLG, LLL). In a similar fashion, but using standard deviation-based features, the same authors presented a method combining the kNN algorithm with the DFT technique, requiring four modules [64]. The demand for

Table 1 Summary of the transmission line and regression scheme assessed by related work

References	Transmission line	Regression scheme
Ray et al. [54]	400 kV, 50 Hz, 300 km	Neural network
Ekici et al. [19]	380 kV, 360 km	Neural network
Fei et al. [23]	400 kV, 50 Hz, 300 km	SVR
Swetapadma and Yadav [63]	400 kV, 50 Hz, 100 km	Four modules with one Decision tree each
Swetapadma and Yadav [64]	400 kV, 50 Hz, 100 km	Four modules with one kNN each
Ferreira et al. [25]	500 kV, 60 Hz, 181 km	Neural network
Bhatnagar et al. [5]	IEEE 14-bus (line details not specified)	Four modules with one Random Forest each
Valabhoju et al. [66]	400 kV, 50 Hz, 198 km	Ensemble of trees

Table 2 Summary of the features usually created and used for fault analysis by related work

Paper	Features	Signal processing technique
Swetapadma and Yadav [63]	SVD	DWT and DFT
Ray et al. [54]	Energy, standard deviation	DWT and WPT
	Arithmetic mean, kurtosis	
	Skewness, entropy	
Ekici et al. [19]	Energy and entropy	WPT
Fei et al. [23]	Amplitudes of voltage data	None
Swetapadma and Yadav [64]	Standard deviation	DFT
Ferreira et al. [25]	RMS, max, min	None
	Arithmetic mean	
	Energy, integral	
Bhatnagar et al. [5]	Teager energy	None
Haq et al. [33]	Energy and standard deviation	DWT
Valabhoju et al. [66]	Standard deviation	DWT
This paper	Energy, distance, RMS, kurtosis	None
	Skewness, maximum frequency	
	Power spectrum density bandwidth	
	AUC, peak-to-peak distance, slope	
	Shannon entropy, arithmetic mean	
	Geometric mean, harmonic mean	
	Variance, standard deviation, median	
	Covariance among the phases	
	Correlation among the phases	
	Maximum and minimum	

more than one model is a drawback of the approaches since they require a fault type classification as a prerequisite to select the most appropriate module to estimate the fault location. Both studies were tested on a 400 kV, 50 Hz, 100 km transmission line, with a sampling rate of 4 kHz.

Ferreira et al. [25] proposed a method that adopted voltage and current data for a single terminal. This method comprises six features (root-mean-square, min, max, mean, energy, and integral) obtained from voltage and current waveforms for the three cycles before and three cycles after the fault incidence for the three phases separately. Thus, a total of 72 features were used as input to a neural network to estimate the fault position. The authors assessed the method in a transmission line with 500 kV, 60 Hz, 181 km, which data were sampled at a frequency of 100 kHz.

Bhatnagar et al. [5] presented a Random Forest (RF) and Teager–Kaiser energy operator (TKEO)-based fault location scheme [7]. The Teager energy of both voltage and current signals of one pre-fault cycle and one post-fault

cycle signal was given as an input to the RF regressor modules, one for each fault type, similarly to the works mentioned before. The number of trees in the RF modules varies depending on the fault type: 300 for LG, 100 for LL, 200 for LLG, and 200 for LLL. The method was evaluated considering the IEEE 14-bus power system, without specifying the transmission line details, for a sampling rate of 1 kHz.

Valabhoju et al. [66] also proposed a method using an ensemble algorithm to estimate the fault distance. Their approach applies DWT over one post-fault cycle of the current and voltage waveforms, extracting standard deviation features from the decomposed signals to use them as input to an ensemble of regression trees, which number of regressors is not explicitly defined by the authors. The method was assessed on a 400 kV, 50 Hz, 198 km transmission line for a sampling rate of 1 kHz.

Most recently, other works in the literature have been using Deep Learning (DL) models for fault location. Basically, those methods usually use raw current and voltage data as input to a model based on algorithms like Long Short-Term Memory (LSTM) and also Convolution Neural Networks (CNN) [6, 20, 22, 62, 68].

Although diverse works that use AI algorithms present accurate methods for fault analysis, some limitations can be identified. Several authors use signal processing techniques, such as those based on wavelet (DWT and WPT) and Fourier (DFT) transforms, to decompose the original current/voltage signal to extract coefficients and attributes [12, 24]. However, these techniques may demand a considerable amount of samples, usually requiring a higher sampling frequency, making their application unfeasible in some real protection systems that have some restrictions on data availability (lower sampling rates, for example) [21, 28, 31, 33]. Another aspect is the fact that the wavelet-based techniques increase the complexity progressively, especially for rising levels of signal decomposition, as well as the determination of the mother wavelet [51], resulting in a higher computational time-consuming and lack of generalizability of the model for other transmission systems [12].

Moreover, the use of ensemble algorithms is a virtually unexplored method in fault location task, demonstrating a gap that should be considered with attention. This motivation is related to the arguments described in Sect. 1, in which the combination of different regressors can result in better performances. In this context, we can apply dynamic selection to obtain the most competent regressors from the ensemble to improve the prediction system. To the best of our knowledge, this is the first work that applies the DRS strategy for fault location.

In response to these limitations, we present in this paper a new dataset with representative features without the use of any signal processing technique. Furthermore, we propose a new DRS framework that aims to select the most accurate models from an ensemble to predict the fault location.

3 Fault analysis database with features

This section presents a new dataset composed of several fault examples in a transmission line. Firstly, we show some information about the environment considered to generate the fault events and the parameters used to simulate them (Sect. 3.1). Afterward, we report the feature extraction process along with some details about these features (Sect. 3.2).

3.1 General information

The Fault Analysis Database with Features (FADbF) is a novel dataset developed and introduced in this work. The fault events were generated based on the IEEE 9-bus power system provided by the ATPDraw tool v7.2 [35], which reproduces a real electrical power system network. This database comprises several attributes extracted from time series of fault simulations of a transmission line with 500 kV, 414 km, and 60 Hz. In addition, the data were sampled at a sampling rate of 10 kHz. It is essential to mention that these simulation data were originally described and created by Ensina et al. [21], where it is possible to access the original data with current and voltage time series used for feature extraction for this new dataset (FADbF).

The database contains 168,000 failure examples, combining essential parameters that include:

- **Fault type**:
  - LG: AG, BG, CG;
  - LL: AB, AC, BC;
  - LLG: ABG, ACG, BCG;
  - LLL: ABC.
- **Fault location**: 1 to 100% of line extension, with intervals of 1%;
- **Fault resistance**: 0.01 to 200 Ω, with intervals of 10 Ω;
- **Fault inception time, in seconds (s)**: 0.091 s, 0.093 s, 0.095 s, 0.097 s, 0.099 s, 0.101 s, 0.103 s, and 0.105 s.

The letters A, B, and C symbolize each of the three phases of the transmission line, while the letter G represents the ground involvement in a fault. Thereby, the initial ACG refers to a failure involving phases A and C with the ground, while AB means a fault between phases A and B without the action of the ground. Each of these ten failure types corresponds to a class for the fault-type classification task, and every fault is guaranteed to contain the same

number of examples with the same parameter values. In other words, FADbF is a balanced dataset with the same number of input samples for both possible targets: fault type classification and fault distance estimation (location).

3.2 Feature extraction

The proposed database extends the features commonly used in the literature on fault analysis based on machine learning (arithmetic mean, standard deviation, energy, entropy, root-mean-square, maximum, and minimum) [11, 45, 54]. It is noteworthy that we explored several surveys, reviews, and related papers, such as those works mentioned and referenced in Sect. 2, aiming to identify the main features used in the literature to compose our dataset. Besides these features, we added some new attributes in FADbF, like geometric mean, distance, and covariance, among others that we will list below.

In total, we extracted 21 features separately for each of the three phases for both voltage and current waveforms along two post-fault cycles from a single terminal, resulting in 126 attributes ($21 \times 3 \times 2 = 126$) in addition to the two possible targets, i.e., fault type (classification task) and fault location (regression task). The feature list includes: fault type, fault location given in kilometers (km), energy, distance, root-mean-square (RMS), kurtosis, skewness, maximum frequency, power spectrum density bandwidth, area under the curve (AUC), peak-to-peak distance, slope, Shannon entropy, arithmetic mean, geometric mean, harmonic mean, variance, standard deviation, median, covariance among the phases, correlation among the phases, maximum, and minimum. The feature extraction from the current and voltage time series was performed with the support of the following libraries in Python: TSFEL v0.1.4 [4], PyInform v0.2.0 [46], and statistics v3.11.4 [53].

The choice of the length of this window (i.e., two post-fault cycles, such as depicted in Fig. 1) was decided considering the related work in the fault location literature, reporting that two cycles provide sufficient information for fault diagnosis [45, 51]. Besides, using more than two cycles can implicate data unavailability since the protection system acts as quickly as possible to identify a failure onset and isolate it from the rest of the system [20].

This dataset makes future benchmarks and evaluations possible for electrical engineering and machine learning-based applications, such as classification (fault type classification) and regression tasks (fault location). Particularly regarding the fault location, this task is responsible for accurately estimating the fault occurrence point on the transmission line, usually considering the distance between a substation and the failure point. The list and all details about each feature are available in our repository.[2]

Table 2 summarizes the features commonly created and used by related work. It is worth mentioning that all these datasets are not available to the scientific community except our database. Additional works and their respective datasets information can be found in the references cited at the beginning of Sect. 2.

Some of these databases were developed using signal processing techniques, such as identified by the respective column in Table 2. Consequently, they are affected by the same signal processing problems previously discussed in Sect. 2. On the other hand, our dataset does not rely on any of these techniques. It is also evident that our features are more diversified and interesting to be explored by machine learning methods, contributing to classification and regression problems.

4 Dynamic regressor selection

Consider a regression problem and let $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$ be an ensemble of N experts. The dynamic selection can be seen as a division of the feature space in $K > 1$ competence regions, denoted by $R_1, R_2, \ldots, R_K$. Thereby, for each region $R_j, j = 1, 2, \ldots, K$, at least one predictor from $\mathcal{F}$ manifests superior performance than the others in R_j [59].

The DRS technique consists of selecting the best regressor(s) from the ensemble to predict the target of a particular test pattern. For this purpose, the selection is commonly performed by calculating the error of each regressor in the RoC and selecting the regressors that attain the smallest errors based on a particular criterion.

Regarding the test pattern x_j, its RoC is a set Ψ composed of the K nearest neighbors of x_j in the training set (or validation set) given by $\mathcal{T} = \{t_1, t_2, \ldots, t_K\}$. In other words, RoC is the local region that surrounds the query x_j, which is predominantly defined by the kNN algorithm [15].

The three techniques that DRS strategy can be used are as follows [41, 56], which include both selection and combination phases, depending on the respective scheme:

- **Dynamic Selection (DS)**: selects only the regressor that presents the lowest cumulative error in the RoC. The errors can be weighted by the distance between the neighborhood pattern and test pattern Eq. (1). Owing to a single model is selected, there is no need for model combination;
- **Dynamic Weighting (DW)**: combines all regressors from the ensemble using the weighted mean without any selection. For each pattern in Ψ, a weighted distance d_k is calculated using Eq. (1), giving more importance to the nearest neighbors:

[2] https://github.com/leandroensina/FADbF.

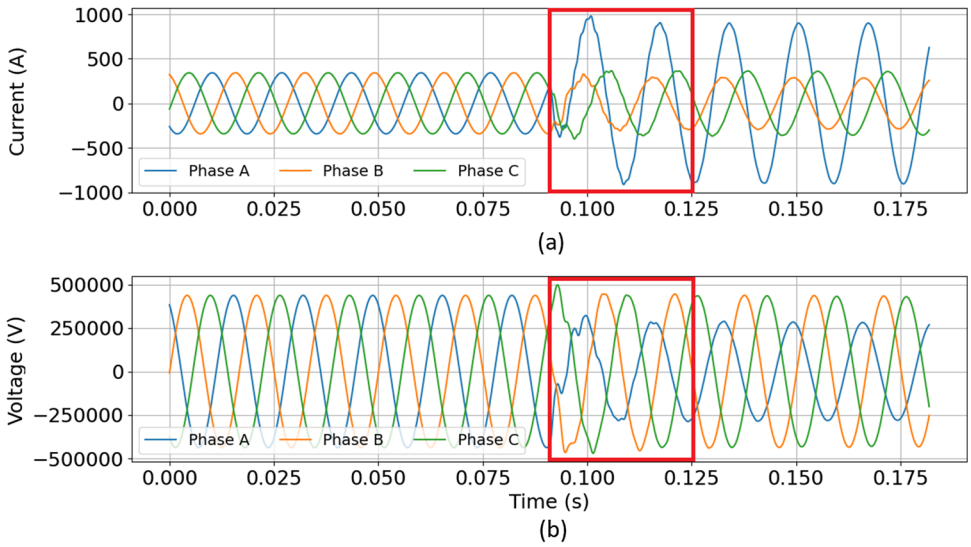

Fig. 1 Example of two post-fault cycles demarcated in a fault simulation of AG type for the **a** current and **b** voltage waveforms. The feature extraction occurred individually for each signal of the three phases for both current and voltage waveforms along two post-fault cycles

$$\mathrm{d}_k = \frac{\frac{1}{\mathrm{dist}_k}}{\sum_{j=1}^{K}(\frac{1}{\mathrm{dist}_j})} \tag{1}$$

where dist_k is the distance between a pattern $t_k \in \Psi$ and the test pattern x_j. The vector $\{d_1, d_2, \ldots, d_k\}$ is used to calculate the weight α_i of the regressor f_i using Eq. (2):

$$\alpha_i = \frac{\frac{1}{\sqrt{\sum_{k=1}^{K}(\mathrm{d}_k \times \mathrm{err}_{k,i})}}}{\sum_{n=1}^{N}(\frac{1}{\sqrt{\sum_{k=1}^{K}(\mathrm{d}_k \times \mathrm{err}_{k,n})}})} \tag{2}$$

where $\sum \alpha_i = 1$, N is the ensemble size, k represents the index of the neighbor, and $\mathrm{err}_{k,i}$ is the error of the regressor i calculated using the pattern $t_k \in \Psi$. Thus, the final prediction corresponds to Eq. (3):

$$\mathcal{F}(x_j) = \sum_{i=1}^{N} f_i(x_j) \times \alpha_i \tag{3}$$

where $\mathcal{F}(x_j)$ is the final decision of the ensemble, $f_i(x_j)$ and α_i are the prediction of the regressor f_i and its respective weight α_i for the test pattern x_j;

- **Dynamic Weighting Selection (DWS)**: selects a subset of models before executing DW. The literature usually uses the standard approach defined by Rooney et al. [56]: the selected models are the ones with the cumulative error below the interval $(E_{\max} - E_{\min})/2$, where $E_{\max}$ and $E_{\min}$ are the largest and lowest cumulative errors of any of the regressors in the ensemble $\mathcal{F}$, respectively. An alternative is the criteria used by Mendes-Moreira et al. [41], in which the selected models are those with an error inferior to a percentage threshold related to the best model (e.g., 10%, 30%, and 90%). However, this last strategy is more complex since several percentage thresholds must be analyzed [43].

A key factor in dynamic selection is the definition of a metric to represent the performance of each model of the ensemble in the RoC, known as the competence measure. This metric is important to serve as a reference to the before mentioned techniques for the selection step (i.e., DS and DWS), which will select only the model(s) that satisfy a given criterion. Moura et al. [49] evaluated eight competence measures using the DS, DW, and DWS techniques, showing that the choice of the best measure is problem-dependent. Moreover, the neighborhood size that determines the RoC also varies depending on the problem [41].

In this scenario, we believe that the use of different distance measures than Euclidean one and different criteria to select the regressors in the DWS technique can also impact the performance of a DRS method, depending on the problem. These analyses aim to explore the diversified and particular data distribution, aiming to attain the most appropriate RoC. At least for the classification problem, there is a correlation between the quality of the neighbors selected to compose the RoC and the models selected to predict the target, directly affecting the final performance [13, 14, 67]. For this purpose, our experiments used only the DWS strategy in order to assess the impact of different criteria to select the regressors.

5 New DRS framework

An overview of the proposed architecture is presented in Fig. 2. In the following subsections, we describe each phase that composes our method.

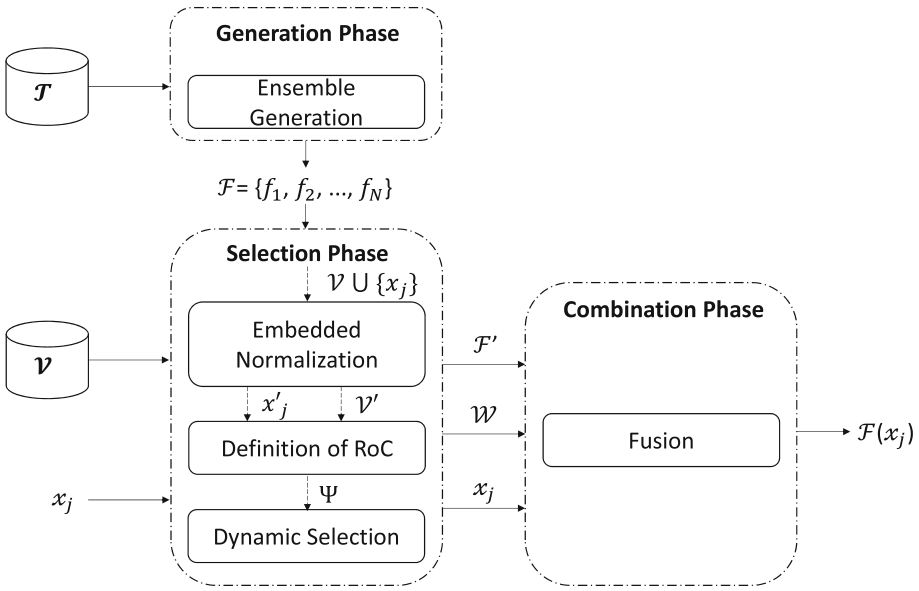

Fig. 2 Overview of the proposed DRS framework

5.1 Generation phase

This phase, composed of a single module (ensemble generation), is responsible for generating the ensemble of regressors $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$. In the context of multiple predictive systems, we have to ensure that each model in the ensemble contains a substantial level of disagreement among them, i.e., diversity in the models.

Therefore, different strategies can be found in the literature for this purpose, mainly for the homogeneous ensemble, such as Bagging [8], Boosting [58], and Random Subspace [34]. An alternative is to use the models from algorithms like Random Forest [9], Extremely Randomized Trees (ERT) [30], and Gradient Boosting [26]. The number of estimators for the pool of models is not considered a hyperparameter of our method since the ensemble generation module does not perform the training process. For example, regarding the use of the ERT's trees to compose our pool of regressors, the hyperparameter N (number of estimators) belongs to the ERT algorithm, which will train the models and provide the trained trees to our DRS method.

5.2 Selection phase

The second phase is responsible for selecting dynamically the most competent regressors from the pool of models, constituted by three modules: embedded normalization; definition of RoC; and dynamic selection. This phase contains the valuable hyperparameters that our method explores in order to achieve better performances, which are adequately identified along the description of the modules. Moreover, tuning hyperparameters automatically to define the best configuration is not part of the method. Hence, the researchers should manually evaluate them based on the comparison of multiple configurations, such as we will perform in the sequence of this manuscript.

Embedded Normalization. The first module corresponds to the normalization of the validation set $\mathcal{V}$ used to determine the RoC Ψ. Usually, the proposed methods in the literature perform the normalization step before the beginning of the entire process as a preprocessing [50, 61] in view of this process is essential to distance-based algorithms, such as kNN used to determine the RoC.

Before executing the normalization process, we added the test pattern x_j into the original validation set ($\mathcal{V} \cup \{x_j\}$), and then we conducted the normalization of this set, being removed from the validation set right after the data scaling. In this framework, any normalization technique can be used to transform the original data into the same range of values, such as MinMax scaler and Z-score scaler. This independence of the scaling procedure is crucial since it can also be problem-dependent [3]. It is noteworthy that this process is repeated individually for each new test pattern. Therefore, we used the normalization step only to determine the neighbors around the scaled test pattern x'_j considering the transformed validation set $\mathcal{V}'$ during the second module (definition of RoC). In this module, there is no hyperparameter to be tuned.

It should be emphasized that this kind of normalization integrated into the DRS method is proposed and exclusive of our framework. Due to its characteristics, we call this process embedded normalization, after all it occurs individually for each test pattern and is inside the framework.

Table 3 Hyperparameters of the proposed DRS framework

Name	Values, but not limited to
Distance function	Euclidean, cosine, geodesic, **cityblock**
K	5, 10, 15, 20, **30**, 40, 50
Competence measure*	Variance, sum absolute error, sum squared error, minimum squared Error, **maximum squared error**, neighbor's similarity, root sum Squared error, closest squared error
Selection criterion	Standard criterion ($(E_{\max} - E_{\min})/2$), median, arithmetic mean **Geometric mean**, harmonic mean

*See Moura et al. [49] for details about each competence measure

Definition of RoC. We used the kNN algorithm to define the RoC similarly as already carried out in the state of the art, but in this work we assessed a hyperparameter unexplored in the literature of DRS strategy: the metric for distance computation. Although the specialized literature uses the Euclidean metric as the standard, the choice of this measure should not be neglected because this hyperparameter can also reverberate in a performance improvement. In response to this issue, we also evaluate other distance metrics, such as Cityblock, Geodesic, and Cosine.

Furthermore, the second module obtains the indexes of the neighbors in $\mathcal{V}'$ to retrieve the respective samples in the original validation set $\mathcal{V}$ to create the RoC Ψ. As can be observed, the normalization in the first module was used only to determine the closest samples around the scaled test pattern x'_j, while the RoC is composed of the original samples. The hyperparameters of this module are the distance function and the value of K for the kNN algorithm that defines the RoC size.

Dynamic Selection. In turn, the third module performs the dynamic selection concerning the competence of the regressors in the RoC Ψ. For this purpose, several competence measures can be used to determine the performances of each model in the pool, such as those proposed by Moura et al. [49].

The criteria for selecting the best regressors can be done in different ways, but the standard approach corresponds to that defined by Rooney et al. [56], as mentioned in Sect. 4, which selects only the regressors with a cumulative error below the interval $(E_{\max} - E_{\min})/2$. In contrast, we can also use other strategies to establish the criterion for the selection, such as median, harmonic mean, arithmetic mean, and geometric mean. In this scenario, we can calculate the median/mean value among the performances based on the competence measure used and use this median/mean value as a threshold, selecting only the regressors that present an error inferior to this limit.

Another task of the third module is the definition of the weights $\mathcal{W} = \{\alpha_1, \alpha_2, \ldots, \alpha_M\}$ of the M selected regressors $\mathcal{F}'$ to the final prediction $\mathcal{F}(x_j)$. This task occurs similarly as described in Sect. 4 through Eqs (1) and (2). The hyperparameters of this module include the competence measure and the selection criterion.

Table 3 summarizes the hyperparameters of the proposed DRS framework. We identify in bold the combination of values determined as the best configuration of the method for the FADbF dataset, as it will be justified in Sect. 7.

5.3 Combination phase

In this last phase, composed of a single module, the fusion of the predictions of the regressors selected in the previous phase occurs as described in Sect. 4 through Eq. (3), using the subset of the selected models $\mathcal{F}'$ and their respective weights $\mathcal{W}$. This module has no hyperparameter to be tuned.

6 Experimental protocol

In this paper, we trained the Extremely Randomized Trees (ERT) algorithm [30] with $N = 700$ estimators and used its trees to compose our pool of regressors $\mathcal{F}$. We used ERT with 700 estimators because preliminary experiments manifested that the ensemble with less number of trees ($<$ 700) presented worst results, while a higher number of estimators ($>$ 700) did not reveal a significant performance improvement. See Supplementary Material[3] for complete details. For all experiments in this study, we used the implementations provided by the Scikit-learn library v1.2.0 [52].

The algorithm was trained and examined with shuffled examples from the dataset. Thus, the FADbF instances were randomly divided into training, validation, and testing sets. The training set $\mathcal{T}$ comprises 50% of all examples (84,000 instances), while validation $\mathcal{V}$ and testing $\mathcal{X}$ sets contain 25% of all cases each (42,000 examples). This partition process was performed ten times, resulting in

[3] https://github.com/leandroensina/SM_NCAA_FaultLocation_DRS.

different sets with distinct instances among each repetition. So, the results in Sect. 7 represent the average and standard deviation based on these ten repetitions. Additionally, the following section used the same pre-trained ERT models for every investigation in order to have fair results.

Moreover, the model was trained with the target in terms of the percentage of the transmission line length. Consequently, the model output consists of a real value between [1%, 100%], corresponding to the percentage of the transmission line length where the fault was located. This value is converted to distance in kilometers (km) after the prediction by Eq. (4). So, all results reported and discussed in the paper correspond to the target in kilometers.

$$\text{distance}_{km} = \frac{TL_{\text{length}} * \text{distance}_{\text{percentage}}}{100} \tag{4}$$

where distance_{km} is the fault location in kilometers, TL_{length} is the transmission line length in km (414 km), and $\text{distance}_{\text{percentage}}$ is the fault location in percentage.

The choice for training in terms of percentage instead of directly in kilometers is justified by the fact that the substations can supervise several transmission lines with different lengths simultaneously, which could affect the model's predictions. Therefore, the use of percentages aims to enable our method to deal with other transmission lines independently of their lengths, making the method more flexible and trying to avoid the requirement of new training individually for each line under supervision [21]. Although in this study we consider a single transmission line with 414 km, we intend to assess the generalization capacity of our method to predict the fault distance for other transmission lines with different lengths in future experiments.

Considering the module "embedded normalization" in the second phase of the proposed DRS (Sect. 5.2), we applied the MinMax technique to scale the attributes into the interval [0, 1]. Also, we established $K = 30$ for the "definition of RoC" module since it presented better results than other neighborhood sizes in preliminary experiments (see Supplementary Material). We analyzed four distance metrics to assess the impact of the variation in this hyperparameter used in the kNN algorithm to determine the RoC: Euclidean, Cosine, Geodesic, and City-block. This evaluation is essential to determine if we can attain better results by varying the distance metric.

In the "dynamic selection" module, the competence measure employed was the maximum squared error proposed by Moura et al. [49] since this metric presented better results in preliminary experiments than the other measures listed in their study. Besides, we also assessed other strategies to select the most competent models in addition to the standard approach mentioned in Sect. 4 (selection criteria). Notably, we evaluated the median, arithmetic mean, harmonic mean, and geometric mean together with the traditional technique.

All experiments were evaluated with the Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics defined by Eqs (5) and (6), respectively, in which n is the number of examples, y_i is the real fault localization, and $\hat{y}_i$ is the predicted fault localization for the sample i.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

The results of the several comprehensive experiments performed in this study were statistically compared, aiming to provide a solid conclusion about the best strategies and configurations. We used the D'Agostino & Pearson normality test (p-value < 0.05) for all groups of data to then select the most appropriate hypothesis test based on the data distribution (parametric or nonparametric). The respective hypothesis tests used are presented in Sect. 7, all with a confidence interval of 95%. Thus, we used different tests along the analyses regarding three essential aspects for choosing the best test individually for each comparison scenario: number of groups, data distribution, and matched/unmatched groups.

7 Results and discussion

In this section, we present and discuss the results of our method with comprehensive experiments. Section 7.1 presents the experiments concerning the normalization process, justifying the decision for an embedded normalization and its importance to the method. Section 7.2 evaluates the impact on the performance when we vary the distance measure used to determine the RoC. Section 7.3 presents the results concerning the use of different criteria to select the most competent regressors from the ensemble. Section 7.4 assesses the impact of the proposed DRS technique using the FADbF dataset compared to the use with only the attributes usually used by the state of the art. Section 7.5 compares the performance of the proposed method against some related methods for fault location, traditional ensemble algorithms, and one of the most recent DRS techniques in the literature, in addition to justifying the choice of using the ERT algorithm as the pool of regressors. Finally, Sect. 7.6 describes the complexity and the main drawbacks of the proposed method.

Table 4 Mean and standard deviation, in parentheses, of the results in kilometers comparing the use of the normalization step, calculated in ten replications

Scaling	MSE	MAE
Without normalization	1.4161 (0.0796)	0.7163 (0.0062)
Standard normalization	2.5465 (1.0858)	0.8250 (0.0156)
Embedded normalization	1.3973 (0.0942)	0.7086 (0.0068)

7.1 Assessing the normalization process

Table 4 reports the results comparing three different scenarios in our DRS technique:

1. **Without normalization**: consists of the use of the original features without any preprocessing (i.e., data scaling);
2. **Standard normalization**: consists of the validation set scaled before the beginning of the entire process of the DRS, using its parameters to normalize each test pattern. In other words, the normalization occurs outside the DRS technique, in which the features for each new example are scaled based on the validation set distribution. Usually, related DRS methods perform this process as a preprocessing [50, 61];
3. **Embedded normalization**: proposed in this work, this scenario is presented in detail in Sect. 5.2. Basically, it scales the test pattern and the validation set together individually for each new example to define the RoC. Before executing the normalization process, we added the test pattern inside the original validation set, and then we conducted the normalization of this set, being removed from the validation set right after the data scaling. This process is carried out individually for each new example. Here, the data scaling is used only to determine the RoC, while the original data, without preprocessing, are used as features for the ensemble.

These results demonstrate that the data scaling reverberates in a performance gain with this process embedded in the DRS approach, manifesting its importance. The statistical comparison with paired t-tests corroborated the better operation of our strategy. Despite there being no statistical difference between "embedded normalization" and "without normalization" for the MSE measure (p-value $= 0.2075$), we can identify statistical difference for the MAE metric (p-value < 0.0001), the main metric used in the literature for fault location.

We also assessed the impact of keeping the validation set normalized and scaling the sample test based on the parameters of the transformed validation set (standard normalization). However, this scenario performed worse than the embedded normalization proposed in this work. These inferior results can be related to the discrepant values of the attributes of the test pattern, i.e., lower minimum values and higher maximum values in the features of the sample test compared to those inside the validation set. On the other hand, this problem does not affect our embedded normalization since the method adds each test pattern inside the original validation set before the normalization. So, this analysis is crucial to endorse the importance of normalizing the test pattern together with the validation set. The statistical tests, paired t-test for MAE and Wilcoxon test for MSE, confirmed the differences among the results for both measures (p-value < 0.0001 for MAE and p-value $= 0.002$ for MSE).

The embedded normalization scales the test pattern and the validation set together, individually for each new example, to define the RoC, leading to better results of the DRS to estimate the fault location. The reason for this online process was related to the fact that the kNN algorithm (distance-based) used to determine the RoC can be affected by the scale of the data, which can influence the examples returned to compose this region. In other words, features with discrepant scales tend to dominate the others and may lead to an imperfect RoC and, consequently, in a performance loss, such as happened to the standard normalization.

7.2 Assessing the distance metrics

Table 5 presents the performances of our method concerning different distance measures. It is possible to notice that the method with Cosine and Cityblock distances performed slightly better than using Euclidean distance. The repeated-measures one-way ANOVA test manifested statistical differences among the performances (p-value $= 0.0007$ for MSE and p-value < 0.0001 for MAE), while the Tukey post-test manifested that there are statistical differences between Cityblock with each other metric, i.e., superior performance. On the other hand, the Geodesic measure showed inferior overall results than the other metrics.

Table 5 Mean and standard deviation, in parentheses, of the results in kilometers comparing the different distance metrics, calculated in ten replications

Distance measure	MSE	MAE
Euclidean	1.4677 (0.0943)	0.7250 (0.0073)
Cosine	1.4583 (0.1233)	0.7196 (0.0072)
Geodesic	1.4822 (0.1154)	0.7253 (0.0071)
Cityblock	1.3973 (0.0942)	0.7086 (0.0068)

This analysis is interesting because we attain better results just by changing this measure, evidencing its importance. Besides, the use of distinct distance metrics aims to explore their influence on the behavior of each database, which can present a diversified and particular data distribution. In this scenario, other measurements like Canberra and Mahalanobis can be used alternatively [18].

7.3 Assessing the selection criteria

Table 6 shows the results considering different criteria to establish the threshold for selecting the most competent regressors from the ensemble based on their performance on the RoC. The performances evidenced that the standard approach usually used in the literature ($(E_{\max} - E_{\min})/2$) is overcome by all other criteria, confirmed by the repeated-measures one-way ANOVA (p-value < 0.0001 for both MSE and MAE metrics) and Tukey post-test.

Although there are no statistical differences among geometric mean, harmonic mean, and median for the MSE measure, geometric mean performed better than them all for the MAE metric. Even though the literature on DRS evidences more attention to the definition of the criterion to measure the competence of the models in dynamic selection [49, 50], we could demonstrate through this work that we should not neglect other hyperparameters of a DRS method, exploring them in order to achieve better performances. We can identify with the presented results that defining the criterion to select the best regressors from the ensemble is crucial, reverberating directly in the performance improvement.

Considering the results presented for the three components evaluated in this paper (i.e., normalization, distance function, and selection criterion), we can establish that, for the FADbF dataset, the distance function led to a bigger variance in performance, followed by normalization and selection criterion, respectively. It is interesting because these comprehensive experiments demonstrate the importance of searching for the best combination of hyperparameters usually neglected or unexplored by the DRS state of the art.

Table 6 Mean and standard deviation, in parentheses, of the results in kilometers comparing the different selection criteria, calculated in ten replications

Selection criteria	MSE	MAE
Standard criterion	1.4445 (0.1004)	0.7127 (0.0073)
Median	1.3968 (0.0921)	0.7097 (0.0067)
Arithmetic mean	1.4157 (0.0979)	0.7097 (0.0072)
Geometric mean	1.3973 (0.0942)	0.7086 (0.0068)
Harmonic mean	1.3922 (0.0885)	0.7127 (0.0064)

7.4 Assessing FADbF dataset

In this subsection, we evaluate the importance of the features presented in the FADbF dataset, which comprises the features usually used in related work for fault analysis, as much as new ones included in this work. So, we compare the performance of the DRS technique for two contexts:

1. Using only the features usually used in the specialized literature composed by arithmetic mean, standard deviation, energy, entropy, RMS, maximum, and minimum for both current and voltage signals;
2. Using all features from the FADbF dataset (Sect. 3).

Table 7 presents the results for both contexts mentioned before, demonstrating a better performance with the application of all features included in the FADbF dataset. The statistical comparison with paired t-test confirmed this statement (p-value < 0.0001 for both MSE and MAE measures).

This behavior can be justified somehow by the analysis conducted in relation to the importance of the features for the ERT algorithm used in this work to compose the pool of regressors for the DRS technique. Table 8 shows the top 10 features ordered by their importance for the ERT algorithm, in which the higher the value, the more relevant the attribute. The scores presented in this table represent the average and the standard deviation for each feature considering the ten pre-trained models, the same used for the analyses in the previous subsections. The relevance of a feature is computed based on the Gini importance with a normalized range of scores [36].

In this analysis, we grouped the features initially separated for each phase (A, B, or C) into a single feature in Table 8, but separately per signal type (current or voltage waveform). For example, considering the standard deviation attribute, there are originally three features based on it, one for each phase (A, B, or C). So, we summed the Gini scores for these three features into a single one to enable an analysis per feature, individually for each signal type. See

Table 7 Mean and standard deviation, in parentheses, of the results in kilometers comparing the features in FADbF dataset and the features used in the literature

Context	MSE	MAE
Literature features	2.1146 (0.0386)	0.9599 (0.0065)
FADbF features	1.3973 (0.0942)	0.7086 (0.0068)

Table 8 Top 10 features ordered by their importance for the ERT algorithm, considering the mean and standard deviation, in parentheses, of the Gini score

Ranking	Feature	Gini score
#1	Standard deviation (current signal)	0.0410 (0.0016)
#2	Variance (current signal)	0.0285 (0.0008)
#3	Geometric mean (current signal)	0.0257 (0.0018)
#4	Distance (voltage signal)	0.0215 (0.0018)
#5	AUC (current signal)	0.0200 (0.0007)
#6	Covariance (current signal)	0.0196 (0.0003)
#7	Energy (voltage signal)	0.0178 (0.0002)
#8	Variance (voltage signal)	0.0176 (0.0005)
#9	RMS (current signal)	0.0156 (0.0005)
#10	RMS (voltage signal)	0.0152 (0.0003)

Supplementary Material for complete details about the Gini scores, including the values individually for each feature.

This analysis evidences that six of the top 10 features are original from the FADbF dataset: variance (current signal); geometric mean (current signal); distance (voltage signal); AUC (current signal); covariance (current signal); and variance (voltage signal). Although the standard deviation and variance have a certain relationship, they still present different statistical perspectives. As a result, variance in this context is considered a new feature. Notice that we chose not to group the features of both signal types into a single feature, e.g., RMS for current and voltage signals only as RMS, and examine them separately because current and voltage waveforms manifest distinct properties. An example is the variance feature, the second most important feature with the current signal, but only in the eighth position for the voltage waveform.

In this work, we did not carry out any feature selection, which could reverberate in a performance gain. We could, for example, use only standard deviation or variance since both of them are used to measure the data dispersion around the arithmetic average. Nevertheless, this kind of preprocessing is out of the scope of this work and will be considered in future works.

7.5 Comparison with the state of the art

Finally, in this section, we compare the performance of our method against five methods reviewed in Sect. 2. Next, we discuss some properties of the ensemble used as the pool of regressors, along with the comparison with traditional ensemble algorithms like Random Forest, ERT, and Gradient Boosting.

Table 9 presents the performance comparison among some state-of-the-art works reviewed in Sect. 2 and the proposed method. We reproduced these methods using the same dataset (FADbF) used to evaluate our approach for a fair comparison, with the same division in training and testing sets as we presented in Sect. 6. As a result, the preprocessing procedures and the features used as input to the algorithms may differ from their original proposals. For example, signal processing techniques like DWT, WPT, and DFT were not performed for the methods that use them since these procedures have to be carried out on the current and voltage signals before feature extraction, which does not comprise the FADbF dataset for the reasons discussed at the end of Sect. 2. On the other hand, we carried out steps like the normalization of the feature vector [54] or the use of multiple modules [5, 63, 64] for the methods that require any of them. Considering that our objective is to evaluate the proposed dataset, we used the whole FADbF set as input for all methods, while each regression scheme (Table 1) was reproduced with the same configuration described by the respective works.

Our method demonstrated superiority over all other methods, statistically corroborated by the Friedman test (p-value < 0.0001 for both MSE and MAE metrics) and Dunn's post-test. These high error rates of the related methods can be explained partly by the fact that these methods were originally developed for different transmission lines than the one used for the FADbF dataset, as well as using different sampling rates for data collection. Besides, the power systems' characteristics and their complexities also differ. Consequently, all these aspects result in different fault signatures (e.g., distinct signal amplitudes even for transmission lines with the same length) among the datasets [21, 55]. As stated by Ferreira et al. [24], the performance of the fault diagnosis system is guaranteed only for the transmission line and fault conditions for which the fault diagnosis system was developed. This statement is reflected in the results of Bhatnagar et al. [5], which manifested the second-best performance for the FADbF dataset. The authors used the IEEE 14-bus power system to develop and assess their approach (Sect. 2), which presents similarities to the IEEE 9-bus system used to simulate the fault events for the FADbF dataset (Sect. 3.1). Consequently, it can explain the better performance of their method compared to the others. Moreover, the performance of their method for the FADbF dataset is similar to the performance reported to the power system considered in their original study, which represented MAE of around 1 km against MAE of 1.3697 km in our dataset.

The choice of using the decision tree as the base model to compose the pool of predictors is justified in part by the results reported in Table 10. Beyond decision trees, other learning algorithms were considered in this test: Multilayer Perceptron (MLP) with two configurations, which structures were based on the works presented by Ferreira et al.

Table 9 Mean and standard deviation, in parentheses, of the results in kilometers for related methods and the proposed method

References	MSE	MAE
Ray et al. [54]	183.6866 (38.0976)	10.6342 (1.2281)
Swetapadma and Yadav [63]	34.7050 (5.0698)	3.0321 (0.0277)
Swetapadma and Yadav [64]	183.0063 (5.7464)	6.6843 (0.0715)
Ferreira et al. [25]	70.6623 (10.6062)	5.6639 (0.5397)
Bhatnagar et al. [5]	5.1203 (0.3550)	1.3697 (0.0155)
Proposed method	1.3973 (0.0942)	0.7086 (0.0068)

Table 10 Mean and standard deviation, in parentheses, of the results in kilometers for each base algorithm, calculated in ten replications

Algorithm	MSE	MAE
CART	40.6147 (6.0779)	3.4148 (0.0451)
MLP [25]	70.6623 (10.6062)	5.6639 (0.5397)
MLP [54]	183.6866 (38.0976)	10.6342 (1.2281)
SVR (RBF kernel)	394.5469 (4.8037)	16.1122 (0.0855)
SVR (linear kernel)	745.6150 (9.1590)	21.3690 (0.1246)
kNN ($k = 3$)	246.0872 (9.5253)	7.3927 (0.0582)
kNN ($k = 7$)	294.7072 (9.4705)	8.7875 (0.0440)

[25] and Ray et al. [54], such as previously presented in Table 9; SVR for two types of kernel; kNN with $k = 3$ and $k = 7$. The decision tree implementation used in this study was the CART (Classification and Regression Tree) algorithm [10]. These initial experiments comprised the FADbF dataset using only the training and validation sets.

The CART algorithm performed better than the other models, as evidenced by the repeated-measures one-way ANOVA test followed by the Tukey post-test, which manifested statistical differences between CART and each predictor. Besides, these algorithms require a superior computational cost than decision trees. Consequently, these aspects explain the choice of the decision tree as the base predictor.

In turn, Table 11 shows the performances of different ensembles, all comprising the CART algorithm.

Table 11 Mean and standard deviation, in parentheses, of the results in kilometers for each ensemble algorithm, calculated in ten replications

Approach	MSE	MAE
ERT	1.7830 (0.1237)	0.7832 (0.0087)
Random forest	9.8331 (1.6850)	1.5183 (0.0216)
Bagging of CART	9.7880 (1.6249)	1.5158 (0.0194)
Gradient boosting	90.7166 (1.6878)	6.5164 (2.2906)

As can be observed, ERT performed better than the other ensembles, justifying its use as the baseline to obtain the most promising pool of regressors since it allows us to obtain its list of trained models. The ERT algorithm uses a random subset of candidate features, drawing the thresholds randomly for each candidate feature and selecting the best of these randomly generated thresholds as the splitting rule. Also, this algorithm can avoid a particular feature predominating over the other features since only part of the attributes is available for each tree during its creation process. As a result, it can reduce the variance of the model and control overfitting. These aspects may have contributed to a better performance of this approach over the other techniques to compose our pool of models.

The choice of ERT is also justified by the performance of its oracle (row "Oracle" in Table 12), achieving a perfect performance, i.e., at least one tree in the ensemble could attain the exact location of the fault. The oracle is an abstract strategy that always selects the model that predicts the correct label for the given query sample if such a model exists [15, 39]. In other words, the oracle is the best possible performance that the ensemble can attain, representing the ideal predictor selection scheme.

To determine the performance of the oracle, the following steps were carried out: first, we calculated the absolute error between the real fault position and the predicted fault position individually for each estimator in the ensemble; next, we selected only the regressor that presented the lowest error for the particular test pattern to predict its label. It is essential to clarify that we had to check the predictions of each model directly with the actual target value of the test pattern to then select the regressor

Table 12 Mean and standard deviation, in parentheses, of the results in kilometers for each DRS method, calculated in ten replications

Approach	MSE	MAE
DWS-MS [50]	1.4421 (0.0815)	0.7184 (0.0071)
Proposed DWS	1.3973 (0.0942)	0.7086 (0.0068)
Oracle	0.0000 (0.0000)	0.0000 (0.0000)
Oracle of the selected models	0.0009 (0.0007)	0.0002 (0.0002)

that presented the lowest error, which could not be possible in a real scenario because the actual target value obviously will not be available. That is why the oracle is considered an abstract strategy.

Table 12 presents the results of the experiments comparing our method against the DRS proposed by Moura et al. [50], called Dynamic Weighting Selection with Measure Selection (DWS-MS). It is essential to mention that we provided exactly the same pre-trained pool of regressors for both methods with the size of $N = 700$.

The hyperparameters of DWS-MS were defined similarly to our technique in this work, with the same competence measure (i.e., maximum squared error), $k = 30$, and the same trained pool of models. We tried to achieve the best possible configuration of this method for the FADbF dataset in order to enable a fair comparison. For the results of our method reported in Table 12, we used maximum squared error as the competence measure, Cityblock as the distance measure, and geometric mean as the criterion to determine the threshold, in which we select only the regressors that present an error lower than this limit. This configuration demonstrates the better performance of our method for the FADbF dataset.

Our technique demonstrated superior performance for all comparisons, considering the paired t-test with a confidence interval of 95%. Notice that we compared our method against DWS-MS and all ensembles listed in Table 11 without considering the oracle models (last two rows in Table 12). The hypothesis test revealed statistically significant differences comparing our approach individually with each of the other techniques for both MSE and MAE metrics (p-valor < 0.0001 for most comparisons).

The better performance of our technique can be explained by the fact that we went further in the variation of the DRS hyperparameters, together with the addition of the embedded normalization. While DWS-MS aims to obtain the best competence measure, our method looks for the ideal distance metric and selection criteria along with the best competence measure.

Even if the DWS-MS hyperparameters were configured with the same distance function and selection criterion used in our method, Cityblock and geometric mean, respectively, it would perform similarly as our framework without the embedded normalization, such as already demonstrated in the row "without normalization" in Table 4. These analyses evidence the crucial role of embedded normalization, leading our method to superior performance. With this process incorporated, it was possible to reach a more appropriate RoC and, consequently, enable the method to select the most competent regressors to predict the test pattern.

Although the oracle shows no error (row "Oracle" in Table 12), with at least one regressor in the pool of models predicting exactly the fault location of each test pattern, the main challenge is selecting this regressor. This is our main motivation in the use of the ensemble technique with dynamic selection against the use of individual models widely used in the state of the art of fault location.

Likewise, the oracle's performance evidenced that we can further investigate the dynamic selection strategies. For this purpose, we explored other distance metrics and other selection criteria aiming to gain performance, as demonstrated in the previous subsections. Nevertheless, we should emphasize that achieving the same performance of the oracle is a complex task since we select multiple regressors to define the target value unless they all predict exactly the same (and correct) value, which is unlikely.

In addition to the oracle of the whole ensemble, we assessed the oracle only among the selected models by our DRS scheme (row "Oracle of the selected models" in Table 12). This analysis aims to determine the best possible performance regarding only the dynamically chosen models by our DRS technique.

Considering the original pool of models with 700 estimators, the proposed DRS method selected about 382 ± 51 regressors per query with the best configuration (embedded normalization, Cityblock as distance metric, and geometric mean as selection criterion). Also, if we observe the oracle among these selected models (row "Oracle of the selected models" in Table 12), it is possible to identify that our technique selected accurate predictors. Besides, these results demonstrate that new strategies can be added as a multi-step dynamic selection, using different competence measures for each new step in order to achieve better results [48].

Our method presented satisfactory performance, achieving an MAE of 0.7086 km $\pm$ 0.0068 km, representing $0.1712\% \pm 0.0016\%$ of the transmission line length (414 km). Figure 3 illustrates the dispersion of the errors along the transmission line, which distribution occurs equally over the line. For this analysis, we sorted the real location vector of the testing set, which varies from 4.14 km to 414 km, simultaneously with the prediction vector in ascending order, respecting the corresponding positions of the predicted and actual targets between both vectors. The result of this ordering is the continuous curve plotted in Fig. 3.

Table 13 shows more details about the error distribution considering different ranges. Errors bigger than 10 km represent only about 35 of 42,000 failure events in the testing sets, corresponding only to about 0.0833% of the test examples. On the other hand, most errors are less than or equal to 0.5 km and correspond to 52.6762% of all test examples (about 22,124 of 42,000 occurrences).

A perspective for further investigation in future works is related to the study of DL predictors like LSTM and CNN

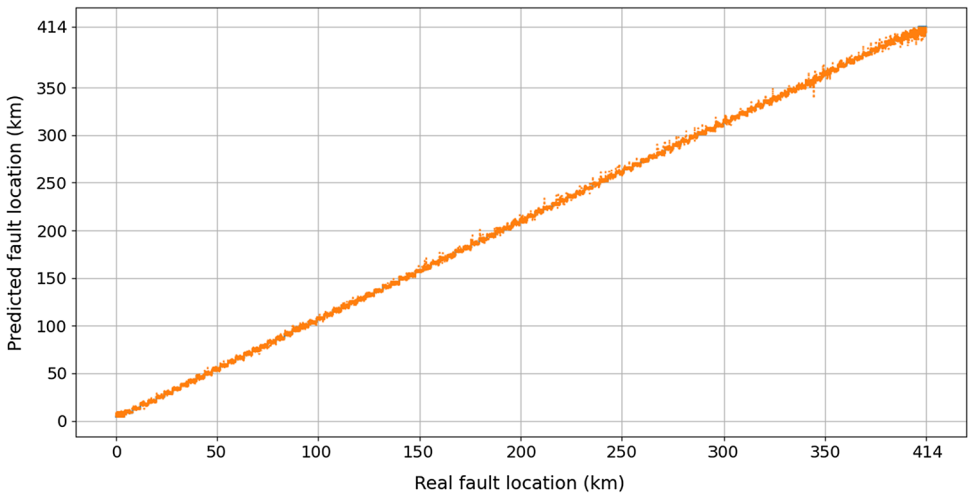

Fig. 3 Representation of the prediction errors

Table 13 Distribution of the errors for different intervals. The values represent the mean and standard deviation, in parentheses, of the number of occurrences over the ten repetitions

Interval of error	Number of occurrences
error $<=$ 0.5 km	22,124 (96)
0.5 km $<$ error $<=$ 1 km	10,817 (109)
1 km $<$ error $<=$ 5 km	8,731 (90)
5 km $<$ error $<=$ 10 km	293 (19)
error $>$ 10 km	35 (9)

as base learners. This analysis is interesting because methods that use DL algorithms and others that use classical ones like decision trees may manifest different dispersion of the errors since they use different features along with the different nature of the algorithms. In this scenario, we can merge DL-based models with classical algorithms to compose the pool of regressors (heterogeneous ensemble), similarly as already used in the literature for time series forecasting [17]. Moreover, we can use simpler structures of DL models to compose the pool instead of requiring a single and complex DL architecture, such as proposed in some works in the literature [6, 20, 22, 68].

Owing to the key property of the DRS strategy being the diversity among the models, we can train DL algorithms with raw current and voltage signals data and train classical algorithms (e.g., CART and SVR) with features manually extracted, such as those from FADbF dataset (Sect. 3). So, we can not only diversify the kind of data used to train the algorithms but also explore the different characteristics of each algorithm, ensuring diversity from several perspectives. With that in mind, we believe that we can achieve better results and also reduce the size of the ensemble (currently with $N = 700$). Last but not least, despite the evaluation only with the DWS technique, our DRS framework can also be adapted to work with DS and DW, which will be assessed in future experiments.

7.6 Final remarks

Although the good performance and effectiveness of the proposed method for fault location, as demonstrated in this study, we can identify two main drawbacks: the computational cost and the number of models required in the pool of regressors.

The former weakness is related to the fact that the method needs to carry out a sequence of steps individually for each new example to be predicted (Sect. 5): (1) normalize the validation set together with the test pattern by the MinMax scaler ($O(n+1)$, where n is the number of examples in the validation set, plus the test pattern); (2) define the RoC by the kNN algorithm ($O(k \times n \times d)$, where k is the number of neighbors, n is the number of examples in the validation set, and d is the data dimensionality); (3) compute the performance of each regressor in the ensemble for the RoC based on the competence measure ($O(r)$, where r is the number of regressors); (4) determine the performance threshold using the selection criterion ($O(r)$); (5) select the regressors that present an error inferior to this limit ($O(r)$); (6) calculate the weights for each selected model ($O(m)$, where m is amount of selected models); and (7) then perform the final prediction ($O(m)$).

Consequently, there is an increase in the processing time needed to predict the target compared with static ensemble algorithms like Random Forest and ERT, which basically average the prediction of all models to get the final decision. Nonetheless, this time is not a problem considering that our method was developed to be used as an offline application in a real protection system, which does not

demand a restricted time for a prediction. We can estimate the prediction time complexity of our method by summing the complexities of the steps mentioned before, which are identified in parentheses in the last paragraph. In summary, the approximate prediction complexity of the method is $O(n + (k \times n \times d) + r + m)$.

Another aspect associated with the computational cost is the requirement to keep the whole validation set together with the pool of models during the operation of the DRS method, similar to any other dynamic selection approach. It is required because the DRS method needs the validation set to use in the lazy learning procedure carried out by the kNN algorithm, which is responsible for obtaining the RoC for each new pattern. As a result, there is a demand for extra storage space. In this study, for example, each validation set file corresponds to about 88 MB with 42,000 instances and 126 attributes, which do not represent a significant size. Of course, the file size should vary depending on the database, but it will not make the method unfeasible. A possible solution for this problem that will be further investigated is the prototype selection [29]. This technique seeks a reduced dataset, serving as a condensed view of the whole original database, and can even improve performance [37].

The latter weakness is the high number of models in the pool required for the fault location task, allowing the method to achieve the performances presented. This problem also reverberates in the computational cost since it demands that each of these regressors is evaluated. So, future efforts will be directed at improving the generation phase (Sect. 5.1), following directions like those mentioned at the end of Sect. 7.5.

It is important to emphasize again that, even with these limitations, our method proved to be accurate and effective. These problems do not affect the viability of our method since it was developed to be used as an offline application. Ferreira et al. [24] emphasize that the computational cost is not critical for offline applications, allowing the use of more sophisticated techniques for fault location.

8 Conclusion

This paper presented a new dataset and a novel DRS scheme, characterized by introducing an embedded normalization and evaluating multiple configurations, in which we varied hyperparameters like the distance metric used to define the RoC and different selection criteria to obtain the most competent regressors. We also introduced the concept of oracle in DRS, widely used for the classification task [15], serving as a reference to how good can be the performance of a DRS method.

The performance of the proposed method achieves an MAE of 0.7086 km ± 0.0068 km, representing 0.1712% ± 0.0016% of the transmission line length, attaining superior results than one of the latest DRS methods in the literature. So, the results show that the proposed approach complies with our objective. These results are crucial to demonstrate that we should also look carefully for other criteria besides the competence measure considered in related work [49, 50]. In this scenario, we can mention the distance function used in the kNN algorithm and the selection criterion used to select the most competent regressors from the ensemble, in which we were able to achieve statistically better performances by changing these DRS hyperparameters. Another essential property that should be highlighted is the embedded normalization in the proposed DRS framework. Our results and analyses demonstrated that this process also impacted a significant performance gain.

However, the method presented some drawbacks that include the computational cost and the high number of models required in the pool of regressors. Nonetheless, these aspects are not critical due to the fact that our method is developed to be used as an offline application.

Future works include (1) the expansion of the proposed technique to automatically determine the best combination of hyperparameters, as well as (2) the evaluation using a heterogeneous ensemble, and (3) the analysis with other normalization techniques than the MinMax. We also aim to (4) explore new strategies for defining the RoC and (5) propose and assess new criteria for selecting the most competent models from the ensemble, evaluating the method with different datasets.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-023-09155-y.

Acknowledgements This study was financed in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brazil (CNPq) under Grant 401992/2022-5.

Data availability The data used in this paper can be obtained by the following link: https://github.com/leandroensina/FADbF.

Declarations

Conflict of interest We declare that there is no conflict of interest regarding the publication of this paper.

References

1. Aleem SA, Shahid N, Naqvi IH (2015) Methodologies in power systems fault detection and diagnosis. Energy Syst 6:85–108. https://doi.org/10.1007/s12667-014-0129-1
2. Alencar GT, Santos RC, Neves A (2022) Euclidean distance-based method for fault detection and classification in transmission

lines. J Control Autom Electr Syst 33:1466–1476. https://doi.org/10.1007/s40313-022-00918-x
3. Amorim LB, Cavalcanti GD, Cruz RM (2023) The choice of scaling technique matters for classification performance. Appl Soft Comput 133:109924. https://doi.org/10.1016/j.asoc.2022.109924
4. Barandas M, Folgado D, Fernandes L et al (2020) Tsfel: time series feature extraction library. SoftwareX 11:100456. https://doi.org/10.1016/j.softx.2020.100456
5. Bhatnagar M, Yadav A, Swetapadma A (2023) Random forest regression-based fault location scheme for transmission lines. In: Dash RN, Rathore AK, Khadkikar V et al (eds) Smart technologies for power and green energy. Springer Nature Singapore, Singapore, pp 201–210
6. Bon NN, Dai LV (2022) Fault identification, classification, and location on transmission lines using combined machine learning methods. Int J Eng Technol Innov 12(2):91–109. https://doi.org/10.46604/ijeti.2022.7571
7. Boudraa AO, Salzenstein F (2018) Teager-kaiser energy methods for signal and image analysis: a review. Digit Signal Process 78:338–375. https://doi.org/10.1016/j.dsp.2018.03.010
8. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140. https://doi.org/10.1007/BF00058655
9. Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
10. Breiman L, Friedman J, Stone CJ et al (1984) Classification and regression trees. CRC Press, Boca Raton
11. Chen K, Huang C, He J (2016) Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. High Volt 1(1):25–33. https://doi.org/10.1049/hve.2016.0005
12. Chen K, Hu J, He J (2018) Detection and classification of transmission line faults based on unsupervised feature learning and convolutional sparse autoencoder. IEEE Trans Smart Grid 9(3):1748–1758. https://doi.org/10.1109/TSG.2016.2598881
13. Cruz RM, Oliveira DV, Cavalcanti GD et al (2019) Fire-des++: Enhanced online pruning of base classifiers for dynamic ensemble selection. Pattern Recognit 85:149–160. https://doi.org/10.1016/j.patcog.2018.07.037
14. Cruz RMO, Cavalcanti GDC, Ing Ren T (2011) A method for dynamic ensemble selection based on a filter and an adaptive distance to improve the quality of the regions of competence. In: International joint conference on neural networks, pp 1126–1133, https://doi.org/10.1109/IJCNN.2011.6033350
15. Cruz RMO, Sabourin R, Cavalcanti GDC (2018) Dynamic classifier selection: recent advances and perspectives. Inf Fusion 41:195–216
16. Cruz RMO, Sabourin R, Cavalcanti GDC (2018) Prototype selection for dynamic classifier and ensemble selection. Neural Comput Appl 29(2):447–457. https://doi.org/10.1007/s00521-016-2458-6
17. de Oliveira JFL, Silva EG, de Mattos Neto PSG (2022) A hybrid system based on dynamic selection for time series forecasting. IEEE Trans Neural Netw Learn Syst 33(8):3251–3263. https://doi.org/10.1109/TNNLS.2021.3051384
18. Deza MM, Deza E (2016) Encyclopedia of distances, 4th edn. Springer, Berlin
19. Ekici S, Yildirim S, Poyraz M (2008) Energy and entropy-based feature extraction for locating fault on transmission lines by using neural network and wavelet packet decomposition. Expert Syst Appl 34(4):2937–2944. https://doi.org/10.1016/j.eswa.2007.05.011
20. Ensina LA, Karvat PEM, Almeida EC, et al (2022a) Fault location in transmission lines based on lstm model. In: X Symposium on knowledge discovery, Mining and Learning, pp 1–8
21. Ensina LA, Oliveira LES, Almeida EC, et al (2022b) Fault classification in transmission lines with generalization competence. In: IECON 2022 - 48th Annual conference of the IEEE industrial electronics society, pp 1–6
22. Fan R, Yin T, Huang R, et al (2019) Transmission line fault location using deep learning techniques. In: 2019 North American Power Symposium (NAPS). IEEE, Wichita, KS, USA, pp 1–5
23. Fei C, Qi G, Li C (2018) Fault location on high voltage transmission line by applying support vector regression with fault signal amplitudes. Electr Power Syst Res 160:173–179. https://doi.org/10.1016/j.epsr.2018.02.005
24. Ferreira VH, Zanghi R, Fortes MZ et al (2016) A survey on intelligent system application to fault diagnosis in electric power system transmission lines. Electr Power Syst Res 136:135–153. https://doi.org/10.1016/j.epsr.2016.02.002
25. Ferreira VH, Zanghi R, Fortes MZ et al (2020) Probabilistic transmission line fault diagnosis using autonomous neural models. Electr Power Syst Res 185:1–10. https://doi.org/10.1016/j.epsr.2020.106360
26. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378. https://doi.org/10.1016/S0167-9473(01)00065-2
27. Furse CM, Kafal M, Razzaghi R et al (2021) Fault diagnosis for electrical systems and power networks: a review. IEEE Sensors J 21(2):888–906. https://doi.org/10.1109/JSEN.2020.2987321
28. Gafoor SA, Rao PVR (2006) Wavelet based fault detection, classification and location in transmission lines. In: 2006 IEEE International power energy conference., pp 114–118
29. Garcia S, Derrac J, Cano J et al (2012) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 34(3):417–435. https://doi.org/10.1109/TPAMI.2011.142
30. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42. https://doi.org/10.1007/s10994-006-6226-1
31. Godse R, Bhat S (2020) Mathematical morphology-based feature-extraction technique for detection and classification of faults on power transmission line. IEEE Access 8:38459–38471. https://doi.org/10.1109/ACCESS.2020.2975431
32. Gururajapathy SS, Mokhlis H, Illias HA (2017) Fault location and detection techniques in power distribution systems with distributed generation: a review. Renew Sustain Energy Rev 74:949–958. https://doi.org/10.1016/j.rser.2017.03.021
33. Haq EU, Jianjun H, Li K et al (2021) Improved performance of detection and classification of 3-phase transmission line faults based on discrete wavelet transform and double-channel extreme learning machine. Electr Eng 103:953–963. https://doi.org/10.1007/s00202-020-01133-0
34. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844. https://doi.org/10.1109/34.709601
35. Høidalen HK, Prikler L, Peñaloza F (2019) ATPDraw version 7.0 for Windows—users' manual
36. Ishwaran H (2015) The effect of splitting on random forests. Mach Learn 99(1):75–118. https://doi.org/10.1007/s10994-014-5451-2
37. Kordos M, Arnaiz-González Á, García-Osorio C (2019) Evolutionary prototype selection for multi-output regression. Neurocomputing 358:309–320. https://doi.org/10.1016/j.neucom.2019.05.055
38. Krogh A, Vedelsby J (1994) Neural network ensembles, cross validation and active learning. In: Proceedings of the 7th international conference on neural information processing systems. MIT Press, Cambridge, MA, USA, NIPS'94, pp 231-238

39. Kuncheva L (2002) A theoretical study on six classifier fusion strategies. IEEE Trans Pattern Anal Mach Intell 24(2):281–286. https://doi.org/10.1109/34.982906
40. Mallaki M, Dashti R (2012) Fault locating in transmission networks using transient voltage data. Energy Procedia 14:173–180. https://doi.org/10.1016/j.egypro.2011.12.914
41. Mendes-Moreira J, Jorge AM, Soares C, et al (2009) Ensemble learning: A study on different variants of the dynamic selection approach. In: Perner P (ed) machine learning and data mining in pattern recognition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 191–205
42. Mendes-Moreira J, Soares C, Jorge AM et al (2012) Ensemble approaches for regression: a survey. ACM Comput Surv 10(1145/2379776):2379786
43. Mendes-Moreira J, Jorge AM, Freire de Sousa J et al (2015) Improving the accuracy of long-term travel time prediction using heterogeneous ensembles. Neurocomputing 150:428–439. https://doi.org/10.1016/j.neucom.2014.08.072
44. Mirzaei M, Vahidi B, Hosseinian SH (2018) Fault location on a series-compensated three-terminal transmission line using deep neural networks. IET Sci Meas Technol 12(6):746–754. https://doi.org/10.1049/iet-smt.2018.0036
45. Mishra DP, Ray P (2018) Fault detection, location and classification of a transmission line. Neural Comput Appl 30(5):1377–1424. https://doi.org/10.1007/s00521-017-3295-y
46. Moore DG (2019) Pyinform. https://elife-asu.github.io/PyInform/, accessed 18 August 2023
47. Morais J, Pires Y, Cardoso C et al (2010) A framework for evaluating automatic classification of underlying causes of disturbances and its application to short-circuit faults. IEEE Trans Power Deliv 25(4):2083–2094. https://doi.org/10.1109/TPWRD.2010.2052932
48. Moura TJ, Cavalcanti GD, Oliveira LS (2021) Mine: a framework for dynamic regressor selection. Inf Sci 543:157–179. https://doi.org/10.1016/j.ins.2020.07.056
49. Moura TJM, Cavalcanti GDC, Oliveira LS (2019) Evaluating competence measures for dynamic regressor selection. In: 2019 International joint conference on neural networks (IJCNN), pp 1–8
50. Moura TJM, Cavalcanti GDC, Oliveira LS (2020) On the selection of the competence measure for dynamic regressor selection. In: 2020 IEEE International conference on systems, Man, and Cybernetics (SMC), pp 1630–1637
51. Mukherjee A, Kundu PK, Das A (2021) Transmission line faults in power system and the different algorithms for identification, classification and localization A brief review of methods. J Inst Eng India Ser B 102(4):855–877. https://doi.org/10.1007/s40031-020-00530-0
52. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
53. Python (2023) statistics—mathematical statistics functions. https://docs.python.org/3/library/statistics.html, accessed 18 August 2023
54. Ray P, Panigrahi BK, Senroy N (2013) Hybrid methodology for fault distance estimation in series compensated transmission line. IET Gener Transm Distrib 7(5):431–439. https://doi.org/10.1049/iet-gtd.2012.0243
55. Raza A, Benrabah A, Alquthami T et al (2020) A review of fault diagnosing methods in power transmission systems. Appl Sci 10(4):1–27. https://doi.org/10.3390/app10041312
56. Rooney N, Patterson D, Anand S, et al (2004) Dynamic integration of regression models. In: Roli F, Kittler J, Windeatt T (eds) Multiple classifier systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 164–173
57. Sani US, Malik OA, Lai DTC (2022) Dynamic regressor/ensemble selection for a multi-frequency and multi-environment path loss prediction. Information 13(11):519
58. Schapire RE, Freund Y, Bartlett P et al (1998) Boosting the margin: a new explanation for the effectiveness of voting methods. Ann Stat 26(5):1651–1686. https://doi.org/10.1214/aos/1024691352
59. Sergio AT, de Lima TP, Ludermir TB (2016) Dynamic selection of forecast combiners. Neurocomputing 218:37–50. https://doi.org/10.1016/j.neucom.2016.08.072
60. Shakiba FM, Azizi SM, Zhou M et al (2023) Application of machine learning methods in fault detection and classification of power transmission lines: a survey. Artif Intell Rev 56(7):5799–5836. https://doi.org/10.1007/s10462-022-10296-0
61. Silva EG, De Mattos Neto PSG, Cavalcanti GDC (2021) A dynamic predictor selection method based on recent temporal windows for time series forecasting. IEEE Access 9:108466–108479. https://doi.org/10.1109/ACCESS.2021.3101741
62. Swaminathan R, Mishra S, Routray A et al (2021) A cnn-lstm-based fault classifier and locator for underground cables. Neural Comput Appl 33(22):15293–15304. https://doi.org/10.1007/s00521-021-06153-w
63. Swetapadma A, Yadav A (2017) A novel decision tree regression-based fault distance estimation scheme for transmission lines. IEEE Trans Power Deliv 32(1):234–245. https://doi.org/10.1109/TPWRD.2016.2598553
64. Swetapadma A, Yadav A (2018) A novel single-ended fault location scheme for parallel transmission lines using k-nearest neighbor algorithm. Comput Electr Eng 69:41–53. https://doi.org/10.1016/j.compeleceng.2018.05.024
65. Ueda N, Nakano R (1996) Generalization error of ensemble estimators. In: Proceedings of international conference on neural networks (ICNN'96), pp 90–95
66. Valabhoju A, Yadav A, Pazoki M et al (2021) Optimized ensemble of regression tree-based location of evolving faults in dual-circuit line. Neural Comput Appl 33(14):8795–8820. https://doi.org/10.1007/s00521-020-05628-6
67. Wang B, Mao Z (2020) A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. Inf Fusion 63:30–40. https://doi.org/10.1016/j.inffus.2020.05.001
68. Wang X, Zhou P, Peng X et al (2022) Fault location of transmission line based on CNN-LSTM double-ended combined model. Energy Rep 8:781–91. https://doi.org/10.1016/j.egyr.2022.02.275
69. Wang Y, Cui Q, Weng Y et al (2023) Learning picturized and time-series data for fault location with renewable energy sources. Int J Electr Power Energy Syst 147:108853. https://doi.org/10.1016/j.ijepes.2022.108853
70. Yadav A, Dash Y (2014) An overview of transmission line protection by artificial neural network: fault detection, fault classification, fault location, and fault direction discrimination. Adv Artif Neural Syst 2014:1–20. https://doi.org/10.1155/2014/230382