IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 29, 2019, accepted August 5, 2019, date of publication August 13, 2019, date of current version August 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934938

Transmission Line Fault Classification Using Hidden Markov Models

JEAN CARLOS AROUCHE FREIRE[1,2], ADRIANA ROSA GARCEZ CASTRO[1], MARCIA SALOMÃO HOMCI[2], BIANCHI SERIQUE MEIGUINS[2], AND JEFFERSON MAGALHÃES DE MORAIS[2]

[1]Faculty of Electrical Engineering, Federal University of Pará, Belém 66.075.110, Brazil
[2]Faculty of Computer, Federal University of Pará, Belém 66.075.110, Brazil

Corresponding author: Jean Carlos Arouche Freire (jeanarouche@gmail.com)

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) Foundation of the Brazilian Ministry of Education and Culture.

ABSTRACT The maintenance of power quality in electrical power systems depends on addressing the major disturbances that may arise during generation, transmission and distribution. Many studies aim to investigate these disturbances by analyzing the behavior of the electrical signal through the classification of short circuit faults in power transmission lines as a way to assist the administration and maintenance of the electrical system. However, most fault classification methods generate a high computational cost that do not always yield satisfactory results; these methods utilize front ends in data processing before being processed by conventional classification algorithms such as Artificial Neural Network (ANN), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF) that are adopted into the Frame Based Sequence Classification (FBSC) architecture that uses the front ends Waveletenergy, Waveletconcat, RAW, Root Mean Square front ends (RMS) and ConcatFrontEnd. An alternative method for classifying faults without having to use front ends employs the UFPAFaults database and the Hidden Markov Model (HMM) algorithm that directly treats the electrical signal in the form of multivariate time series. The results indicate the HMM algorithm as a potential classifier because its comparatively low error rate of 0.03% exceeds the performance of the conventional classifiers ANN, SVM, KNN and RF as used with the FBSC architecture. When the statistical test with a significance of $\alpha = 5\%$ is applied, only the ANN and RF classifiers present a result close to what the HMM algorithm provides. Another relevant factor is that the HMM algorithm considerably decreases the computational cost by more than 90% of processing time as compared to the conventional classifiers of the FBSC architecture, thereby validating its potential in the direct classification of faults in electric power system transmission lines.

INDEX TERMS Electric power quality, electrical power systems, short circuit, classification of faults, hidden markov model.

I. INTRODUCTION

The transmission line is the component of an Electric Power System (EPS) that is most vulnerable to faults, especially when considering that from one end to another, the line is subject to various diverse types of natural phenomena such as atmospheric discharge, forest fires, windstorms, and more. Such situations can cause disturbances (faults) in the transmission line, which can then cause an interruption in the transmission of electric energy.

The associate editor coordinating the review of this article and approving it for publication was Chao Shen.

Among the faults that can occur in a transmission line, short circuit type faults are the ones that have the greatest impacts on consumers. Studies have shown that these faults account for about 70% of electrical system disturbances and blackouts [1]. Thus, the need for EPSs to adopt mechanisms for diagnosing and identifying such faults is evident, as is the necessity for analyzing the electrical signal behavior through short circuit fault classification as a way to assist in the maintenance and restoration of the power supply.

The transmission in-line fault classification systems can be divided into two types: online classification systems and post-fault classification systems [2]. Online classification systems make a decision (classification) in a short period of time, with

the analysis segment (or frame) occurring approximately at the instant in which the fault occurs. Post-fault classification can be executed offline, and its input consists of a multivariate variable duration (length) time series, which differs from online classification in that entry is a vector of fixed size. Online and post-fault systems attempt to solve problems that can be treated as problems of conventional classification and of sequence, respectively [3]. This paper focuses on the classification of sequence.

In the sequence classification representing faults, it is possible to use machine learning techniques such as Artificial Neural Networks (ANN) [4], [5] and Support Vector Machine (SVM) [6], [7], among others. In this case, the classification process requires pre-processing, or a front end stage that converts the raw data into sensitive parameters to feed into the back end (in this case, the classifier).

Various works based on machine learning and different types of front ends have been proposed for classifying short-circuit faults in transmission lines. For example, impulse response filter two-stage finite element in conjunction with SVM are used in [1]. In some research, wavelet transform was combined with other techniques such as k-nearest neighbor (KNN), ANN, SVM and Fuzzy Systems, composing hybrid frameworks for fault classification [8]–[11]. In [12], an approach for fault classification is proposed that combines independent component analysis (ICA) with the theory of traveling waves (TW) and SVM. In [13], another methodology using empirical mode decomposition (EMD) and SVM is proposed. Other front ends, such as Fourier [14] and root mean square (RMS) [15], are also used in fault classification.

Initiated in [16] and updated in [17], the frame based sequence classification (FBSC) architecture was proposed for fault classification in transmission lines. The main idea of the FBSC architecture is to segment an input sequence into fixed-length vectors called frames and repeatedly invoke a conventional classifier (eg. ANN and SVM) to process each frame. The FBSC classifier considers the outputs of the conventional classifier and, through a voting process, comes to a final decision by observing which class is more frequent. The problem with the FBSC architecture is that it has many degrees of freedom in the design of the model (front end plus classifier), and should be evaluated using a complete data set and a rigorous methodology in order to avoid biased conclusions.

As an alternative to the FBSC architecture, this study proposes the adoption of the Hidden Markov Model (HMM) algorithm. HMM is an efficient standards recognizer that has the ability to classify the event directly in a probabilistic way based on the characteristics and waveform of the fault [18], [19]. The differentiated ability of HMM to handle classification problems with small, non-linear, and high-dimensional sample numbers makes this algorithm a potential choice for application in fault classifications in EPSs. These differentiated features of the HMM algorithm make it special in relation to other classification algorithms because it is able to directly recognize an electrical signal with varying sizes even if they belong to the same fault class.

Some studies suggest that HMM can be associated with other techniques such as wavelet and vector quantization [20], [21], partial discharge pulse pattern recognition [22], accident identification and decision making in power plants [23], and power transformer failure diagnosis based on gas [24]. In the studies of [25], [26], HMMs were specifically applied to the differential protection of the power transformer. Reference [27] highlights the inherent probabilistic characteristic of HMM that makes it flexible for different operating conditions and requires low computational cost for decision making (classification) after training. This advantage makes HMM a very attractive option for fault diagnosis in power transmission lines.

The main objective of this study is to apply the HMM algorithm for fault classification in power transmission line as a way of auxiliary administration and maintenance of EPSs, accomplished by observing its performance with the UFPAFaults database. The results were compared with the FBSC architecture, and statistical tests were conducted to verify the degree of significance in the results of the classifiers used in the study.

This work is organized as follows: Section II explains the reasons that motivated the study, the front ends that were used, the FBSC architecture, and details of the HMM algorithm and statistical tests. In Section III, the research methodology is presented together with basic information about the UFPAFaults database. Section IV presents the results and discussions, and finally, Section V discusses the conclusions and future work of fault classification using the HMM algorithm.

II. BACKGROUND

The growing demand for electric power and the increase in consumption of electro-electronic equipment vulnerable to electrical disturbances entails a greater need for good, quality energy. This reality requires that EPSs possess more acceptable configurations of the physical, operational, and control infrastructures that can help avoid and reduce electrical disturbances, hereafter referred to as short circuit faults in transmission lines [4].

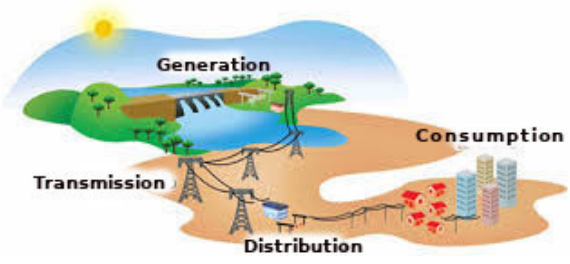

FIGURE 1. Representation of the electric power system.

A typical EPS, as illustrated in Figure 1, is generally divided into three functional zones before the energy reaches

the final consumer. These three functional zones are: generation, transmission and distribution [16]. Such functional zones are subject to the occurrence of natural or man-made disturbances. As a consequence, the voltage or current waveforms undergo certain changes and deviate from their nominal values, resulting in the so-called quality of energy events.

Thus, the need for EPSs to adopt increasingly efficient fault classification mechanisms is evident, so as to assist the decision making process at the operational level responsible for restoring the EPSs [19].

In transmission lines, fault classification is represented by a sequence, generally found in three-phase electrical systems, whereby each fault can be considered a multivariate temporal series of variable duration. The nth fault X_n can be represented by a matrix $Q \times N$. The column x_t of $X_n, t = 1, \ldots, T_m$ is a multidimensional sample represented by a vector of Q elements, where Q is the number of signals and T_m is the number of samples of the fault. This study adopts $Q = 6$ (voltage and current waveforms of phases A, B and C) in the experiments. Since the number of multivariate samples depends on n, a conventional classifier is not feasible; this is one of the main reasons in favor for adopting the HMM algorithm, because it exmploys a direct classification of multivariate samples.

Most power transmission systems have three phases: A, B and C. For example, a short circuit between phases A and B may be identified as a fault AB. Considering the possibility of a short circuit with the earth phase (T), this work takes into account the sequence classification in 11 possible faults that will be used in the classifiers, which are: AT, BT, CT, AB AC, BC, ABC, ABT, ACT, BCT and ABCT. However, only 10 classes will be considered (ABC = ABCT) because the simulated data in the UFPAFaults database is based on a balanced system where there is no current flowing through the earth [8].

The classification of faults in transmission lines corresponds to a special classification problem where the voltage and current waveforms represent sequences (time series) of varied size, that is, the input data is represented by a matrix of variable size. Thus, the conception of a classification model that deals with this input matrix can be done in several ways: indirectly, with the use of front ends associated with conventional classification algorithms such as ANN, KNN, Random Forest (RF) and SVM used by the FBSC architecture, or with algorithms that use mechanisms more flexible than front ends, such as the HMM algorithm adopted in this study. Figure 2 presents the classification model with the HMM algorithm, the variable X_n of entries (time series representing the base faults UFPAFaults), and their respective outputs Y_n (Types of Faults).

A. THE FBSC ARCHITECTURE

The FBSC architecture makes use of front ends that organize the data into a fixed-size array to be processed by a conventional classifier. A conventional classifier F is a mapping $f : \mathbb{R}^K \rightarrow \{1, \ldots, Y\}$, where K is the dimension of the input vector $z \in \mathbb{R}^K$ and the label $y \in \{1, \ldots, Y\}$ is the class. A training set $T = \{(z_1, y_1), \ldots, (z_v, y_v)\}$ containing V samples of (z, y) is used to train a conventional classifier [16], [17]. The classification process takes place after the execution of the F classifier, following a sequence starting with the submission of the samples of variable size to a parameter extractor (front end), which converts a matrix X representing the faults in a matrix Z with dimension K x N_n, where K is the number of parameters and N_n is the number of parameter vectors of the *nth* example.

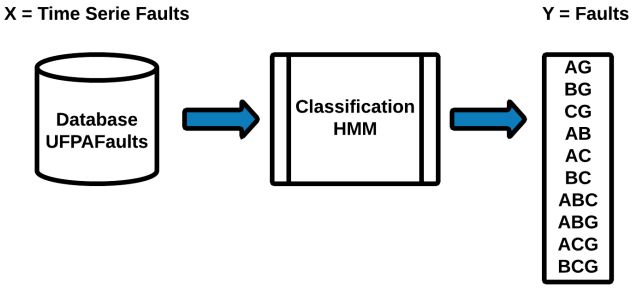

FIGURE 2. Classification model used in the study.

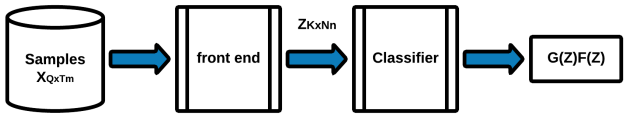

FIGURE 3. FBSC architecture processing flow. the output of the $G(Z)$ sequence classifier depends on the classifier's decisions $F(z_n)$. source: [16].

As seen in Figure 3, this architecture performs classification in the Z matrix and not in X. The X matrix is composed of the concatenation of the original and organized samples into a matrix with dimensions Q x L, where L is the size of the frame L_{min} specified by the user, and its concatenation is $\hat{Z} = [F_1, \ldots, F_{N_n}]$, thereby generating a matrix with dimensions Q x L_n, where N is the number of frames. An overlap is considered a displacement S (or quantity of samples between the beginning of two consecutive frames), and can be smaller than the size of the window with a value of S_{min} specified by the user. A fault X_n is represented by the number of frames N_n.

$$N_n = 1 + \lfloor (T_n - L)/S \rfloor \tag{1}$$

The *floor* function is represented by $\lfloor - \rfloor$. When the offset is equal to the frame size, $S = L$ (no overlap) and a concatenation of samples represent a frame where the matrices $X = \hat{Z}$ coincide.

The classification error rate is a parameter used to evaluate the performance of a sequence classifier $G(Z)$.

$$E_s = \frac{1}{R} \sum_{r=1}^{R} I(G(Z_r)! = y_r) \tag{2}$$

I is the indicator function, which is 1 if the argument is true and 0 otherwise. R represents the test sequence. In the

case of the FBSC architecture, E_s depends directly on the performance of the F classifier.

B. FRONT ENDS

Any sample that represents a fault does not contain sufficient characteristics that allow for a decision using a classification algorithm. For this reason, a front end has the function of converting the samples into parameters (features), thereby generating a sequence that allows for reasonable decisions through the conventional classifiers.

The front end Raw is the simplest, because its output parameters correspond to values of the original sample without needing any other processing that organizes the samples into the Z matrix where the classification will occur [17].

Another widely used front end is RMS, which organizes the data that allow for an approximate estimate of the amplitude of the fundamental frequency of the waveform. This front end consists of calculating the windowed RMS value for each of the waveforms [13], [15].

The front ends wavelets are discussed by much of the existing research, as they enable a number of implementations. On the other hand, special care must be taken in replicating a front end, in order to avoid jobs with unfeasible results. In this study, the front end Wavelet concatenates and organizes all coefficients in a Z matrix, taking into account the coefficients that have different sampling frequencies and forming a table with organized coefficients. This process is called the Waveletconcat front end. Another front end called Waveletenergy is an alternative for organizing Wavelet coefficients that uses the average energy of each coefficient, and, similar to Waveletconcat, treats the signals of different sampling frequencies. The main difference between these front ends is that instead of concatenating all the coefficients, the front end Waveletenergy calculates the energy of short intervals, representing X by means of energy E in each frequency band obtained by the composition Wavelet [11].

There is also the front end ConcatFrontEnd that is presented in [17], which is a combination of all previous front ends and is differentiated from the others in that it takes advantage of all effective characteristics to provide a more precise classification with better results.

C. HIDDEN MARKOV MODEL FOR FAULT CLASSIFICATION ON TRANSMISSION LINES

An HMM $X_t : t \in \mathbb{N}$ is defined as a set of dependent observations with temporal records in the time 1 to t, $X^{(t)}$ and $C^{(t)}$, where $X^{(t)}$ represents the sequences $X_1, X_2, \ldots, X_t$ of observed values, and $C^{(t)}$ represents the sequence $C_1, C_2, \ldots, C_t$ of hidden states. The structure of an HMM model is defined by the relation of its hidden-state probabilities P_{ocult} (Equation 3) and probabilities of observed sequences P_{obs} (Equation 4) as shown in Figure 4. Being an algorithm with probabilistic characteristics, its main advantage is to recognize and adjust accordingly to new data where there is incomplete information about the source from which the sequences with varied sizes are generated, even though

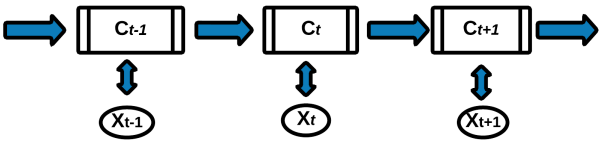

FIGURE 4. Diagram of the relationship between hidden states $C^{(t)}$ and obeserved values $X^{(t)}$ of an HMM model. source: [29].

they belong to the same model (class) [18], [28].

$$P_{ocult} = (C_t|C^{(t-1)}) = Pr(C_t|C_{t-1}), \quad t = 2, 3, \ldots \tag{3}$$

$$P_{obs} = (X_t|X^{(t-1)}, C^{(t)}) = Pr(X_t|C_t), \quad t \in \mathbb{N}. \tag{4}$$

The formal parameters of a discrete HMM model with $X^{(t)}$ sequences of observed values and $C^{(t)}$ hidden states can be represented by $\lambda = (\delta, \Gamma, \pi)$, where Γ is the state transition matrix, δ is the set of probability functions of observations (usually normal distribution - Gaussian), and π is a vector of dimension S, composed of a probability density function from the initial states. The element a_{ij} of δ is the probability of performing a $\sum$ of transition from state i to j, with i $= 1, \ldots,$ S $a_{ij} = 1, \forall\ i$. In some models (in the software R adopted in this study, for example), the vector π has been incorporated into a diagonal matrix P, from the definition of two states that do not emit an exit symbol. That is, $\lambda = (\delta, \Gamma)$ is used in HMM models so that condition to remain in the current state C_2 or go on to the next state C_1 obeys a left-right topology that is most commonly used to model time series [29]. Figure 5 shows the topology used in the study for the ten HMM models that represent the fault classes in transmission lines.

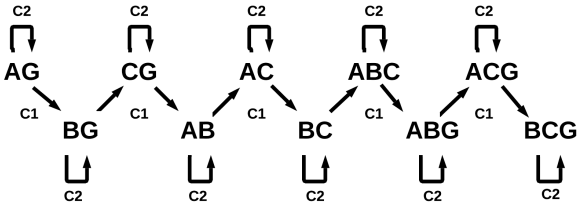

FIGURE 5. Structure of the HMM algorithm adopted in the study with left-right topology.

D. TRAINING PROCESS WITH THE HMM ALGORITHM

The training with the HMM algorithm is performed by estimating the expected model. Suppose a sequence of observations $X^{(t)}$ originates from an HMM (a class Y) model. The HMM associated with such a model has m states, an initial distribution δ, and transition matrix Γ, with the functions of probability (or density) of dependent-states $p_i, i = 1, 2, \ldots, i_n$. The probability of observing such a sequence L_X under the HMM (Y) model described is said to be the **likelihood function** $L_X Y$, given by Equation 5.

$$L_X Y = \delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \ldots \Gamma P_{(x_t)} 1' \tag{5}$$

For [29], the likelihood calculation requires a higher quantity of mathematical operations, but the use of recursive means causes this number to decrease considerably. In the study, one way to achieve this goal was to make use of the forward probability (Equation 6) and backward probability (Equation 7) through the Baum-Welch algorithm to perform the estimation of the maximum likelihood of samples with size T of each HMM model.

$$\alpha_t = \delta P_{(x_1)} \Gamma P_{(x_2)} \ldots \Gamma P_{(x_t)} = \delta P_{(x_1)} \prod_{s=2}^{T} \Gamma P_{(x_s)} \tag{6}$$

$$\beta_t = \Gamma P_{(x_{x+1})} \Gamma P_{(x_{x+2})} \ldots \Gamma P_{(x_t)} 1' = \left(\prod_{s=t+1}^{T} \right) 1' \tag{7}$$

The Baum-Welch algorithm is a frequently used optimization method to find maximum likelihood estimators when data is incomplete, considering the fact that the likelihood (or log-likelihood) of the complete data is probably simpler to maximize than only evaluating the (incomplete) data observed. The algorithm consists of two steps: (E = (Expectation) and M = (Maximization)). To use the algorithm, consider that a sequence of size T of an HMM has been observed. Then, in the E step, the definition will be $C_1, C_2, \ldots, C_t$ as the sequence of HMM states with the function

$$u_j(t) \begin{cases} 1, & only \quad c_t = j, \ for \ t = 1, 2, \ldots T \\ 0, & otherwise \end{cases}$$

where $u_j(t)$ is the estimate of the probability that the process is in the state j at time $t(C_t = j)$, given what is observed in sequence X_t. This is calculated by Equation 8.

$$u_j = \frac{\alpha_t(j)\beta_t(j)}{\delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \ldots \Gamma P_{(x_t)} 1'} \tag{8}$$

Another function

$$v_{jk}(t) \begin{cases} 1, & only \quad c_{t-1} = j, \ for \ k = 2, 3 \ldots T \\ 0, & otherwise \end{cases}$$

represents the estimation of the probability of the process to walk from state j to k at times $t-1$ and $t(C_{t-1} = j, C_t = k)$, given what is observed in sequence X_t. This is calculated in Equation 9

$$v_{jk} = \frac{\alpha_{t-1}(j)\gamma_{jk}\rho_k\beta_t(j)}{\delta \Gamma P_{(x_1)} \Gamma P_{(x_2)} \ldots \Gamma P_{(x_t)} 1'} \tag{9}$$

where γ_{jk} are the transition probabilities that make up Γ, and ρ_k are the initial state distributions that make up δ. Thus, for the M step, after calculating $u_j(t)$ and v_{jk}, one must maximize each term, since term 1 depends only on δ, term 2 on Γ and term 3 on the parameters of the dependent-state distribution. Thus, the parameter estimators are calculated by Equation 10

$$\delta = \frac{\hat{u}_j(1)}{\sum_{j=1}^{m} \hat{u}_j(1)} \tag{10}$$

which defines an estimate for the initial probability of the state j (or the estimation of the probability of the process being in the state j in time $1(C_1 = j)$) since the sequence X_t was observed. Thus, the maximum likelihood of the complete data in a sample $L_X Y_{max}$ is given by Equation 11.

$$L_X Y_{max} = \left(\delta_{c1} \prod_{t=2}^{T} \gamma_{c_{t-1},c_t} \prod_{t=1}^{T} \rho_{c_t}(x_t) \right) \tag{11}$$

The model $M_{L_X Y_{max}}$ of each expected HMM is obtained by the arithmetic mean (μ) of the maximum likelihood $L_X Y_{max}$ of the set of samples $X^{(T)}$ of each class Y.

As so, in the training process, the HMM algorithm uses a model $M_{L_X Y_{max}}$ for each class with the default set of λ parameters for all classes that were previously estimated. This involves, in addition to the use of the initial and transition probabilities, the means (μ) and variances (σ) of each observed sequence that, depending on their variation, may provide a good result [18], [29]. A sequence X_t will be classified as of a given class Y if the maximum likelihood of this sequence for the model of this class $(M_{L_X Y_{max}} | \lambda_X = L_X Y)$ is greater than the maximum likelihood of all other models of the other classes. That is, for Y classes, the sequence X_t is of class Y if

$$L_X Y > M_{L_X Y_{max}}, L \neq M = 1, 2, \ldots M_n \tag{12}$$

and also if the label of the class of the sequence X_{Label} is equal to the label of the class of M_{Label} HMM model. Figure 6 describes the steps in the HMM algorithm training process.

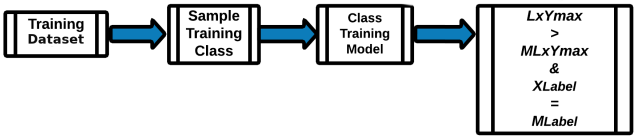

FIGURE 6. Process of training the HMM algorithm.

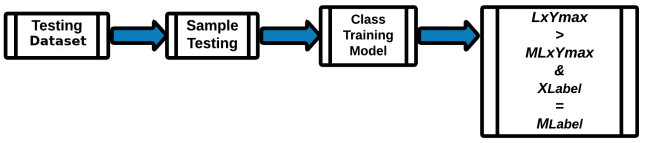

FIGURE 7. Process of testing the HMM algorithm.

E. PROCESS OF TESTING THE HMM ALGORITHM

The testing process of the HMM algorithm follows the same mechanism used in the training. That is, a test sequence X_t will be classified as of a given class Y if the maximum likelihood $L_X Y$ of this sequence for the model of this class $M_{L_X Y_{max}}$ estimated in training is greater than the maximum likelihood of all other models of the other classes, and also if the label of the class of the test sequence X_{Label} coincides with the label of the class M_{Label} HMM obtained in the training. Figure 7 shows the steps of the testing process.

As an example of the testing process, consider five sequences of the test dataset X_1, X_5, X_{11}, X_{37} and X_{42} as

TABLE 1. Example of classification of maximum likelihood of a class.

Sequences	X_1	X_5	X_{11}	X_{37}	X_{42}
Training Base Origin	AG	**AG**	BG	ABC	BG
Testing Base $L_X Y_{max}$	13.1	**9.14**	11.10	9.42	10.90
M_{L_X} AG	13.12	13.12	13.12	13.12	13.12
M_{L_X} BG	11.2	11.2	11.2	11.2	11.2
M_{L_X} CG	9.23	**9.23**	9.23	9.23	9.23
M_{L_X} AB	13.02	13.02	13.02	13.02	13.02
M_{L_X} AC	12.04	12.04	12.04	12.04	12.04
M_{L_X} BC	10.70	10.70	10.70	10.70	10.70
M_{L_X} ABC	9.56	9.56	9.56	9.56	9.56
M_{L_X} ABG	11.79	11.79	11.79	11.79	11.79
M_{L_X} ACG	12.93	12.93	12.93	12.93	12.93
M_{L_X} BCG	7.88	**7.88**	7.88	7.88	7.88
Classification	AG	**CG**	BG	ABC	BG

shown in Table 1, where the sequence X_5 has a maximum likelihood of 9,14 that originated and was trained from the model AG. After being compared with the maximum likelihoods of the other models, it becomes evident that it is larger than the BCG model of mean 7.88, and smaller than the CG model with average 9.23 and the others that also have higher averages. In this scenario, the sequence X_5 creates an error because it is classified as the CG model. The same does not happen with the other sequences that are classified correctly.

F. STATISTICAL TEST

Statistical tests correspond to a decision rule that allows for accepting a hypothesis based on the results of samples. The authors of [30] state that the statistical test is a mechanism of precise significance for prediction samples, since it contains statistical calculation resources more suited for classification tasks. This technique was adopted in the study to compare the classification results between two classifiers for which the following hypotheses below were tested:

- $H_0 : \overline{X}_0 = \overline{X}_1$ there is no significant difference between the mean error rates of classifiers 1 and 2.
- $H_1 : \overline{X}_0 \neq \overline{X}_1$ there is a significant difference between the mean error rates of classifiers 1 and 2.

From Equation 13

$$t = \frac{(\bar{X}_1 - \bar{X}_2)}{S_{x_1x_2} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{13}$$

where $S_{x_1x_2}$ is

$$S_{x_1x_2} = \sqrt{\frac{(n_1 - 1)S_{x_1}^2 + (n_2 - 1)S_{x_2}^2}{n_1 + n_2 - 2}} \tag{14}$$

being that the degree of freedom for these cases equal is to

$$df = n_1 + n_2 - 2 \tag{15}$$

where

- $\bar{X}_1$ corresponds to the mean of the error rate of classifier 1.
- $\bar{X}_2$ corresponds to the mean of the error rate of classifier 2.
- n_i is the number of *folds* (experiments) for each classifier.
- S_{x_i} corresponds to the standard deviation of the classifier error rate i.

The basic idea is that there is a null hypothesis H_0 and another alternative H_1 to then compare the results obtained in the table values t_{tab} of the statistical distribution according to the degrees of freedom df. The calculated value t_{calc} is extracted from Equation 13 (difference of the error rates between two classifiers) and the value of t_{tab} is obtained according to the t-Student [27]. Table 2 presents a few values of significance α with degrees of freedom of 1, 8, 9 and 10 units. The study takes into account five experiments (*folds*) for each classifier. Then, the degree of freedom is conditioned by a reference value of 8 after the calculation of Equation 15 in the distribution table for the statistical tests.

TABLE 2. Part of table t-Student with values of t, according to degree of freedom and the value of α equal 10%, 5% and 1%. source: [30].

Degree of Freedom (df)	10%	5% (α)	1%
1	6,31	12,71	63,66
8	1.86	**2.31**	3,36
9	1.83	2.26	3.25
10	1.81	2.23	3.17

III. METHODOLOGY

As was previously discussed, the purpose of this paper is to apply the HMM algorithm in the classification of faults in electric power transmission lines, with the study's differentiating factor being that it eliminates pre-processing steps (front ends). The research methodology is presented in Figure 8.

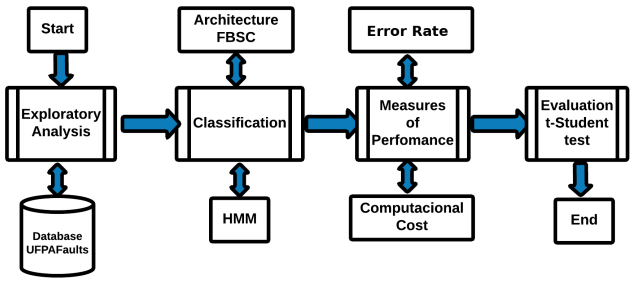

FIGURE 8. Flowchart of the methodology used in the paper.

Given the information in Figure 8, the methodology was divided into four steps. In the first step, an exploratory analysis of the UFPAFaults database was performed. Next, the experiments were carried out to classify faults in power transmission lines with the HMM algorithm and the FBSC architecture. Afterwards, the error rate and the computational cost of the algorithms involved in the classification were analyzed. Finally, the statistical test was applied to compare and validate the obtained results.

A. EXPERIMENT SETTINGS

In the general configurations of all the experiments, 10 training datasets with different amounts of samples distributed in 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 signals and one testing dataset with 1000 samples were used. It was also necessary to use a trigger that discarded the samples when the waveform did not present any type of anomaly, and allowed only the segments where a fault was detected to pass to the subsequent stages. All the training datasets were trained in five experiments, individually tested with the test sample of 1000 samples in a machine with an intel i7 processor and 16 GB of RAM with varying values of the classifiers parameters.

For experiments with the FBSC architecture, after defining the general settings, a Java language script was developed in order to format the ''txt'' files of the UFPAFauts database, generated by each front end, to the file pattern ''Arff'' of the WEKA software, an open source General Public License (GPL) tool composed of a set of machine learning algorithms [16]. Subsequently, the experiments of [17] with the conventional classifiers ANN, SVM, KNN and RF were replicated in the classification of short circuit type faults in power transmission lines made available in WEKA software.

Regarding the HMM algorithm and due to working directly with the samples in the database, after defining the general configurations of the experiments a script in Java language was developed to format the original files of the UFPAFaults database into new files with a ''txt'' extension. The generated file has already been used to give input directly into the HMM classifier. The algorithm is implemented in R language in the R STUDIO software, an open source tool under the General Public license (GPL), and has been properly tested and validated according to the specifications set out in [29]. Finally, we applied statistical tests to the results obtained between conventional classifiers and HMM to compare and prove the equality or difference of the classification results.

B. UFPAFAULTS DATABASE

This study uses a database with short circuit fault simulations on transmission lines called UFPAFaults. This database was developed by the research group of the Laboratory of Signal Processing (LaPS) of the Federal University of Pará (UFPA), is a database of public domain, is properly labeled, and can be found for consultation at https://github.com/jeanarouche/HMM-KNN-DTW-FaultClassification. The additional scripts used in this paper can also be found for reference.

The base has 27500 simulations, organized into five sets of 100, 200, . . . , 1000 faults each. All simulations are performed within the 1 second interval in voltage and current waveforms representing short-circuit faults in transmission lines. The waveforms of the signal were generated by the software simulators Alternative Transient Program (ATP) and AmazonTP, where they had a sampling period equal to 0.25 microseconds (in ATP, deltaT = 2.5E-5), which corresponds to a sampling frequency of f = 40kHz [8]. The voltage and current phases are represented by A, B, and C, and an ''AB'' fault is identified when the short-circuit occurs between phases A and B. Considering the possibility of a short-circuit with the ground phase (G), ten possible causes are found: AG, BG, CG, AB, AC, BC, ABC, ABG, ACG and BCG.

IV. RESULTS AND DISCUSSIONS

The results and discussions of the experiments regarding fault classifications in power transmission lines in an offline scenario are laid out in this section to further clarify what was proposed by the research.

A. RESULTS OF THE FBSC ARCHITECTURE EXPERIMENTS

This section presents the results of replicate experiments using the FBSC architecture and the UFPAFaults dataset. The front ends adopted were Waveletconcat, Waveletenergy, Raw, RMS and ConcatFrontEnd, while the conventional classifiers employed were ANN, RF, KNN and SVM and were all from the WEKA software package. These classifiers were chosen because they are popular representatives of different learning paradigms [16], [17], [31].

The logical combination of the front ends and the choice of values that define the parameters of each classifier was made from an automatic model selection adopted in [16]. Table 3 presents the best values for the parameters of each conventional classifier.

TABLE 3. Result of grid of selection of model of conventional classifiers.

Classifier	Parameters	Values Grid
ANN	H	11, 5, 10, 20, 40, 80 and **160**
	L	0,1, 0,5, and **0,9**
	M	0,1, 0,2 and **0,4**
RF	I	**100**, 200, 300, ..., and 1000
SVM	G	0.1, 1, **100**
	C	**0.1**, 1, 10 and 100
KNN	K	**1**, 3, 5, 7, 9, 11, 13 e 15

As can be seen in Table 3, the replications of the experiments performed for each classifier had a reasonable amount of variation in the values of their parameters until they reached an optimal value. For ANN in the *H* parameter, 160 neurons in the hidden layer was the value needed for better network convergence. On the other hand, RF needed 100 trees in *I* for better performance, while SVM used 100 and 0.01 in *G* and *C* respectively to reach the same goal. And finally, in KNN, one nearest neighbor in the *K* parameter was enough to reach a better rating value.

The error rate of each conventional classifier and its respective front ends in the FBSC architecture had lower performances in the training datasets with fewer samples. As these samples increase in the posterior datasets, the error decreases. It is observed that on average, most of the classifiers performed better with the front end ConcatFrontEnd with its

TABLE 4. Values used in the parameters of the HMM algorithm, error rate and average error rate in the datasets from 100 to 1000.

		Datasets									
Parameters/Error	**Exp**	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**
Probability of State	1	0.45	0.27	0.38	0.42	0.57	0.38	0.54	0.48	0.49	0.48
	2	0.61	0.75	0.36	0.68	0.72	0.66	0.86	0.36	0.56	0.77
	3	0.36	0.41	0.86	0.20	0.71	0.36	0.19	0.53	0.52	0.53
	4	0.22	0.09	0.53	0.24	0.48	0.56	0.84	0.62	0.52	0.53
	5	0.54	0.85	0.36	0.44	0.47	0.64	0.82	0.38	0.64	0.54
Probability of Transitions	1	0.39	0.17	0.22	0.03	0.31	0.41	0.35	0.20	0.06	0.56
	2	0.63	0.72	0.24	0.61	0.68	0.68	0.91	0.83	0.56	0.70
	3	0.64	0.12	0.87	0.01	0.61	0.14	0.09	0.70	0.31	0.68
	4	0.36	0.52	0.48	0.18	0.21	0.42	0.84	0.41	0.36	0.48
	5	0.25	0.88	0.003	0.01	0.52	0.45	0.82	0.03	0.65	0.59
Average (μ)	1	-1.47	-1.55	-0.99	-1.09	-0.36	-1.42	-1.38	-0.56	-0.94	-0.90
	2	-0.95	-1.50	-1.23	-0.80	-0.64	-0.03	-0.61	-0.23	-1.46	-0.25
	3	-0.97	-1.44	-0.26	-0.06	-1.23	-0.11	-1.08	-1.13	-1.21	-1.52
	4	-1.30	-1.12	-0.07	-0.96	-1.27	-0.25	-1.57	-1.47	-0.42	-0.80
	5	-0.71	-0.23	-0.56	-1.17	-1.43	-1.02	-1.28	-0.79	-0.79	-0.06
Variance (λ)	1	0.34	0.48	0.43	0.46	0.32	0.43	0.58	0.33	0.46	0.32
	2	0.10	0.46	0.52	0.31	0.41	0.08	0.23	0.56	0.62	0.19
	3	0.08	0.17	0.04	0.43	0.18	0.18	0.45	0.48	0.53	0.09
	4	0.52	0.10	0.52	0.57	0.32	0.09	0.20	0.56	0.27	0.34
	5	0.29	0.19	0.17	0.24	0.27	0.47	0.05	0.42	0.38	0.34
Error rate	1	0.03	0.04	0.03	0.05	**0.02**	0.04	0.04	0.04	0.03	0.04
	2	0.03	0.02	0.03	0.04	**0.04**	0.03	0.05	0.04	0.03	0.04
	3	0.03	0.04	0.04	0.04	**0.03**	0.04	0.03	0.04	0.04	0.04
	4	0.03	0.03	0.03	0.05	**0.02**	0.04	0.03	0.04	0.03	0.05
	5	0.04	0.02	0.03	0.05	**0.02**	0.04	0.03	0.04	0.03	0.04
Average Error rate	=	0.03	0.03	0.03	0.04	**0.03**	0.04	0.04	0.04	0.03	0.04

value parameters $L_{min} = 9$ and $S_{min} = 4$, in which the ANN and RF classifiers presented the best performances in the experiments, with error rates of 0.2% and 0.1% respectively, while KNN and SVM provided inferior results, with error rates of 4.6% and 7.9%. Table 6 presents the best results of the experiments involving all front ends and conventional classifiers used in the FBSC architecture.

Another factor evaluated in the study for FBSC architecture was the computational cost in the replication of the experiments. The classification in this architecture requires a significant amount of time for the execution of the experiments. Table 6 displays the execution time in seconds for all conventional algorithms associated with their respective front ends.

In accordance with the observations in Table 6, the execution time with the FBSC architecture is relatively high for fault classification in transmission lines. The front end with the highest computational cost is the ConcatFront and Waveletconcat associated with the conventional ANN classifier. This is certainly due to the fact that they have features of other front ends. Comparatively, Waveletenergy, Raw and RMS have simpler structures and present lower computational costs than those associated with most classifiers. The FBSC architecture had a total time of 877150 seconds to process all replications in the experiments performed on the fault classification.

B. RESULTS OF THE HMM ALGORITHM EXPERIMENTS

In this section the results of the HMM algorithm experiments using the UFPAFaults dataset described in section III will be presented. For the training processes and tests that represent the faults in transmission lines, the mean (μ), variance (λ), state probability, and transition probability were used through the Baum-Welch algorithm in models HMMs [29]. Table 4 displays the best values for each of the four parameters of the HMM algorithm used in the study, the error rates in each experiment per dataset, and the average error rate taken from the five experiments together.

In agreement with Table 4, it is observed that the HMM algorithm performed better on the dataset of 500 samples with a mean error rate of 0.03%, and also had fewer errors in the experiments performed. Being that the probability of state for this dataset varied between 0.47 and 0.72, the transition probability was between 0.21 and 0.68, the mean in the range between -0.36 and -0.43, and the variance between 0.18 and 0.32. This indicates that the balancing and quantity of samples arranged for each class in this dataset may have influenced the result. This characteristic is also noticeable in the other datasets that on average had a performance close to or equal to the dataset of 500 samples. In general, the performance of the HMM algorithm is satisfactory in the classification of faults in transmission lines, staying within the range of 0.03% in most datasets with different sample quantities.

The results of the experiments contained in Table 4 also indicate numerous advantages of using the HMM algorithm in fault classifications, including its differentiated ability to recognize patterns more precisely as compared to other conventional algorithms. On the other hand, in addition to being able to directly handle information without using front

TABLE 5. E_i, M_v, AE_i e G_{Ac} on the datasets of 100 to 1000.

Datasets												
Class	=	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**	**900**	**1000**	G_{Ac}
AG	E_i	0,00	0.00	0.00	0.12	0.00	0.02	0.00	0.07	0.03	0.06	
	AE_i	0.04	0.03	0.02	0.13	0.01	0.07	0.02	0.09	0.04	0.13	0.06
	M_v	16.80	12.63	14.48	13.80	16.27	16.77	15.42	16.46	14.15	16.30	
BG	E_i	0.06	0.00	0.00	0.07	0.00	0.00	0,01	0,00	0,00	0,00	
	AE_i	0.07	0.05	0.04	0.06	0.00	0.01	0.08	0.01	0.00	0.01	0.03
	M_v	16.23	16.44	15.82	13.61	14.97	16.23	15.67	16.43	14.66	14.61	
CG	E_i	0.06	0.00	0.08	0.06	0.00	0.08	0.06	0.06	0.00	0,06	
	AE_i	0.03	0.01	0.00	0.06	0.02	0.06	0.06	0.07	0.00	0.06	0.04
	M_v	16.89	17.11	14.06	14.08	14.66	14.64	16.56	16.80	14.83	14.47	
AB	E_i	0.04	0.06	0.00	0.01	0.08	0.01	0.05	0.06	0.04	0,02	
	AE_i	0.03	0.01	0.00	0.06	0.02	0.06	0.06	0.07	0.00	0.06	0.04
	M_v	14.12	15.25	14.25	16.93	13.92	15.63	16.02	14.59	15.62	15.45	
AC	E_i	0.01	0.01	0.00	0.05	0.04	0.02	0.02	0.05	0.06	0,00	
	AE_i	0.01	0.02	0.07	0.05	0.04	0.05	0.02	0.07	0.06	0.01	0.04
	M_v	14.12	15.25	14.25	16.93	13.92	15.63	16.02	14.59	15.62	15.45	
BC	E_i	0,00	0.01	0.06	0.02	0.00	0.00	0.00	0.00	0.00	0.00	
	AE_i	0.01	0.03	0.05	0.04	0.01	0.01	0.02	0.02	0.01	0.01	**0.02**
	M_v	14.23	16.42	14.97	15.10	15.85	15.19	13.16	15.63	16.32	14.71	
ABC	E_i	0.02	0.00	0.06	0.00	0.00	0.03	0,00	0,05	0,06	0,06	
	AE_i	0.04	0.01	0.04	0.00	0.02	0.05	0.01	0.05	0.06	0.07	0.04
	M_v	13.97	14.27	13.99	15.95	14.86	14.63	16.84	15.48	14.63	14.67	
ABG	E_i	0.00	0.00	0.01	0.02	0.00	0.01	0.06	0.00	0.00	0,01	
	AE_i	0.03	0.01	0.02	0.03	0.00	0.03	0.02	0.01	0.02	0.01	**0.02**
	M_v	16.23	16.44	15.82	13.61	14.97	16.23	15.67	16.43	14.66	14.61	
ACG	E_i	0.06	0.08	0.00	0.05	0.03	0.00	0.05	0.00	0.07	0,01	
	AE_i	0.07	0.06	0.00	0.04	0.05	0.00	0.07	0.00	0.07	0.02	0.04
	M_v	14.01	16.18	12.51	17.81	15.69	14.00	14.89	15.18	18.09	17.80	
BCG	E_i	0.01	0.00	0.002	0.02	0.06	0.08	0.02	0.06	0.06	0,06	
	AE_i	0.03	0.01	0.00	0.04	0.05	0.07	0.02	0.05	0.06	0.04	0.04
	M_v	16.30	15.73	13.33	16.77	13.81	14.53	15.25	15.47	16.16	16.95	

ends, its probabilistic characteristic favors balancing of data keeping the performance quite satisfactory with lower error rate of 0.02% in the datasets of 200 and 500, 0.03% for the 100, 300, 600, 700, 900, and 0.04% data sets for the 400, 800, and 1000 data sets.

Another factor evaluated in the study was the performance of each class in the experiments performed, where it was possible to obtain a more precise idea of the learning of the HMM algorithm in the classification of faults. Table 5 shows the maximum likelihood (M_v) obtained, the result of the lowest error rate of the classes between the experiments (E_i), mean error rate of the class between the experiments (AE_i), and the general error rate of the experiments for each class among all the datasets (G_{Ac}).

As can be seen in Table 5, the performances of the fault classification between the datasets using the HMM algorithm stood out in the BC and ABG classes that obtained, on average, the lowest error rate of 0.02%. For these classes, the maximum likelihood ranged from 13.16 to 16.42 and 13.16 to 16.14 respectively. The AG class was the one with the highest error rate of 0.06%, with maximum verisimilitude ranging from 12.63 to 16.80. The other classes had an intermediate performance of 0.04% with a maximum likelihood that was different from the AG, BC, and ABG classes. It is noted that, in the same way as with the results obtained by the dataset, the classification by class also indicates that the balance and quantity of samples arranged for each class in these datasets may have influenced all results.

HMM algorithm performance results in each fault class observed in Table 5 reflect another advantage of adapting to different contexts by preserving greater accuracy over to conventional algorithms. So much so that the average rate error classes AG, BG, CG, AB, AC, BC, ABC, ABG, ACG and BCG in all datasets obtained great performance ranging from 0.02% to 0.06%. Being that the largest variation of the average error rate is perceived in the AG class with 0.06%, certainly due to distribution of samples by class are less balanced to better learning of the HMM algorithm between sets of data. While classes CG, AB, AC, ABC and ACG have a more uniform distribution of samples between classes, achieving similar performance with an intermediate variation in the average error rate of 0.04%. And the smallest variation in the average error rate was in the BC and ABC classes with 0.02%, this is certainly due better balance of the samples in the classes between the datasets relative to other classes with smaller performance.

As with the FBSC architecture, the computational cost for the HMM algorithm was also noted in the experiments that were performed in order to classify faults. Due to the HMM algorithm dealing directly with samples of a time series,

TABLE 6. Best results and computational cost using the HMM algorithm and the FBSC architecture.

Classifier	Front End	Error rate (%)	Time Per Experiment (sec)
ANN	Waveletconcat	0.5	68496
	Waveletenergy	4.1	36092
	Raw	5.7	41522
	RMS	8.7	42113
	ConcatFrontEnd	**0.2**	82343
RF	Waveletconcat	7.1	34075
	Waveletenergy	5.0	34049
	Raw	6.6	48449
	RMS	13.5	38792
	ConcatFrontEnd	**0.1**	65418
SVM	Waveletconcat	1.0	34182
	Waveletenergy	10.3	34049
	Raw	5.3	48449
	RMS	8.3	38792
	ConcatFrontEnd	7.9	65418
KNN	Waveletconcat	6.9	34016
	Waveletenergy	6.4	34008
	Raw	6.4	34006
	RMS	9.1	38258
	ConcatFrontEnd	4.6	56306
HMM	————	**0.03**	**3564**

the number of operations could be decreased, resulting in a lower computational cost. Table 6 displays the execution time of the HMM for the processing of the experiments performed on the same machine used in the FBSC architecture.

According to Table 6 the HMM algorithm presents a very acceptable computational cost, where it loads just once all the datasets in 274 seconds. Then, the experiments are carried out with each dataset with different sample quantities. The whole process takes a total time of 3564 seconds to classify the faults.

C. COMPARISON OF RESULTS BETWEEN FBSC ARCHITECTURE AND HMM

The comparison of the results between FBSC architecture and the HMM algorithm took into account the results of the classification by datasets, computational cost, error rate and statistical test for all classifiers, and also accounted for the computational cost with front end and parameter selection in the FBSC architecture and the loading of datasets of the HMM algorithm. That is to say, the experiments started from the same point so that the performance measurements are equivalent. Table 6 presents the best error rate results of the HMM algorithm and the FBSC architecture with their respective Waveletconcat, Waveletenergy, Raw, RMS and ConcatFrontEnd front ends associated with the conventional ANN, RF, KNN, and SVM classifiers.

In accordance with Table 6, the error rate results indicate that depending on the combination of the front end and the associated conventional classifier to FBSC architecture, the HMM algorithm in general presents superior performance with an error rate of 0.03%. If only the best results between the front ends with ConcatFrontEnd associated with the ANN and RF classifiers are taken into consideration, with error rates of 0.2% and 0.1% respectively, they are statistically equal to the HMM algorithm with a significance level of $\alpha = 5\%$. However, those results do not show the same performance in all datasets. Table 7 presents the results of the application of t-Student Statistical test between the HMM algorithm and the conventional classifiers of the FBSC architecture associated with the front end ConcatFrontEnd.

TABLE 7. Results of the statistical test comparison between the HMM algorithm and the FBSC architecture with concatfrontend associated with the ANN, SVN, KNN and RF classificators with significance of $\alpha = 5\%$.

Classifiers	Degree of Freedom	T_{cal}	T_{tab}	t-Student	Average
HMM->ANN	8	0.17	2.31	$T_{cal} < T_{tab}$	Equal
HMM->RF		0.07			
HMM->KNN	8	4.70	2.31	$T_{cal} > T_{tab}$	Different
HMM->SVM		7.87			

According to the results presented in Table 7, the statistical test had a table value t_{tab} of 2.31 and a degree of freedom value of 8 with a significance of $\alpha = 5\%$, which corresponds with the comparison between two classifiers with five experiments per each. The difference of the error rate t_{cal} between the HMM algorithm and ANN was 0.17%, between HMM and KNN was 4.70%, between HMM and RF was 0.07%, and between HMM and SVM was 7.87%. So for this scenario, HMM, ANN, and RF have statistically similar performances, although they present different values in their results. SVM and KNN result in inferior performances to the other classifiers, with significant differences in their error rates.

Regarding the computational cost, Table 6 displays the processing time of all classifiers used in the study. It is observed that the HMM algorithm requires 3564 seconds to process all experiments in the classification of faults. By comparison, the next best performance using the FBSC architecture was with the conventional classifier RF associated with the Waveletenergy front end that took 34002 seconds to make the same classifications. The difference between the two of 30438 seconds equals, on average, 90% less time needed by the HMM algorithm to process the same experiments in comparison to the conventional classifiers of the FBSC architecture, presenting itself as a potential classifier for the classification of faults in power transmission lines.

V. CONCLUSION

This study aimed to investigate a problem that compromises the quality of electric power and propose a solution that does not use the processing phases that precede fault classification. The study utilized techniques that were used to analyze a public dataset called UFPAFaults with simulations of short circuits in transmission lines. The HMM algorithm was used, which allowed for the direct classification of an event without the use of pre-processing steps (front ends) with multivariate characteristics of variable duration. The efficiency of the results was proven by comparing them with

an architecture called FBSC developed to classify the faults from the UFPAFaults dataset.

The performance of the HMM algorithm exceeded that of the FBSC architecture using the Waveletconcat, Waveletenergy, Raw, RMS and ConcatFrontEnd front ends associated with the conventional classifiers ANN, RF, KNN and SVM. The HMM algorithm also obtained lower error rates as compared to the conventional classifiers of the FBSC architecture. Only when the t-Student statistical test was applied did the ANN and RF classifiers associated with the front end ConcatFrontEnd present performances equivalent to the HMM algorithm, considering a significance $\alpha = 5\ \%$. In terms of computational cost, HMM is about 90% faster in its processing time in relation to all classifiers of the FBSC architecture, thus presenting itself as a potential classifier in the classification of faults in transmission lines.

The classification with the HMM algorithm presented a superior accuracy and performance in relation to the FBSC architecture that used front ends and parameter selection, which are two processes that precede the use of conventional classifiers. This paper proposes the direct classification of faults with the HMM algorithm that presented results generated in a machine with processor i7 and 16G of memory. Further research can be conducted concerning the evaluation of other types of faults in power systems with the HMM algorithm and also the application in real data sets to help the decision-making process at the operational level.

ACKNOWLEDGMENT

The authors would like to thank Eletrobras-Eletronorte for the technical support.

REFERENCES

[1] H. Fathabadi, "Novel filter based ANN approach for short-circuit faults detection, classification and location in power transmission lines," *Int. J. Elect. Power Energy Syst.*, vol. 74, pp. 374–383, Jan. 2016. Accessed: Jan. 11, 2019. [Online] Available: http://www.sciencedirect.com/science/article/pii/S014206151500335X

[2] M. H. J. Bollen and I. Y. H. Gu, *Signal Processing of Power Quality Disturbances*, vol. 30. Hoboken, NJ, USA: Wiley, 2006.

[3] G. A. Sudha and T. Basavaraju, "A comparison between different approaches for fault classification in transmission lines," in *Proc. IET-UK Int. Conf. Inf. Commun. Technol. Elect. Sci. (ICTES)*, Dec. 2007, pp. 398–403. Accessed: Jan. 18, 2019. [Online] Available: https://ieeexplore.ieee.org/document/4735829

[4] A. Yadav and Y. Dash, "An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination," *Adv. Artif. Neural Syst.*, vol. 2014, Dec. 2014, Art. no. 230382.

[5] C. H. Fontes and O. Pereira, "Pattern recognition in multivariate time series—A case study applied to fault detection in a gas turbine," *Eng. Appl. Artif. Intell.*, vol. 49, pp. 10–18, Mar. 2015.

[6] M. Singh, B. K. Panigrahi, and R. P. Maheshwari, "Transmission line fault detection and classification," in *Proc. Int. Conf. Emerg. Trends Elect. Comput. Technol.*, Mar. 2011, pp. 15–22.

[7] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2018.

[8] P. Ray, D. P. Mishra, K. Dey, and P. Mishra, "Fault detection and classification on a transmission line using wavelet multi resolution analysis and neural network," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2017, pp. 178–183.

[9] K. Hosseini, "Short circuit fault classification and location in transmission lines using a combination of wavelet transform and support vector machines," *Int. J. Elect. Eng. Inform.*, vol. 7, pp. 353–365, Jun. 2015.

[10] B. Sreewirote and A. Ngaopitakku, "Classification of fault type on loop-configuration transmission system using support vector machine," in *Proc. 6th IIAI Int. Congr. Adv. Appl. Inform.*, Jul. 2017, pp. 892–896.

[11] M. Nayeripour, A. H. Rajaei, M. M. Ghanbarian, and M. Dehghan, "Fault detection and classification in transmission lines based on a combination of wavelet singular values and fuzzy logic," *Cumhuriyet Üniversitesi Fen-Edebiyat Fakültesi Fen Bilimleri Dergisi (CFD)*, vol. 36, no. 3, pp. 69–82, 2015.

[12] A. R. Almeida, O. M. Almeida, B. F. S. Junior, L. H. S. C. Barreto, and A. K. Barros, "ICA feature extraction for the location and classification of faults in high-voltage transmission lines," *Electr. Power Syst. Res.*, vol. 1, no. 1, pp. 254–263, 2017.

[13] N. R. Babu and B. J. Mohan, "Fault classification in power systems using EMD and SVM," *Ain Shams Eng. J.*, vol. 8, no. 2, pp. 103–111, 2017.

[14] A. Abdollahi and S. Seyedtabaii, "Comparison of Fourier & wavelet transform methods for transmission line fault classification," in *Proc. 4th Int. Power Eng. Optim. Conf. (PEOCO)*, Jun. 2010, pp. 579–584.

[15] C. Cardoso, Y. Pires, J. Morais, and A. Klautau, "Hierarchical agglomerative clustering of short-circuit faults in transmission lines," in *Proc. 10th Brazilian Symp. Neural Netw.*, Salvador, Brazil, 2008, pp. 87–92.

[16] J. Morais, Y. Pires, C. Cardoso, and A. Klautau, "A framework for evaluating automatic classification of underlying causes of disturbances and its application to short-circuit faults," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2083–2094, Oct. 2010.

[17] M. Homci, P. Chagas, B. Miranda, J. Freire, R. Viégas, Jr., Y. Pires, B. Meiguins, and J. Morais, "A new strategy based on feature selection for fault classification in transmission lines," in *Proc. IBERAMIA*. San Jose, CA, USA: Costa Rica. Sociedad Iberoamericana de Inteligencia Artificial, 2016.

[18] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[19] Y. P. Pires, "Data mining applied to electrical systems: Classification of short circuits in transmission lines," M.S. thesis, Inst. Technol., Federal Univ. Pará, Belém, Brazil, 2009.

[20] J. Chung, E. J. Powers, W. M. Grady, and S. C. Bhatt, "Power disturbance classifier using a rule-based method and wavelet packet-based hidden Markov model," *IEEE Trans. Power Del.*, vol. 17, no. 1, pp. 233–241, Jan. 2002.

[21] T. K. Abdel-Galil, E. F. El-Saadany, A. M. Youssef, and M. M. A. Salama, "Disturbance classification using Hidden Markov Models and vector quantization," *IEEE Trans. Power Del.*, vol. 20, no. 3, pp. 2129–2135, Jul. 2005.

[22] T. K. Abdel-Galil, A. M. Youssef, and M. M. A. Salam, "Disturbance classification using hidden Markov models and vector quantization," *IEEE Trans. Power Del.*, vol. 20, pp. 2129–2135, 2005.

[23] K.-C. Kwon and J.-H. Kim, "Accident identification in nuclear power plants using hidden Markov models," *Eng. Appl. Artif. Intell.*, vol. 12, pp. 491–501, Aug. 1999.

[24] Q. Suxiang, J. Weidong, H. Hongsheng, and Y. Gongbiao, "Transformer power fault diagnosis system design based on the HMM method," in *Proc. IEEE Int. Conf. Automat. Logistics*, Aug. 2007, pp. 1077–1082.

[25] X. Ma and J. Shi, "A new method for discrimination between fault and magnetizing inrush current using HMM," *Electr. Power Syst. Res.*, vol. 56, pp. 43–49, Oct. 2000.

[26] S. Jazebi, B. Vahidi, and S. H. Hosseinian, "A novel discriminative approach based on hidden Markov models and wavelet transform to transformer protection," *Simulation*, vol. 86, no. 2, pp. 93–107, 2010.

[27] P. A. Scudino, "The use of some statistical tests to analyze the variability of the price of honey in the municipalities of Angra dos Reis and Mangaratiba," Ph.D. dissertation, Inst. Exact Sci., Dept. Math., Federal Rural Univ. Rio de Janeiro, Seropédica, Brazil, 2008.

[28] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models* (Springer Series in Statistics). Springer, 2005.

[29] I. M. Frondana, "Biopotential classification with hidden Markov chains," Ph.D. dissertation, Inst. Exact Sci., Dept. Statist., Univ. Brasília, Brasília, Brazil, 2012.

[30] N. Ravikumar, A. Gooya, S. Çimen, A. F. Frangi, and Z. A. Taylor, "Group-wise similarity registration of point sets using Student's t-mixture model for statistical shape models," *Med. Image Anal.*, vol. 44, pp. 156–176, Feb. 2018.
[31] B. G. Costa, J. C. A. Freire, H. S. Cavalcante, M. Homci, A. R. G. Castro, R. Viegas, B. S. Meiguins, and J. M. Morais, "Classification on transmission lines using KNN-DTW," in *Proc. 17th Int. Conf. Comput. Sci. Appl. (ICCSA)*, Trieste, Italy, 2017, pp. 174–187.

JEAN CARLOS AROUCHE FREIRE received the degree in computer science from CEUCLAR, and the master's degree in computer science from Federal University of Pará (UFPA), where he is currently pursuing the Ph.D. degree in applied computation with the Institute of Technology. He is currently a professor with the University of the Amazon, Belém, Brazil. He teaches classes in computer networks, systems analysis and development, information systems, and mathematical discipline courses, such as probability, statistics, calculus I, II, III, and numerical and differential equations. His research interests include the application of computational intelligence, machine learning, and data mining in domains of different problems.

ADRIANA ROSA GARCEZ CASTRO completed the master's degree in electrical engineering from the Federal University of Pará (UFPA). She received the Ph.D. degree in electrical engineering from the University of Porto. She is currently an Associate Professor with UFPA. Her areas of academic interest include applied computational intelligence, control of electronic processes, and power systems.

MARCIA SALOMÃO HOMCI received the degree in data processing technology from the University of the State of Pará (CESUPA) with a Specialization in computer networks from the University of the Amazon (UNAMA), and the master's degree in computer science from Federal University of Pará (UFPA), where she is currently pursuing the Ph.D. degree in computer science. She is currently a Professor and a Course Coordinator with the University of the Amazon, where she also researches computer networks, computational intelligence, machine learning, and data mining. Her master's concentration area involved computer systems, while her research interest includes computational intelligence.

BIANCHI SERIQUE MEIGUINS received the master's degree in information systems management from the Pontifical Catholic University of Campinas, in 1999. He is currently pursuing the Ph.D. degree in computer science with the Graduate Program in Electrical Engineering, Federal University of Pará (UFPA), where he has been an Associate Professor with the Faculty of Computer Science, since 2004. He was an IT Manager with Companhia Docas do Pará, from 2009 to 2013, coordinating several IT modernization projects for the Pará ports. He is also a Professor of the Graduate Program in Computation with UFPA, where he teaches courses in virtual and augmented reality, and visualization of information. At the undergraduate level, he teaches courses in computer graphics, programming, and basic software, and among others. He acts as a consultant for several projects of technological innovation with Paraense companies with financing from various development agencies. His main areas of interests include information visualization, virtual and augmented reality, and human–computer interaction. He also coordinated the VIII Symposium on Virtual Reality held at Belém, in 2006, promoted and supported by the Brazilian Computer Society.

JEFFERSON MAGALHÃES DE MORAIS received the bachelor's degree in computer science from the University Center of Pará, and the master's degree in computer Science and the Ph.D. degree in electrical engineering from the Federal University of Pará (UFPA), where he is currently an Associate Professor and a Researcher with the Laboratory of Visualization, Interaction, and Intelligent Systems (LabVIS). He has extensive experience in computer science, with an emphasis on computer intelligence and data mining, mainly in the following subjects: data mining in bioinformatics and power systems, computational vision, brain–computer interface, and information visualization.