Research Article

Identification of Power Transformer Currents by Using Random Forest and Boosting Techniques

Tamer Khatib¹ and Gazi Arar²

¹Department of Energy Engineering and Environment, An-Najah National University, 97300 Nablus, State of Palestine

²Faculty of Graduate Studies, An-Najah National University, 97300 Nablus, State of Palestine

Correspondence should be addressed to Tamer Khatib; t.khatib@najah.edu

Received 20 August 2020; Revised 9 September 2020; Accepted 12 September 2020; Published 18 September 2020

Academic Editor: Ying-Yi Hong

Copyright © 2020 Tamer Khatib and Gazi Arar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this research, a differential protection technique for a power transformer is proposed by using random forest and boosting learning machines. The proposed learning machines aim to provide a protection expert system that distinguishes between different transformer status which are normal, inrush, overexcitation, CT saturation, or internal fault. Data for 20 different transformers with 5 operating cases are used in this research. The utilized random forest and boosting techniques are trained using these data. Meanwhile, the proposed models are validated by other measures such as out-of-bag error and confusion matrix. In addition, variable importance analysis that shows signal's component importance inside a transformer at different instances is provided. According to the result, the proposed random forest model successfully identifies all of the current cases (100% accuracy for the conducted experiment). Meanwhile, it is found that it is less accurate as a conditional monitoring element with accuracy in the range of 97%–98%. On the other hand, the proposed boosting model identifies all of the currents for both cases (100% accuracy for the conducted experiment). In addition to that, a comparison is conducted between the proposed models and other AI-based models. Based on this comparison, the proposed boosting model is the simplest and the most accurate model as compared to other models.

1. Introduction

Electrical power system contains different elements including a power transformer which links the whole system. A power transformer exhibits different oscillatory power flow features including faults. The differences in these oscillatory power features are due to the nonlinearity of the magnetic core, where it is difficult to diagnose these features, and consequently maloperation may occur.

The efficiency of power transformer protection is usually described based on its selectivity, security, dependability, and speed of operation. This is due to the fact that power transformers have different current status with different characteristics depending on many operation factors. Transformer's normal operation, faulty operation, external fault, magnetizing inrush, and overexcitation are different transformer statuses. In most conditions, a power

transformer can supply differential current with close characteristics, while only internal fault current is required to be cleared amongst other conditions [1].

Traditional protection uses the harmonic restrain method to differentiate between these currents so as to avoid unwanted protection operation. The second harmonic order is used, for example, to differentiate inrush current status from other statuses. However, the novel improvement of transformers' core material decreases the level of second harmonic which may be classified then as a faulty condition. Moreover, current transformer (CT) saturation as well as shunt or distributive capacitance in long high voltage transmission lines increases second harmonic level in faulty condition which may cause a failure in detecting the fault status. Based on that, differential protection is usually applied in such cases so as to differentiate between different currents [2].

Conventional differential protection considers current transformer (CT) transformation errors, CT mismatch, and tap variation. It usually follows an increasing curve with positive specific slope for a relation between differential and restrain currents. Here, whenever the point of operation is above this curve, fault status is classified and tripping is imposed to a circuit breaker. Meanwhile, novel differential protection implies many classification techniques including artificial intelligence, transitory feature detection, hybrid systems, and many others [1–3].

In general, there are two methodologies to address this problem as listed below:

- (1) Signal analysis using time-frequency analysis such as wavelet transform, s-transform, and Huang–Hilbert transform [4–10]
- (2) Classification algorithms and learning machines such as artificial neural networks and tree-based algorithms

In this research, the focus is given to the second methodology which is classification algorithms and learning machines. In general, many research studies have been done on this issue. In [11], a decision tree (DT) is used to classify different current data composition so as to compare between these currents which are differential current, restrain current, and percentage differential current. In [3], two ANN networks have been used, for internal fault detection. The utilized networks classify different operation statuses such as normal, inrush, overexcitation, and CT saturation. In addition, the authors utilized the particle swarm optimization algorithm (PSO) to optimize the number of hidden layers and neurons in the proposed ANN networks. Similarly, in [12], the authors have suggested digital differential protection using the optimal probabilistic neural network (PNN). By this model, external fault and normal operation are identified by comparing two consecutive peaks. Meanwhile, overexcitation is differentiated by comparing the voltage to frequency ratio with rated voltage to frequency ratio. In [13], differential current samples are fed to a decision tree to classify two statuses of operation which are inrush and internal fault. Meanwhile, the authors in [2] classified internal fault, inrush, CT saturation, overexcitation, and normal condition by using the Bayesian classifier (BC) with normal distribution by means and variance for single dimension and means, variance, and covariance for multi-dimension. The authors in [14] used support vector machine (SVM) and radial basis function (RBF) kernel to classify standard deviation of the detail 1 coefficient. Meanwhile, in [15], differential protection such as the wavelet-neurosystem is performed. The proposed FFNN is trained by feature vector dimension and standard deviation of the detail coefficient so as to differentiate between two statuses which are internal and inrush current.

In [16], three membership functions which are differential, restrain currents, and inrush detector are generated. The inrush detector is assumed to be operated based on primary dead angle detection which is to set the inrush current flag. These three inputs are used to train the fuzzy

inference engine rule base code. Here, if the detector is activated, then the algorithm acts as an inrush detector. Otherwise, the relay issues trip/no trip action based on restrain and differential membership function. In [17], maximal overlap discrete wavelet transform is used with two ANN networks. A spectral energy of the sliding window is calculated first and compared then to some threshold as a disturbance detector. If disturbance is detected, the ANN network is initiated and warning is issued regarding external fault. Meanwhile, internal fault warning is issued, while the second ANN classifies the fault. Finally, in [18], internal fault, external fault, CT saturation with fault, deep saturation, and data sampling by means of the similarity differentiation algorithm are proposed. They use pseudocharacteristic to extract the core operating region, and thus linearity is checked by means of orthogonal polynomial representation to model the inrush/fault detector. This algorithm consists of three sequential steps which are amplitude check, harmonic check, and pseudocharacteristic check.

From the reviewed research studies, most of the methods are based on ANN techniques. However, it is claimed that random forests (RFs) and boosting have better ability for classification as compared to ANN considering model accuracy and simplicity. RFs model incorporates random decision trees and bagging. Meanwhile, bagging is a technique for reducing the variance of an estimated prediction function. On the other hand, boosting is another ensemble technique which has some similarity with bagging and random forest. Following this, researchers have utilized RF for such a purpose. In [19], RF is utilized for fault discrimination for a power transformer. The proposed scheme relies on extracting features from the measured data of differential current signals of a power transformer. An overall fault discrimination accuracy of more than 98% is achieved. Moreover, in [20], internal and external electrical faults and inrush current of the transformer are predicted. The fault current signals are analyzed first, and then these features are considered for training the classifiers of the decision tree and the artificial neural network. According to [20], the proposed procedure is capable of classifying and discriminating among winding mechanical defects, internal and external electrical faults, and inrush current with a good accuracy. Meanwhile, in [21], the Hausdorff distance (HD) algorithm is used to reflect the waveform sinusoidal similarity of the differential current so as to distinguish internal faults, magnetizing inrushes, and faults accompanied by CT saturation of the transformer. However, RF should be investigated more for this purpose considering other classification techniques and development methodologies. Therefore, the main objective of this research is to provide an accurate and simple tool to identify power transformer currents and therefore classify the operation of a power transformer into two main statuses. First is “no trip” status which implies normal, inrush, overexcitation, and current transformer (CT) saturation. Second is “trip” status which implies internal fault. The main contribution of this proposal is the application of different classification techniques including RF. Moreover, performance evaluation of these

techniques is provided considering sensitivity analysis for proposed models. Finally, analysis of required data for training is proposed so as to develop an accurate model with simple computational needs.

2. Types of Currents in Power Transformer

Transformer's nonlinearity nature complicates the understanding of power transformer performance. This is because in any abnormal operation status, the system loses its steady state operation phase. Consequently, the transient operation phase may imply different types of currents with nonlinear behaviour including internal fault currents, inrush magnetizing current, overexcitation current, and CT saturation current [22].

Internal and external fault currents mainly occur because of main internal faults due to insulation deterioration and breakdown. Other reasons for internal faults are faults in winding, core, tap changer, cooling, bushing, and casing. Meanwhile, external fault current is a high current that passes through transformer sides. In general, a passing-through fault current that has a value of 10 times the rated current may cause a differential current of 1-2 times the power transformer-rated current. Moreover, high passing-through current may cause internal faults due to overheating of insulation [23].

Another type of transformer currents is overexcitation current which is due to the increase of flux flowing through the core above some design limit. Magnetizing current characteristics follow core characteristics distorting the current signal. Thus, the current from the source to load has two components which are core magnetizing current and load current. Transformer overexcitation in transmission and distribution networks is caused by overvoltages in the network. This current has high percent of the fifth harmonic order. On the other hand, inrush current occurs because of an overexcited core with special case of saturation during the initial excitation. Finally, deep saturation current is similar to external fault current, and it occurs when a transformer is driven more into saturation [23].

3. Modelling of Power Transformers

In general, transformer's model has two parts, namely, winding model and core model. When modelling a transformer, it is necessary to consider transient analysis as it has two main aspects which are the nonlinearity and frequency dependency. Nonlinearity arises from the magnetic core saturation region, whereas frequency affects winding and core sections [24]. On the other hand, the steady state transformer model based on matrix representation can be described as follows:

$$V = ZI. \quad (1)$$

The transient phenomena including the inductance effect case can be described then as follows:

$$\left[\frac{di}{dt} \right] = [L]^{-1} [v] + [L]^{-1} [R] [i]. \quad (2)$$

Here, this representation is valid for frequencies that are up to 1 kHz.

Transformer model for a three-phase, three-legged, two-winding transformer can be represented, where a simplified model is available [23]. In this model, which is based on self- and mutual inductances, this technique has a problem of close values for self- and mutual inductances [25]. However, the transformer model can be solved by state space matrices as follows:

$$\left[\frac{dX}{dt} \right] = [A] [X(t)] + [B] [U(t)], \quad (3)$$

where $X(t)$ is the state variables = $[i_{L1} \ i_{L2} \ i_{L0} \ \phi_1 \ \phi_2 \ \phi_3]$, $[U(t)]$ is the input vector = $[e_1 \ e_2 \ e_3 \ S_{h1j} \ S_{h2k} \ S_{h3l}]$, A is the coefficient matrix, and B is the coefficient matrix.

One of the main concerns in transient analysis is stiffness since the problem may swing from nonstiff to extremely stiff situation which may affect the algorithm used and the number of steps [23]. Since some numerical methods require small step size to ensure stability, the explicate numerical method needs to avoid the stiff system [24]. Further development of the model can be achieved by taking into account winding and core topology, frequency dependency, and capacitor effect. Winding resistance can be approximated using [24]

$$R = R_{dc} + \Delta R_{ac} \left(\frac{f}{f_0} \right)^m, \quad (4)$$

where m is a factor that has a value in the range of 1.2–2.

The general model needs manufacturing data and tests to estimate its characteristics. In this research, data are obtained from [3]. This dataset includes data for 20 different transformers. More details about adapted transformer specifications can be found in [3]. In this dataset, differential current samples are extracted. Hence, a discrete signal was sampled with 16 samples/cycle, and each sample in the resulting data is denoted by P1, P2, P3, ..., P16 where symbol P denotes point. Meanwhile, the output is denoted as type, which has values of 1–5. Full data can be obtained from [3].

The five numbers (1, 2, 3, 4, and 5) indicate the five different cases of a transformer which are normal operating condition (Case 1), inrush magnetizing current (Case 2), overexcitation (Case 3), CT saturation (Case 4), and internal fault (Case 5).

4. Classification Trees

Over the past years and in light of artificial intelligence techniques, different classification tree-based techniques have been developed. These techniques include single tree, bagging, random forests, and boosting. Random forests and boosting techniques perform better than single tree and bagging. Trees are a good candidate classifier for the random forests technique, as it reduces variance. Random forest is an ensemble technique which uses the tree-based algorithm. It is an extension to the bagging algorithm that is considered as a predecessor technique. This technique has internal

measures that can be used to judge the algorithm including error, strength, correlation, and variable importance [25].

4.1. Tree-Based Algorithm. These techniques partition the problem space into separate domains, whereas each domain or region is a rectangle. There are two elements that are needed to perform this operation which are variable (feature) to split and point to split on.

Different measures are used to guide the tree building algorithm based on trees' building goal such as regression, classification, and purpose of using (growing tree or pruning). In regression, sum of squares is a good impurity measure. Meanwhile, in classification of other measures of impurities, the Gini index, cross entropy, and misclassification error are used. Gini index and cross entropy are more sensitive, so they are more suitable for growing trees. Tree is interpretable since the whole space is described by some inequalities [25].

4.2. Bagging. This technique uses bootstrap samples to build trees that are averaged over the ensemble to reduce variance leaving bias unchanged. For classification, every tree casts a vote, and the majority vote presents the result. The main idea in bagging is to have independent identical distribution which implies zero correlation between a pair of trees in the ensemble and the same bias, so it is suitable for high-variance low-bias examples [26].

4.3. Out-of-Bag. It describes the average for observation (single) over classifiers which bootstrap does not contain. These out-of-bag classifiers are used to estimate generalization error, strength, correlation, and variable importance since they are similar to the test set with approximately one-third of the training set since they mimic the validation set. If there is a set of n samples and m extracted samples, the average number of different examples is given by following relation [26].

4.4. Random Forest. Random forest construction is similar to bagging, rather it differs in introducing more randomness to the model using different methods such as random feature selection, random linear combination of input, and random noise in output. Figure 1 shows the random forest model.

Common technique uses random feature selection where the number of features selected is less or equal to the total number of features, where these features are used to split on by selecting the best split. More randomness implies less correlation and more strength.

Increasing number of the tree generalization error will have upper limit:

$$\text{generalization error} = \frac{\rho(1 - s^2)}{s^2}, \quad (5)$$

where ρ is the average correlation between the tree vector conditioned to training data and s is the strength of classifiers given by the margin function.

It is evident that error is a combination of two trade-off values; increased correlation will increase error, hence low classifier performance and vice versa. Meanwhile, increased strength will reduce error, hence better classifier performance. Lower number of randomly selected input will reduce the correlation, for example, see [25].

Another way to understand the principle is in terms of variance; for an independent identical distributed random variable with variance σ^2 , variance of the average is given by

$$\text{variance} = \frac{\sigma^2}{N}, \quad (6)$$

where N is the number of trees in the ensemble.

The variables are not independent rather identically distributed, and the variance of the average is given by

$$\text{variance} = \rho\sigma^2 + \frac{(1 - \rho)\sigma^2}{N}, \quad (7)$$

where ρ is the sampling correlation between a pair of trees rather than average and σ is the sampling variance of a single tree.

Equation (7) clearly states that the second term vanishes with the increasing number of trees and is limited to the value given by the first term. As stated earlier, random forest reduces variance keeping bias unchanged, and hence variance is limited to multiplication of correlation and tree variance. This reduces correlation without affecting variance too much by insertion of more randomness like random selection.

4.5. Boosting. Boosting is another ensemble technique; nevertheless, it has superficial similarity with bagging and random forest. Since previous techniques build classifiers in a parallel way, boosting builds them in serial granting weights in two different steps. Figure 2 illustrates the boosting algorithm concept.

Boosting represents a family of algorithms which are adaboost, adaboost.M1, and SAMME. Adaboost is a biclass boosting technique. Thus, boosting represents the move to multiclass using the forward stage-wise additive model and the fitting additive model as follows:

$$\text{prediction} = \sum \beta b(x; \gamma), \quad (8)$$

whereas a loss function that uses exponential loss is applied for adaboost.M1 and SAMME as follows:

$$L(y, f(x)) = e^{-yf(x)}. \quad (9)$$

Adaboost.M1 algorithm starts by initializing the weight as follows:

$$w_i = \frac{1}{N}, \quad (10)$$

where i indicates the weight for each respective observation. Next, weighted error is calculated by

FIGURE 1: Random forest model.

$$\text{error} = \frac{\sum_{\text{observations}} w_i I(h(x) \neq y)}{\sum_{\text{observations}} w_i}. \quad (11)$$

Training rate is calculated based on error which represents the contribution of each classifier in the final result, and it represents learning rate as follows:

$$\alpha = \ln \frac{1 - \text{error}}{\text{error}}. \quad (12)$$

The previous equation is based on Freund and Schapire, while other boosting multiply the previous equation by half as suggested by Breiman [25]. Then, weights are adjusted so that wrongly classified examples have more attention hence more weight, while correctly classified examples have less weights using the following equation:

$$w_{i_{\text{new}}} = w_{i_{\text{old}}} e^{\alpha I(h(x) \neq y)}. \quad (13)$$

After that, a new training set is used to train the new classifier. These steps are repeated where each classifier gives weighted vote using α .

SAMME only differs from adaboost.M1 as it takes into account the number of classes by meaning of training rate and modifying as follows:

$$\alpha = \ln \frac{1 - \text{error}}{\text{error}} + \ln k - 1. \quad (14)$$

5. Results and Discussion

5.1. Model Architecture and Data Rehabilitation. In this research, RF and boosting models are developed using the R language which uses the random forest package. Data were divided into two groups, namely, training and testing with different percentages (100%:0%, 80%:20%, and 60%:40%). The utilized data in this research can be described as 16 features (discrete samples) of differential current with typically 4 variables to split and response of 5 classes available.

The developed RF model has 500 trees to be grown (ntree), while the number of variables (mtry) is selected randomly as a candidate for splitting. This value is different for classification ($\sqrt{p}$) and for regression ($p/3$), where p is the number of variables. On the other hand, the minimum size of the terminal node (nodesize) for classification is set to be 1 and 5 for regression. Meanwhile, the maximum number of terminal nodes (maxnodes) is assumed to be limited by node size.

The proposed model was trained, and OOB error was investigated for the best number of trees (ntree) and the number of variables to split (mtry). Moreover, the accuracy of the proposed model is enhanced by changing the number of nodes by tuning ((nodesize) and (maxnodes)). This process is evaluated by extracting trees and observing its parameters.

On the other hand, R function gbm is also tested. The main features of gbm are distribution which represents the multinomial because of the multiclass classification problem, ntree (701), bag.fraction (0.5) to perform OOB

FIGURE 2: Boosting algorithm.

estimation, shrinkage (0.1), and cv.folds (5) to perform cross validation.

As for the boosting model, the best number of trees (ntree) is 701, bag.fraction is 0.5, shrinkage is 0.1, and cv.folds is 5.

The proposed RFs and boosting models are trained by differential current samples, where 16 data columns are needed to be investigated. The output of this process is the type of current which is represented by five classes (1, 2, 3, 4, and 5). At the beginning, RFs and boosting models are trained based on 100% of the data. This is to check the general ability of these models to handle the data and the suitability of the dataset to be processed by such a model. Considering the nature of these learning machines, it is expected to have high accuracy of prediction in case of testing them using the same data that they were trained based on. Meanwhile, high error in such a prediction means that either the utilized tools are not suitable for such data or the data need to be rearranged or processed to as to be handled by such tools.

Anyway, after conducting this precheck, the performance of the models was noticed to be with around one-third of data misclassified. The prediction error in classifying normal, inrush current, overexcitation, CT saturation, and internal fault was 40%, 55%, 30%, 25%, and 15%, respectively, for the RF model. Meanwhile, the overall accuracy of the proposed boosting model was 27.5%.

Here, it is very clear that the utilized data should be rehabilitated so as to be suitable for training and prediction. To do so, current samples with different structures and relation are given new weights such as $(s/\sqrt{3} V_p)$, $(s/\sqrt{3} V_s)$, and $(s/\sqrt{3} V_{average})$. In addition to that, apparent power of five power transformers has been replicated with the same distribution over five cases for all datasets with different structures by using an exponential relation $(s/\sqrt{3} V_{average})$.

5.2. Model Training Results and Validity of Utilized Data. After rehabilitating the data, the model was trained again using the new set of data with a single candidate to be split

FIGURE 3: OOB error vs. the number of trees for the modified dataset.

and 701 trees, whereas 100% of data were used in training. This process resulted in minimum OOB error, 0%. Figure 3 shows the OOB error development with respect to the number of trees.

Moreover, Table 1 shows the confusion matrix of this process which gave perfect situation, whereas all numbers are diagonal and off-diagonal are zeros. Hence, error rate of all cases is assumed to be zero.

In addition to that, variable importance which is a feature that gives insight over significance and contribution of each feature to system's accuracy is provided in Figure 4. It reflects error decrease contribution of each variable to the overall accuracy. Two measures of variable importance are shown in Figure 4 which are mean decrease accuracy (MDA) and mean decrease Gini (MDG).

In Figure 4, the MDA value is measured using OOB samples by permuting these samples for each tree. Base OOB prediction error is recorded, and the OOB error is recorded after permuting the variable. The sum of differences between these two values is averaged over all trees in the ensemble. The larger the MDA, the more important the variable. Meanwhile, MDG measures sum decrease of node impurity by splitting on that variable. The larger the MDG, the purer the variable.

Anyway, from Figure 4, the importance of samples is approximately the same for all. However, it also can be seen that early moments of the signal have higher importance than late moments of the signal (P_4 has higher importance than P_{16}). Thus, before developing the final model, sensitivity analysis should be done to determine the most important values as in the following section.

5.3. Sensitivity Analysis of Models Input. As stated previously, different moments of the signal hold different amount of information and consequently it varies in importance. This fact could be used to reduce the number of required inputs and therefore ensure faster performance and simple model. In this research, different datasets are used in training

TABLE 1: Confusion matrix for the modified dataset.

		Prediction					Class error
		1	2	3	4	5	
True	1	20	0	0	0	0	0.0
	2	0	20	0	0	0	0.0
	3	0	0	20	0	0	0.0
	4	0	0	0	20	0	0.0
	5	0	0	0	0	20	0.0

which are (3/4), (1/2), and (1/4) of the data. Figures 5(a)–5(c) show the results of variable importance with (3/4), (1/2), and (1/4) of the input data, respectively.

In order to provide a simple model, minimum number of inputs should be considered. To do so, the capture OOB error development with respect to the number of trees for each case is calculated. Here, the OOB error of using (3/4) of the data (P1–P12) is 0%; meanwhile, the OOB error of using half of the input data (P1–P8) is almost 1%, while the OOB error of using the quarter of the data (P1–P4) is 2%. Therefore, in this research, (3/4) of the data (P1–P12) are used to train the models; this process provides similar results to the process, whereas 100% of the data is used. This simplifies the model especially in the embedding process.

5.4. RFs Model Testing Results. The previous analysis is based on OOB error which is good enough to judge models in terms of validity and ability considering the number of inputs and ability of handling the problem. However, the previously developed models cannot be used for testing as they have been developed using 100% of the data. Thus, to propose a realistic model, the model should be tested by data not used in training. Therefore, two training-to-testing ratios of 80/20 and 60/40 are selected, and the proposed model is developed based on that. The idea here is to minimize the percentage of the training data subject to high accuracy so as to minimize the need capacity during the embedding process.

FIGURE 4: Variable importance.

First, the RF model is trained with 80% of data and tested by using the remaining data. The developed model in this case has an error of 1.25% as indicated in the confusion matrix in Table 2.

Secondly, the proposed RF model is trained based on 60% of data, while 40% are used for testing. The error noticed with this model is 5% as indicated in the confusion matrix that is shown in Table 3.

Figures 6(a) and 6(b) show the results of both cases. In general, the first model almost predicted all cases as it failed in predicating only one case, while the second model successfully predicted all cases. Here, these results do not mean that the second model (60/40) is better than the first model (80/20); on the contrary, the first model should be better in fact as it is trained more than the second model. However, both models fail sometimes as there is a small margin of error indicated in both cases. However, it is very clear that both models can predict these cases successfully.

5.5. Boosting Model Testing Results. Boosting is another powerful ensemble technique, whereas the performance of this technique is highly comparable to random forest that surpasses in some cases, and it is used to further investigate system performance. It is important to select an optimal number of trees for the developed model so as to achieve the fast model which do to not overfit data. Table 4 shows the number of trees used in developing the boosting model considering different dataset situation and training process.

Even so, the optimal number changes from single run to another, using optimal number of trees each case at a time. With the original dataset used, the system will be trained the same as previous parameters while prediction is achieved by an optimal number of trees with a train-to-testing ratio of 60/40. As a result, when the boosting model is trained using the modified data, the model showed very high accuracy

where all cases were identified correctly. This in fact slightly exceeds the proposed RF model. Moreover, it required modifying the parameters of the RF model and consequently complicates it so as to make it more accurate. Meanwhile, the proposed boosting model was more accurate and simpler as it did require any parameter modification. Moreover, it was developed based on less training-to-testing ratio.

5.6. Comparison between Proposed Models and ANN-Based Models. As mentioned in Section 1, most of the researchers utilize ANN for classification of power transformer currents. Moreover, some other researchers utilized optimization techniques to optimally select the number of hidden layers and hidden layers' neuron numbers so as to achieve minimum error. Thus, in order to show the superiority of the proposed models, different types of ANN-based models are taken as the benchmark. The conventional ANN model in [2] as well as ANN/PSO and ANN/IGSA in [2, 3], respectively, are taken as benchmarks in this research.

Table 5 shows a comparison between different techniques. It is clear from the table that all of the models are very accurate except for the RF model which has an accuracy that is slightly below other models (boosting, ANN, ANN/PSO, and ANN/IGSA). After all, we can say that all models are accurate. However, when it comes to AI-based techniques that are applied to physical systems such as electrical power systems, other issues should be considered. One of the most important issues is the easiness of embedding this technique. In fact, embedding control algorithms are essential so as to implement physically. As far as the algorithm is complex and large, the embedding process is challenging. Thus, by looking at the models in Table 5, we can say that optimised ANN models are more complex than conventional ANN, RF, and booting models. Meanwhile, 60-to-40 RF and boosting models are preferred as compared to 80-to-20 RF and boosting models. Meanwhile, the 60-to-40 boosting

FIGURE 5: (a) Variable Importance for (3/4) of the dataset. (b) Variable Importance for (1/2) of the dataset. (c) Variable importance for (1/4) of the dataset.

TABLE 2: Confusion matrix for the train-to-test ratio 80/20.

		Prediction					Class error
		1	2	3	4	5	
True	1	16	0	0	0	0	0.0
	2	0	16	0	0	0	0.0
	3	1	0	11	0	0	0.08333333
	4	0	0	0	18	0	0.0
	5	0	0	0	0	18	0.0

TABLE 3: Confusion matrix for the train-to-test ratio 60/40.

		Prediction					Class error
		1	2	3	4	5	
True	1	12	0	0	0	0	0.0
	2	0	12	1	0	0	0.07692308
	3	1	0	10	0	0	0.09090909
	4	0	1	0	10	0	0.09090909
	5	0	0	0	0	13	0.0

(a)

(b)

FIGURE 6: (a) Prediction results for the 80/20 RF model. (b) Prediction results of the 60/40 RF model.

TABLE 4: Optimum number of trees for different datasets.

	100% original dataset	100% modified dataset	80/20% modified dataset	60/40% modified dataset
C.V.	58	87	89	70
OOB	15	63	42	40

TABLE 5: Comparison between different techniques (accuracies mean the number of successful identification cases as compared to the total number of testing cases).

	ANN (%)	PSO-ANN (%)	IGSA-ANN (%)	RF 100 (%) training	RF 80-to-20 (%)	RF 60-to-40 (%)	Boosting 80-to-20 (%)	Boosting 60-to-40 (%)
Protection element	98	100	100	100	100	100	100	100
Conditional monitoring element	98				98	97	100	100

model is classified as the simplest and easiest to be embedded.

6. Conclusion

In this research, differential protection and conditional monitoring based on a sampled differential signal by using random forest and boosting models is done. An experimental dataset for a power transformer was used in this research. Then, these data samples are used to train the selected ensemble techniques. These models are assumed to issue tripping status for internal fault and no trip status otherwise. Meanwhile, all cases are classified including normal, inrush, overexcitation, CT saturation, and internal fault as a conditional monitoring system. The utilized dataset has been modified first so as to achieve maximum accuracy. After that, different training-to-testing ratios for validating the model 80-to-20% and 60-to-40% were applied. Results showed that for the proposed RF model, the accuracy of protection element was 100%, while less accurate conditional monitoring element (97%–98%) was noticed. On the other hand, the proposed boosting model showed better accuracy by achieving 100% of correct decisions for both cases. Finally, a comparison was conducted between the proposed models and other AI-based models, and the proposed boosting models were found to be the simplest and the most accurate models as compared to other models.

Data Availability

The data are available upon requests to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. O. Oliveira, A. S. Bretas, and G. D. Ferreira, "Adaptive differential protection of three-phase power transformers based on transient signal analysis," *International Journal of Electrical Power & Energy Systems*, vol. 57, pp. 366–374, 2014.
- [2] M. Yazdani-Asrami, M. Taghipour-Gorjilolaie, S. Mohammad Razavi, and S. Asghar Gholamian, "A novel intelligent protection system for power transformers considering possible electrical faults, inrush current, CT saturation and over-excitation," *International Journal of Electrical Power & Energy Systems*, vol. 64, pp. 1129–1140, 2015.
- [3] M. Geethanjali, S. Raja Slochanal, and R. Bhavani, "PSO trained ANN-based differential protection scheme for power transformers," *Neurocomputing*, vol. 71, no. 4–6, pp. 904–918, 2008.
- [4] M. Žarkovic and Z. Stojkovic, "Analysis of artificial intelligence expert systems for power transformer condition monitoring and diagnostics," *Electric Power Systems Research*, vol. 149, pp. 125–136, 2017.
- [5] A. Behvandi, S. G. Seifossadat, and A. Saffarian, "A new method for discrimination of internal fault from other transient states in power transformer using Clarke's transform and modified hyperbolic S-transform," *Electric Power Systems Research*, vol. 178, Article ID 106023, 2020.
- [6] B. Saravanan and A. Rathinam, "Inrush blocking scheme in transformer differential protection," *Energy Procedia*, vol. 117, pp. 1165–1171, 2017.
- [7] S. Hasheminejad and S. Esmaeili, "Transient actions analysis of power transformers based on S-transform and hidden Markov model," *International Transactions on Electrical Energy Systems*, vol. 24, no. 6, pp. 826–841, 2013.
- [8] M. Rasoulpoor and M. Banejad, "A correlation based method for discrimination between inrush and short circuit currents in differential protection of power transformer using Discrete Wavelet Transform: theory, simulation and experimental validation," *International Journal of Electrical Power & Energy Systems*, vol. 51, pp. 168–177, 2013.
- [9] D. Shi, J. Buse, Q. Wu, L. Jiang, and Y. Xue, "Fast identification of power transformer magnetizing inrush currents based on mathematical morphology and ANN," in *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–6, Detroit, MI, USA, July 2011.
- [10] E. Ali, A. Helal, H. Desouki, K. Shebl, S. Abdelkader, and O. P. Malik, "Power transformer differential protection using current and voltage ratios," *Electric Power Systems Research*, vol. 154, pp. 140–150, 2018.
- [11] Y. Sheng and S. Rovnyak, "Decision trees and wavelet analysis for power transformer protection," *IEEE Transactions on Power Delivery*, vol. 17, no. 2, pp. 429–433, 2002.
- [12] M. Tripathy, R. P. Maheshwari, and H. K. Verma, "Power transformer differential protection based on optimal probabilistic neural network," *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 102–112, 2010.
- [13] S. R. Samantaray and P. K. Dash, "Decision Tree based discrimination between inrush currents and internal faults in

- power transformer," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 1043–1048, 2011.
- [14] A. M. Shah and B. R. Bhalja, "Discrimination between internal faults and other disturbances in transformer using the support vector machine-based protection scheme," *IEEE Transactions on Power Delivery*, vol. 28, no. 3, pp. 1508–1515, 2013.
 - [15] O. Ozgonenel and S. Karagol, "Transformer differential protection using wavelet transform," *Electric Power Systems Research*, vol. 114, pp. 60–67, 2014.
 - [16] V. Barhate, K. Thakre, and M. Deshmukh, "Adaptable differential relay using fuzzy logic code in digital signal controller for transformer protection," in *Proceedings of the 57th International Scientific Conference on Power and Electrical Engineering*, Riga, Latvia, October 2016.
 - [17] J. Fernandes, F. Costa, and R. Medeiros, "Power transformer disturbance classification based on the wavelet transform and artificial neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 640–646, Vancouver, Canada, July 2016.
 - [18] H. Weng, S. Wang, Y. Wan, X. Lin, Z. Li, and J. Huang, "Discrete Fréchet distance algorithm based criterion of transformer differential protection with the immunity to saturation of current transformer," *International Journal of Electrical Power & Energy Systems*, vol. 115, Article ID 105449, 2019.
 - [19] A. M. Shah and B. R. Bhalja, "Fault discrimination scheme for power transformer using random forest technique," *IET Generation, Transmission & Distribution*, vol. 10, no. 6, pp. 1431–1439, 2016.
 - [20] H. Weng, S. Wang, X. Lin, Z. Li, and J. Huang, "A novel criterion applicable to transformer differential protection based on waveform sinusoidal similarity identification," *International Journal of Electrical Power & Energy Systems*, vol. 105, pp. 305–314, 2019.
 - [21] S. Bagheri, Z. Moravej, and G. B. Gharehpetian, "Classification and discrimination among winding mechanical defects, internal and external electrical faults, and inrush current of transformer," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 484–493, 2018.
 - [22] E. Cardelli, E. Della Torre, V. Esposito, and A. Faba, "Theoretical considerations of magnetic hysteresis and transformer inrush current," *IEEE Transactions on Magnetics*, vol. 45, no. 11, pp. 5247–5250, 2009.
 - [23] J. A. Martinez, R. Walling, B. A. Mork, J. Martin-Arnedo, and D. Durbak, "Parameter determination for modeling system transients-Part III: transformers IEEE PES task force on data for modeling system transients of IEEE PES working group on modeling and analysis of system transients using digital simulation (general systems subcommittee)," *IEEE Transactions on Power Delivery*, vol. 20, no. 3, pp. 2051–2062, 2005.
 - [24] A. Tokic, V. Milardic, I. Uglešić, and A. Jukan, "Simulation of three-phase transformer inrush currents by using backward and numerical differentiation formulae," *Electric Power Systems Research*, vol. 127, pp. 177–185, 2015.
 - [25] F. de Leon and A. Semlyen, "Complete transformer model for electromagnetic transients," *IEEE Transactions on Power Delivery*, vol. 9, no. 1, pp. 231–239, 1994.
 - [26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.