International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/IJSRCSEIT

Diagnosis of Transformer Faults Using the Multi-Class Adaboost Algorithm

U. Preetha[1], Dr. K. Venkataramana[2]

PG Scholar[1], Associate Professor[2]

Department of CSE, Vemu Institute of Technology, P. Kotha Kota, Chittoor, Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted: 10 July 2023

Published: 28 July 2023

Publication Issue

Volume 9, Issue 4

July-August-2023

Page Number

211-225

ABSTRACT

Traditional shallow machine learning algorithms are incapable of exploring the link between fault data of oil-immersed transformers, resulting in low fault diagnostic accuracy. In answer to this challenge, this study provides a transformer defect diagnostic approach based on Multi-class AdaBoost Algorithms. First, the AdaBoost technique is merged with Support Vector Machines (SVM). The SVM is improved by the AdaBoost algorithm, and the transformer defect data is thoroughly investigated.The dynamic weight is then introduced into the Particle Swarm Optimization (PSO); by realtime updating of the particle inertia weight, the particle swarm optimization algorithm's search accuracy and optimization speed are improved, and the improved particle swarm optimization algorithm (IPSO) is used to optimize the parameters of the SVM. Finally, the uncoded ratio technique creates a new gas group collaboration by studying the link between the dissolved gas in the transformer oil and the fault type.As the input feature vector, the better ratio technique is built. Based on simulations of 117 sets of IECTC10 standard data and 419 sets of transformer fault data collected in China, the diagnosis method proposed in this paper has a strong search ability, a fast convergence speed, and a significant improvement in diagnostic accuracy when compared to traditional methods.

Keywords : Power transformers, fault detection, a support vector machine and a multi-class AdaBoost method are all included.

I. INTRODUCTION

The oil-immersed transformer is a critical component of the power grid, performing critical functions such as power transmission and conversion. When the fault arises, it will result in significant economic losses. As a result, transformer fault diagnosis is performed to detect hidden flaws in real time and undertake maintenance based on the kind of fault. It is critical to limit the loss and damage caused by transformer failure

Copyright © 2023 The Author(s): This is an open-access article distributed under the terms of the Creative Commons Attribution **4.0 International License (CC BY-NC 4.0)** which permits unrestricted use, distribution, and reproduction in any medium for non-commercial use provided the original author and source are credited.

and to increase the electrical grid's stability and reliability .In normal operation, the oil-immersed transformer generates a small quantity of gas owing to the ageing and cracking of the insulation, which is dissolved in the transformer oil. These gases' primary constituents are hydrogen (H2), methane (CH4), and ethane (C2H4). Specific gas components will rapidly grow when transformers have distinct defects. When insulating oil is warmed, for example, the ratio of CH4 to C2H4 rapidly increases; during high-energy discharge, the concentrations of H2 and C2H2 rise. This displays the kind and severity of transformer problems. A substantial link exists between changes in gas composition. DGA technology employs non-electrical quantities as diagnostic indications, is not impacted by electromagnetics, is versatile, and is commonly utilized in the online diagnosis of oil-immersed transformers. According to some experts, the three-ratio Rogers ratio technique Duval triangle method and other DGA-based principles are straightforward and have played an important role. Nonetheless, they all have insufficient state coding and over-absolute coding constraints. In actual applications, such issues have specific limitations. In recent years, as artificial intelligence technology has advanced, diagnostic methods have evolved from the classic IEC three-ratio approach and enhanced three-ratio method to machine learning and other artificial intelligence methods such as neural networks and Support Vector Machines (SVM).

In response to this problem, the ensemble learning AdaBoost algorithm builds multiple weak classifiers over multiple iterations, affects the weight of the next-generation classifier samples based on the classification results, performs deep mining of the samples by assigning different weights to the samples, and finally weights Voting to produce a strong classifier for transformer fault diagnosis. Zhou and colleagues employed a cloud diagnostic model, decision tree algorithm, extreme learning machine, and other weak classifiers before employing the AdaBoost method to diagnose transformer faults. Because of the small number of fault samples from big oil-immersed transformers, the AdaBoost method improves the accuracy of cloud models, decision trees, and other algorithms. The setup of SVM hyperparameters, on the other hand, necessitates previous empirical knowledge, and the optimal hyperparameter selection remains an unresolved topic in related study domains. Zhang produced good results by using the enhanced krill algorithm and genetic algorithm to set the hyperparameters of SVM. Still, optimization efficiency and hyperparameter accuracy need to be improved. This work offers an Improved Particle Swarm Optimization (IPSO) to improve SVM core parameters and penalty factors on this premise. It iteratively integrates the AdaBoost method with SVM to generate numerous IPSOSVM weak classifiers, and other conduct in-depth mining of transformer failure data to increase SVM classification impact.

II. RELATED WORKS

Users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a de ated (or in ated) overall average rating of an app which can a ect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. is is especially signi cant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating

drastically Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. is is especially signicant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drastically Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. is is especially signicant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star

drastically Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. is is especially signicant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to ve stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star

ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; "Awesome App". One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; "Awesome App". One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; "Awesome App". One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for

the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five stars, with one star denoting a high sense of dissatisfaction with the app and five stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, often the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be five stars, but the actual rating is one star! Such inconsistent ratings can lead to a deflated (or inflated) overall average rating of an app which can affect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive A biased feedback about the application that does not represent ground reality. This is especially significant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Mobile app distribution platforms such as Google Play Store allow users to share their feedback about downloaded apps in the form of a review comment and a corresponding star rating. Typically, the star rating ranges from one to five

stars, with one star denoting a high sense of dissatisfaction with the app and ve stars denoting a high sense of satisfaction. Unfortunately, due to a variety of reasons, oen the star rating provided by a user is inconsistent with the opinion expressed in the review. For example, consider the following review for the Facebook App on Android; “Awesome App”. One would reasonably expect the rating for this review to be ve stars, but the actual rating is one star! Such inconsistent ratings can lead to a deated (or inated) overall average rating of an app which can aect user downloads, as typically users look at the average star ratings while making a decision on downloading an app. Also, the app developers receive a biased feedback about the application that does not represent ground reality. is is especially signicant for small apps with a few thousand downloads as even a small number of mismatched reviews can bring down the average rating drasticall Power transmission networks [1] are critical components of smart grids. Fast and accurate faulty-equipment identification is crucial for power system fault diagnosis; nevertheless, it is challenging due to ambiguous and partial fault alarm messages in fault occurrences. In the scope of membrane computing, this research provides a new defect diagnostic approach for transmission networks. We first propose a class of self-updating spiking neural P systems based on the biological apoptosis process and its self-updating matrix reasoning method. The successfully unifies the attribute reduction ability of rough sets and the apoptosis mechanism of biological neurons in a P system for the first time, where the apoptosis algorithm for condition neurons is designed to reduce redundant information in fault signals.

Compared with conventional methods [2] of fault diagnosis for power transformers, which have defects such as imperfect encoding and too absolute encoding boundaries, this paper systematically discusses various intelligent approaches applied in fault diagnosis and decision making for large oil-immersed power transformers based on dissolved gas analysis (DGA), including expert system (EPS), artificial neural network (ANN), fuzzy theory, rough sets theory (RST), grey system theory (GST), swarm intelligence (SI) algorithms, data mining technology, machine learning (ML), and other intelligent diagnosis tools, and summarizes existing problems and solutions. According to the results of this study, a single intelligent technique for fault diagnosis may only reflect the operating state of the transformer in one specific aspect, resulting in varied degrees of inadequacies that cannot be successfully remedied. This paper provides a full and systematic overview of numerous intelligent techniques to power transformer failure diagnosis and decision making, in which their benefits and drawbacks are extensively analyzed, as well as improvement plans and future development trends offered. Furthermore, the study proposes that a range of intelligent algorithms should be coupled for mutual complementation to construct a hybrid defect diagnostic network, so that these algorithms do not fall into a local optimum. Furthermore, the detecting devices must be improved in order to get appropriate characteristic gas data samples.

Deep learning, as a revolutionary [3] approach in machine learning, has produced outstanding achievements in every discipline with its great self-feature extraction potential in recent years. Because power transformers are at the heart of energy conversion and transmission in power grids, learning its fault diagnosis procedures is critical for detecting defects sooner and making the system safer. Because the power transformer's fault mechanism is complicated, the thesis investigates its fault diagnostic techniques primarily using the vibration signal of the power transformer, the partial discharge signal, and the gas content dissolved in the oil dissolved on the basis of the deep learning theory. The vibration signal is non-stationary and changes over time. If a conventional signal is evaluated. We can better identify the characteristics of the vibration signal in case of defects by detecting the vibration modes within the signals, which can aid in fault identification later. The sparse auto encoder can effectively abstract data

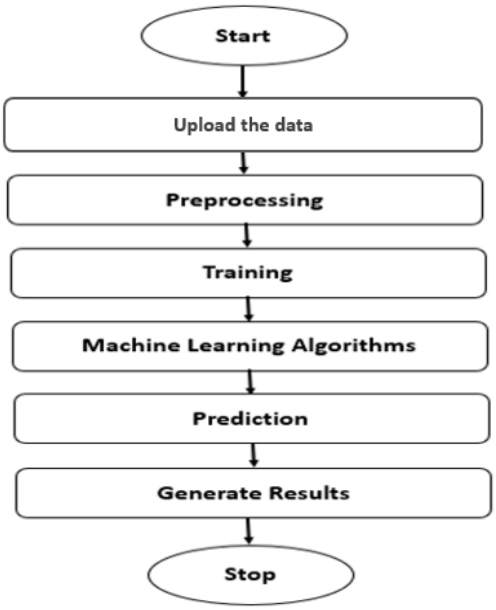
properties to make classification easier, and it has a higher optimization capacity than the typical BP neutral network.

Transformers are the primary piece [4] of equipment used in power system functioning. Undiagnosed failures in the transformer's internal components will increase downtime and result in considerable economic losses. Efficient and accurate transformer failure diagnostics is a critical component of power grid research that contributes to the safe and stable operation of the power system. Traditional transformer fault diagnosis methods suffer from low accuracy, difficulties in properly processing fault characteristic information, and superparameters that impair transformer fault identification. We present a transformer defect diagnostic approach based on improved particle swarm optimization (IPSO) and multigrained cascade forest in this work (gcForest). Given the association between the characteristic gas dissolved in oil and the kind of fault, the noncode ratios of the characteristic gas dissolved in oil are first determined as the model's characteristic parameter. The IPSO technique is then used to repeatedly refine the gcForest model parameters to find the best parameters with the maximum diagnostic accuracy.

III. METHODOLOGY

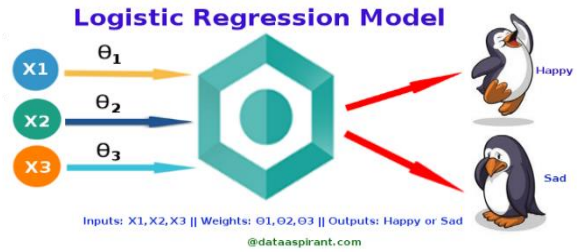
Proposed system:

The proposed system for diagnosing transformer faults utilizes multi-class machine learning algorithms, including logistic regression, MLP classifier, SVC, random forest, ANN, CNN, AdaBoost, decision tree, and LGBM classifier. These algorithms enable accurate fault classification, allowing for effective identification and troubleshooting of transformer issues. By leveraging the power of these diverse algorithms, the system enhances diagnostic accuracy and facilitates efficient maintenance of transformer systems.



IV. IMPLEMENTATION

1. Logistic Regression:



Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical. For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)
- Consider a scenario where we need to classify whether an email is spam or not. If we use linear regression for this problem, there is a need for setting up a threshold based on which classification can be done. Say if the actual class is malignant, predicted continuous value 0.4 and the threshold value is 0.5, the data point will be

classified as not malignant which can lead to serious consequence in real time.

- From this example, it can be inferred that linear regression is not suitable for classification problem. Linear regression is unbounded, and this brings logistic regression into picture. Their value strictly ranges from 0 to 1.

Where to use logistic regression

Logistic regression is used to solve classification problems, and the most common use case is binary logistic regression, where the outcome is binary (yes or no). In the real world, you can see logistic regression applied across multiple areas and fields.

- In health care, logistic regression can be used to predict if a tumor is likely to be benign or malignant.
- In the financial industry, logistic regression can be used to predict if a transaction is fraudulent or not.
- In marketing, logistic regression can be used to predict if a targeted audience will respond or not.

The three types of logistic regression

1. **Binary logistic regression** - When we have two possible outcomes, like our original example of whether a person is likely to be infected with COVID-19 or not.
2. **Multinomial logistic regression** - When we have multiple outcomes, say if we build out our original example to predict whether someone may have the flu, an allergy, a cold, or COVID-19.
3. **Ordinal logistic regression** - When the outcome is ordered, like if we build out our original example to also help determine the severity of a COVID-19 infection, sorting it into mild, moderate, and severe cases.

2. Random Forest Classifier:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

A random forest eradicates the limitations of a decision tree algorithm. It reduces the over fitting of datasets and increases precision. It generates predictions without requiring many configurations in packages (like Scikit-learn).

Features of a Random Forest Algorithm:

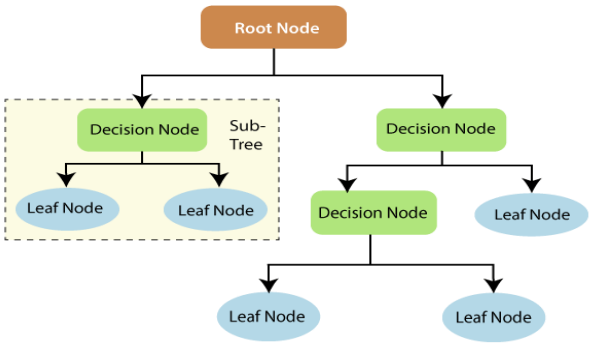
- It's more accurate than the decision tree algorithm.
- It provides an effective way of handling missing data.
- It can produce a reasonable prediction without hyper-parameter tuning.
- It solves the issue of over fitting in decision trees.
- In every random forest tree, a subset of features is selected randomly at the node's splitting point.

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work.

A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This

sequence continues until a leaf node is attained. The leaf node cannot be segregated further.

The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.



The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An overview of these fundamental concepts will improve our understanding of how decision trees are built.

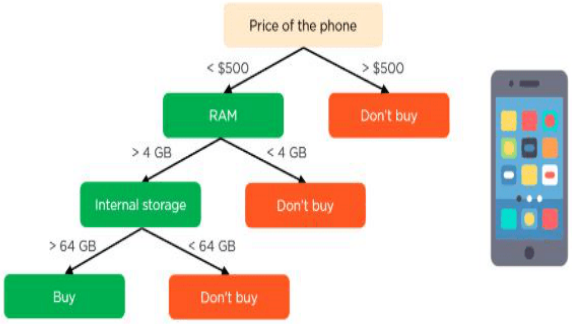
Entropy is a metric for calculating uncertainty. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables.

The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y.

Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees.

Let's take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram.

The root node and decision nodes of the decision represent the features of the phone mentioned above. The leaf node represents the final output, either *buying* or *not buying*. The main features that determine the choice include the price, internal storage, and Random Access Memory (RAM). The decision tree will appear as follows.



Applying decision trees in random forest

The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction.

Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let's assume we have only four decision trees. In this case, the training data comprising the

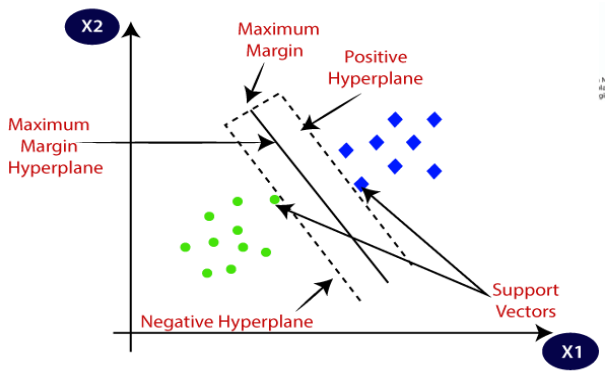
phone's observations and features will be divided into four root nodes.

The root nodes could represent four features that could influence the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees.

The outcome chosen by most decision trees will be the final choice. If three trees predict *buying*, and one tree predicts *not buying*, then the final prediction will be *buying*. In this case, it's predicted that the customer will buy the phone.

V. SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane:

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need

to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

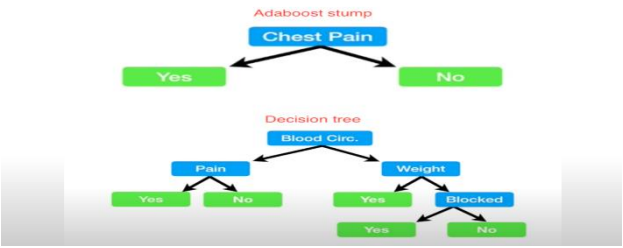
We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

VI. ADABOOST ALGORITHM

AdaBoost, also known as Adaptive Boosting, is a Machine Learning approach that is utilized as an Ensemble Method. The most frequent AdaBoost method is decision trees with one level, which is decision trees with just one split. These trees are often referred to as Decision Stumps.Boosting is an ensemble modeling approach introduced by Freund and Schapire in 1997. Since then, Boosting has been a popular technique for dealing with binary classification issues. These algorithms boost prediction power by transforming a large number of weak learners into strong learners.The basic idea behind boosting techniques is that we build a model on the training dataset first, then build a second model to correct the faults in the first model. This technique is repeated until the mistakes are reduced and the dataset is accurately predicted.To illustrate, imagine you created a decision tree algorithm using the Titanic dataset and obtained an accuracy of 80% from it.



Boosting algorithms have grown in prominence in data science and machine learning contests in recent years. To attain great accuracy, the majority of these competition winners employ boosting algorithms. These data science challenges give a global forum for learning, researching, and solving corporate and government problems. Boosting algorithms combine many low accuracy (weak) models to produce high accuracy (strong) models. It may be used in a variety of fields, including credit, insurance, marketing, and sales. Machine learning techniques such as AdaBoost, Gradient Boosting, and XGBoost are extensively utilized to win data science contests. In this course, you will learn about the AdaBoost ensemble boosting method.

Decision Tree:

A decision tree is a machine learning algorithm that builds a hierarchical structure of decisions and conditions to classify or predict outcomes. It starts with a root node and splits data based on features at each internal node, resulting in a tree-like structure. At each leaf node, a decision or prediction is made. The decision tree algorithm selects the best features and splits based on criteria such as information gain or Gini impurity to maximize the tree's predictive accuracy. Decision trees are interpretable, can handle numerical and categorical data, and are widely used for classification and regression tasks in various domains.

CNN

Convolutional Neural Network (CNN) is a deep learning algorithm widely used for image classification tasks. It comprises multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract relevant features from input images by applying filters and performing element-wise multiplications. The pooling layers reduce the spatial dimensions of the feature maps, aiding in capturing important patterns. The fully connected layers integrate the extracted features and make predictions based on learned parameters. CNNs excel in capturing spatial dependencies, making them ideal for image recognition. Their hierarchical structure and ability to learn complex patterns have propelled their success in various computer vision tasks, including object detection and image segmentation.

MLP

The MLP (Multi-Layer Perceptron) classifier is a type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons. It is a feedforward neural network, where information flows in a single direction from the input layer to the output layer. Each neuron in the MLP receives inputs from the previous layer and computes a weighted sum, followed by applying an activation function. This non-linear activation function allows the MLP to learn complex patterns and make nonlinear decisions. The MLP classifier is trained using backpropagation, which adjusts the weights and biases of the neurons to minimize the error between the predicted and actual outputs. Its versatility and ability to handle non-linear data make it a popular choice for various classification tasks.

LightGBM

LightGBM is a gradient boosting framework that stands for Light Gradient Boosting Machine. It is a popular machine learning algorithm known for its high-performance and efficiency. LightGBM uses a novel gradient-based approach for tree splitting and handles large datasets efficiently by utilizing histogram-based algorithms. It supports various types of tasks, including classification, regression, and ranking. LightGBM uses leaf-wise tree growth, which leads to faster training times and lower memory usage compared to other boosting algorithms. It also incorporates features like bagging, feature importance, early stopping, and categorical feature handling. These qualities make LightGBM a valuable tool for tackling complex machine learning problems with large-scale datasets.

ANN

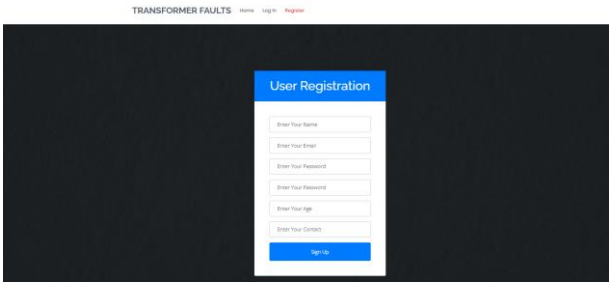
ANN (i.e., Artificial Neural Network) is a component of a computer system created to mimic how the human brain evaluates and interpret data. It serves as the cornerstone of artificial intelligence (AI) and finds solutions to issues that would be unattainable or challenging by human or statistical standards. As more data becomes available, ANNs' self-learning abilities allow them to provide better results.Processing units, the artificial neurons found in an ANN, number in the hundreds or thousands and are linked together by nodes. These processing units have input and output components. Based on an internal weighting mechanism, the input units receive information in a variety of forms and structures, and the neural network makes an effort to study regarding the actual information supplied in order to generate a result of problems. Backpropagation, which stands for backward propagation of error, is a set of learning rules that ANNs use to refine their output results, just as people use rules and guidelines to produce a result or output.

VII. RESULTS AND DISCUSSION

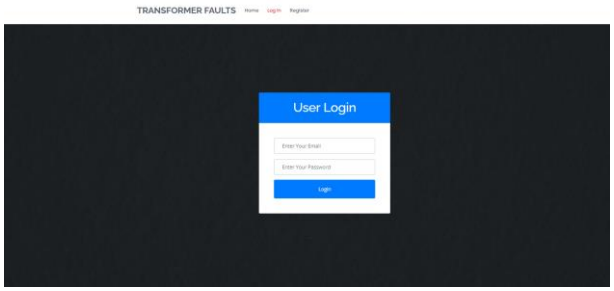
The following images will visually depict the process of our project.

Home page: In this home page we can see the logo designing of our website.

Register:

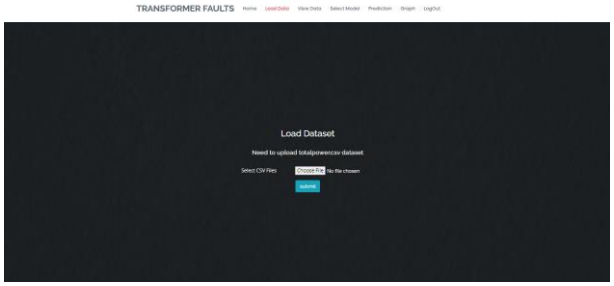


Login:

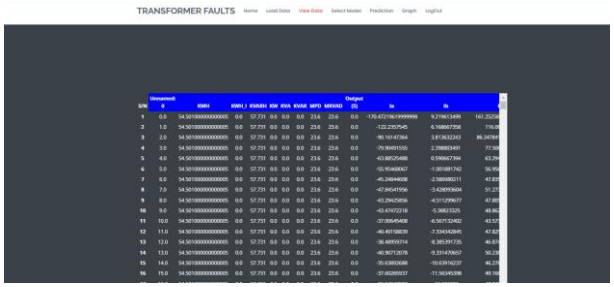


Userhome:

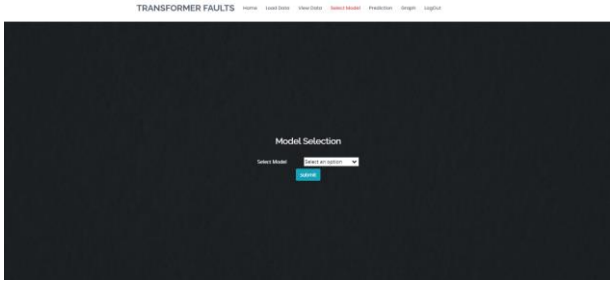
Load Dataset: This is page to load the dataset.



View Dataset: To view the dataset in the project.



Model training and Evaluation: This page increases the model training process and evaluation and prediction page.



Prediction:

Graph:

VIII. CONCLUSION

The AdaBoost technique is utilized in this study to improve the improved particle swarm optimization (IPSO) optimized support vector machine (SVM) transformer fault diagnostic model.And, using the IEC TC 10 data set, perform fault diagnosis with domestic transformer data by analyzing the relationship between the dissolved gas in the transformer oil and the fault type, applying the uncoded ratio method to form a new gas combination as the characteristic parameters of the fault model, and establishing an improved ratio method as the input feature vector.The IPSO-SVM fault diagnostic model, augmented by the AdaBoost algorithm, can diagnose the kind of transformer problem effectively and precisely. It outperforms the classic SVM and AdaBoost algorithms in classification accuracy.During the optimization process, the classical PSO method is prone to the problem of local optimum and premature convergence. The search performance of the PSO method is improved by incorporating linearly declining weights to replace the quantitative weights of the classic PSO algorithm. When compared to the standard PSO method, the IPSO algorithm has superior search capabilities.Based on the examination of DGA fault data, the suggested enhanced ratio technique has a

considerable improvement in accuracy when compared to the traditional ratio method and the DGA fault gas data.

IX. REFERENCES

[1]. Y. F. Wang, 400 Cases of Fault Diagnosis of Common Electrical and Electronic Control Equipment. Beijing, China: China Electr. Power Press, 2011, p. 1020.

[2]. Y. Liu and Y. P. Ni, "Transformer fault diagnosis method based on grey correlation analysis of three ratios," High-Voltage Technol., no. 10, pp. 16–17 and 27, 2002, doi: 10.3969/j.issn.1003-6520.2002.10.007.

[3]. T. F. Yang, P. Liu, J. L. Li, and Y. Hu, "New fault diagnosis method of power transformer by combination of FCM and IEC three-ratio method," Chin. J. Anal. Chem., vol. 33, no. 6, pp. 66–70, Aug. 2007, doi: 10.1016/S1872-2040(07)60059-0.

[4]. M. Duval, "Dissolved gas analysis: It can save your transformer," IEEE Elect. Insul. Mag., vol. 5, no. 6, pp. 22–27, Nov. 1989, doi: 10.1109/57.44605.

[5]. Mineral Oil-Filled Electrical Equipment in Service–Guidance on the Interpretation of Dissolved and Free Gases Analysis, Standard UNE-EN 60599- 2016.AENOR ES,AENOR, 2016.

[6]. R. Rogers, "IEEE and IEC codes to interpret incipient faults in transformers, using gas in oil analysis," IEEE Trans. Electr. Insul., vol. EI-13, no. 5, pp. 349–354, Oct. 1978, doi: 10.1109/TEI.1978.298141.

[7]. X. Shi, Y. L. Zhu, X. G. Ning, L. Wang, Q. G. Shun, and G. Q. Chen, "Fault diagnosis of power transformer based on deep self-encoding network," Electr. Power Autom. Equip., vol. 36, no. 5, pp. 122–126, 2016, doi: 10.16081/j.issn.1006-6047.2016.05.021.

[8]. Y. Zhang, X. Li, H. Zheng, H. Yao, J. Liu, C. Zhang, H. Peng, and J. Jiao, "A fault diagnosis model of power transformers based on dissolved gas analysis features selection and improved krill herd algorithm optimized support vector machine," IEEE Access, vol. 7, pp. 102803–102811, 2019, doi: 10.1109/ACCESS.2019.2927018.

[9]. C. Y. Zhang, Z. F. Lin, D. Liu, and J. L. Huang, "Transformer fault diagnosis based on normal cloud model and improved Bayesian classifier," Elect. Meas. Instrum., vol. 54, no. 4, pp. 50–56, 2017.

[10]. J. Y. Wu, G. Q. Xia, T. Li, B. Gao, and G. N. Wu, "Evaluation of the aging state of transformer oil-paper insulation based on time-domain dielectric method and dynamic Bayesian network," High Voltage App., vol. 55, no. 10, pp. 196–203, 2019, doi: 10.13296/j.1001-1609.hva.2019. 10.032. VOLUME 10, 2022 1531 J. Li et al.: Transformer Fault Diagnosis Based on Multi-Class AdaBoost Algorithm

[11]. T. Yi, Y. Xie, H. Zhang, and X. Kong, "Insulation fault diagnosis of disconnecting switches based on wavelet packet transform and PCA-IPSOSVM of electric fields," IEEE Access, vol. 8, pp. 176676–176690, 2020, doi: 10.1109/ACCESS.2020.3026932.

[12]. H. Keskes and A. Braham, "Recursive undecimated wavelet packet transform and DAG SVM for induction motor diagnosis," IEEE Trans. Ind. Informat., vol. 11, no. 5, pp. 1059–1066, Oct. 2015, doi: 10.1109/TII.2015.2462315.

[13]. H. Xu and H. Yuan, "An SVM-based adaboost cascade classifier for sonar image," IEEE Access, vol. 8, pp. 115857–115864, 2020, doi: 10.1109/ACCESS.2020.3004473.

[14]. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, 1995, doi: 10.1006/jcss.1997.1504.

[15]. F. Wang, Z. Li, F. He, R. Wang, W. Yu, and F. Nie, "Feature learning viewpoint of adaboost and a new algorithm," IEEE Access, vol. 7, pp. 149890–149899, 2019, doi: 10.1109/ACCESS.2019.2947359.

Cite this article as :

U. Preetha, Dr. K. Venkataramana, "Diagnosis of Transformer Faults Using the Multi-Class Adaboost Algorithm", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 4, pp.211-225, July-August-2023.