DIABETES PREDICTION USING FEATURE EXTRACTION AND MACHINE LEARNING MODELS

Bhargavi Poyekar*, Rishita Mote†, Dr. Preeti Vinayakray‡
Department of Computer Engineering, Sardar Patel Institute Of Technology, Mumbai, India
Email: *bhargavi.poyekar@spit.ac.in, †rishitamote@gmail.com,
‡ preeti.vinayakray@spit.ac.in

***Abstract*—Diabetes is a disorder wherein the production of insulin is stopped or it is not responded to by the body. This results in high blood glucose. Our food produces blood glucose and insulin produced by the pancreas makes it possible for the glucose to reach the cells. High blood sugar may lead to kidney failure or heart disease. The risk and the severity of diabetes can be reduced by predicting it at an early stage. In this research paper, we strive to propose machine learning algorithms namely KNN, Decision Tree, Random Forest, LGBM, and Adaboost that help predict diabetes. All these algorithms are built using the Pima Indian Diabetes (PID) dataset by taking several parameters such as glucose, skin thickness, insulin, age, etc. that help locate diabetes. We trained our models and observed that LGBM outperformed all other models by giving an accuracy of 89.85% and AUC as 0.95. Thus, LGBM is more efficient algorithm in distinguishing diabetic and non-diabetic person.**

***Index Terms*—Diabetes, LGBM, Machine learning, PID dataset.**

I. Introduction

Diabetes is an escalating problem for both developing and developed countries. The alarming rise of diabetes across all age groups is mainly because of urbanization, unhealthy diet as people are consuming fast processed food rather than healthy food, consumption of alcohol and tobacco use, stress levels, growing environmental pollution, and changing lifestyles. Diabetes is known to be a disease that is almost impossible to cure. The illness is progressive and hence a diabetic person is bound to maintain his/her body weight, do regular physical activity and minimize consumption of alcohols and sugar products in order to sustain life. In recent years, India has become the capital of diabetes. The estimates state that one in every six people who have diabetes is identified as Indian. India has topped the second position after China which has over 116 million diabetics as revealed by [1]. It is a sad reality that India is a developing country and due to poor healthcare facilities, most people don't even know that they have diabetes. As per the International Diabetes Federation (IDF), 95% of the patients are diagnosed with type 2 diabetes and the numbers are high among the young population. Hence it is very necessary that the diabetes is predicted at an early stage so that the patient is able to take the required measures in order to reduce the severity. In one of the government surveys conducted during 2015-2019 [2], it was observed that India has 11.8% of the prevalence of diabetes. The male prevalence was observed as 12% while that of females as 11.7%. According to the mentioned survey, it was found that out of 63,000, around 56,771 or 90.1% were diagnosed with diabetes. Hence, early diagnosis of diabetes is indispensable. In order to gain control over such large numbers, it is essential to detect diabetes and thus prevent the person to control the sugar level at early stage. After extensive research, we implemented machine learning algorithms which can be applied in real life to detect diabetes. In this literature, we propose a new framework for predicting diabetes using the PIMA Indian Diabetes Dataset. Feature extraction and data cleaning are the heart of our proposed methodology. We first cleaned our data which included detection of outliers and handling missing values of the dataset. Then by analyzing the correlation matrix graph we combined features that are highly correlated with each other. Then before splitting the dataset into testing and training, we standardized our data to bring each feature to the same scale. Then we implemented six Machine learning classification models which are KNN, Decision Tree, Random Forest, LGBM, and Adaboost. The accuracy and area under the curve (AUC) were considered for evaluating the performance of each model. We observed that the LGBM classifier gave the best accuracy among all the models. Our proposed methodology for diabetes outperforms the articles and papers mentioned in the literature survey. Our research gave us positive results and can be used by researchers for further study on this topic.

In the following sections we have discussed: Section II presents the literature survey related to our study, Section III presents the Methodology, Section IV elaborates our results and Section V concludes our paper.

II. Literature Survey

In this section, various algorithms that have been applied for diabetes classification and diagnosis have been studied. In [3], they used the Pima Indian Diabetes dataset and used ML algorithms like Decision tree, Logistic regression, and Naive Bayes. They achieved the best results with Logistic regression.

Similarly in [4], instead of Logistic Regression, Support Vector Machine (SVM) was used. They used various measures

for evaluating the performance of the model and got the best accuracy using Naive Bayes.

Hasan et al. in [5] performed data cleaning by handling missing values and outlier rejection. Then they pre-processed the data using feature selection, and data standardization. And then they used different ml models and MLP. The weighted ensembling of various ML models was proposed by this literature for improving the prediction of diabetes. Area Under ROC Curve (AUC) was used as the metric of comparison and optimized using Grid Search while performing parameter tuning. A number of data preprocessing models and ML models were used to maximize the AUC score of prediction of diabetes for the same dataset. They used the best ML classifier as the baseline model for comparison with their proposed classifier. An ensembling classifier by combining ML models for improving diabetes prediction was proposed by them.

Kishore G. et al. in [6] used the five algorithms namely SVM, Decision Tree, Naive Bayes, Logistic Regression, and KNN for diabetes prediction. They found out that random forest had a better accuracy rate of nearly 75% and also a better accuracy can be achieved by the addition of any other algorithm.

Syed [7] et al. in their paper, studied the existence and association between the exposures and outcomes using conventional risk factors of diabetes. The techniques that they used to analyze Type 2 Diabetes mellitus (T2DM) risk prediction were the Chi-Squared test and binary logistic regression. Synthetic Minority Over-sampling Technique (SMOTE) was used as a class balancer to classify patients with a higher risk of diabetes with the help of the F1 Score measure. They obtained an improved score by tuning the best classifier's hyper-parameters using 10-fold cross-validation.

In [8], the authors proposed Diabetes Mellitus classification on the imbalanced data with Missing values (DMP_MI) algorithm. Naive Bayes was used for substituting the missing values for data normalization. After that, an adaptive synthetic sampling method (ADASYN) was used to reduce the impact of class imbalance on diabetes prediction performance. The comprehensive set of evaluation indicators were used to evaluate predictions generated using Random Forest (RF) classifier.

The proposed algorithm in [9] uses Goldberg's Genetic algorithm in the pre-processing stage to select the essential features from the dataset and a Multi-Objective Evolutionary Fuzzy Classifier is applied to it. The principle of maximum classifier rate and minimum rules is the basis of this algorithm. The feature selection with GA reduced the number of features to 4 from 8 and the rate of the classifier was enhanced to 83.0435 % with NSGA II in training and testing rate of 70% and 30% respectively.

III. Methodology

A. Dataset

The dataset used for training and testing the models is the publically available Pima Indians Diabetes Dataset [10] which was obtained from Kaggle. This dataset consists of 768 females of Pima Indian Heritage with an age no less than 21 years old. It consists of one target variable, Outcome, and eight features such as skin thickness, pregnancies, Diabetes Pedigree Function and Age, glucose, blood pressure, BMI, insulin as shown in Table-I.

TABLE I
Description of Attributes in Dataset

Attributes	Description
Pregnancies	Number of pregnancies
Glucose	Concentration of Plasma glucose
Blood Pressure	Diastolic blood pressure (mm Hg)
Skin Thickness	Skin fold thickness of triceps (mm)
Insulin	2 hours Serum Insulin (mu U/ml)
BMI	Body mass index (kg/m^2)
Diabetes Pedigree Function	Diabetes pedigree function
Age	Age of a person (years)
Outcome	Class Variable: 0 or 1 (268 of 768 are 1,others 0)

B. Proposed Methodology

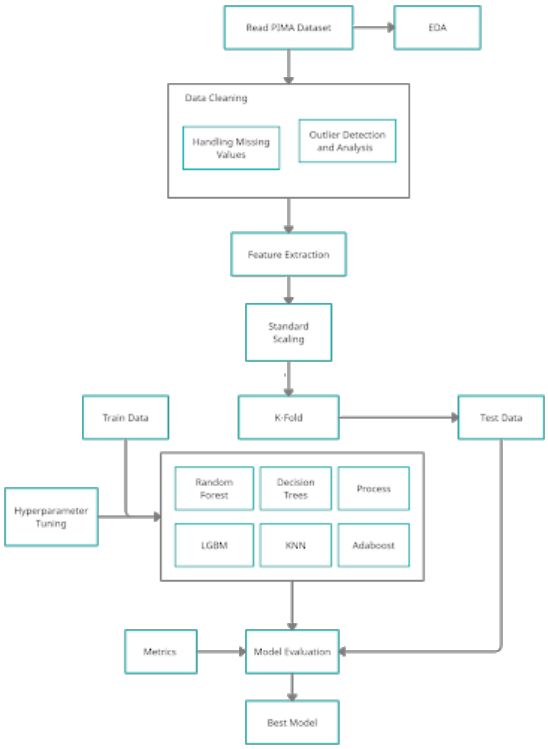

Fig. 1. Proposed System Diagram for Diabetes Prediction

The proposed system diagram shown in Fig. 1 first performs data cleaning, feature extraction, and data preprocessing. Then the models are trained and the hyperparameters are tuned. Finally, the models are evaluated and the best model is selected. These steps are further elaborated in the following sections.

a) Data Cleaning: The two main steps in Data Cleaning include handling Missing Values and detecting the outliers.

An outlier is a data point that deviates from other data points. Outliers, if not handled correctly, can greatly affect the accuracy of a model. Hence they have to be handled carefully. The Pima Indians Dataset has many outliers, especially for the 'Insulin' attribute. Hence the removal of these outliers can lead to a loss of important information from the dataset. The proposed system handles outliers by replacing the upper and lower whiskers with the upper and lower limit respectively. The lower and upper limit is shown in (1) and (2):

$$lowerlimit = Q3 - 1.5 * IQReq \tag{1}$$

$$upperlimit = Q3 + 1.5 * IQReq \tag{2}$$

where Q3 and IQR stand for 3rd Quartile and Interquartile Range respectively.
The outlier handling function F(x) as shown in (3):

$$F(x) = \begin{cases} \text{upper limit}, & \text{if x} \geq upperlimit \\ \text{lower limit}, & \text{if x} \leq lowerlimit \end{cases} \tag{3}$$

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Fig. 2. No. of Missing values per column

The number of missing values per column are shown in Fig. 2. The features 'Insulin' and 'Skin Thickness' have maximum missing values.

Outcome	Glucose	BloodPressure	SkinThickness	Insulin	BMI
0.0	107.0	70.0	27.0	102.5	30.1
1.0	140.0	74.5	32.0	169.5	34.3

Fig. 3. Median Value grouped by Outcome Variable

These features show differences in their median values when analyzed with respect to the 'Outcome' variable as shown in Fig. 3. The missing values present in a column are handled by replacing them with the median value of that column, grouped by the 'Outcome' variable.

b) Feature Extraction and Pre-processing: The correlation matrix of all features with one another is shown in Fig. 4. The features 'BMI' and 'Skin Thickness' have the highest correlation (0.58). Also, 'Insulin' and 'Glucose' have a correlation of 0.53. The correlation between these features can be better visualized from the scatter plot shown in Fig. 5 and Fig. 6.

These feature pairs are combined to form new features as shown in (4) and (5):

$$IG = Insulin * Glucoseeq \tag{4}$$

$$BS = BMI * SkinThicknesseq \tag{5}$$

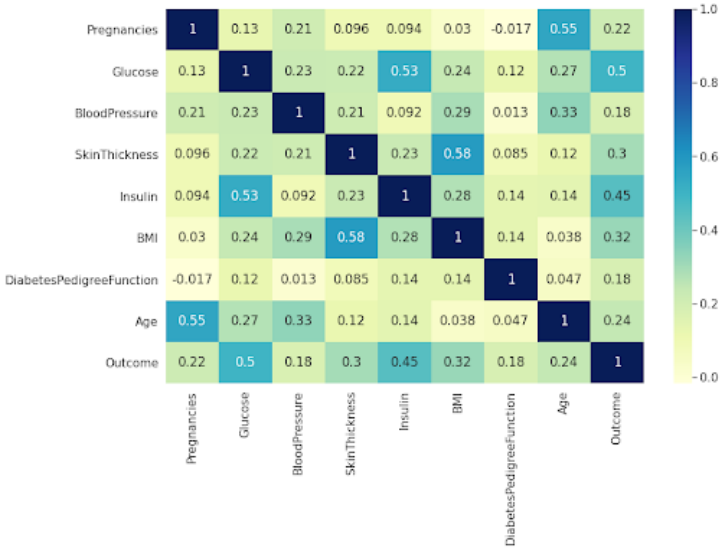

Fig. 4. Correlation Matrix

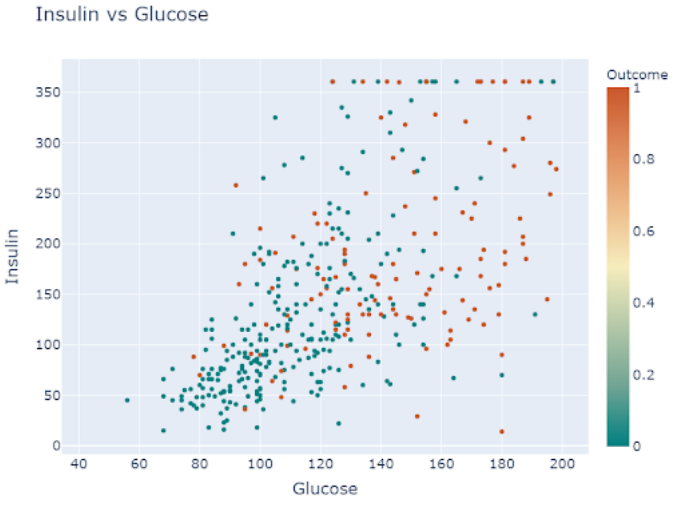

Fig. 5. Scatter Plot of Glucose vs. Insulin

The new feature 'IG' has a correlation of 0.49 with the 'Outcome' variable which is more than the correlation of

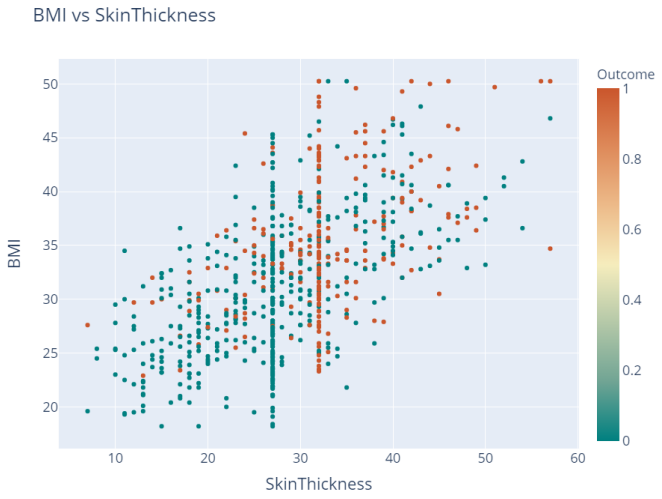

Fig. 6. Scatter Plot of Skin Thickness and BMI

'Insulin' with 'Outcome' (0.45). Similarly, the feature 'BS' has a correlation of 0.33 with the 'Outcome' variable which is more than the correlation of 'Skin Thickness' with 'Outcome' (0.3). Thus the two new features 'IG' and 'BS' are added to the dataset.

Then the dataset is standardized using StandardScalar. Standardization as shown in (6):

$$Z = (x - \bar{x})/\sigma \tag{6}$$

where x is a row of the feature vector and $\bar{x}$ is the mean of the feature vector, and σ is the standard deviation of the feature vector.

Standardization is required when the features have different scales and do not contribute equally to the model fitting thus creating a bias during model training.

c) Model Tuning: The classification models like KNN, SVM and tree-based models like Decision Tree, Random Forest, LGBM, and Adaboost were used for Diabetes Prediction. The advantage of classifying data is that it helps in decision-making and categorizing the data seperately. It helps in segregating our large dataset into distinct categories such as 0/1 or True/False. The classification models that we implemented are briefly explained as follows:

K Nearest Neighbours (KNN):
K Nearest Neighbours [11] is a supervised learning model in which a data point is classified based on the majority class among its K nearest neighbors. The length of the space between two points is used to determine their proximity. The initial value of K is set as the count of neighbors that are chosen. The distance between the test data point and every other data point is calculated and sorted in ascending order by distance. From the sorted list, the first K entries are considered. Then based on the most appearing class of these entries, a class is assigned to the test data point. KNN differs from K-means clustering algorithm as, K-means is an unsupervised algorithm which clusters the similar data points together.

Decision Tree:
Decision Tree is a tree-based supervised machine learning model whose internal nodes are taken as test cases and leaf nodes are taken as categories [13]. This process is recursively repeated for every subtree. The accuracy of the decision tree depends upon the strategy used to split a node. The decision tree splits the nodes by all the features in the dataset and then selects the split which gives the most homogeneous sub-nodes. Various algorithms like Information Gain, Gini Index, Gain Ratio, etc. are used to select an appropriate feature for the split.

Random Forest:
Ensemble models combine multiple models in order to obtain better prediction results. Random Forest is an ensemble model which produces many trees in place of a single tree [14]. It then predicts the final class of the data point by calculating the majority of votes obtained from each decision tree. Since Random Forest combines many decision trees which are prone to noise, it reduces the overall effect of noise thus giving better results [14].

Light Gradient Boosting Machine (LGBM):
Light GBM is a gradient boosting framework that uses a leaf-wise strategy to grow trees and makes a split based on the largest gain of variance [15]. It uses two techniques like Exclusive Feature Bundling (EFB) for downsampling the features and data and Gradient-based One Side Sampling (GOSS) [16].

GOSS is a downsampling method that downsamples data instances based on their gradients [14]. Since data points with small gradients are trained properly and those having large gradients are undertrained, LightGBM randomly samples instances with small gradients and retains instances with large gradients [16]. GOSS first arranges the data in descending order of gradients and selects top a *100% instances and randomly samples b * 100% instances from the remaining data (a, b is the sampling ratio for large and small gradient data resp.) [16]. In order to maintain original data distribution, the contribution of samples with small distribution is amplified by a constant (1-a)/b to concentrate on undertrained instances [16].

LightGBM uses the EFB algorithm to improve the speed of the training phase [15]. The mutually exclusive features are bundled together into a single feature called Exclusive feature Space [15].This reduces the complexity from O(#data * #features) to O(#data * #bundle) [16]

Adaboost:
Adaptive Boost is an ensemble model which uses Boosting technique. This model helps to train weak classifiers, thus constructing a strong classifier. [15]. It gives more attention to the wrongly categorized samples during the training process. [17]. Firstly, the data initialization is done with weight as

TABLE II
BEST PARAMETERS FOR CLASSIFIERS

Classifier	Best Parameters (Grid Search)
Decision Trees	max_depth = 5 min_samples_split = 2
Random Forest	bootstrap = True max_depth = 80 max_features = 3 min_samples_leaf = 3 min_samples_split = 8 n_estimators = 100
KNN	n_neighbors = 28
LGBM	num_leaves = 50 max_bin = 1000 learning_rate = 0.06 class_weight = None boosting_type = gbdt
Adaboost	learning_rate = 0.2 n_estimators = 300

TABLE III
PERFORMANCE OF CLASSIFIERS IN TERMS OF ACCURACY AND AUC

Classifier	Accuracy	AUC
Decision Trees	87.2430 %	0.890982
Random Forest	87.7650 %	0.944585
KNN	83.7238 %	0.900855
LGBM	89.8515 %	0.951947
Adaboost	87.2413 %	0.937591

1/N where N denotes the total number of instances [18]. Then the actual influence for the individual classifier [18] is determined as shown in (7):

$$\alpha_t = \frac{1}{2} \ln \frac{(1 - TotalError)}{TotalError} \tag{7}$$

where the total error is the total of the number of miscalculations for the training set divided by the total number of entries in the training set. Higher weight is given to the incorrectly classified instance so that subsequent classifiers can pay more attention to the misclassified instances.

Parameter Tuning using GridSearchCV:
The Grid Search process is used for the hyperparameter tuning of the above models. Grid Search constructs a new model for each combination of specified hyperparameters and returns a set of parameters that give the highest validation accuracy [19]. A five-fold Cross-Validation Splitting strategy is used in Grid Search. The details of the best parameters obtained from hyperparameter tuning performed using Grid Search are given in Table-II.

IV. RESULTS

In this section, we are presenting the results of our research study. Table-III Performance of classifiers in terms of Accuracy and AUC

The performance results of various classifiers in terms of metrics like Accuracy and AUC (Area Under ROC Curve) are given the Table-3. The percentage of correct predictions for the test data is called the accuracy of the classifier. It is calculated by dividing the total number of correct predictions by the total of the number of predictions. In the Receiver Operating Characteristics curve i.e. ROC curve, the False Positive Rate (FPR) is plotted on the x-axis and True Positive Rate (TPR) is plotted on the y-axis. The more the ROC curve closer to the y-axis, the performance of the model is better. Another metric used for evaluating the performance is the area under the ROC curve (AUC). The metrics scores for all the models were calculated using a cross-validation score for 5 folds. The values of accuracy and AUC reported in Table III are the mean values of all folds.

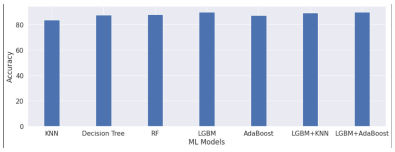

Fig. 7. Comparison of Accuracy for models

We can see from Fig. 8 that the LGBM model outperformed all the other models with the highest accuracy of 89.85%. It means that the LGBM model could correctly classify the 89.85% of the test cases given to is as a diabetic or a healthy person.

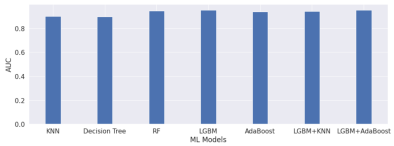

Fig. 8. Comparison of AUC for models

Also, Fig. 9 shows that the AUC value of 0.95 for the LGBM classifier is the highest amongst all the classifiers. This value of AUC suggests that the model is capable of distinguishing between the classes of diabetic and healthy people. The larger value of AUC shows that the FPR is lower which means Type-I error is less.

CONCLUSION

In this paper, we trained various ensemble models, tree-based models, and KNN for diabetes prediction using the Pima dataset. The dataset contained many missing values and outliers which were carefully handled in our proposed framework. The correlation-based feature selection method improved the performance of the model. Hyperparameter tuning was one of the key methods that helped in improving the performance of models. Tuned LGBM outperformed all the models. The validation of the model was done using five-fold cross-validation. In the future, we will train our model on a larger dataset to get more accurate results.

ACKNOWLEDGMENT

We are thankful to the National Institute of Diabetes and Digestive and Kidney Diseases for providing PIMA Indian Dataset for the research study of diabetes prediction.

REFERENCES

[1] Ramya Kannan, "India is home to 77 million diabetics, second highest in the world", November 2019.

[2] Neetu Chandra Sharma, "Government survey found 11.8% prevalence of diabetes in India", October 2019.

[3] Parvin Soleymani, "Prediction of diabetes", Research Gate, May 2020.

[4] Deepti Sisodia, Dilip Singh Sisodia, "Prediction of Diabetes using Classification Algorithms", International Conference on Computational Intelligence and Data Science, 2018

[5] M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," IEEE Access, vol. 8, pp. 76516-76531, 2020, doi: 10.1109/ACCESS.2020.2989857.

[6] Naveen Kishore G., V. Rajesh, A. Vamsi Akki Reddy, K. Sumedh, T. Rajesh Sai Reddy, "Prediction Of Diabetes Using Machine Learning Classification Algorithms", International Journal of Scientific & Technology Research, Jan 2020.

[7] A. H. Syed and T. Khan, "Machine Learning-Based Application for Predicting Risk of Type 2 Diabetes Mellitus (T2DM) in Saudi Arabia: A Retrospective Cross-Sectional Study," IEEE Access, vol. 8, pp. 199539-199561, 2020, doi: 10.1109/ACCESS.2020.3035026.

[8] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng and D. N. Davis, "DMP_MI: An Effective Diabetes Mellitus Classification Algorithm on Imbalanced Data With Missing Values," in IEEE Access, vol. 7, pp. 102232-102238, 2019, doi: 10.1109/ACCESS.2019.2929866.

[9] Vaishali R. Dr. R. Rasikala, S. Ramasubbareddy, S. Remya, Sravani Nalluri, "Genetic algorithm based feature selection and MOE fuzzy classification algorithm on Pima Indians Diabetes dataset", IEEE, 2017.

[10] Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., and Johannes, R.S, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus", Proceedings of the Symposium on Computer Applications and Medical Care, 1998.

[11] K. Taunk, S. De, S. Verma and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1255-1260, doi: 10.1109/ICCS45141.2019.9065747.

[12] S. Ghosh, A. Dasgupta and A. Swetapadma, "A Study on Support Vector Machine based Linear and Non-Linear Pattern Classification," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 24-28, doi: 10.1109/ISS1.2019.8908018.

[13] A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, 2011, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826.

[14] S. V. Patel and V. N. Jokhakar, "A random forest based machine learning approach for mild steel defect diagnosis," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, pp. 1-8, doi: 10.1109/ICCIC.2016.7919549.

[15] M. R. Machado, S. Karray and I. T. de Sousa, "LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry," 2019 14th International Conference on Computer Science & Education (ICCSE), 2019, pp. 1111-1116, doi: 10.1109/ICCSE.2019.8845529.

[16] G. Ke, Q. Meng and T. Finley, "LightGBM: A highly efficient gradient boosting decision tree", Proc. Annu. Conf. Neural Inf. Process. Syst., pp. 3146-3154, 2017.

[17] X. Shu and P. Wang, "An Improved Adaboost Algorithm Based on Uncertain Functions," 2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration, 2015, pp. 136-139, doi: 10.1109/ICIICII.2015.117.

[18] Y. Zhang et al., "Research and Application of AdaBoost Algorithm Based on SVM," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2019, pp. 662-666, doi: 10.1109/ITAIC.2019.8785556.

[19] Z. Jin, W. Chaorong, H. Chengguang and W. Feng, "Parameter optimization algorithm of SVM for fault classification in traction converter," The 26th Chinese Control and Decision Conference (2014 CCDC), 2014, pp. 3786-3791, doi: 10.1109/CCDC.2014.6852839.