International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

# Comparative Investigation and Determination of Partial Discharge source using GNN and XGBOOST Techniques

Dr. Priyanka Kothoke
Assistant Professor, Electrical Engineering
MGMs College of Engineering and Technology,
Panvel, Maharashtra, India

Mrs. Kajol Nitin Chaudhari
Research Scholar, Electrical Engineering
Shri J. J. T. University,
Rajasthan , India

***Abstract*— The outline of partial discharge (PD) is an substantial instrument for high-voltage insulation systems diagnostics. In different PD data representations, human experts can detect possible isolation defects. PD (PRPD) patterns are one of the most commonly used representations. To ensure the confident operation of HV-equipment, the statistical properties of PDs must be linked to the defect properties and the type of defect determined. Classifier Naive Bayes and Xgboost are a family of simple 'probabilistic classifiers,' which use Bayes theorem with robust presumptions of independence among features. At the moment the GNB's importance in assessing its appropriateness for PD actions is being considered. The model is built in Google's Python co-laboratory and can be generated from the PD source on a graphical user interface, whether it is void, surface or crown discharge, using statistically based parameters.**

**Keywords— *Gaussian Naive Bayes; Xgboost; Google co-laboratory; Partial Discharge; Phase-Resolved.***

## INTRODUCTION

Partial Discharge is a confined electrical discharge which bonds part of the insulation among electrodes and can occur head-to-head to a conductor or otherwise [1]. PDs usually concern dielectric materials used and bridge the voltage between the electrodes[2] in some respects. Strong, liquid or gaseous materials or any mixture of them may consist of the insulation. The PD is the main reason for high-voltage electrical equipment electrical aging and insulation. Different PD sources affect the isolation performance differently. In the assessment of the destructiveness of the discharge [2], PD classification is therefore important.

The PD classification is intended to identify unknown source discharges. For many years, the process has been carried out by examining the pattern of the flux on an oscilloscope screen using a well-known ellipse, which was crudely observed by the eye. Currently, extensive research has been published in order to recognize PD sources through smart technology, such as artificial neural networks, fluorescent logic and acoustic emissions [2].

In this part a new, fast-track digital and computer-based techniques and algorithm for the processing and analysis of PD-based measurement signals have driven the recent rise in the research on DP phenomena. With sufficiently advanced digital technology, it seems to be anticipated that no only new insights into the physical and chemical foundation of PD phenomena can be gained, but also that pd's patterns can be used to identify featments of the 'deficiencies' in the insulation of the observable PD [3]. A computer-aided measuring system's having a ability to process a large amount of information and convert it into a comprehensible output is an undeniable advantage [4]. Many kinds of patterns can be used to identify PD sources. Where the statistical parameters of these differences can be used, it can be possible to identify the defect type from the observable PD pattern. Since each defect has its own special breakdown mechanism, the link between the release patterns and the type of defect is important to know [5]. As a result, progress in quality control in insulating systems is becoming increasingly important in recognizing internal discharge and its correlation with this type of defect [6]. The study of fractional discharge sources was carried out using statistical numerical and neural network techniques [7]. During the experiments, there are three distinct classifications of PD pulse data patterns from digital PD detectors: phase data, time-resolved data, and data that lacks information on either phase or time.

1. Three-dimensional discharge epochs, f charge allocation, q discharge rate, n patterns (α~q, ~n and ~n patterns), at a specific test voltage, are the data that are phase-resolved.
2. The time-solved data constitutes a certain time interval, i.e. q~t data pattern, for the individual release pulse magnitudes.
3. The third information classification includes changes in pulse discharge amplitude to V-test voltage amplitude (for increasing and decreasing levels), i.e. q ~V data, respectively.

Out of these three classifications, phase resolved data method is used in this work for statistical method whose pre-processed data is used as an input to this GNB Technique. Phase resolved data is used since it is drawing 2D patterns from raw data showing variations between either phase angle and charge, phase angle and number of pulses or number of pulses and charge and then comparison of these patterns amongst themselves will result in improved accuracy of the outputs.

At present, in any industry, just the source of PD is notified by using a buzzer. However, the buzzer indicates only the place of its occurrence and not the type of discharge, which is very essential for removal of insulation defects. The technique suggested in this research enable to know the type of the PD accurately so that the insulation defects can be removed from any high voltage equipment by specific techniques to avoid chances of supply failure.

PDs should be marked with the phase angle ̈, PD charge magnitude q and PD pulse numbers n being the key parameters. These three parameters consist of PD distribution patterns. For phase-resolved (f-q), (p-n), and statistic parameters are obtained (n-q).

PDs are characterized by phase angle ·, magnitude q of PD load, and by PD n. With respect to the 50 (± 5) Hz sine wave, pulse pulses are grouped by their phase angle. The voltage cycle is therefore divided into phase windows that represent the phase angle axis (0 to 360'). If observations for multiple voltage cycles are made, in each phase window a statistical distribution of individual PD events can be determined. Throughout this phase angle axis, the average values of these statistical distributions lead to two dimensional patterns for the observed PD patterns [8]. A 2-dimension (2-D) α-q and Ś-n distribution represents 'Q' and 'n' pulses as a function of the angle of phase, as the phase angle '5-0' PD charge magnitude [9]. The medium pulse allocation Hqn (ć) is the average magnitude of the PD charge in every window, depending on the angle of phase ±. The distribution of the pulse counts Hn (ć) is the number of PD pulses within the window depending on the angle of phase ±. Both of these quantities are further split into 2 separate negative and positive half cycle distributions, resulting in the emergence of 4 different distributions: positive half of the Hqn+ (5-0) and Hn+ (−) voltage cycles, negative half of the Hqn-(−2) and Hn–(−3) voltage cycles[9]. The normal distribution can be described as PD quantities for a single defect. Hqn's (ć) and Hn's (±) distribution profiles were modeled by the time of normal distribution: skewness and kurtosis. Pulse umber n. These three parameters consist of PD distribution patterns. For phase-resolved (f-q), (p-n), and statistic parameters are obtained (n-q).

The skewness and curtos is of the reference normal distribution are evaluated. Asymmetry or tilt degree of the data is a measure of skewness in terms of normal distribution. Sk=0 is a symmetrical distribution; left is asymmetric, Sk>0; right, Sk<0 is asymmetrical. Kurtosis represents an indicator of distribution sharpness. If the sharpness of the distribution is equal to the normal distribution, Ku=0. Sharing Ku>0, and Flatter Ku<0[10], if more sharp than normal.

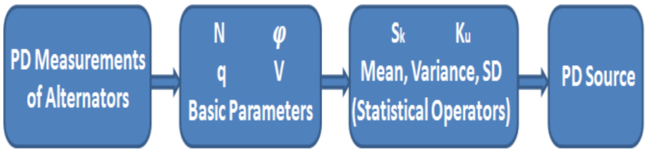


Fig.1. A Systematic Flow of discharge analysis for (n-q)[15-17]

Where,
S.D = standard deviation
Sk = skewness
Ku = kurtosis
For the calculation of several statistical operators, statistical analysis is applied. The following definitions are described for most of such statistical operators. You can use a general function, i.e. yi=f(xi)[25-3], as a profile of all these discrete distribution functions. It is possible to calculate the statistical function:

The skewness and curtosis of the reference normal distribution are evaluated. Asymmetry or tilt degree of the data is a measure of skewness in terms of normal distribution. Sk=0 is a symmetrical distribution; left is asymmetric, Sk>0; right, Sk<0 is asymmetrical. Kurtosis represents an indicator of distribution sharpness. If the sharpness of the distribution is equal to the normal distribution, Ku=0. Ku>0 is sharper than normal and Ku<0 [11] [20] is flatter.

In our study, internal and external discharges of various kinds such as the void, the surface and the crown were tested using statistical parameters such as average, standard deviation, variance, skeshedness and kurtosis for (n-q), as explained above in fig. 1. After entering the five data values of statistical parameters as an input in Google's collaboratory in Python, GUI displays the type of discharge as input.

## I. Gaussian Naïve Bayes and XGBoost Classifiers

### A. *Gaussian Naïve Bayes Classifier*

The classifying group based on Bayes Theorem is called Naive Bayes Classifiers, also referred to as simple Bayes and Bayes for independence. The common classification principle shares all classifiers in this category. The reason it was referred to as Naive Bayes is its assumption that there is no correlation between all attributes of a dataset and each attribute is independent. Classifiers from Naive Bayes could be easily scaled. The number of functions in a classification problem requires linear parameters. Naive Bayes training could be carried out with maximum similarity. Training in this kind of classifiers is fast because in classifiers in Naïve Bayes iterative approximation is not done. Naive Bayes is an easy classification technique. It models a classifier that gives instances of the test datasets a class label. The class label is drawn from the training set by each instance to be a vector of the feature values. There is no unique algorithm, but a common principal family of algorithms, to train such classification systems. All classifications in Naive Bayes assume that the value of a specific characteristic is independent of the value of any other feature's class variable.
For some types of probability models, the Naive Bayes classifier can
be very effectively trained in a supervised learning environment.

Parameter approximation for models of Naive Bayes practices the highest probability technique in many practical. Naive Bayes classification systems have worked quite well in a number of compound situations despite their naïve project and simple assumptions. The analysis of the Bayesian problem of classification showed that behind the apparently unbelievable

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 03, March-2023

efficiency of classifier types there are sound theoretical reasons. In 2006 an extensively comparing classification algorithms showed that the classification of bays was outperformed by other forests, such as boosted trees and random forests. The advantage and application of Naive Bayes is to approximation the parameters necessary for classification only by using a small number of training data. Naive Bayes works with a discrete value, which is the fundamental property. It is recommended to use the Gaussian Naive Bayes classifier [12] if attribute values are continuous.

Naive Bayes is a group of supervised techniques used for classification machines. Bayes Theorem is the core of this method of classification. For each class in the dataset, it predicts membership probabilities such as the probability that a data point is in a specific class. The most likely class of data points is the class with the highest probability of membership. On the basis of a training data set [13].

### B. *Xgboost Classifier*

As portion of the Distributed (Deep) Machine Learning Community (DMLC), Tianqi Chen began working on XGBoost as a research project. An early version of the programme may be modified using a configuration file for the Linux virtual machine (libsvm). The Higgs Machine Learning Challenge's winning answer made it a household name in the field of machine learning competitions. In addition to Python and R packages, XGBoost now supports Java, Scala, Julia, Perl and more languages through package implementations. Because of this, the library gained popularity on Kaggle, where it has been used in numerous competitions, and was made available to additional developers explained by Shikha Agarwal (2019) et al and Dana Bani-Hani (2019) et al.

Several other packages were quickly added to make it easier for people to utilise in their own groups. The caret package for R users and scikit-learn for Python users have both been integrated. There are a number of frameworks for Data Flow that integrate with the abstracted Rabit and XGBoost4J. For FPGAs, XGBoost is also supported by OpenCL. Tianqi Chen and Carlos Guestrin have released a paper describing an implementation of XGBoost that is both efficient and scalable.

"Extreme Gradient Boosting" is the name given to this machine learning package, which stands for "Extreme Gradient-Boosted Decision Trees." For regression, classification, and ranking issues, it is the go-to machine learning package because of its parallel tree boosting capabilities.

Boosting Algorithm for Extreme Gradients When it comes to solving classification and regression problems with ensemble machine learning techniques, gradient boosting is a good choice. Decision trees are used to build ensembles. One of the most popular machine learning frameworks for Python data scientists is Scikit-learn, and XGBoost provides frameworks for both languages. Classification and regression problems may be solved with it, making it a good fit for the great majority of data science tasks.

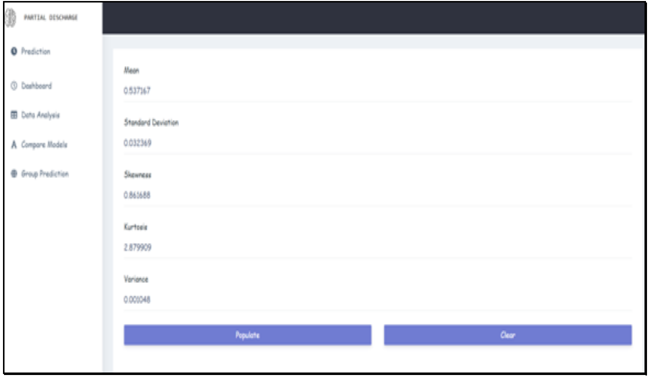
M. Junshui (2003) explained about how the gradient boosted trees algorithm is well-implemented in XGBoost, a popular open-source implementation. With the help of a number of smaller, less accurate models, a method known as gradient boosting seeks to improve the accuracy of a target variable's predicted value.

## II. METHODOLOGY

1. Google colab is used for this study as open-source software is cheaper and easier to reach and can be adapted for future purposes. Five required parameters viz. mean, standard deviation, variance, skewness and kurtosis for both known (void, surface and corona) and unknown discharges (data1,data2 and data3) are given as an input for GNB and Xgboost methods.
2. Initially, in Google Co- laboratory, common model is built. For model building, the obtained statistical required parameters are used for both training and testing.
3. Secondly and important the final processed file (merging all six known and unknown data) is created.
4. Thirdly, this built model of google colab is taken in python server for creating Graphical User Interface (GUI). GUI will display the type of discharge by entering the statistical parameters viz mean, standard deviation, variance, skewness and kurtosis[18][19] .

## III. EXPERIMENTAL RESULTS

The output (type of discharge) for the above methods will be shown by entering the input values (statistical parameters) as shown in figures below. Fig 2. below is showing the screenshot of window which will be used for entering input values and Fig 3. is used to select the method and Fig 4. shows the output.



Mean
0.537167

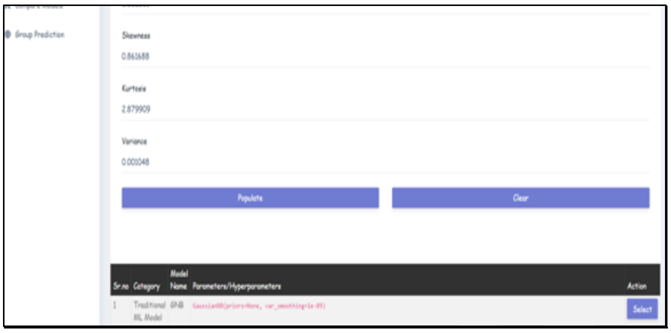
Standard Deviation
0.032369

Skewness
0.861688

Kurtosis
2.879909

Variance
0.001048

Fig.2. Figure showing the entered input values on GUI



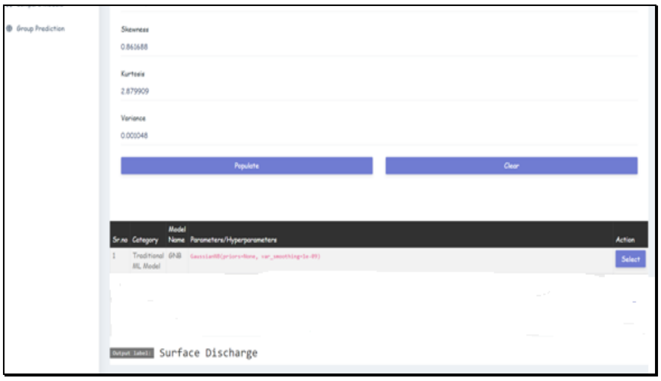
Skewness
0.861688

Kurtosis
2.879909

Variance
0.001048

| Sr.no | Category | Model Name | Parameters/Hyperparameters | Action |
|---|---|---|---|---|
| 1 | Traditional ML Model | GNB | GaussianNB(priors=None, var_smoothing=1e-09) | Select |

Fig.3. Figure showing the icon to select the Technique on GUI



Skewness

0.861688

Kurtosis

2.879909

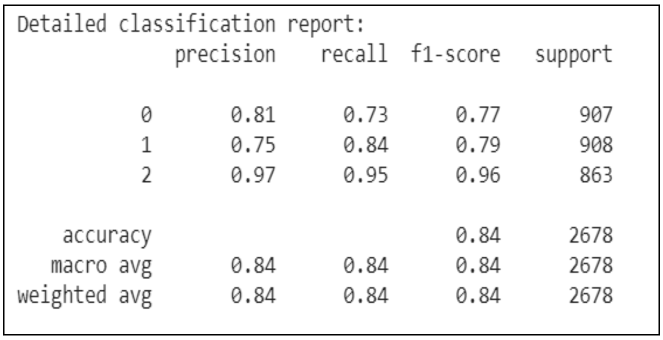
Variance

0.001048

| Sr.no | Category | Model Name | Parameters/Hyperparameters | Action |
|---|---|---|---|---|
| 1 | Traditional ML Model | GNB | GaussianNB(priors=None, var_smoothing=1e-09) | Select |

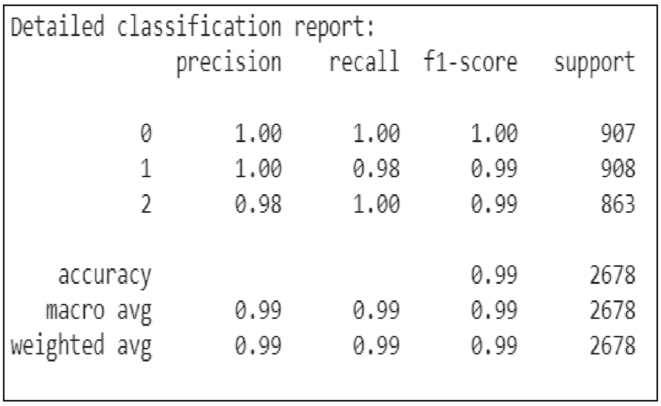
Output label: Surface Discharge

Fig.4. Figure showing the output as surface discharge on GUI



```
Detailed classification report:
              precision    recall  f1-score   support

           0       0.81      0.73      0.77       907
           1       0.75      0.84      0.79       908
           2       0.97      0.95      0.96       863

    accuracy                           0.84      2678
   macro avg       0.84      0.84      0.84      2678
weighted avg       0.84      0.84      0.84      2678
```

Fig.5. Figure showing the detailed classification report i.e. precision, recall, f1-score and support of GNB Technique



```
Detailed classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       907
           1       1.00      0.98      0.99       908
           2       0.98      1.00      0.99       863

    accuracy                           0.99      2678
   macro avg       0.99      0.99      0.99      2678
weighted avg       0.99      0.99      0.99      2678
```

Fig.6. Figure showing the detailed classification report i.e. precision, recall, f1-score and support of Xgboost Technique

In Fig.5 and Fig.6, "0" stands for known discharge viz. corona discharge, "1" stands for surface discharge and "2" stands for void discharge.
Comparative analysis of all the parameters is shown in Fig.7 below.

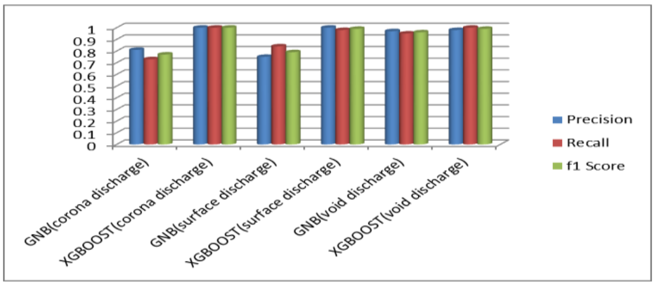


Fig.7. Comparative plot of Precision, Recall and f1-score for all three known discharges from GNB Method and Xgboost method

Table I shows the hyper-parameters used in both the Methods. Any algorithm that has hyper-parameters contributes significantly to the model output, therefore it is best to determine the optimal (or near-optimal) hyper-parameter combination. Hyperparameters are defined as the features of a model that can be defined by the user. It differs from parameters as during the workout, the parameters are changed internally, not before the workout, by the user.

Table - I. Table showing hyper-parameters used in both the Methods

| Sr.no | Hyperparameters | Description | Value |
|---|---|---|---|
| 1 | Priors | “Priorities of the classes If this option is selected, the priors are not adjusted based on the data” | NONE |
| 2 | Var (“Smoothing”) | “A portion of the largest variance of all the features is added to variances to calculate stability” | 1.00e-09 |

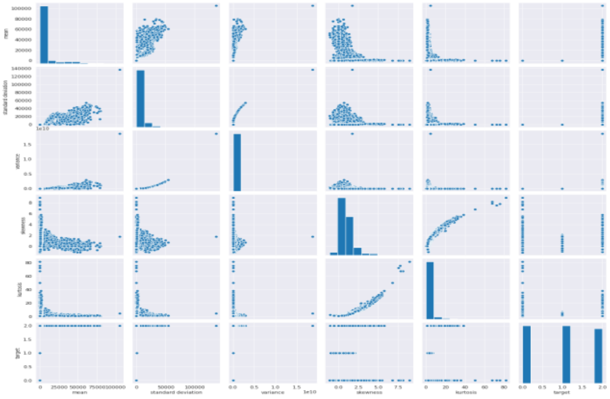
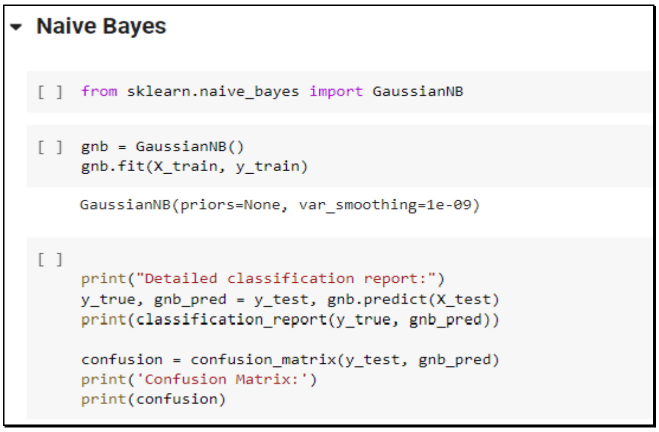


Fig.8. Attributes disseminations and statistical breakdown of the used dataset

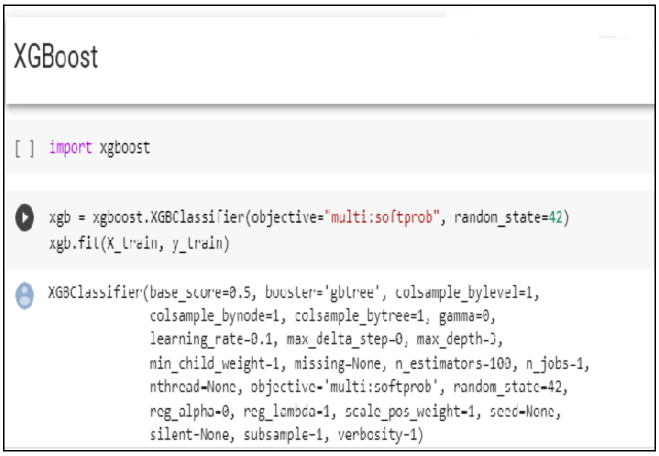
Fig.8. shows the Attributes disseminations and statistical break down of the used dataset used as we can see from the last column and last row figure that how we can reach to target.



Naive Bayes

```
[ ] from sklearn.naive_bayes import GaussianNB

[ ] gnb = GaussianNB()
    gnb.fit(X_train, y_train)

    GaussianNB(priors=None, var_smoothing=1e-09)

[ ]
    print("Detailed classification report:")
    y_true, gnb_pred = y_test, gnb.predict(X_test)
    print(classification_report(y_true, gnb_pred))

    confusion = confusion_matrix(y_test, gnb_pred)
    print('Confusion Matrix:')
    print(confusion)
```

Fig.9. Steps for execution of GNB method on google co-laboratory window



XGBoost

```
[ ] import xgboost
```

```
xgb = xgboost.XGBClassifier(objective="multi:softprob", random_state=42)
xgb.fit(X_train, y_train)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate-0.1, max_delta_step-0, max_depth-3,
              min_child_weight-1, missing-None, n_estimators-100, n_jobs-1,
              nthread-None, objective-'multi:softprob', random_state-42,
              reg_alpha-0, reg_lambda-1, scale_pos_weight-1, seed-None,
              silent-None, subsample-1, verbosity-1)
```

Fig.10. Steps for execution of Xgboost method on google co-laboratory window

Above figure shows the programming steps for execution of both the methods, after uploading the pre-processed data obtained from statistical method, click on these 3 steps so that the program will get executed and output can be seen on GUI (as shown in Fig.2, Fig 3 and Fig.4)

## IV. DISCUSSION AND CONCLUSION

It was seen those results of unknown data, data2 were not clearly confirming the source of discharge as void or surface using Statistical method in MATLAB software. But now, it can be finally concluded using GNB method using Python software that the data2 discharge is definitely a surface discharge. The accuracy from GNB method is 87.26% and from XGBOOST is 99 %.

Created a common model and confirmed the results accurately for type of PD.

Use of google co-laboratory gives accurate results. The advantage of google co-laboratory in Python is to reduce the error to zero faster.

TABLE - II. RESULTS

| Unidentified data | Partial discharge Source |
|---|---|
| Unknown Discharge 1 | Surface Discharge |
| Unknown Discharge 2 | Surface Discharge |
| Unknown Discharge 3 | Void Discharge |

Advantage of using GNB is to increase the speed of PD type recognition and avoid a lot of power supply failures in industry.

If we want to avoid power supply problems in industry, XGBOOST is an excellent tool to use. Finally, we may say that — It is recommended that the Xgboost approach be used for the identification of partial discharge type.

## REFERENCES

[1] “Statistical Pattern Analysis of Partial Discharge Measurements for Quality in High Assessment of Insulation Systems Voltage Electrical Machinery Birsen Yazici Symporium on Diagnostin far Electric Machines”, Power Electronics and Drives Atlanta, Georgia, GA USA 24-26, August 2003.

[2] Akimasa Hirata, Syou Nakata, and Zen-Ichiro Kawasaki, “Toward Automatic Classification of Partial Discharge Sources With Neural Networks”, IEEE transactions on power delivery, vol. 21, no. 1, january 2006.

[3] J. R Hyde and I. J. Kemp, “Partial Discharge Pattern Recognition and Associated Technologies”, IEEE Colloquium on Monitoring Technologies for Plant Insulation,1994.

[4] S.H.Park, K. W.Leel, K.J.Lim. andS.H.Kang, “Classification of External and Internal PD Signals Generated in Molded Transformer by Neural Networks”, Proc. of the 7th Int. Conf. on Properties and Applications of Dielectric Materials June 1-5 Nagoya P2-41, 2003.

[5] Shan Ping', XuDake', Wang Guolil, Li Yanming, “Application of Neural Network with Genetic Algorithm to UHF PD Pattern Recognition in Transformers”, Annual Report Conference on Electrical Insulation and Dielectric Phenomena, 2002.

[6] E. Gulski and F. H. Kreuger, “Computer-aided recognition of Discharge Sources”, IEEE Trans. on Electrical Insulation Vol. 27 No. 1, February, 2001.

[7] S. Ghosh and N.K. Kishore, “Modelling Of Partial Discharge Inception And Extinction Voltages Of Sheet Samples Of Solid Insulating Materials Using an Artificial Network”, Proc. Of IEE Volume: 149 , Issue: 2 2002.

[8] Gagan Deep Meena, Dr. Girish Kumar Choudhary, Manoj Gupta, "Neural Network Based Recognition Of Partial Discharge Patterns", International Journal of Advanced Engineering Research and Studies. Vol. I/ Issue II/January-March, 2012/121-126.

[9] V. M. Catterson, B. Sheng, "Deep Neural Networks for Understanding and Diagnosing Partial Discharge Data", Institute for Energy and Environment, University of Strathclyde, Glasgow, United Kingdom.

[10] M. M. A. Salama, and R. Bartnikas," Determination of Neural-Network Topology for Partial Discharge Pulse Pattern Recognition", IEEE Transactions On Neural Networks, Vol. 13, No. 2, March 2002.

[11] Radio Location of Partial Discharge Sources: A Support Vector Regression Approach K. P. Bennett, and C. Campbell, “Support vector machines: hype or hallelujah,” ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 1-13, 2000.

[12] M. Junshui, T. James and P. Simon, “Accurate On-line Support Vector Regression", Neural Computation, vol. 15, no. 11, pp. 2683--2703, 2003.

[13] Shikha Agarwal, Balmukumd Jha, Tisu Kumar, Manish Kumar, Prabhat Ranjan, " Hybrid of Naive Bayes and Gaussian Naive Bayes for Classification: A Map Reduce Approach", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-6S3, April 2019.

[14] Dana Bani-Hani, Pruthak Patel & Tasneem Alshaikh, " An Optimized Recursive General Regression Neural Network Oracle for the Prediction and Diagnosis of Diabetes", Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence Volume 19 Issue 2 Version 1.0 Year 2019.

[15] Namrata Bhosale, Priyanka Kothoke Amol Deshpande, Dr. Alice Cheeran, “Analysis of Partial Discharge using Phase-Resolved(φ-q) and (φ-n) Statistical Techniques”, International Journal of Engineering Research and Technology, Vol. 2 (05), 2013,ISSN2278-0181.

[16] Priyanka Kothoke, Namrata Bhosale, Amol Deshpande, Dr. Alice Cheeran, “Analysis of Partial Discharge using Phase-Resolved (n-q) Statistical Techniques”, International Journal of Engineering Research and Applications.

[17] Yogesh R. Chaudhari, Namrata R. Bhosale, Priyanka M. Kothoke, “Composite Analysis of Phase Resolved PD Patterns using Statistical Techniques”, International Journal of Modern Engineering Research (IJMER) Vol. 3, Issue. 4, pp-1947-1957, 2013.

[18] . Priyanka Kothoke, Dr. Anupama Deshpande, Yogesh Chaudhari, “Investigation and Determination of PD source utilizing SVR and RF”, International Conference on Inventive Computation Technologies (ICICT-2020), inclusion into IEEE Xplore and Scopus.

[19] Priyanka Kothoke, Dr. Anupama Deshpande, Yogesh Chaudhari, “Examination and Determination of Partial Discharge Source utilizing SOM and BPM Techniques of ANN”, International Journal of Advanced Science and Technology, Vol. 29, No. 8s, (2020), pp. 3299-3306.

[20] Priyanka Kothoke, Dr. Anupama Deshpande, Yogesh Chaudhari, “Analysis and Determination of Partial Discharge Type using Statistical Techniques and Back Propagation Method of Artificial Neural Network for Phase Resolved Data;”, International Journal of Engineering Research and Technology(IJERT),Vol. 08, Issue 08, August 2019.