# XGBOOST BASED GENDER RECOGNITION SYSTEM USING VOICE DATA

[***Muhammad***](https://edas.info/showPerson.php?p=914927&c=27789) ***Yasin1\**, *Ayesha Ali2*, *Bilal Khan3, Muhammad Arshad4***

1,2,3,4 Department of Computer Science, City University of Science and Information Technology, Peshawar 25000, Pakistan

Email: [yasin\_icl@yahoo.com1](mailto:yasin_icl@yahoo.com1), ashu.hasho@gmail.com2, bilalsoft63@gmail.com3, m.arshad@cusit.edu.pk[4](mailto:zainab.abubakar@mediu.edu.my4)

# ABSTRACT

*Gender recognition is the extraction of gender information from the speaker’s voice. Voice is formed by a biological mechanism using various body parts. It is facile for the human brain to recognize gender differences by listening to a person’s voice, but it’s not the same case for computers. Gender Recognition can be used in numerous real-time applications to improve* *human-computer interaction. From aerospace to interactive voice response systems and robotics, from mobile and telematics to health care, gender recognition has greatly helped in the enhancement of human-computer interaction. For gender recognition, different approaches are proposed by researchers for many years. But still, there is a need to improve accuracy in voice-based gender recognition. However, this study proposes a LightGBM-based Model for gender recognition on the voice dataset taken from Kaggle. Going through overall analysis, it has been observed that the projected model performs better analysis in terms of achieving the accuracy of 98.5 %.*

***Keywords: Extreme Gradient Boosting, Human-Computer Interaction, Gender Recognition, Telematics.***

# 1.0 INTRODUCTION

Humans' everyday decision-making is hugely affected by their emotions. In addition, emotional states can have an unintentional impact on human communication, attention, and the ability to comprehend information [1]. While it is common for humans to recognize and interpret emotional states but it is not always the case, these tasks pose significant challenges for computers [2].

Emotion recognition plays a vital role in the area of artificial intelligence and human-computer interaction.

Different techniques are used to identify the emotions of a human such as body movements, facial expressions, heartbeat, blood pressure, and textual information [3]. From a practical standpoint, the detection of human emotions from text has become highly important in computational linguistics. There is an immense amount of textual data on the internet nowadays [4]. Nowadays, people are increasingly depending on computers to do daily tasks, necessitating the improvement of human-computer interactions. The lack of commonsense knowledge makes emotion difficult for a computer to recognize and generate. Therefore, there has been a lot of research done on emotion recognition. Emotion recognition can be broken down into three main categories: emotion recognition from speech, emotion recognition from facial expressions, and emotion recognition from text [5].

Emotion recognition from text plays a significant role in human-computer interactions in the form of forums, emails, product reviews, text messages, blogs, and social media platforms, including Facebook, YouTube, Reddit, Twitter, and Instagram. Emotion recognition from text has applications in education, business, psychology, and a variety of other sectors where understanding and interpreting emotions are necessary[6].

# RELATED WORK

Different approaches have been used to recognize emotions from text, which includes the following related work:

Kudakwashe Zvarevashe et al. [7] presented a gender recognition technique using feature selection using the Random Forest (RF) recursive feature elimination. The classification was performed on gradient boosting machines (GBMs) algorithm. The dataset used was taken from the Kaggle repositories comprising of 1584 males and 1584 females’ voices. Without feature selection, the GBM algorithm obtained an accuracy of 97.58% while achieving almost 100% accuracy after applying feature selection.

Fatih Ertam et al. [8], proposed a three-stage model. It used a Long Short-Term Memory (LSTM) model for the prediction of gender. It utilized 10 different parameters based on weights that were calculated using the relief method. Dataset used was obtained from Kaggle repositories which included 21 features. Evaluation metrics used were based on accuracy, sensitivity, specificity, and Gmean, and concluded that the proposed technique performed better with an accuracy of 98.4%.

Mucahit Buyukyilmaz and Ali Osman Cibikdike et al. [9], used a multilayer perceptron (MLP) model. They used the acoustic properties present in the voice to recognize the gender. The dataset used was obtained from Kaggle repositories containing 21 features. The evaluation metric used was accuracy and concluded that the proposed algorithm had an accuracy of 96.74%.

A comparative analysis performed by Amaz Uddin et al. [10] on K-Nearest Neighbors (KNN), and Support vector machine (SVM) on three different datasets i.e. The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT) dataset, BGC (Self-created), and RAVDESS consisting of 6300, 1440, and 300 audio files respectively. The evaluation metric used was accuracy. Post assessment it was concluded that KNN had a higher accuracy on each dataset.

Alexandre A. Cheuiche Martins et al. [11]proposed a gender recognition algorithm, Artificial Neural Network (ANN) based on Eigen-filtering trained via Generalized Hebbian Learning. TIMIT speech dataset was utilized for training and testing purposes, and out of 630 speakers only 100 speakers were used, 50 men and 50 women. The evaluation metric used was accuracy and it was concluded that ANN-based on maximum Eigenfilter system performed better with an accuracy of 100% for both males and females.

A comparative analysis was performed by Arijit Ghosal et al. [12] on ANN and Random Sample and Consensus (RANSAC). The dataset used was self-made comprising of 200 male voice and 200 female voice files. The evaluation metric used was accuracy and it was concluded that RANSAC performed better compared to ANN with an accuracy of 92.5%.

Pahwa et al. [13] proposed a gender recognition model to identify gender using voice samples taken from 46 speakers. They performed feature selection by extracting the most dominant and researched speech features, including the first and second-order derivatives, and Mel coefficients. The proposed model consisted of Neural Network Classiﬁer and Support Vector Machine where the results of both algorithms were stacked together to make a combined prediction made by Naïve Bayes (NB). The classiﬁcation accuracy obtained was 93.48%.

In 2018 L. N. Pondhu et al. [14]performed comparative analysis using techniques LR, KNN, NB, DT, RF, SVM, and ANN on Speech dataset which comprises 3168 voice samples of males and females, and used Accuracy for result evaluation. It was concluded that SVM and ANN were performing better. By applying Parameter tuning they achieved 98.6% and 99.87% for SVM and ANN respectively.

In 2020 S. S. I. Badhon et al [15] proposed a method to automate voice using techniques LR, RF, Gradient Boosting (GB) on Speeches collected from different sources. For evaluation F-measure, accuracy, precision, recall, and error rate were used. The authors achieved an accuracy of 91.62, 98.25, and 99.13 for LR, RF, and GB respectively.

In 2018 S. Chaudhary et al. [16] proposed a gender recognition system based on voice signal characteristics. The characteristics include Mel frequency cepstral coefficients (MFCC), Energy, and Pitch. The technique used was SVM. The dataset used was TIMIT from which some important characteristics were extracted. The evaluation matrix used was Accuracy. Post assessment accuracy of 96.25% was achieved

Ensuring accuracy and improvement in speed are the two basic things required to predict the gender of a person more quickly and effectively. On the assumption of ensuring accuracy, choosing a speedy iteration update model helps in making the predictions quicker and more effective. When the relevant data attributes are paired with the proper model, they can be utilized effectively. Therefore, selecting the best machine learning strategy is important. In this paper different machine learning strategies are compared with the proposed LightGBM model from a theoretical and engineering perspective. Fig 1 depicts a comparison of traditional, bagging, and boosting strategies. From the traditional strategy we have taken SVM, from the Bagging strategy we have selected Random Forest, and from Boosting strategy we have selected XGBoost.

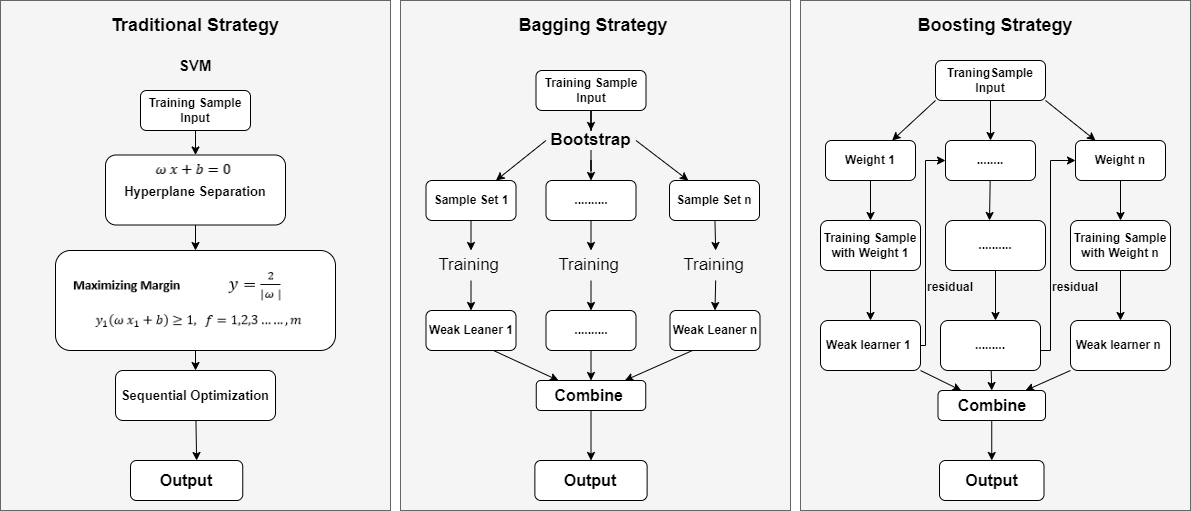


Figure 1: Flow chart of traditional, bagging, and boosting strategy

# Machine Learning Models

# 3.1.1 Support Vector Machine

An SVM is a controlled machine learning technique developed in the mid-1990s that solves two categorization problems using classification strategies. The SVM is a unique small-sample-based data acquisition approach that outperforms previous techniques in many ways since it is based on the notion of structural risk reduction rather than the standard empirical risk mitigation approach. It was generally used for class classification issues, but with the rapid advancement of computer technology, data network technology, and data processing technology for the identification and monitoring of massive amounts of data, the class classification technique no longer fulfills individual’s requirements. There are several hyperplanes from which to choose to split the two kinds of data points. The hyperplane (which in two dimensions) is essentially a line that optimally differentiates the identifiers is produced by a support vector machine to utilize data. There are two types of SVM.

**Linear SVM:** Linear support vector machine (SVM) is an algorithm that is used for the data which can be linearly separable, which implies that if a dataset can be categorized in two different categories using only a single horizontal line, it can be then referred as linearly separable data, and the classifier used is known as linear SVM.

**Non-Linear SVM:** Non-linear support vector machine (SVM) algorithm is used to classify the data that is non-linearly separated, the non-linear dataset consists of the data that cannot be categorized as a straight line, and the classifier used for the classification purpose is called as non-linear support vector machine (SVM) [17]

# 3.1.2 Random Forest

A random forest is a classification system made up of several clustered tree classifiers such as:

where (k), represents the individual symmetrically distributed random vectors. “As the name indicates, a random forest (RF) is made up of a huge number of discrete DT that work together as an array as shown in Figure 3.9. Each decision tree in the RF algorithm produces a class prediction and points are associated with each class. The class with maximum points becomes the prediction of the class”. Developing an array of trees and allowing them to select the most preferred class results in a significant increase in classifying reliability. A random vector that regulates the development of each tree in the array is frequently constructed to develop these populations. The RF is a bagging technique modification that uses both bagging and feature randomization to generate an unassociated decision tree forest. Feature randomization is also called feature bagging or “the random subspace method” that produces a unique set of attributes, resulting in little association between decision trees. This depicts a distinction between RF and DT. Moreover, DT analyzes all potential data split whereas, RF picks just a set of data splits.

There is a distinction between RF utilization for classification and regression purposes. When RF is used for the classification purpose each tree generates and sends a class of votes towards it and then it is classified based on the vote’s majority. Whereas when RF is used for regression purposes each tree generates its prediction and then it is averaged at a common point x. Similarly, the input value (m) changes for both classification and regression. The value of m for classification becomes m = √p whereas, the value of m for regression is p/3.

# 3.1.3 XGBoost

XGBoost is one of the family of boosting algorithms that help poor learners to improve their performance. A poor learner is only marginally better than guessing at random. Boosting is the sequential method in which trees are produced one after the other utilizing information from the previously developed tree. XG Boost is utilized to solve both classification and regression.

Classification Problems: To overcome such challenges, it employs the booster = gbtree parameter, which entails growing the tree one by one to lower the rate of misclassification in the subsequent iterations. The next tree is constructed by providing a larger weight to points that were misclassified by the prior tree [18].

In conventional Gradient Boosting Decision Tree (GBDT) algorithms, the information used is only the first-order derivative. Furthermore, due to dependency between weak learners, training the data in parallel becomes difficult for GBDT. But what XGBoost does is that it takes the Taylor expansion of loss function till second order to obtain optimum results, which helps balance the complexity of the model and decline of the objective function, thus helping avoid overfitting [19].

The XGBoost model is:

Where K is the number of decision tree, is input function for the k-th decision tree, F is a group of all conceivable CART, and is the forecasted value. XGBoost’s objective function comprises of two parts i.e., regularization and training error, that is

Where is the function to calculate the variance between the loss function’s actual value and predicted value. is the regularization term and .

is leaf penalty coefficient, T is the total amount of leave nodes, w is the score achieved by leaf node, and makes certain that the score achieved by leaf node by no means is exceedingly immense.

The XGBoost algorithm works on gradient boosting decision tree, it adds trees one by one instead of adding trees all at once and using the residuals of the last prediction to constantly repair the prior test results.

…………

Combining the equation (2) and (3), the objective function for the t-th decision tree becomes

To find the objective function, Taylor's expansion of the loss function is taken to the second order. The equation becomes:

Now adding loss function value of each data, the steps are:

Where , and are first and second derivative functions respectively.

The objective function and optimal values acquired are as follow

Where, ,

Throughout the training process, node loss is continuously calculated to choose the leaf node which has the highest information gain loss. New trees are constructed by XGBoost by constantly dividing features. To use the residual of the previous prediction, a new tree is added each time using the function.

K trees are obtained after training, each tree will consist of a leaf node containing the features of prediction samples, and every leaf node coincides with a score. Ultimately, the score of individual trees is put together to acquire the results in form of predictions value [20].

# ` EXPERIMENTAL SETUP

All training, test, and prediction codes have been written by using Python libraries. After the collection of the dataset, data is loaded from CSV file into Python array. Each row contains 20 parameters and 1 label. Then K fold cross-validation has been applied where training and testing of data is done 18 times and, on each loop, different data is selected for testing. Also, the last column i.e., the label has been converted to integer 0 for males and 1 for females. Then classification techniques are applied. The classification techniques including Support Vector Machine (SVM), Random Forest (RF), Light Gradient Boosting Machine(LightGBM), and Extreme Gradient Boosting (XGBoost) were trained using the above-mentioned K fold-cross validation on the dataset to predict the result. Post prediction analysis of comparison is performed among all mentioned techniques to check which technique has higher accuracy. The overall methodology for GR is shown in Figure 2

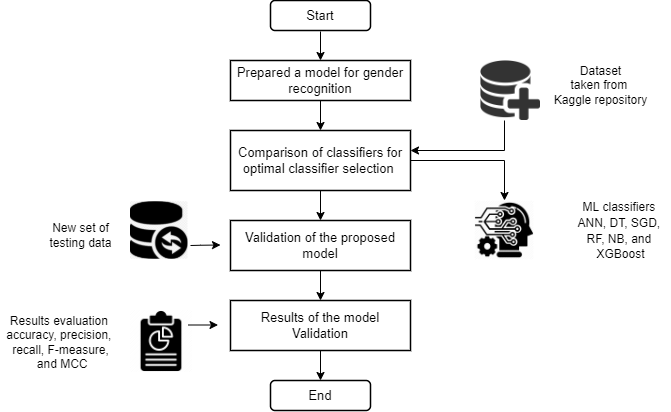


Figure 2: Methodology Workflow

# Dataset

The dataset used for training and testing is taken from Kaggle Repository which comprises of total 3168 instances and 21 attributes, out of which 1584 are male voice parameters and 1584 female voice parameters. Out of 21 attributes, 20 attributes are for each feature and the last column is for label i.e., male or female. Some of the important features in the dataset are meanfun which is the average of frequency of voice signal , Q25 which is the frequency point where the signal gets divided into two intervals of 25% and 75% energy, IQR which is the time range between Q75 and Q25 and is measured in secs, dfrange which is the frequency range that is most dominant in the voice signal, sd which is the standard deviation of frequency, sfm which is the measure of characterizing an audio spectrum, and skew which is a measure of distortion in voice. The list of attributes and ranges are listed in Table 1

Table 1: List of Attributes and Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SNo | Variable | Description | SNo | Variable | Description |
| 1 | Meanfreq | Mean frequency | 11 | Mode | Mode frequency of the voice signal |
| 2 | Sd | Standard deviation of frequency | 12 | Centroid | Frequency centroid |
| 3 | Median | Median frequency | 13 | Meanfun | Average of the base frequency |
| 4 | Q25 | First quantile | 14 | Minfun | Minimum of base frequency signal |
| 5 | Q75 | Third quantile | 15 | Maxfun | Maximum of fundamental frequency |
| 6 | IQR | Interquantile range | 16 | Meandom | Mean of dominant frequency |
| 7 | Skew | Skewness | 17 | Mindom | Minimum of dominant frequency |
| 8 | Kurt | Kurtosis | 18 | maxdom | Maximum of dominant frequency |
| 9 | Sp.ent | Spectral entropy of the voice signal | 19 | dfrange | Dominant frequency range |
| 10 | Sfm | Spectral flatness | 20 | modindx | Modulation index of voice signal |

# Training and Testing

For training and testing purposes K-fold cross-validation has been utilized. K-fold cross-validation is a technique used to evaluate the performance of a classifier. K-fold divides data into several equally sized folds. One-fold (k) is used for validation while others (k-1) are used for training purposes. This process is repeated k-times and each time it selects a different fold for testing and training purposes until each portion is utilized for validation. A value of K=10 is generally used for several folds [21].

However, we started by setting the value of K=10 and incremented it until the value reached K=20. We found that by setting the value of K=18, we achieved the best results in terms of higher accuracy, precision, recall, and MCC.

# Performance Assessment

Model evaluation is the core task of any research work. It is important to evaluate with some standard evaluation measures/models. For evaluation of algorithms assessment metrics namely accuracy, precision, recall, F-measure, and MCC were used.

* + 1. **Accuracy**

Accuracy refers to how close measurements are to a certain value. It is the ratio of the number of correct predictions to the total number of input samples.

* + 1. **Precision**

Precision is the closeness of the measurements to each other.  The ratio of True Positives to all Positives is known as precision. For our problem statement, that would be the measure of male voices that were correctly identified out of all the male voices.

* + 1. **Recall**

The recall is the percentage of correct findings divided by all other class findings.

* + 1. **F-Measure**

It is the average of Precision and Recall. It is calculated by taking false negatives and false positives into account.

# Mathews Correlation Coefficient

It is also known as the phi coefficient. It is a matrix that is used to assess the quality of binary classifications. A confusion matrix's four values are used to compute MCC.

# 5.0 RESULT AND DISCUSSION

After applying the machine learning models, from Table 2 and Figure 4 it can be seen that higher accuracy and better results can be obtained than traditional machine learning models like SVM. However, the Bagging strategy has a certain difference in results as compared to Boosting Strategy. For this reason, we choose a machine learning model based on Boosting strategy for further verification.

Table 2 Performance Analysis of each Technique using Precision, Recall and F-Measure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Technique | Precision | Recall | F-Measure | MCC | Accuracy |
| SVM | 0.69 | 0.74 | 0.70 | 0.74 | 68.72% |
| RF | 0.97 | 0.96 | 0.96 | 0.93 | 96.52% |
| XGBoost | **0.98** | **0.97** | **0.98** | **0.96** | **98.07%** |

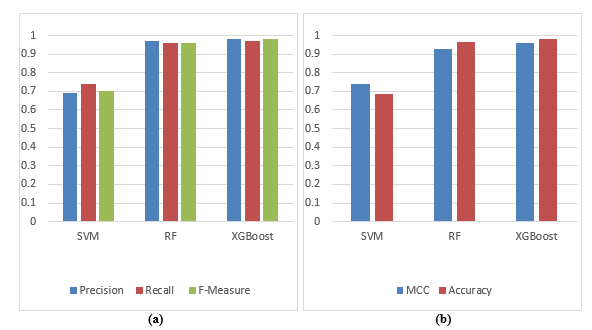
****

Figure 3: Comparison of Techniques based on: (a) Precision, Recall, and F-Measure. (b) MCC and Accuracy

# 5.1 Selection of XGBoost and LightGBM Algorithms

To address the need for a quick iteration model to predict the gender of a person, we need a gradient boosting framework that is fast, distributed, and high performing based on the tree model. Simultaneously, the training pace of the model is increased while maintaining a high level of accuracy, and the model's iteration periods are increased to ensure that the task of gender recognition is completed faster and more effectively. This study chooses a quicker model after determining to use the Boosting strategy. A faster LightGBM model is chosen to make selection and validation.

# 5.1.2 LightGBM

LightGBM is a type of Gradient Boosting decision tree (GBDT) that is used in classification, regression and supports efficient parallel training. It is also known for its light computational burden. It is based on a decision tree algorithm proposed using exclusive feature bundling (EFB) and gradient-based one-side sampling. Traditional GBDT algorithms use first-order derivative information for the optimization of the loss function, which means that each tree in the decision tree learns from the residuals and conclusions of previous trees. But, the increase in data volume affects the accuracy and efficiency. While LightGBM adopts a histogram algorithm that divides the eigenvalues into k intervals and selects the decision point between the k values. The leaf-wise tree growth strategy with multi-thread optimization and depth limitation help solve unnecessary memory consumption and processes big data with higher accuracy and efficiency. It also greatly helps in reducing the depth of the decision tree and avoiding over-fitting.

The LightGBM model is

(1)

Here, the difference between the predicted value and target value is calculated using the logistic loss function.

(2)

Then a Regression tree is used:

The representation of the regression tree can be done in another form, where is the weight, J is the total number of leave nodes, q is the decision rule, and the function can be written as:

(4)

In traditional GBDT only the gradient of the loss function is considered which uses a steepest descent method, while LightGBM uses a Newton's method to rapidly approximate the objective function. After simplification of Equation (4), the objective function can be written as:

(5)

Where are the first and second-order loss function, subsequently.

(6)

Using in place of leaf to represent sample set, the Equation (6) can be written as:

(7)

By using the structure of tree q(x), the limit and optimal weight of each node acquired are as follow:

(8)

(9)

Then the gain calculation formula is

(10)

LightGBM uses a multi-thread optimization approach to boost efficiency and save time, using maximum tree depth to reduce the depth of the decision tree and avoid over-fitting.

# 5.2 Comparing XGBoost and LightGBM Models

Figure 5 presents the comparison of both models. Whether it's LightGBM or XGBoost, the area where most of the work is done is divided into two parts. In the first part preprocessing of data takes place for both models. In the Second part location of the segmentation point is calculated which helps build the decision tree. The model's training speed and accuracy are determined by how well these two parts are processed and optimized.

First, LightGBM adopts a histogram algorithm that divides the eigenvalues into k intervals and selects the decision point between the k values. This method reduces the amount of data and prepares the data for later calculations.

XGBoost's pre-sorting method, on the other hand, is unable to reduce the quantity of data needed to optimize.

The pre-sorting technique of XGBoost requires determining a one-time segmentation gain for each eigenvalue to cope with the same features at the same time. On the other hand, LightGBM's histogram algorithm just needs to calculate the partition gain of the partition bucket, which saves a significant amount of time. Thus, LightGBM’s feature partitioning process is way quicker than XGBoost. XGBoost uses a level-wise tree growth strategy. While LightGBM adopts the leaf-wise tree growth strategy with multi-thread optimization and depth limitation help solve unnecessary memory consumption and processes big data with higher accuracy and efficiency.

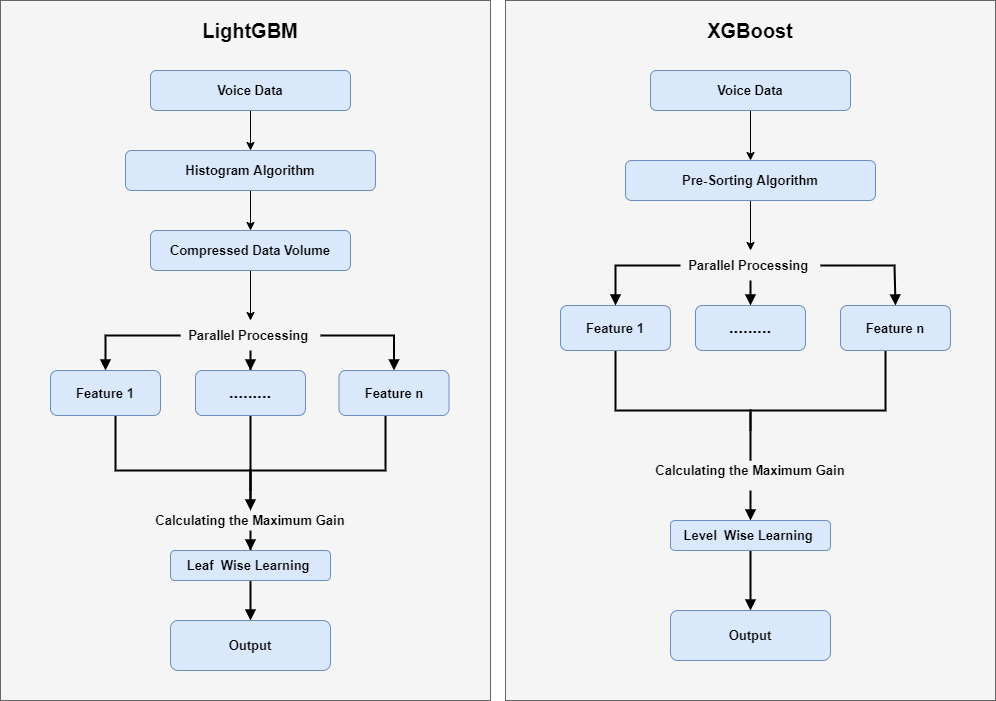


Figure 4: Flow chart of LightGBM and XGBoost

To further verify which algorithm is better, both models were trained on the voice data taken from Kaggle. Table 3 and Figure 6 presents the performance analysis of XGBoost and LightGBM in terms of Precision, Recall, F-Measure, MCC, and Accuracy.

From Table 3 and Figure 6 we can see that LightGBM has performed better than XGBoost. LightGBM not only achieved better accuracy results but also outperformed XGBoost in terms of precision, recall, and MCC. However, LightGBM achieved the same level of F-Measure score as compared to XGBoost. These results show the better performance of the LightGBM algorithm.

Table 3: Performance Analysis using Precision, Recall and F-Measure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Technique | Precision | Recall | F-Measure | MCC | Accuracy |
| XGBoost | 0.98 | 0.97 | 0.98 | 0.96 | 98.07% |
| LightGBM | **0.99** | **0.98** | **0.98** | **0.97** | **98.5%** |

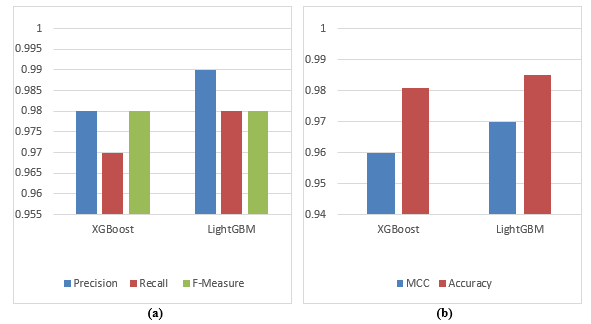
****

Figure 5: Comparison of XGBoost and LightGBM based on: (a) Precision, Recall, and F-Measure. (b) MCC and Accuracy

# 5.2 Threads to validity

# 5.2.1 Internal Validity

The analysis of this work is based on distinct and recognized evaluation standards that have been used in different studies in the past. Thus, the advent of new assessment criteria as a substitute for employed standards can affect results. Besides, the techniques utilized in this research can be replaced with the latest and superior techniques.

# 5.2.2 External Validity

We perform experimental analysis on a dataset taken from the Kaggle repository. A threat to validity may appear if the predicted techniques are associated with the data comprised by organizations or swap datasets with a distinct dataset which in turn may decrease accuracy. Similarly, the projected technique may be incompetent in forecasting desired results using other datasets.

# 5.2.3 Construct Validity

In this study, various ML techniques are used, on the voice dataset occupied from the Kaggle based on several assessment measures. The variety of techniques used in this study is at the core of their advancements above other techniques used by researchers in recent decades. Though, the threat might be that if a new technique is utilized, then there is a possibility that new techniques might cripple the presented techniques. Furthermore, dividing the dataset into training and testing, together with increasing or decreasing the number of folds for the experiments might affect results and result in the decrease of accuracy. Using the latest evaluation standards may result in better outcomes beating the current accomplished results.

1. **CONCLUSION**

Over the years, different machine learning models were studied by experts for gender recognition. Despite the wide research done in this field, still challenges exist. And there is room for improvement. This paper focuses on improving the accuracy rate for gender recognition and provides a novel LightGBM model for gender recognition. LightGBM’s unique EFB, GOSS, and Histogram-based approach is more effective in solving unnecessary memory consumption and processes big data with higher accuracy and efficiency, and has better generalization capabilities. The performance of LightGBM and other models are evaluated using evaluation metrics namely precision, recall, f-measure, MCC, and accuracy. The LightGBM’s performance evaluation results are better than other techniques with higher accuracy and precision, recall, and MCC results. Thus, the recommendation of this paper for gender recognition using voice is LightGBM. Suggestions for future studies might include

# REFERENCES

1. G. Sharma and S. Mala, “Framework for gender recognition using voice,” *Proceedings of the Confluence 2020 - 10th International Conference on Cloud Computing, Data Science and Engineering*, pp. 32–37, 2020, doi: 10.1109/Confluence47617.2020.9058146.
2. M. Hasan, M. Mclaren, H. Van, and D. A. Van Leeuwen, “Engineering Applications of Artificial Intelligence Speaker age estimation using i-vectors,” *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 99–108, 2014, doi: 10.1016/j.engappai.2014.05.003.
3. G. De Collongue and E. Cedex, “VOICE-BASED GENDER IDENTIFICATION IN MULTIMEDIA APPLICATIONS Hadi Harb , Liming Chen LIRIS CNRS FRE 2672 Dept . Mathématiques Informatique, Ecole Centrale de Lyon,” pp. 1–17, 1997.
4. Y. M. Zeng, Z. Y. Wu, T. Falk, and W. Y. Chan, “Robust GMM based gender classification using pitch and RASTA-PLP parameters of speech,” *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics*, vol. 2006, no. August, pp. 3376–3379, 2006, doi: 10.1109/ICMLC.2006.258497.
5. H. Harb and L. Chen, “A general audio classifier based on human perception motivated model,” *Multimedia Tools and Applications*, vol. 34, no. 3, pp. 375–395, 2007, doi: 10.1007/s11042-007-0108-9.
6. K. Zvarevashe and O. O. Olugbara, “Gender Voice Recognition Using Random Forest Recursive Feature Elimination with Gradient Boosting Machines,” *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems, icABCD 2018*, pp. 1–6, 2018, doi: 10.1109/ICABCD.2018.8465466.
7. F. Ertam, “An effective gender recognition approach using voice data via deeper LSTM networks,” *Applied Acoustics*, vol. 156, pp. 351–358, 2019, doi: 10.1016/j.apacoust.2019.07.033.
8. M. Buyukyilmaz and A. O. Cibikdiken, “Voice Gender Recognition Using Deep Learning,” vol. 58, no. Msota, pp. 409–411, 2016, doi: 10.2991/msota-16.2016.90.
9. M. A. Uddin, M. S. Hossain, R. K. Pathan, and M. Biswas, “Gender Recognition from Human Voice using Multi-Layer Architecture,” *INISTA 2020 - 2020 International Conference on INnovations in Intelligent SysTems and Applications, Proceedings*, pp. 0–6, 2020, doi: 10.1109/INISTA49547.2020.9194654.
10. R. D. R. Fagundes, A. A. C. Martins, F. Comparsi De Castro, and M. C. Felippetto De Castro, “Automatic gender identification by speech signal using eigenfiltering based on Hebbian learning,” *Proceedings - Brazilian Symposium on Neural Networks, SBRN*, vol. 2002-Janua, pp. 212–216, 2002, doi: 10.1109/SBRN.2002.1181476.
11. A. Ghosal and S. Dutta, "Automatic Male-Female Voice discrimination," *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT*), 2014, pp. 731-735, doi: 10.1109/ICICICT.2014.6781371.
12. A. Pahwa and G. Aggarwal, “Speech Feature Extraction for Gender Recognition,” *International Journal of Image, Graphics and Signal Processing*, vol. 8, no. 9, pp. 17–25, 2016, doi: 10.5815/ijigsp.2016.09.03.
13. L. N. Pondhu and G. Kummari, “Performance Analysis of Machine Learning Algorithms for Gender Classification,” *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, no. Icicct, pp. 1626–1628, 2018, doi: 10.1109/ICICCT.2018.8473192.
14. S. M. S. I. Badhon, M. H. Rahaman, and F. R. Rupon, “A Machine Learning Approach to Automating Bengali Voice Based Gender Classification,” *Proceedings of the 2019 8th International Conference on System Modeling and Advancement in Research Trends, SMART 2019*, pp. 55–61, 2020, doi: 10.1109/SMART46866.2019.9117385.
15. S. Chaudhary and D. K. Sharma, “Gender Identification based on Voice Signal Characteristics,” *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, pp. 869–874, 2018, doi: 10.1109/ICACCCN.2018.8748676.
16. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.
17. R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao, and D. Gao, “Degradation state recognition of piston pump based on ICEEMDAN and XGBoost,” *Applied Sciences (Switzerland)*, vol. 10, no. 18, Sep. 2020, doi: 10.3390/APP10186593.
18. Y. Chen *et al.*, “Classification of short single lead electrocardiograms (ECGs) for atrial fibrillation detection using piecewise linear spline and XGBoost,” *Physiological Measurement*, vol. 39, Jan. 2018, doi: 10.1088/1361-6579/aadf0f.
19. A. Koul, C. Becchio, and A. Cavallo, “Cross-validation approaches for replicability in psychology,” *Frontiers in Psychology*, vol. 9, no. JUL, pp. 1–4, 2018, doi: 10.3389/fpsyg.2018.01117.
20. M. Aljanabi, M. Qutqut, and M. Hijjawi, “Machine Learning Classification Techniques for Heart Disease Prediction: A Review,” *International Journal of Engineering and Technology*, vol. 7, no. October, pp. 5373–5379, 2018.
21. A. Fattah, M. M., P. S., and T. F., “A Decision Tree Classification Model for University Admission System,” *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 10, pp. 17–21, 2012, doi: 10.14569/ijacsa.2012.031003
22. E. Ortiz, “Sgd-Qn,” *Journal of Machine Learning Research*, vol. 10, p. 56, 2009.
23. V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, “An assessment of the effectiveness of a random forest classifier for land-cover classification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, no. 1, pp. 93–104, 2012, doi: 10.1016/j.isprsjprs.2011.11.002.
24. I. Rish, “An empirical study of the naive Bayes classifier,” *Physical Chemistry Chemical*, vol. 3, no. 22, pp. 4863–4869, 2001, doi: 10.1039/b104835j.