electronics

MDPI

Article

A Hybrid Ensemble Stacking Model for Gender Voice Recognition Approach

Eman H. Alkhammash [1,*], Myriam Hadjouni [2] and Ahmed M. Elshewey [3]

[1] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
[2] Department of Computer Sciences, College of Computer and Information Science, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; mfhaojouni@pnu.edu.sa
[3] Computer Science Department, Faculty of Computers and Information, Suez University, Suez, Egypt; elshewy86@gmail.com
* Correspondence: eman.kms@tu.edu.sa

Abstract: Gender recognition by voice is a vital research subject in speech processing and acoustics, as human voices have many remarkable characteristics. Voice recognition is beneficial in a variety of applications, including mobile health care systems, interactive systems, crime analysis, and recognition systems. Several algorithms for voice recognition have been developed, but there is still potential for development in terms of the system's accuracy and efficiency. Recent research has focused on combining ensemble learning with a variety of machine learning models in order to create more accurate classifiers. In this paper, a stacked ensemble for gender voice recognition model is presented, using four classifiers, namely, k-nearest neighbor (KNN), support vector machine (SVM), stochastic gradient descent (SGD), and logistic regression (LR) as base classifiers and linear discriminant analysis (LDA) as meta classifier. The dataset used includes 3168 instances and 21 features, where 20 features are the predictors, and one feature is the target. Several prediction evaluation metrics, including precision, accuracy, recall, F1 score, and area under the receiver operating characteristic curve (AUC), were computed to verify the execution of the proposed model. The results obtained illustrated that the stacked model achieved better results compared to other conventional machine learning models. The stacked model achieved high accuracy with 99.64%.

Keywords: machine learning; stacking model; ensemble learning; k-nearest neighbor; stochastic gradient descent; support vector machine; logistic regression; linear discriminant analysis

check for updates

Citation: Alkhammash, E.H.; Hadjouni, M.; Elshewey, A.M. A Hybrid Ensemble Stacking Model for Gender Voice Recognition Approach. *Electronics* **2022**, *11*, 1750. https://doi.org/10.3390/electronics11111750

Academic Editors: Yu-Chen Hu and Gemma Piella

Received: 30 March 2022
Accepted: 25 May 2022
Published: 31 May 2022

1. Introduction

Voice recognition is a significant technique for humans to communicate with each other to express their emotions, cognitive processes, and objectives. Humans produce distinct voices by a natural biological system in which the lungs exhale air and convert it to sounds via several organs including the lips, tongue, and teeth [1]. One of the most important systems for voice recognition is the ear, where the ear can differentiate between the gender voice based on various properties such as loudness and frequency. One of the most significant properties of voice is sexual dimorphism, particularly in pitch, which is particularly marked in human beings [2]. Gender recognition is used in several applications such as human-to-machine interaction, automatic salutations, speech emotion recognition and sorting the phone calls via gender categorization [3,4]. According to the acoustic properties, information regarding voice can be acquired by several acoustic factors, such as in the spectral formant frequencies and perceptual relevance frequencies [5,6]. Software for voice recognition converts analog signals to digital signals, known as analog-to-digital conversion [7]. In order to decode a signal, a computer needs a vocabulary or a dictionary of syllables, and also a way to compare the data to the signals. There is a hard disk that stores the speech patterns and loads them into the memory when running the program.

These patterns are checked by a comparator against the outcome of the analog-to-digital convertor; this process is known as pattern recognition. Artificial intelligence (AI), machine learning, and deep learning have made advances in speech recognition technology [8,9]. Machine learning plays vital role in addressing various problems in many fields, including such as medicine, banking, and finance and has been used in many studies involving gender voice recognition and classification by using data mining techniques and machine learning [10–12].

Ensemble learning has expanded over recent decades and is used to obtain high accuracy and better results for classification [13,14]. Ensemble learning has resolved the problems of traditional machine learning models by using multiple classifiers to gain better results for the predictive performance rather than using one single classifier [15,16]. Stacking learning is an ensemble model that performs a combination of multiple classifiers using a meta classifier [17]. In this paper, a novel stacked model is used to obtain improved results for the process of classification between male and female voices. This model uses four classifiers, namely, support vector machine (SVM), k-nearest neighbor (KNN), logistic regression (LR), and stochastic gradient descent (SGD), as base classifiers and linear discriminant analysis (LDA) as meta classifier. The stacked model results are compared with other machine learning models and also compared with other research studies which used the same dataset used in this work. Experimental results show that the proposed stacked model achieved high accuracy and proved to be a suitable model for gender voice recognition.

The rest of this paper is organized as follows: Section 2 summaries the related work on gender voice recognition. Section 3 describes the materials and methods. Section 4 demonstrates the machine learning models used in this study. Section 5 presents the experimental results and discussion. Section 6 presents the conclusion.

2. Related Work

Voice recognition and classification have been utilized for a long period of time. In recent decades, a lot of data mining and machine learning models have been conducted for gender voice recognition. A system to recognize gender voice is proposed in [1], using data from 46 speakers. The model consists of two classifiers, neural network classifier and support vector machine (SVM) via a stacking model. The accuracy obtained by the proposed model was about 93.48%. In [6], an android platform for speech was produced as a smartphone application for language and gender classification via more than one support vector machine model. The challenge in this work was utilizing dynamic training via the characteristics extracted by each user via installing spectra on the smartphone. This developed a robust classifier that achieved high accuracy during the process of classification. In [10], a multilayer perceptron (MLP) model with deep learning was produced to identify voice gender. The dataset contained 3168 voice samples of males and females, where the samples were developed via acoustic analysis. The accuracy obtained by the model was about 96.74%. In [11], the authors reported a study using 438 males and 192 females for gender voice recognition using different scenes (indoor and outdoor). The experimental results demonstrated that using non-linear smoothing improved accuracy to about 99.4%. In [18], the hyper parameters of the random forest model were optimized using the grid search method and used for gender voice classification. The experimental results conducted on the gender voice dataset indicated that the accuracy was 96.90%. In [19], a gaussian mixture model (GMM) classifier was used to differentiate gender and age. The classifier model's accuracy for gender recognition was above 90%. In [20], a gender voice classification model using the feature selection method via random forest recursive feature elimination and a gradient boosting model was also used. The dataset consisting of 1584 males and 1584 females was collected from various gender voices. The experimental results achieved an accuracy of 97.58%. In [21], the authors demonstrated an ensemble-based self-labeled algorithm (iCST-Voting) for voice gender classification. The algorithm performs a combination of three efficient self-labeled methods, namely, co-training, tri-training, and self-training, using an ensemble as a base classifier. The proposed algorithm achieved an accuracy of 98.4%. In [22], a deeper long short-term memory (LSTM) model was used to recognize gender voice. The model consisted of 3 steps. First, 10 efficacious attributes of the data were selected; second, a deep learning-based network was constructed with a double-layer LSTM frame, and third, the values for specificity, sensitivity, and accuracy were calculated. The study obtained an accuracy of 98.4%. In [23], several machine learning models, namely, k-nearest neighbor (KNN), artificial neural network (ANN), logistic regression, support vector machine (SVM), naïve Bayes, decision tree, and random forest were used for gender voice recognition. The results demonstrated that ANN achieved the best accuracy, with 98.35%.

3. Materials and Methods

In this paper, a new stacked model was constructed to demonstrate how different machine learning models can be stacked. In this stacked model, four machine learning models are utilized as the base classifiers and one machine learning model is applied as a meta classifier. Data preprocessing and k-fold cross-validation are used to obtain the best-predicted output. Figure 1 illustrates the six steps used to develop the stacked model. These steps are: (1) Dataset Description, (2) Data preprocessing, which includes data cleaning, label encoding, and data normalization, (3) k-fold cross-validation, (4) Using four base classifiers, namely, KNN, SVM, SGD, and LR models, (5) Using LDA model as the meta classifier, and (6) Calculating final prediction.

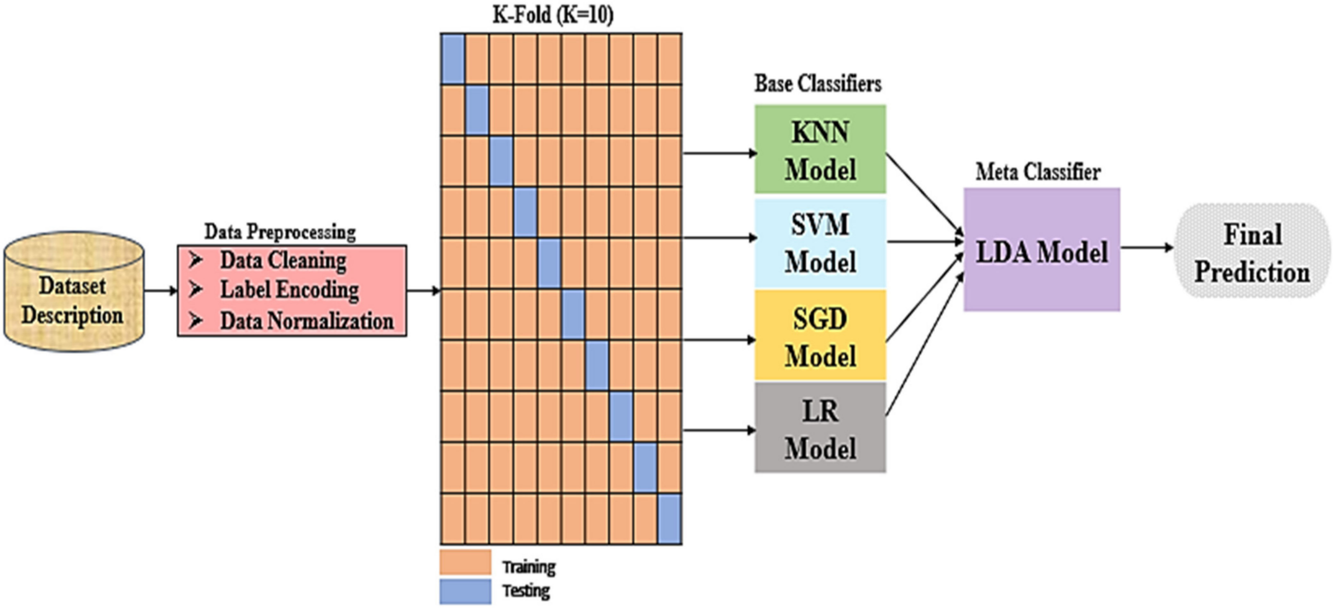


Figure 1. The stages for the proposed stacked model.

3.1. Dataset Description

The dataset used for this study is available at [24] and consists of 3168 instances of voice samples (1584 male voices and 1584 female voices). This dataset is especially used for gender voice recognition and classification and consists of 21 features, where 20 features are predictors, and one feature is the target. The format of the voice samples is a WAV file, which is preprocessed via acoustic analysis using tuneR and seewave packages in software R. The features of the dataset comprise standard deviation of the frequency, mean frequency, median frequency, quantile no.1, quantile no. 3, interquantile range, skewness, kurtosis, spectral entropy, spectral flatness, mode frequency, frequency centroid, minimum of fundamental frequency, maximum of fundamental frequency, average of fundamental frequency, minimum of dominant frequency, maximum of dominant frequency, average of dominant frequency, modulation index, range of dominant frequency, and the label, which is 1 for male and 0 for female.

3.2. Data Preprocessing

The data preprocessing step is vital in artificial intelligence and machine learning [25]. The quality of the data can affect the learning of machine learning models. Thus, it is very important to preprocess the data before importing it into the machine learning model as inputs [26]. In this study, the data preprocessing stage includes data cleaning, label encoding, and data normalization.

3.2.1. Data Cleaning

The The process of data cleaning is carried out to make the data ready for the process of analysis. The concept of data cleaning is not only aimed at eliminating the unnecessary parts of the data, but it also involves eliminating incorrect information which can affect the results of the machine learning model [27], restricting duplication in the dataset, deleting columns that contain one single value, and deleting columns that have low variance [28].

3.2.2. Label Encoding

Label encoding is the process of converting text or categorical values (non-numerical values) to numerical values to make them readable by the machine [29] and is thus a vital step in data preprocessing for the dataset in supervised learning [30]. In this study, the target feature contains categorical values (male and female). By using label encoding, a male is converted to 1, and a female is converted to 0.

3.2.3. Data Normalization

Data normalization is used to rescale the original data without changing their nature [25]. It is an essential step of data preprocessing in artificial intelligence and machine learning [26]. The major goal of normalization is to change the original values of the data in the dataset to a scaling value, without large changes in the differences in the ranges of the values of the data [31]. The converted data is commonly in the range [(0,1), (−1,1)]. The normalized data values are dependent on the mean value and standard deviation values. In this study, the data is normalized in the range (−1,1) to ensure all input variables to have the same treatment in the model [31]. The value is normalized, given an attribute and the value of an attribute, using Equation (1):

$$U_i = \frac{V_i - Avg(A)}{Std(A)} \tag{1}$$

where $Avg(A)$ and $Std(A)$ are the values of average and standard deviation, respectively, of the attribute A.

3.3. K-Fold Cross-Validation

K-fold cross-validation is used for several machine learning models; the objective of the cross-validation technique is to elucidate the achievement of accuracy for machine learning models [32]. This technique is widely used to approximate the prediction of machine learning models and k-fold cross-validation is one of the most popular forms of cross validation [33]. The main steps of the k-fold cross-validation approach are as follows:

1. Randomize the dataset.
2. Divide the dataset into groups.
3. For every group:
   (a) One group is used as a test set.
   (b) Remaining groups will be utilized as the training set.
   (c) Use a machine learning model on the training set and then evaluate it on the test set.

(d) Save the evaluation score and reject the model.

4. Establish the effectiveness of the model by using the evaluation scores of the model.

4. Machine Learning Models

In this section, the various machine learning models used in this study are summarized.

4.1. K-Nearest Neighbor

The K-nearest neighbor (KNN) model is a non-parametric statistical learning model [34]. It is widely used in many AI research fields, including classification, prediction [35], audio-visual recognition [36], and many other modern applications. The basic aim of the KNN model is to specify the category of an item of unknown data based on the categories of the other samples. First, it extracts the properties of the data that will be classified and compares them with the known category data in the testing set. It then selects the elements that have the smallest distance from the actual point and counts the frequency of the category. Finally, it takes the nearest neighbor that is the closest and most comparable [37]. There are many distance calculations functions that can be used, such as Manhattan distance, Euclidean distance or Chebyshev distance. In this study, Manhattan distance is utilized, because it gives better accuracy for high dimensional data than any another distances. The Manhattan distance [38] d_M between two points x_i and x_j with respective features i and j is given by Equation (2):

$$d_M(x_i, x_j) = \sum |x_i - x_j| \tag{2}$$

4.2. Support Vector Machine

Support vector machine (SVM) is an advanced supervised learning model that is utilized for both classification and regression problems [39]. SVM is based on structural risk minimization (SRM) principles that help to generalize better than the neural networks (NN) [40]. SVM achieves pattern recognition among two points' classes using support vectors (SV). SV represents a decision surface that is attained from quadratic programming problem solution [41,42]. The main SVM task is to estimate a classification function, as given in Equation (3):

$$f : R^n \rightarrow \{\pm 1\} \tag{3}$$

where f is the function that maps points x to their correct classification y; the input/output training data are from classes $(x_1, y_1), \ldots, (x_i, y_i) \in R^n \times \{\pm 1\}$. The SVM formula is given by Equation (4):

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b \tag{4}$$

where (x_i, y_i) is the ith training point, α_i and b are the learning weights, and $k(x, x_i)$ is the kernel function, where the kernel function transfers the lower dimensional space and returns the dot product of the transformed vectors in the higher dimensional space (transformed space), as shown in Equation (5):

$$k(x, x_i) = \varphi(x) \cdot \varphi(x_i) \tag{5}$$

4.3. Stochastic Gradient Descent

Stochastic gradient descent (SGD) is a type of gradient descent approach. It is a non-deterministic method that simplifies the mini-batch gradient descent (which is another type of gradient descent approach) and is widely used in optimization problems. These problems consist of finding the parameters that minimize a mathematical function [43]. As an amelioration of the batch gradient approaches, the stochastic gradient descent deals with only random instances of the training data to calculate the gradients in each iteration. The gradient descent can be succinctly written in the following form:

$$\overline{W} \Leftarrow \overline{W} - \alpha \nabla J(\overline{W}) \tag{6}$$

where $\overline{W}$ is a d-dimensional vector initialized randomly or using a heuristic chosen point, the step's size is described by the learning rate α, and $\nabla J(\overline{W})$ is the gradient vector, while the stochastic gradient descent formula [44] can be presented as shown in Equation (7):

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \tag{7}$$

where z_t is the random sample and Q is a loss function where the loss function evaluates the loss of the model and also describes the performance of the model.

4.4. Logistic Regression

Logistic regression (LR) is a model where the output variable probability is computed using an assumed collection of features. It provides a mechanism for applying linear regression techniques to classification problems [45]. Linear regression is not convenient for several outcomes, so logistic regression is another choice for these outcomes [46]. Logistic regression may contain one or more independent variables as linear regression, as checking multiple variables is very informative because it demonstrates the contribution of every variable after modifying the others. The logistic regression formula is given by Equation (8):

$$\hat{Y}_i = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_i x_i}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_i x_i}} \tag{8}$$

where $\hat{Y}_i$ is the estimated probability, β_0 is the model intercept, β_i is the coefficients of the regression, and x_i is the independent variables.

4.5. Linear Discriminant Analysis

Linear discriminant analysis (LDA) is an essential data analysis approach that has been widely used for distinguishing between different types of flowers [47]. The idea behind this is to define a subspace of lower dimension, contrasted to that of the original data sample, in which the data points of the initial problem are distinguishable [48]. Having a $d * n$ data matrix, $X = [x_1, x_2, \ldots, x_n] \in R^{d*n}$, $x_i \in R^d$, the purpose is to reduce the data through finding a transformation matrix $W \in R^{d*m}$ that will be used to map the high dimensional data in a series of low-dimensional data, $Y = [y_1, y_2, \ldots, y_n] \in R^{m*n}$, where $m \ll d$. The objective function [49] of LDA is given by Equation (9):

$$\min_{W^T S_t W = I} Tr\left(W^T S_w W\right) \tag{9}$$

where $Tr()$ is the trace of the matrix, S_w is the within-class scatter matrix, and S_t is the total-class scatter matrix. S_w and S_t are given by Equations (10) and (11):

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left(x_j^i - \mu^i\right)\left(x_j^i - \mu^i\right)^T \tag{10}$$

and

$$S_t = \sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T \tag{11}$$

where n_i is the ith class sample number, x_j^i is the jth sample that is in ith class, μ^i is the mean sample of ith class, and μ represents the mean sample of all the data.

4.6. Stacking Learning

Some machine learning models may not perform well for the requirements of difficult tasks, so it is important to combine different machine learning models to construct a learning model. This learning model is called the ensemble learning model. Ensemble learning can be classified into two groups. The first is called the sequential method, where the machine learning models have an intensive dependency and are generated sequentially, like boosting and gradient boosting. The second is called the parallel method, where the machine learning models do not have an intensive dependency and are generated in parallel, like bagging and random forest [50]. Stacking is an ensemble learning model that is different from boosting and bagging [51]. Stacking improves the model's performance, decreases the generalization error, and enables a wide use for the model. Stacking combines several machine learning models through a meta classifier. The stacking model consists of two main layers. The first layer is composed of different machine learning models called base classifiers. The base classifiers proceed with the output prediction process on the input data with various classification performances. Stacking combines all the output predictions to form new inputs to the second layer. The second layer is called the meta classifier, where the new inputs are the input for the meta classifier to obtain the final prediction. In this study, the base classifiers are KNN, SVM, SGD, and LR models. The meta classifier used in this study is the LDA model. The final prediction of the stacked model is 1 (in the case of males) and 0 (in the case of females).

5. Results and Discussion

This section reports on a set of key experiments carried out to evaluate the performance of the stacked model. The execution of the stacked model was carried out using jupyter notebook version (6.4.6). Jupyter notebook assists in the process of executing and writing python codes simply and is widely used as an open source for implementing and executing machine learning models for classification. To evaluate the effectiveness of the stacked model, five metrics were utilized to evaluate the performance of the final prediction for the stacking model. The performance of the stacked model proposed in this study was estimated using the accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC) [52]. Accuracy is the ratio between the number of correct predictions and the total number of predictions. Accuracy is calculated using Equation (12):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{12}$$

where TP if true positive, TN is true negative, FP is false positive, and FN is false negative.

Recall is the proportion between true positives and all the actual positives. Recall is calculated using Equation (13):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

Precision is the proportion between true positives and all the predicted positives. Precision is calculated using Equation (14):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{14}$$

The F1 score is a combination between recall metric and precision metric. The F1 score is a maximum when recall is equal to precision. The F1 score is computed using Equation (15):

$$\text{F1 Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{15}$$

The area under the receiver operating characteristic curve (AUC) is a vital metric for evaluating classification models. The receiver operating characteristic (ROC) curve is a graph that illustrates the execution of a binary classification model [53]. The ROC curve plots two values: the first value is called the true positive rate (TPR), which lies on the y-axis, and the second value is called false positive rate (FPR), which lies on x-axis at various thresholds. The AUC measures the whole area under the ROC curve and demonstrates the ability of the model to differentiate between classes. When the value of AUC is near to 1, this means that the accuracy of the model is good, and when the value of AUC is near to 0, this means that the accuracy of the model is poor.

In this study, the performance of the stacked model was compared with its component base classifier models, namely, KNN, SVM, SGD, and LR. The performance of these classification models was evaluated using accuracy, precision, recall, F1 score, and AUC as the evaluation metrics. All the models were evaluated using 10-fold cross validation to avoid overfitting (OF), where the error of the testing was 95%. Table 1 illustrates the configuration of the parameters for the four classification models, namely, KNN, SVM, SGD, and LR, respectively.

Table 1. Specification of the parameter for the base classifiers.

Models	Parameters
KNN	N_neighbors = 2, distance = Manhattan.
SVM	Kernel = linear, regularization parameter (C) = 0.1.
SGD	Loss = hinge, penalty = l2.
LR	Penalty = l2, fit_intercept = true.

The experimental results of accuracy, F1 score, recall, precision, and AUC for KNN model, SVM model, SGD model, LR model, and the stacked model, respectively, are presented in Table 2. The best results of the evaluation metrics are highlighted in bold.

Table 2. Comparison of prediction performances between the stacked model and its component base classifiers' models.

Models	Accuracy	F1 Score	Recall	Precision	AUC
KNN	97.78%	97.78%	98.73%	96.89%	0.998298
SVM	99.61%	**99.42%**	**99.50%**	**99.60%**	0.999479
SGD	96.20%	96.20%	96.83%	95.62%	0.997797
LR	99.05%	99.05%	**99.50%**	98.74%	0.999439
Stacked model	**99.64%**	**99.42%**	**99.50%**	**99.60%**	**0.999639**

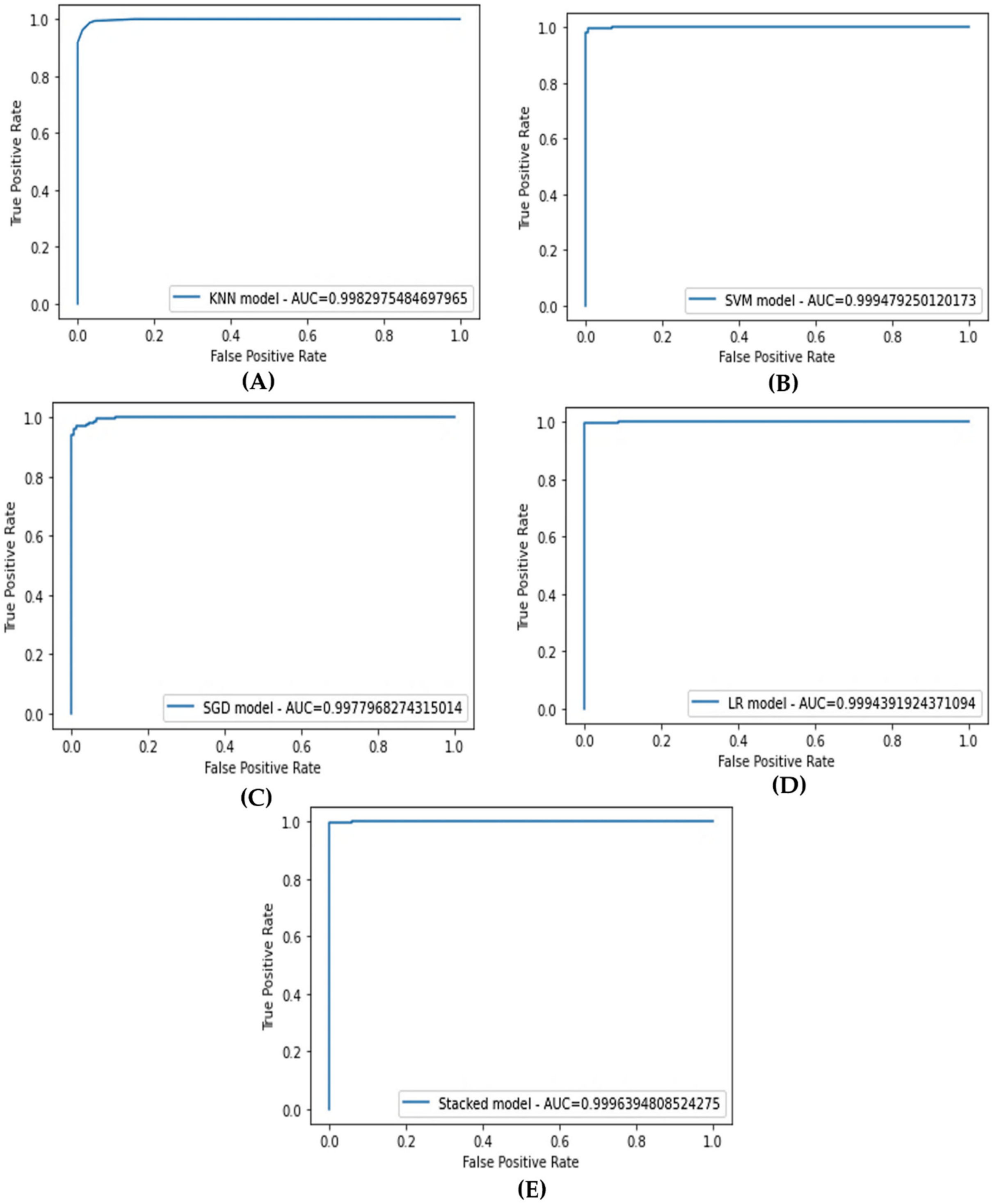
As illustrated in Table 2, the stacked model achieved better results in terms of accuracy, with 99.64%. The lowest accuracy was obtained by the SGD model, with 96.20%. In terms of the F1 score, the stacked model and SVM model produced better results with 99.42%, while the SGD model produced the poorest result, with 96.20%. The stacked model, SVM model, and LR model obtained the best results in terms recall, with 99.50%, while the SGD achieved the lowest recall score, with 96.83%. In terms of precision, the stacked model and SVM model demonstrated the bests results, with 99.60% and the SGD model achieved the worst results, with a score of only 95.62%. The best value for the AUC was achieved via the stacked model, with 0.999639. This value is considered excellent, as it is close to 1. The lowest value for AUC was achieved by the SGD model with only 0.997797. Figure 2 demonstrates the area under the ROC curve for the models, namely, KNN model, SVM model, SGD model, LR model, and the stacked model, respectively. Figure 3 demonstrates a comparison between the actual values and the predicted values for the models, namely, KNN, SVM, SGD, LR, and the stacked model, respectively.

Table 3 demonstrates the accuracy and the running time in milliseconds for the models, namely, the KNN, SVM, SGD, LR, and the stacked model, respectively.

To demonstrate the influence of the stacked model, the performance of the stacked model was compared with the performance of three conventional machine learning models, namely, decision tree (DT) model, random forest (RF) model, and adaptive boosting (Adaboost) model. Decision tree (DT) is a supervised learning model utilized for classification and regression problems [54].

DT is a set of consecutive decisions in order to reach to desired result. DT consists of root nodes, branches, child nodes, and leaf nodes. The significant role of decision tree is to search for the descriptive features that include vital information related to the target feature [55]. The next step is splitting the dataset over the features' values, where the target feature values for the dataset are clear as possible. Random forest (RF) is an ensemble learning model that is constructed using multiple decision trees that have various depth [56]. The decision trees utilize multiple features and variables to produce the final classification.

Adaptive boosting (Adaboost) is an ensemble learning model that is constructed using decision stumps [57]. Decision stumps are similar to decision trees, where the decision stumps include only one node and two leaves. Adaboost utilizes multiple decision stumps, such that every decision stump utilizes one feature or variable.



(A) **(B)** **(C)** **(D)** **(E)**

Figure 2. Area under the ROC curve for the for models: (**A**) KNN model, (**B**) SVM model, (**C**) SGD model, (**D**) LR model, and (**E**) the stacked model.

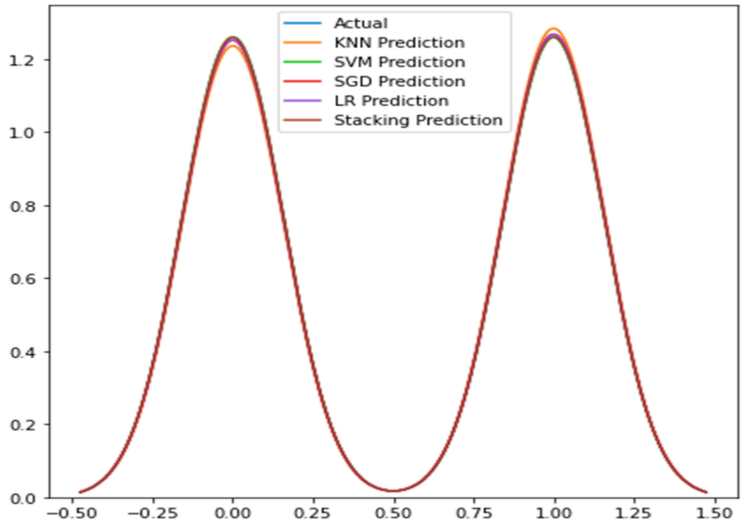


Figure 3. Comparison between the actual values and the predicted values.

Table 3. Running time and accuracy for the models, namely, KNN, SVM, SGD, LR, and the stacked model.

Models	Sample Size (*n*)	Accuracy	Time (*t*)
KNN	3168	97.78%	0.00842
SVM	3168	99.61%	3.304537
SGD	3168	96.20%	0.018447
LR	3168	99.05%	0.115368
Stacked model	3168	99.64%	14.960304

Table 4 shows the configuration of the parameters for the three conventional machine learning models, namely, DT model, RF model, and Adaboost model, respectively.

Table 4. Specification of the parameter for DT model, RF model, and Adaboost model.

Models	Parameters
DT	Max_depth = 2, criterion = entropy.
RF	N_estimators = 100.
Adaboost	N_estimators = 100.

The experimental results for accuracy, F1 score, recall, precision, and AUC for the supervised machine learning models, namely the DT model, RF model, Adaboost model, and the stacked model, respectively, are demonstrated in Table 5.

Table 5. Comparison of prediction performances between DT model, RF model, Adaboost model, and the stacked model, respectively.

Models	Accuracy	F1 Score	Recall	Precision	AUC
DT	95.72%	95.41%	95.53%	95.66%	0.975841
RF	98.79%	98.57%	98.61%	98.71%	0.998697
Adaboost	97.48%	97.27%	97.34%	97.42%	0.998182
Stacked model	99.64%	99.42%	99.50%	99.60%	0.999639

The stacked model achieves the best results compared with the supervised machine learning models as shown in Table 5. The DT model exhibited the poorest performance, in terms of all its scores.

Table 6 compares the results of several studies that used the same dataset used in this study.

Table 6. Comparison of proposed stacked model with several studies.

Studies	Model	Accuracy
Ref. [10]	MLP with deep learning	96.74%
Ref. [18]	Grid search optimization	96.90%
Ref. [21]	iCST-Voting	98.4%
Ref. [22]	Deeper LSTM	98.4%
Ref. [23]	ANN	98.35%
Proposed stacked model	KNN, SVM, SGD, and LR as base classifiers and LDA as meta classifier	99.64%

From Table 6, it can be seen that the proposed stacked model achieved the best performance in the terms of accuracy compared to the previous studies.

6. Conclusions

In this work, an effective stacked model was constructed for gender voice recognition. The proposed model utilizes four models as base classifiers, namely, the KNN model, SVM model, SGD model and LR model, and one model as a meta classifier, namely, the LDA model. Several performance metrics, namely, accuracy, recall, precision, F1 score, and AUC were used to evaluate the impact of the proposed model. The performance of the proposed model was compared with traditional machine learning models, where the proposed model achieved the best results for accuracy (99.64%), F1 score (99.42%), recall (99.50%), precision (99.60%), and AUC (0.999639). The performance of the proposed model was compared with traditional machine learning models, where the proposed model achieved the best results.

Author Contributions: Conceptualization, E.H.A., A.M.E. and M.H.; methodology, E.H.A., A.M.E. and M.H.; software, E.H. and A.M.E.; validation, E.H.A., A.M.E. and M.H.; formal analysis, E.H.A. and A.M; investigation, E.H.A., A.M.E. and M.H.; resources, E.H.A., A.M.E. and M.H.; data curation, E.H.A., A.M.E. and M.H.; writing—original draft preparation, E.H.A., A.M.E. and M.H.; writing—review and editing, E.H.A., A.M.E. and M.H.; funding acquisition, E.H.A. and M.H.; All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia. This work is also supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Gender Recognition by Voice. Available online: https://www.kaggle.com/primaryobjects/ (accessed on 17 February 2022).

Acknowledgments: The authors would like to acknowledge Taif University Researchers Supporting Project number (TURSP-2020/292) Taif University, Taif, Saudi Arabia. The authors would like also to acknowledge Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R193), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pahwa, A.; Aggarwal, G. Speech feature extraction for gender recognition. *Int. J. Image Graph. Signal Process.* **2016**, *8*, 17. [CrossRef]
2. Ericsdotter, C.; Ericsson, A.M. Gender differences in vowel duration in read Swedish: Preliminary results. *Work. Pap. Lund Univ. Dep. Linguist. Phon.* **2001**, *49*, 34–37.
3. Gamit, M.R.; Dhameliya, K.; Bhatt, N.S. Classification techniques for speech recognition: A review. *Int. J. Emerg. Technol. Adv. Eng.* **2015**, *5*, 58–63.

4. Yasmin, G.; Dutta, S.; Ghosal, A. Discrimination of male and female voice using occurrence pattern of spectral flux. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, India, 6–7 July 2017; pp. 576–581.
5. Hautamäki, R.G.; Sahidullah, M.; Hautamäki, V.; Kinnunen, T. Acoustical and perceptual study of voice disguise by age modification in speaker verification. *Speech Commun.* **2017**, *95*, 5.
6. Bisio, I.; Lavagetto, F.; Marchese, M.; Sciarrone, A.; Fra, C.; Valla, M. Spectra: A speech processing platform as smartphone application. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015; pp. 7030–7035.
7. Wang, W.C.; Pestana, M.H.; Moutinho, L. The effect of emotions on brand recall by gender using voice emotion response with optimal data analysis. In *Innovative Research Methodologies in Management*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 103–133.
8. Holzinger, A. Introduction to Machine Learning & Knowledge Extraction (MAKE). *Mach. Learn. Knowl. Extr.* **2019**, *1*, 20.
9. Ferri, M. Why topology for machine learning and knowledge extraction? *Mach. Learn. Knowl. Extr.* **2019**, *1*, 115–120. [CrossRef]
10. Buyukyilmaz, M.; Cibikdiken, A.O. Voice gender recognition using deep learning. *Adv. Comput. Sci. Res.* **2016**, *58*, 409–411.
11. Maka, T.; Dziurzanski, P. An analysis of the influence of acoustical adverse conditions on speaker gender identification. In Proceedings of the XXII Annual Pacific Voice Conference (PVC), Krakow, Poland, 11–13 April 2014; pp. 1–4.
12. Clarke, B.; Ernes, F.; Ha, H.Z. *Principles and Theory for Data Mining and Machine Learning*; Springer: New York, NY, USA, 2009.
13. Livieris, I.E.; Kanavos, A.; Tampakas, V.; Pintelas, P. An ensemble SSL algorithm for efficient chest x-ray image classification. *J. Imaging* **2018**, *4*, 95. [CrossRef]
14. Livieris, I.E.; Kiriakidou, N.; Kanavos, A.; Tampakas, V.; Pintelas, P. On ensemble SSL algorithms for credit scoring problem. *Informatics* **2018**, *5*, 40. [CrossRef]
15. Pławiak, P.; Acharya, U.R. Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals. *Neural Comput. Appl.* **2020**, *32*, 11137–11161. [CrossRef]
16. Pławiak, P. Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals. *Swarm Evol. Comput.* **2018**, *39*, 192–208. [CrossRef]
17. Wang, G.; Hao, J.; Ma, J.; Jiang, H. A comparative assessment of ensemble learning for credit scoring. *Expert Syst. Appl.* **2011**, *38*, 223–230. [CrossRef]
18. Ramadhan, M.M.; Sitanggang, I.S.; Nasution, F.R.; Ghifari, A. Parameter tuning in random forest based on grid search method for gender classification based on voice frequency. *DEStech Trans. Comput. Sci. Eng.* **2017**, *10*. [CrossRef]
19. Přibil, J.; Přibilová, A.; Matoušek, J. GMM-based speaker gender and age classification after voice conversion. In Proceedings of the 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, Denmark, 6–8 July 2016; pp. 1–5.
20. Zvarevashe, K.; Olugbara, O.O. Gender voice recognition using random forest recursive feature elimination with gradient boosting machines. In Proceedings of the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 6–7 August 2018; pp. 1–6.
21. Livieris, I.E.; Pintelas, E.; Pintelas, P. Gender recognition by voice using an improved self-labeled algorithm. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 492–503. [CrossRef]
22. Ertam, F. An effective gender recognition approach using voice data via deeper LSTM networks. *Appl. Acoust.* **2019**, *156*, 351–358. [CrossRef]
23. Prasad, B.S. Gender classification through voice and performance analysis by using machine learning algorithms. *Int. J. Res. Comput. Appl. Robot.* **2019**, *7*, 1–11.
24. Gender Recognition by Voice. Available online: https://www.kaggle.com/primaryobjects/voicegender (accessed on 17 February 2022).
25. Kotsiantis, S.B.; Kanellopoulos, D.; Pintelas, P.E. Data preprocessing for supervised leaning. *Int. J. Comput. Sci.* **2006**, *1*, 111–117.
26. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer International Publishing: Cham, Switzerland, 2015.
27. Chu, X.; Ilyas, I.F.; Krishnan, S.; Wang, J. Data cleaning: Overview and emerging challenges. In Proceedings of the 2016 International Conference on Management of Data, San Francisco, CA, USA, 26 June–1 July 2016; pp. 2201–2206.
28. Gudivada, V.; Apon, A.; Ding, J. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *Int. J. Adv. Softw.* **2017**, *10*, 20.
29. Mottini, A.; Acuna-Agost, R. Relative label encoding for the prediction of airline passenger nationality. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, Spain, 12–15 December 2016; pp. 671–676.
30. Zhuang, F.; Cheng, X.; Luo, P.; Pan, S.J.; He, Q. Supervised representation learning: Transfer learning with deep autoencoders. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
31. Rodríguez, C.K. *A Computational Environment for Data Preprocessing in Supervised Classification*; University of Puerto Rico: Mayaguez, Puerto Rico, 2004.
32. Anguita, D.; Ghelardoni, L.; Ghio, A.; Oneto, L.; Ridella, S. The ‘K’in K-fold cross validation. In Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, 25–27 April 2012; pp. 441–446.

33. Fushiki, T. Estimation of prediction error by using K-fold cross-validation. *Stat. Comput.* **2011**, *21*, 137–146. [CrossRef]
34. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
35. Jamjoom, M.; Alabdulkreem, E.; Hadjouni, M.; Karim, F.; Qarh, M. Early Prediction for At-Risk Students in an Introductory Programming Course Based on Student Self-Efficacy. *Informatica* **2021**, *45*, 6. [CrossRef]
36. Paul, B.; Dey, T.; Adhikary, D.D.; Guchhai, S.; Bera, S. A Novel Approach of Audio-Visual Color Recognition Using KNN. In *Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2022; pp. 231–244.
37. Zhang, C.; Zhong, P.; Liu, M.; Song, Q.; Liang, Z.; Wang, X. Hybrid Metric K-Nearest Neighbor Algorithm and Applications. *Math. Probl. Eng.* **2022**, *2022*, 8212546. [CrossRef]
38. Szabo, F. *The Linear Algebra Survival Guide: Illustrated with Mathematica*; Academic Press: Cambridge, MA, USA, 2015.
39. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1996.
40. Ukil, A. *Intelligent Systems and Signal Processing in Power Engineering*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
41. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425.
42. Yue, S.; Li, P.; Hao, P. SVM classification: Its contents and challenges. *Appl. Math. A J. Chin. Univ.* **2003**, *18*, 332–342. [CrossRef]
43. Ketkar, N. Stochastic gradient descent. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017; pp. 113–132.
44. Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.
45. Sammut, C.; Webb, G.I. Logistic Regression. In *Encyclopedia of Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2010; p. 631.
46. Stoltzfus, J.C. Logistic regression: A brief primer. *Acad. Emerg. Med.* **2011**, *18*, 1099–1104. [CrossRef]
47. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
48. Xanthopoulos, P.; Pardalos, P.M.; Trafalis, T.B. Linear discriminant analysis. In *Robust Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 27–33.
49. Nie, F.; Wang, Z.; Wang, R.; Wang, Z.; Li, X. Adaptive local linear discriminant analysis. *ACM Trans. Knowl. Discov. Data (TKDD)* **2020**, *14*, 9. [CrossRef]
50. Tang, Y.; Gu, L.; Wang, L. Deep Stacking Network for Intrusion Detection. *Sensors* **2022**, *22*, 25. [CrossRef]
51. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
52. Hossin, M.; Sulaiman, M.N. A review on evaluation metrics for data classification evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1.
53. Hoo, Z.H.; Candlish, J.; Teare, D. What is an ROC curve? *Emerg. Med. J.* **2017**, *34*, 357–359. [CrossRef] [PubMed]
54. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
55. Song, Y.Y.; Ying, L.U. Decision tree methods: Applications for classification and prediction. *Shanghai Arch. Psychiatry* **2015**, *27*, 130. [PubMed]
56. Azar, A.T.; Elshazly, H.I.; Hassanien, A.E.; Elkorany, A.M. A random forest classifier for lymph diseases. *Comput. Methods Programs Biomed.* **2014**, *113*, 465–473. [CrossRef]
57. Schapire, R.E. Explaining adaboost. In *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52.