Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3364141, 12 pages
https://doi.org/10.1155/2022/3364141

Research Article

Deep Learning-Based Classification of Spoken English Digits

Jane Oruh and Serestina Viriri

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, South Africa

Correspondence should be addressed to Serestina Viriri; viriris@ukzn.ac.za

Received 27 May 2022; Accepted 20 August 2022; Published 28 September 2022

Academic Editor: Muhammad Fazal Ijaz

Classification of isolated digits is the basic challenge for many speech classification systems. While a lot of work has been carried out on spoken languages, only limited research work on spoken English digit data has been reported in the literature. The paper proposes an intelligent-based system based on deep feedforward neural network (DFNN) with hyperparameter optimization techniques, an ensemble method; random forest (RF), and a regression method; gradient boosting (GB) for the classification of spoken digit data. The paper investigates different machine learning (ML) algorithms to determine the best method for the classification of spoken English digit data. The DFNN classifier outperformed the RF and GB classifiers on the public benchmark spoken English digit data and achieved 99.65% validation accuracy. The outcome of the proposed model performs better compared to existing models with only traditional classifiers.

1. Introduction

Speech is a means of communicating information from one or more speakers to one or more listeners. The speech produced by a speaker carries data in the form of signals, which are being transported from the mouth of the speaker to the ear of the listener. Speech is made up of sequences of phonemes, which are uttered at an average rate of approximately 12 phonemes per second [1]. Speech communication has become a predominant model for information exchange and social interaction among humans. Speech recognition is an emerging technology in the area of natural language processing (NLP) by Jurafsky and Martin [2].

Classification of speech is one of the most essential issues in speech processing [3]. Classification is the procedure of labeling a given set of data into classes. The process is conducted on organized data as well as on unorganized data. It starts with predicting the class of given data points which are known as targets, labels, or categories. The essence of classification predictive modeling is to map input values, x, to category y, output values [4] using a mathematical function. Classification of isolated digits is the basic challenge for many speech classification systems. Limited studies have been conducted on the classification of the English digit data.

The challenge with spoken digit recognition is a result of the following: (1) the spoken digits are of short acoustic duration, normally a few seconds of speech; (2) Some digits are acoustically identical to each other [5], Kopparapu and Rao [6]. The importance of this challenge has led several authors to research how to enhance digit recognition for different languages, which includes English [7], Portuguese [8], Arabic [9], and Mandarin [10].

The model proposes an intelligent-based system that will make use of a deep feedforward neural network (DFNN) with hyperparameter optimization techniques, an ensemble method; random forest (RF), and a regression method; gradient boosting (GB) for the classification of the spoken English digit data. The proposed DFNN performance was evaluated using hyperparameter optimization techniques such as adaptive moment estimation (Adam) optimization algorithm and stochastic gradient descent (SGD) optimization algorithm. Adam's optimization algorithm showed a better result than the classical SGD optimization algorithm. Optimization is a method of finding the best value of some function or model. Optimization for test cases aims at minimizing the number of test cases while delivering the best fault coverage [11]. Short-term Fourier transform (STFT) was used to extract

features from the audio data before performing one-hot encoding to produce the class label.

In our previous work, the proposed model used only DFNN with optimization techniques for the classification of the spoken English digit data [12], but in this work, the classification approach has been extended to use RF, GB, and DFNN for the classification of the spoken English digit data. The essence is to compare the performances of three different machine learning (ML) algorithms and to determine the best approach amongst them for classification purposes. The result from our experiment shows that DFNN is the best classification method compared to RF and GB.

Contributions to this research are as follows:

(1) A brief review of RF, GB, and DFNN methods and techniques is presented.

(2) The paper investigates three (3) different ML algorithms to determine the best approach for the classification of spoken English digit data.

The rest of the paper is arranged in this form: Section 2 considers recent speech classification techniques and their achievements. Section 3 gives a comprehensive description of the proposed ML algorithms, methods, and techniques for the proposed model. Section 4 discusses the experiments, model training, and experimental results and presents a relative analysis of the proposed model. Finally, Section 5 presents the conclusion.

2. Related Works

Speech classification in recent times has left several authors with the challenge of having to investigate the best method for achieving optimum accuracy.

The model proposes a novel approach that can be used to classify Bengali spoken digits using the convolutional neural network (CNN) [13]. The voice recordings of ten (10) individuals were classified considering gender, dialects, and age groups. The result of the classification accuracy is 98.37%, which shows the credibility of the proposed approach. The result here is bounded for Bengali spoken digits.

The work proposes a speech pathology recognition system that will automatically analyze the voice system of patients [14]. NN and deep learning methods were used for the classification of speech signals, to distinguish between a voice signal that is normal or pathological. The Levenberg Marquardt algorithm was used for classifying voice signals, whereas the restricted Boltzmann machine algorithm was used to implement the deep learning classification of the voice signals. The restricted Boltzmann machine algorithm shows an accuracy of 98.00% compared to the Levenberg Marquardt algorithm with 92.00% accuracy. The accuracy of the proposed model can be improved when tested with the other ML algorithms during network training.

The proposed model combines lexicon-based and machine learning methods for the prediction of hate speech, based on sentiment analysis [15]. The emotional facts found in the text assisted in improving the accuracy of hate speech detection, from 41.00% in the previous work to 80.64% on the test result. The proposed model could use deep learning optimization techniques alongside the lexicon-based and ML methods to improve hate speech accuracy.

A progressive rendering of a real-time speech emotion recognition application using the AlexNet image classification network was proposed in [16]. The baseline approach shows the result of 82.00% accuracy on the Berlin emotional speech (EMO-DB) data. The proposed model could not attain a high accuracy even with the AlexNet pretrained network.

The model proposed in [17] introduces a new multimodal deep learning framework that instinctively extracts features from textual-acoustic data for speech intention classification. The proposed system was tested in a real medical setting to serve as a reference for future research. The model achieved an average accuracy of 83.10% when 6 different intentions were detected. The model proposed here has performed better than existing models that used manufactured features. The proposed model accuracy is not very high.

The study in [18] studied a good deal of speech classification algorithms. A comparative analysis of five classification algorithms was conducted. Based on the result of investigations, a multilayer perceptron with 93.00% accuracy by the Robust scaler method was proposed. Achieving such accuracy for the proposed method is restricted to using the Robust scaler method to scale the multilayer perceptron. A deep feedforward multilayer perceptron was proposed in this work, and the accuracy was 99.65%.

In [19], a deep CNN was used to advance Pashto isolated digit recognition. Mel frequency cepstral coefficients (MFCC) were used in extracting features from the speech signal. The result shows an accuracy of 84.17% for testing, which is equivalent to a 7.32% improvement in comparison with existing works. The proposed approach is edged in Pashto isolated digit recognition.

In a recent study on Dari one-word speech recognition, CNN was used in recognition of the isolated words in Dari speech [20] using deep learning algorithms. MFCC was used for feature extraction during training. The test result shows 88.2% accuracy, which reveals that the proposed method predicts visualized words with high accuracy during training. The use of other deep learning techniques for the analysis of the Dari speech can improve the model accuracy.

Marcolla et al. [21] proposed a new approach known as "lie detection" for speech classification using a voice stress analysis method. The authors employed the long short-term memory (LSTM) network to analyze and classify a person's speech as authentic or not. The best neural network model in the proposed method showed a precision accuracy of 72.5%. The result is scientifically remarkable for such problems as voice stress analysis, which implies that it is possible to find patterns in the voice of people who are under stress. The precision accuracy is considerably not high and could be improved.

A multiclass classification was conducted on the spoken English digit dataset using support vector machine (SVM), K-nearest neighbor (KNN), and random forest (RF) [22]. RF performed better than SVM and KNN. With 10% testing data, 97.50% accuracy was obtained. Using ML methods

with hyperparameter optimization techniques as proposed in this work yielded a high accuracy of 99.65% on the same dataset.

A speech classification module was developed in [23] that will identify the appropriate speech for generating a medical report. The evaluation of the proposed model was performed using CNNs and LSTMs. Several parameters were tested and the performance of the model on different speaker features was examined. CNNs show 92.41% validation accuracy on 2709 speech segment data and are more thriving than LSTM networks. The proposed model could possibly be evaluated using different types of machine learning algorithms to obtain optimal validation accuracy, as shown in this work.

Sánchez–Hevia et al. [24] analyzed the performances of various deep neural networks (DNNs) for age estimation and differentiating gender from speech in interactive voice response (IVR) systems. The results of their experiment indicate good results for all the types of networks for gender classification, but combining CNNs and temporal convolutional networks (CTCN) gives a better result for the age group classification. Their best systems showed about 80% and 70% for precision and recall, respectively. Precision and recall accuracy is relatively high.

The proposed research in [25] used an RF and SVM classifier on 200 images of the standard Odia database for simulation. The simulation result shows 96.3%, 98.2%, 88.9%, and 93.6% accuracy on the Odia character and the Odia numerical database, respectively. The result is bounded in the Odia database.

The study by [26] presented an automated recognition system that will accurately classify authentic and forged signatures for offline signature verification. The proposed model was compared with six pretrained CNNs architecture based on transfer learning (TL) across a collection of publicly available signature samples. The outcome of their experiment shows 88% accuracy on the proposed model compared to other related networks and can be approved as a prototype for offline signature verification. The proposed model's accuracy could be improved using different machine learning algorithms to obtain optimal accuracy.

Sethy et al. [27] proposed an automated hybrid system for handwritten character recognition. The proposed model was tested on three benchmark datasets; Odia characters, Bangla numerals, and Odia numerals. The overall performance of handwritten Odia characters is 99.01% and 98.1%, Odia numerals are 98.6% and 97.6%, and Bangla numerals are 97.6% and 96.3%, respectively. The result analysis shows the best performance for least-square (LS)-SVM compared to RF. The performance of the proposed system is high but can be improved.

3. Methods and Techniques

The section initiates a progressive step in developing the proposed model. The proposed method for classifying the English digit data using ML algorithms; DFNN, RF, and GB includes the steps depicted in Figure 1. Data preprocessing is a step in which the raw data is transformed, or encoded, to bring it to a state that is appropriate for a machine or a deep learning model, and it is the first and pivotal step while creating a model.

3.1. Reading Dataset. The audio data is read using the library "Librosa." STFT features were extracted from the audio data [28]. The main idea behind the STFT feature extraction is to break up the longer time signals into smaller fragments of the same length and then compute the Fourier transform independently on each of the smaller fragments. The continuous form of STFT is expressed as

$$STFT\,x(t)(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)\omega(t-\tau)e^{-j\omega t}\mathrm{d}t. \quad (1)$$

The discrete form of STFT is conveyed as

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x(m)\omega(n-m)e^{-j\omega m}. \quad (2)$$

$\omega(n)$ stands for the analysis window [29], and it is assumed to be non-zero. Figure 2 represents extracted STFT features.

3.2. One-Hot Encoding Method. The proposed method in this work used one-hot encoding as a preprocessing step. The method is applied to the categorical data variables to convert them to a form that is appropriate for ML algorithms to perform an improved task of classification. This involves first mapping the categorical variables to integer values. Then each of the integer values is represented as a binary vector, i.e., all are 0's except for the index integer which is 1.

The conversion to this form is very necessary because many ML algorithms cannot work with the categorical data directly, it must first be converted to numbers. Figure 3 shows how each category value is transformed into a new column and assigned a "1" or "0" value, which is a notation for true/false. One-hot encoding of the audio data generates a target class label which was used as input into the proposed model.

3.3. Optimization Techniques

3.3.1. Stochastic Gradient Descent. Gradient descent is a method for minimizing a function $J(\theta)$, where J is the loss function and $\theta \in R^n$ is the model's parameter vector. To minimize $J(\theta)$, one has to calculate the gradient $\nabla J(\theta)$ with respect to the parameter θ. Then the parameter θ is updated as follows:

$$\theta = \theta - \eta \bullet \nabla J(\theta), \quad (3)$$

where the learning rate η controls the size of the steps to reach a local minimum. Formula (3) represents the steepest descent (batch-gradient descent) algorithm for minimizing $J(\theta)$.

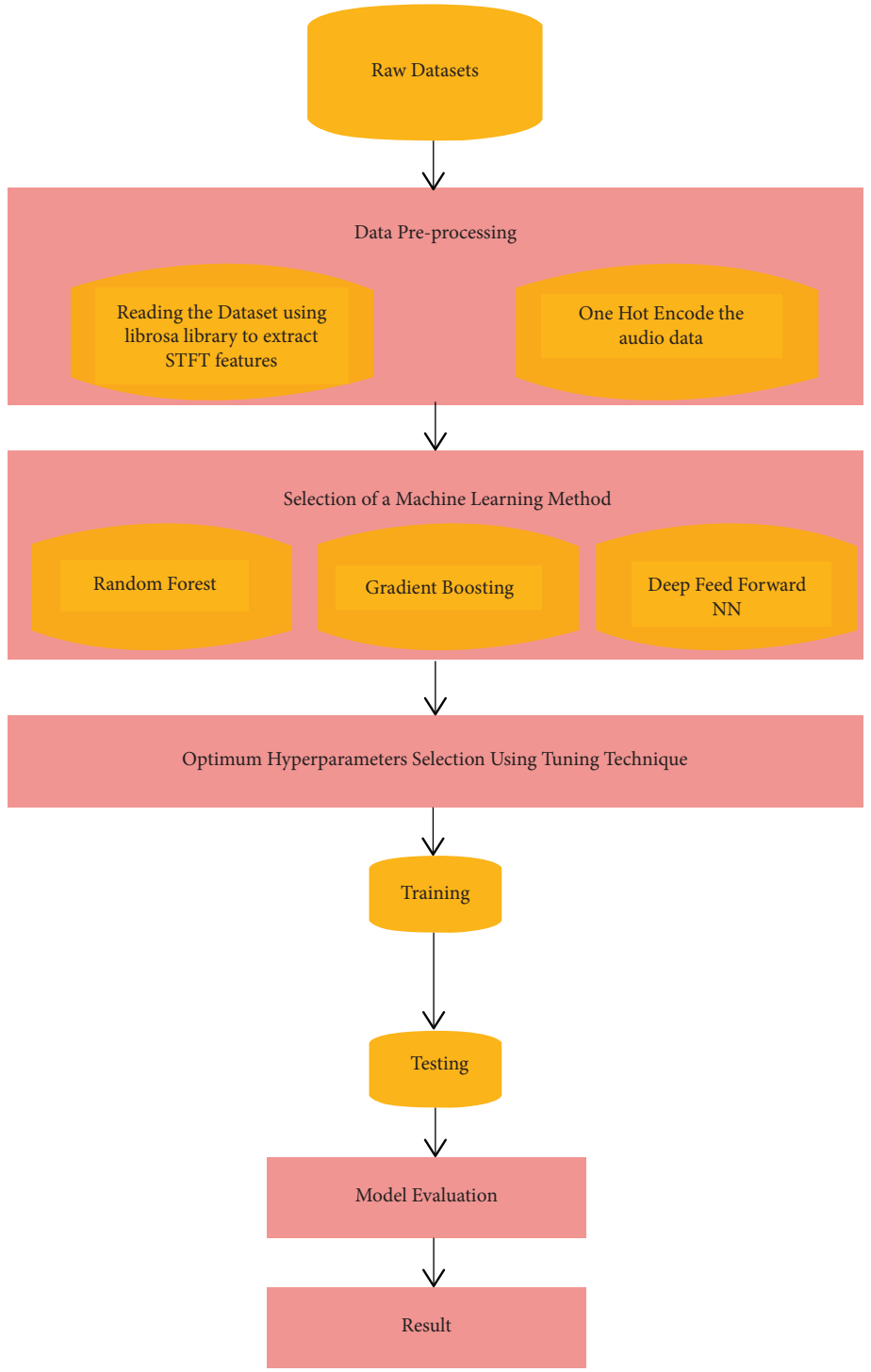

Figure 1: Flowchart of the proposed ML-based model.

For each training example $x(i)$ and label $y(i)$, SGD [30] executes an update of the parameter as

$$\theta = \theta - \eta \bullet \nabla J(\theta; x(i); y(i)). \tag{4}$$

SGD was developed to overcome the pitfalls of batch-gradient descent. SGD is faced with the challenge of having to choose an appropriate learning rate, to avoid shifts at the point of convergence.

3.3.2. Adam Optimization Algorithms. Adam [31], is a dynamic method for stochastic optimization that demands only first-order gradients with minimal memory requirement. Adam shows an edge over SGD by combining two other extensions of SGD; adaptive gradient (AdaGrad) [32], and RMSProp [33].

Suppose, we want to solve an optimization problem of the formula as

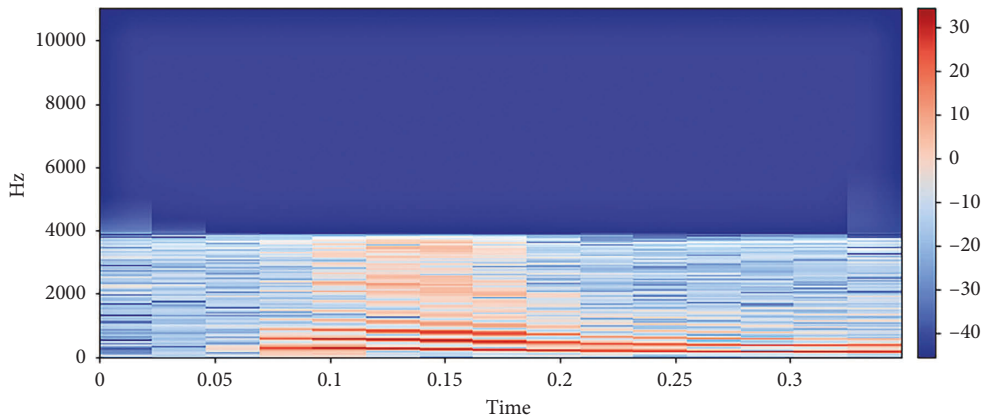

Figure 2: STFT representation of audio data.

	0	1	2	3	4	5	6	7	8	9
0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	1
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1
...	..	..	..	..	..	..	..	..	..	..
2395	0	1	0	0	0	0	0	0	0	0
2396	1	0	0	0	0	0	0	0	0	0
2397	1	0	0	0	0	0	0	0	0	0
2398	0	0	0	0	0	0	1	0	0	0
2399	0	0	1	0	0	0	0	0	0	0

[2400 rows × 10 columns]
(2400, 10)

Figure 3: A sample output showing one-hot encoding of the audio data.

$$\text{Minimize} f(x), \tag{5}$$

where $f(x)$ is a differentiable stochastic scalar function of the parameter x. The underlying idea of the Adam optimization algorithm applied to the above problem is to minimize the expected value, $E[f(x)]$, of the function $f(x)$. Suppose, $f_1(x)$, $f_2(x)$,..., $f_{T-1}(x)$, $f_T(x)$ are the functional values of the stochastic function $f(x)$ at successive time steps $1, 2, \ldots, T-1, T$. The stochasticity could be due to the evaluation of $f(x)$ at random subsamples (minibatches) of data points. The gradient of $f(x)$ at time step t is given by $g_t = \nabla_x f_t(x)$.

The estimate of the first moment (mean) represents the moving average of the gradient. On the other hand, the estimate of the second moment (variance) represents the moving average of the squared gradient. Let m_t be the first moment and v_t the second moment. Then the Adam algorithm computes the first order of momentum (the bias-corrected estimate of m_t) as

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^2}, \tag{6}$$

and the second order of momentum (the bias-corrected estimate of v_t) as

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^2}, \tag{7}$$

$\beta_1, \beta_2 \in [0, 1]$ as in equations (6) and (7) are the hyperparameters that control the exponential decay rates of moving averages. For the equations (6), (7), $\beta_1 = 0.9$, $\beta_2 = 0.999$.

The Adam optimization algorithm used for this study permits the network to achieve high accuracy by regulating the network's weights through an adaptive moment gradient change.

3.4. Random Forest Classifier. RF is a supervised ML algorithm that is used widely in classification and regression tasks. It is derived from the concept of ensemble learning, which involves the combination of various classifiers to resolve complicated problems and improve the model performance.

RF and other ensemble methods do not need as much preprocessing as some other methods. RF consists of multiple decision trees, each of which output a prediction. It is often said that in a given forest, more trees make for more robustness. RF creates decision trees through the selection of data samples randomly to get the prediction from each of the trees and then arrive at the best result using balloting [34].

In RF, each decision tree, otherwise known as the base learner, can benefit from a random subset of feature vectors [35]. Consequently, the feature vector is described in the following formula:

$$x = (x_1, x_2, \ldots, x_n), \tag{8}$$

which is an n-dimensional vector. Let $L(Y, f(x))$ be the loss function. The main objective is to find the function $f(x)$ that predicts the parameter Y.

The goal of the loss function is to minimize the expected value of the loss. Squared error loss and zero-one loss are common choices in regression and classification applications. They are defined in (9) and (10), respectively, [36].

$$L(Y, f(x)) = \left(Y - f(x)^2\right), \tag{9}$$

$$\begin{aligned} L(Y, f(x)) &= 1(Y \neq f(x)), \\ &= \{0, \text{if} Y = f(x), 1, \text{otherwise}. \end{aligned} \tag{10}$$

The steps in implementing the RF algorithm are as follows:

(i) Step 1—First, choose random samples from a given dataset.

(ii) Step 2—Here, the algorithm builds a decision tree for each sample. A prediction outcome is computed from each of the decision trees.

(iii) Step 3—This step performs voting for each of the predicted results.

(iv) Lastly, the final predicted result will be selected as the most voted prediction.

RF was implemented on the proposed model using an RF classifier represented as 'CLF'. Here we set the number of trees in the forest to 100, which is default of n_estimators, while the maximum_depth is set to 5. This implies that the number of decision trees is 100. Then the 'CLF' is fit to X_train and y_train, respectively, to train the model on the data. The model's accuracy when trained with the RF classifier showed a validation accuracy of 73.67%. The proposed RF algorithm and the corresponding flowchart are described in Algorithm 1 and Figure 4, respectively. Figure 5 explains the working of the RF algorithm.

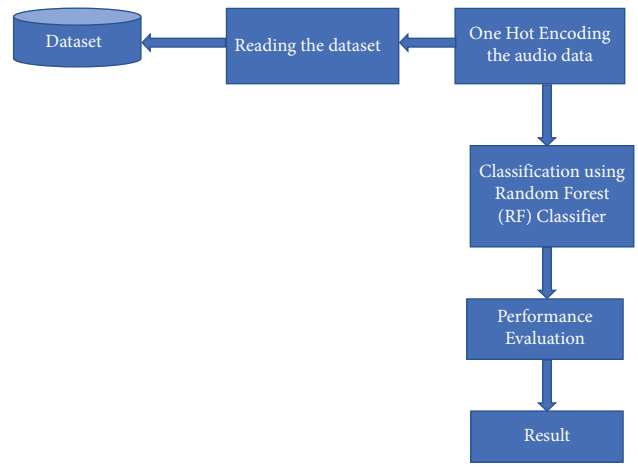

Figure 4: Random forest flowchart.

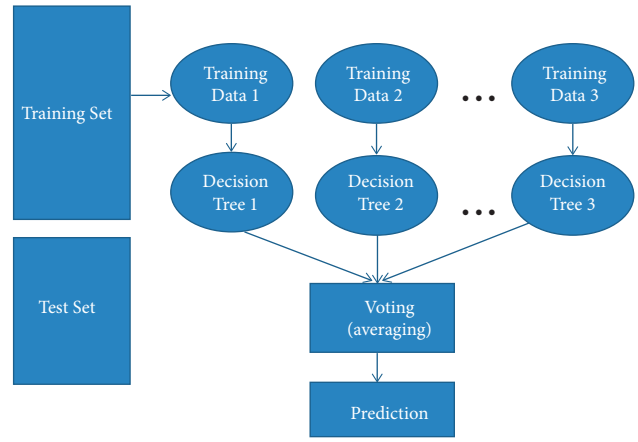

Figure 5: Summary of the workings of the random forest.

3.5. Gradient Boosting Classifier. The gradient boosting (GB) [37] classifiers are groups of ML algorithms that merge numerous weak models to produce a stronger predictive model. It is a concept from ensemble learning for solving regression and classification problems. GB combines several decision trees on subparts of the same dataset to form a stronger predictive model.

It integrates multiple machine learning models (mainly decision trees) and every decision tree model gives a prediction. Decision trees are used as the weak learners in GB. Decision trees solve the problem of ML by converting the data into a tree representation. If we align all the decision trees in a successive order, then it can be said that each subsequent model would minimize errors in the prior decision tree model. For a better understanding of the statements above, Figure 6 was used to illustrate.

The first step in GB is to create an initial constant prediction value F_0, where

$$F_0(x) = \arg_\gamma \min \sum_{i=1}^{n} L(y_i, \gamma), \tag{11}$$

where L is the loss function, γ is the predicted value. Since the target column is continuous, our loss function will be

$$L = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \gamma)^2. \tag{12}$$

Here y_i is the observed value, and γ is the predicted value. There is a need to find the least value of γ that minimizes the loss function.

The proposed GB algorithm is defined in Algorithm 2 with the corresponding flowchart as represented in Figure 7. The parameters that were used for the GB classification in this work are defined in Table 1.

3.6. The Proposed Deep Feedforward Neural Network Architecture. Deep neural networks (DNNs) have become a fundamental part of state-of-the-art ASR systems [38]. The DNN-based classification as proposed in the study enforced acoustic attributes that are plucked from the raw speech data [39]. In a feedforward neural network, information always travels in one direction [40]. There are no feedback connections and no cycles or loops in the network.

The proposed DFNN model used dense sequential fully connected layers that consist of three hidden layers with 256, 128, and 128 dimensions, respectively. The input and output layers are 1025 and 10 dimensions, respectively. In the first layer, the input layer is of 1025 dimensions whereas the input for the second dense layer is the output of the first layer, which is 256 dimensions. The third layer is related, the model instinctively considers the input dimension to be the same as the output of the last layer, which is 256. The last layer also known as the output layer with 10 dimensions represents 10 classes.

(1) **procedure** RandomForest Classifier (X, Y) $(\triangleright)X$ contains the STFT features of each audio sample, while$Y$ contains the target audio class label
(2) Read the dataset using the library "Librosa"
(3) Extract STFT features from the audio
(4) One hot encode the audio data to produce the class label.
(5) Split the dataset into training and testing set with STFT features as the input and audio class as the target label
(6) Start the random forest model
(7) Set up the hyperparameters tuning: n_estimators, max_depth
(8) RandomForestClassifier (Hyperparameters)
(9) Fitting training and the testing dataset
(10) Evaluate the model
(11) end procedure

Algorithm 1: The Random Forest Classification Model.

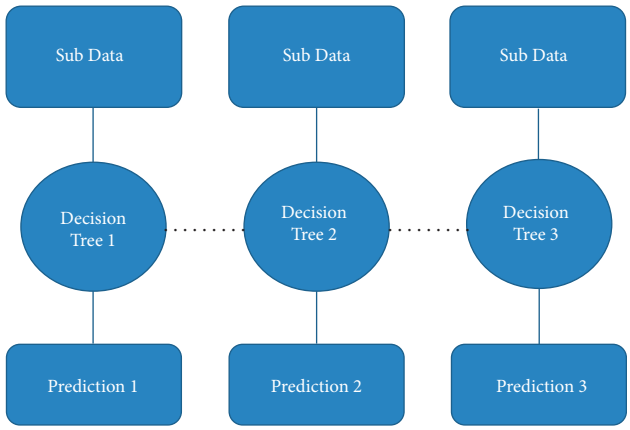

Figure 6: The architecture of gradient boosting.

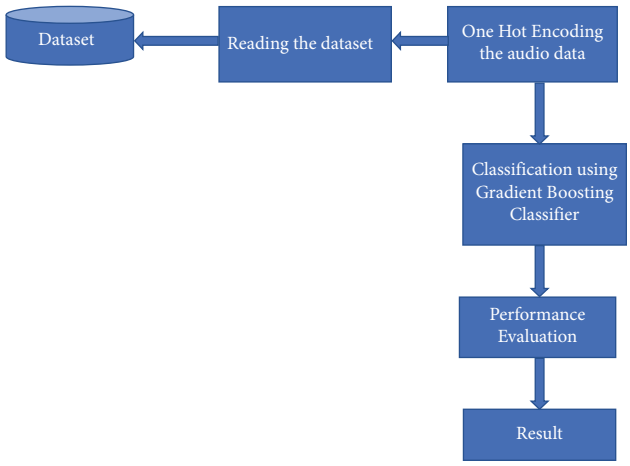

Figure 7: Gradient boost flowchart.

Hyperbolic tangent (tanh) activation was used at each level of the network layer except for the output layer. However, the output layer used the softMax activation function. The softMax activation function is often used in deep learning models as the last activation function of the neural network (NN) to regulate the network's output against the predicted output classes. The tanh activation function was also chosen due to its nonlinearity. The output of tanh is between the range of −1 to +1. Like the sigmoid, tanh in addition has a dispersing inclination issue. Tanh is also zero-centred, which enables modeling of inputs with strongly negative, neutral, and positive values.

Table 1: Parameters used for the GB Classification.

Parameters	Denoted As	Value
Number of estimators	n_estimators	20
Maximum features	max_features	2
Maximum depth	max_depth	5
Learning rate	learning_rate	0.05
Random state	random_state	0

The DNN's top layer is made up of nodes that employ the softMax activation function [39]. The function permits the DNN to produce class probabilities for each node which sums up to 1.

$$P(Y = i \mid x, W, b) = \frac{e^{W_i x + bi}}{\sum_j e^{W_j x + b_j}}, \tag{13}$$

Y represents the target mixtures, while W and, b represents the weight matrix and bias vector, respectively.

$$\ell(\theta = W, b, D) = -\sum_{i=0}^{|D|} \log\Big(P\big(Y = y_i \mid x_i, W, b\big)\Big). \tag{14}$$

The model proposed here used categorical cross-entropy loss, which specifies multiple classes. Hence, it is a loss function for multiclass classification tasks. They are used for optimizing classification models during training, to reduce loss function. Cross entropy loss is a key factor in deciding how many epochs will be used for a particular model.

Adam and SGD are the optimization algorithms for minimizing errors in the proposed DFNN. The results of the two algorithms were compared, and Adam showed better accuracy than SGD. The proposed DFNN structure is represented in Figure 8, whereas the algorithm for the proposed DFNN model is illustrated in Algorithm 3. The flowchart for the proposed DFNN model is illustrated in Figure 9.

(1) **procedure** GradientBoosting Classifier (X, Y) $(\triangleright)X$ contains the STFT features of each audio sample, while$Y$ contains the target audio class label
(2) Read the dataset using the library "Librosa"
(3) Extract STFT features from the audio.
(4) One-hot encode the audio data to produce the class label.
(5) Split the dataset into training and testing set with STFT features as the input and audio class as the target label.
(6) Start the GradientBoosting model
(7) Set up the hyperparameters tuning: n_estimators, learning_rate, max_features, max_depth, and random_state.
(8) GradientBoostingClassifier (Hyperparameters)
(9) Fitting training and the testing dataset
(10) Evaluate the model
(11) end procedure

Algorithm 2: The GradientBoosting Classification Model.

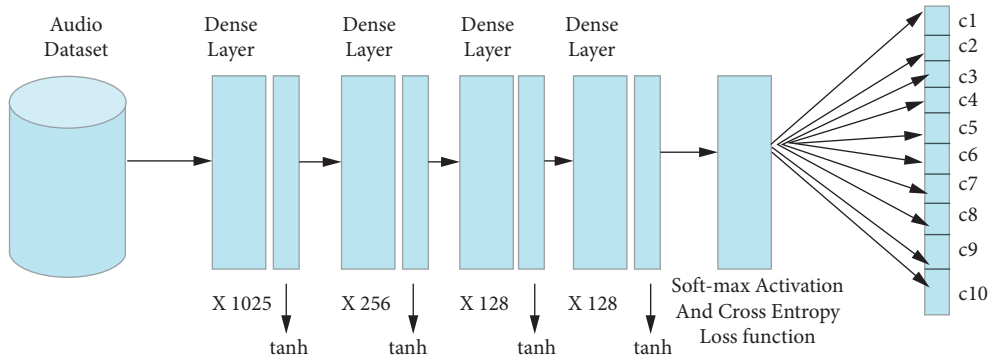

Figure 8: Architectural diagram for the neural network method.

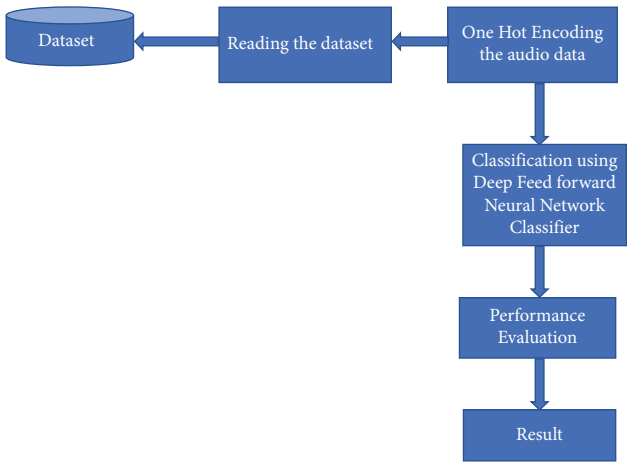

Figure 9: Deep feedforward neural network flowchart.

4. Experimental Results

4.1. Dataset. The dataset used for the model's training and validation is a well-founded freely accessible dataset, a collaboration that works with Pannous [41] to improve speech recognition. The dataset is from the Librosa library [42]. It consists of isolated digits with a total of 2400 different audio files in a WAV format for the model training. For the training of each of the proposed models, a training-validation data split of 75%–25% was used. The validation accuracy is fitted into the model's training output. STFT features were used as input and the audio class as the target label in the proposed model.

The RF technique was the first to be implemented in the work using an RF classifier represented as 'CLF'. The hyperparameter tuning was set up with the number of decision trees = 100, which is the default of n_estimators, while the maximum depth was set to 5. The 'CLF' was fitted to the X_train and y_train, respectively, to train the model on the data. After training, the result of the RF classifier showed a validation accuracy of 73.67%.

The model was trained next using the GB classifier with parameters set for n_estimators, max_features, max_depth, learning_rate, and random_state of values 20, 2, 5, 0.05, and 0, respectively. The learning rate was adjusted to 0.075, 0.1, 0.25, 0.5, 0.75, and 1 during training. The best validation accuracy of 81.80% on the training set and 49.00% on the validation set for a 0.75 learning rate were achieved. A sample screenshot showing the results of each digit's precision, recall, and f1-score were as shown in Figure 10.

For DFNN training, the epoch size was set initially to 20 epochs using the Adam optimization algorithm, and increased later to 30, 50, and 100 epochs, respectively. The same number of epoch sizes were replicated for SGD optimization algorithm training. The accuracy of Adam and SGD optimization algorithms for the various epoch sizes; 20, 30, 50, and 100 were calculated and compared. Both Adam and SGD optimization algorithms showed the best accuracy

(1) **procedure** Deep Feedforward Neural Network Classifier (X, Y) $(\triangleright)X$ contains the STFT features of each audio sample, while Y contains the target audio class label
(2) Reading the dataset using the library "Librosa"
(3) Extract STFT features from the audio
(4) One hot encode the audio data to produce the class label.
(5) Split the dataset into training and testing set with STFT features as the input, and audio class as the target label.
(6) Start the DFNN model
(7) Epoch = N; audio = first audio
(8) for i = 1: N do
(9) First_Layer = Dense (first audio, input_dim = 1025, output_dim = 256)
(10) Second_Layer = Dense (input_dim = 256, output_dim = 128)
(11) Third_Layer = Dense (Input_dim = 128, output_dim = 128)
(12) Fourth_Layer = Dense (input_dim = 128, output_dim = 128)
(13) Output_Layer = Dense (input_dim = 128, output_dim = 10)
(14) if Output_Layer = = the target_Layer then
(15) audio = next audio
(16) end if
(17) end for
(18) end procedure

Algorithm 3: The DFNN Classification Model.

Classification Report

	precision	recall	f1-score	support
0	0.29	0.34	0.31	47
1	0.41	0.37	0.39	65
2	0.32	0.38	0.34	48
3	0.26	0.22	0.24	55
4	0.45	0.51	0.48	53
5	0.49	0.54	0.51	72
6	0.47	0.46	0.47	65
7	0.37	0.29	0.32	59
8	0.75	0.77	0.76	70
9	0.39	0.35	0.37	66
accuracy			0.43	600
macro avg	0.42	0.42	0.42	600
weighted avg	0.43	0.43	0.43	600

Figure 10: A sample screenshot showing results of each digit's precision, recall, and f1-score.

for 100 epochs, as demonstrated in Figures 11 and 12. Figure 11 shows the accuracy and the loss curve diagram of the model's performance for 100 epochs using Adam optimization algorithms. The model has achieved a validation accuracy of 99.65% and a minimal validation loss of 0.25%. Figure 12 is the accuracy and loss curve diagram of the model for 100 epochs using the SGD optimization algorithm. The SGD result has shown a validation accuracy of 98.42% and a validation loss value of 0.54%.

Table 2 shows the accuracy comparison of the different ML classification methods used in training the dataset. In Table 2, it was noticed that the DFNN technique exhibited a validation accuracy of 99.65% compared to the other classification methods. The model's performance was compared with some traditional classifiers [22] such as SVM, KNN, and RF on the same dataset. The model proposed in this work attained 99.65% accuracy compared to the conventional approach, as demonstrated in Table 3.

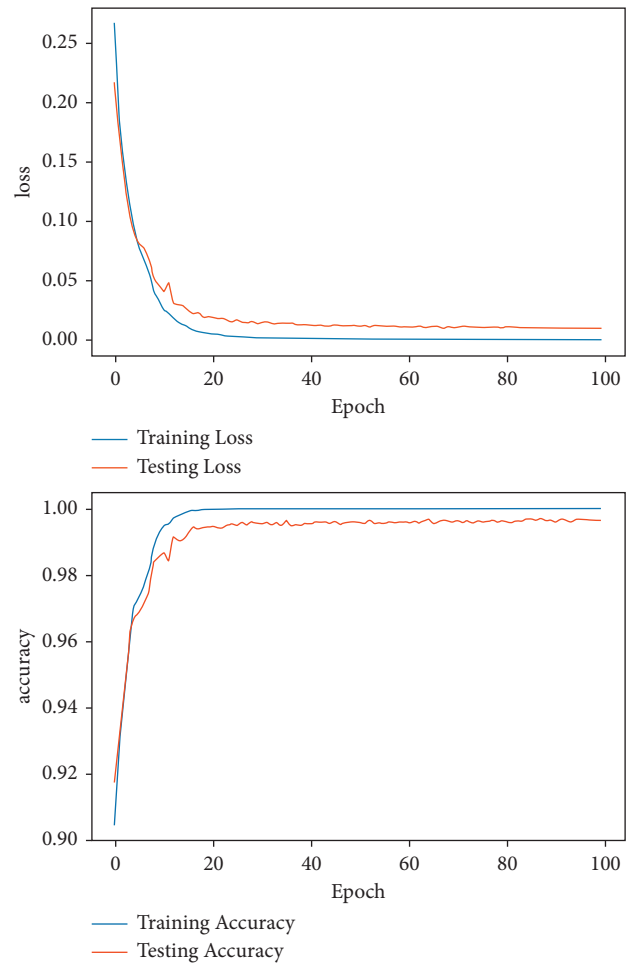

Figure 11: Model accuracy and loss curve diagram using the Adam optimization algorithm.

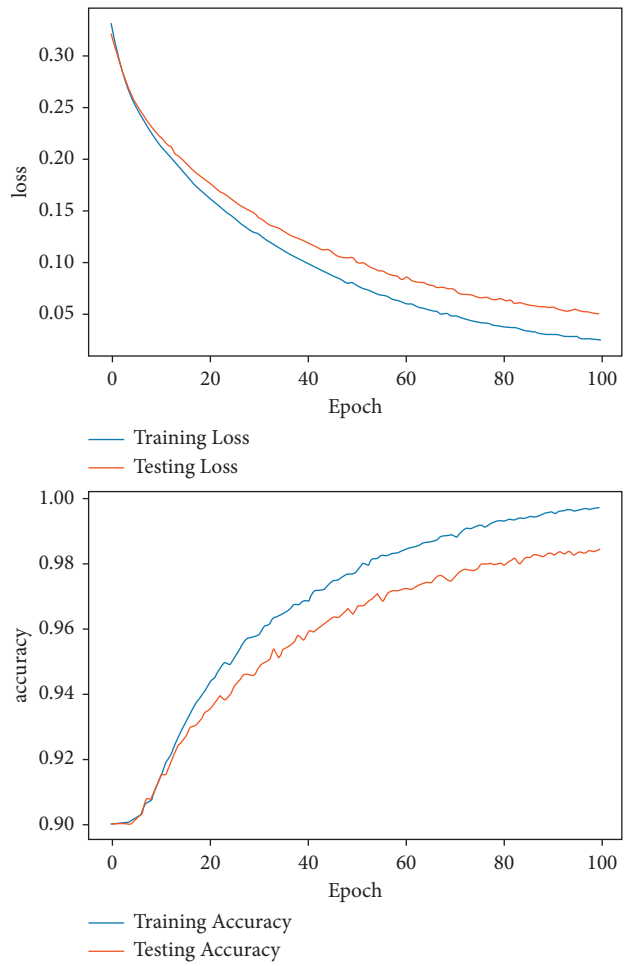

Figure 12: Model accuracy and loss curve diagram using the SGD optimization algorithm.

Table 2: Accuracy comparison of the various classification algorithms.

Algorithm	Random forest	Gradient boosting	Deep feedforward neural network
Accuracy	73.67%	79.70%	99.65%

5. Discussion

Research on speech classification is still an open issue as a result of the limitations of ASR systems. Recognition of isolated words/digits is practically arduous. A classification technique that involved three techniques; DFNN with hyperparameter optimization techniques, an ensemble method, i.e., RF, and a regression method, i.e., GB, was proposed in this study for the classification of spoken English digit data with the primary objective of determining the best method among them.

Figure 5 shows the working of the RF algorithm. For a better understanding of the RF algorithm, knowledge of the decision tree algorithm is vital. The validation accuracy for the RF classifier after training shows a result of 73.67% accuracy, which is not high. This suggests that more decision trees should be created since the greater number of trees in the forest results in greater accuracy while overfitting is avoided.

Figure 6 shows the architecture of the GB. The GB's performance was compared for different learning rates; lr_list = (0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1), where 'lr' represents the learning rate. Training the proposed model with variable learning rates achieved 81.80% on the training set and 49.00% on the validation set for a 0.75 learning rate. This is an indication that the GB was overfitting the training dataset which affects the accuracy.

Figure 8 shows the architectural diagram of the proposed DFNN model. The model was trained first using the Adam optimization algorithm and retrained using the SGD optimization algorithm using variable epoch sizes. Increasing epoch size helps in enhancing the model's network accuracy. Epoch performs an essential function in the network training of a model [43]. The total amount of epochs to be

Table 3: Accuracy Comparison of the Various classification Algorithms.

Method	Dataset	Features	Classifier	Accuracy
Supervised learning	Spoken English digit	MFCC	SVM + KNN + RF	97.50%
The proposed model	Spoken English digit	STFT	RF + GB + NN	99.65%

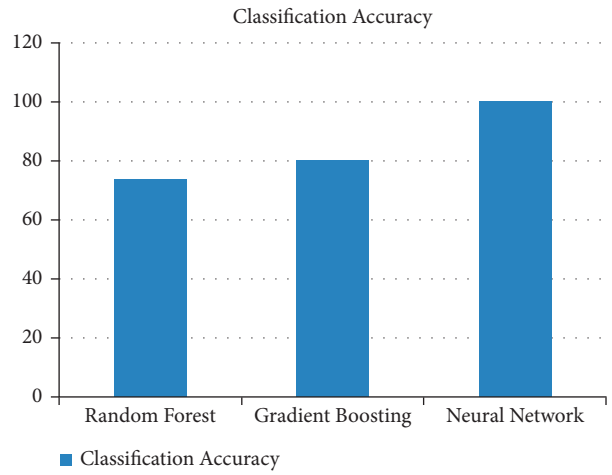

Figure 13: A bar chart for comparing the classification accuracy of the machine learning algorithms.

applied in network training would help to determine whether the data is overtraining or not.

The performance evaluation in Table 3 suggests that the proposed deep feedforward method is optimal for spoken digit classification. A summary of the performance of the ML algorithms used for this work is depicted as a bar chart in Figure 13.

6. Conclusion

Classification of spoken English digit data was conducted using DL methods; ensemble, regression, and a DFNN method with hyperparameter optimization algorithms. STFT feature extraction and a one-hot encoding was implemented on spoken digit data to produce the STFT features as input and the audio class as the target label in the proposed model. Classification results of the training have shown that the DFNN model outperformed the RF and GB models with the validation accuracy of 99.65% compared to the 73.67% and 79.70% accuracy of RF and GB, respectively. Hence, the DFNN model is an efficient approach for the classification of spoken English digit data.

Data Availability

The data used are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] P. N. Nasreen, A. C. Kumar, and P. A. Nabeel, "Speech analysis for automatic speech recognition," in *Proceedings of the International Conference on Computing, Communication and Science*, Pune, India, January 2016.

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Pearson Education, Bengaluru, Karnataka, 3rd edn edition, 2020.

[3] Q. T. Nguyen and T. D. Bui, "Speech classification using SIFT features on spectrogram images," *Vietnam Journal of Computer Science*, vol. 3, no. 4, pp. 247–257, 2016.

[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, Vol. 1, MIT press, , Cambridge, MA, USA, 2016.

[5] D. F. Silva, V. M. A. de Souza, and G. E. A. P. A. Batista, "A comparative study between MFCC and LSF coefficients in automatic recognition of isolated digits pronounced in Portuguese and English," *Acta Scientiarum. Technology*, vol. 35, pp. 621–628, 2013.

[6] S. K. Kopparapu and P. V. S. Rao, "Enhancing spoken connected-digit recognition accuracy by error correction codes—a novel scheme," *Sadhana*, vol. 29, no. 5, pp. 559–571, 2004.

[7] K. Nimje and M. Shandilya, "Automatic isolated digit recognition system: an approach using HMM," *Journal of Scientific and Industrial Research*, vol. 70, 2011.

[8] D. F. Silva, V. M. A. de Souza, G. E. A. P. A. Batista, and R. Giusti, "Spoken digit recognition in Portuguese using line spectral frequencies," in *Ibero-American Conference on Artificial Intelligence*Springer, New York, NY, USA, 2012.

[9] Y. A. Alotaibi, "Investigating spoken Arabic digits in speech recognition setting," *Information Sciences*, vol. 173, no. 1-3, pp. 115–139, 2005.

[10] R.-C. Shyu, J.-F. Wang, and J.-Y. Lee, "Improvement in connected Mandarin digit recognition by explicitly modeling coarticulatory information," *Journal of Information Science and Engineering*, vol. 16, pp. 649–660, 2000.

[11] K. Tyagi and K. Tyagi, "A comparative analysis of optimization techniques," *International Journal of Computer Application*, vol. 131, no. 10, pp. 6–12, 2015.

[12] J. Oruh and S. Viriri, "Deep learning with optimization techniques for the classification of spoken English digit," in *International Conference on Computational Collective Intelligence*Springer, New York, NY, USA, 2021.

[13] R. Sharmin, S. K. Rahut, and M. R. Huq, "Bengali spoken digit classification: a deep learning approach using convolutional neural network," *Procedia Computer Science*, vol. 171, pp. 1381–1388, 2020.

[14] D. S. S. Megala, R. Padmapriya, B. Jayanthi, and M. Suganya, "Detection and classification of speech pathology using deep learning," *International Journal of Scientific & Technology Research*, vol. 8, pp. 3045–3051, 2019.

[15] R. Martins, M. Gomes, J. J. Almeida, P. Novais, and P. Henriques, "Hate speech classification in social media using emotional analysis," in *Proceedings of the 2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 61–66, IEEE, Sao Paulo, Brazil, October 2018.

[16] M. Lech, M. Stolar, C. Best, and R. Bolia, "Real-time speech emotion recognition using a pre-trained image classification network: effects of bandwidth reduction and companding," *Frontiers of Computer Science*, vol. 2, p. 14, 2020.

[17] Y. Gu, X. Li, S. Chen, J. Zhang, and I. Marsic, "Speech intention classification with multimodal deep learning," *Adv Artif Intell*, vol. 10233, pp. 260–271, 2017.

[18] O. Mamyrbayev, N. Mekebayev, M. Turdalyuly, N. Oshanova, T. I. Medeni, and A. Yessentay, "Voice identification using classification algorithms," in *Intelligent System And Computing*IntechOpen, London, UK, 2019.

[19] B. Zada and R. Ullah, "Pashto isolated digits recognition using deep convolutional neural network," *Heliyon*, vol. 6, no. 2, Article ID e03372, 2020.

[20] M. Dawodi, J. A. Baktash, T. Wada, N. Alam, and M. Z. Joya, "Dari speech classification using deep convolutional neural network," in *Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, September 2020.

[21] F. M. Marcolla, R. de Santiago, and R. L. S. Dazzi, "Novel lie speech classification by using voice stress," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020)*, pp. 742–749, Jaipur, India, 2020.

[22] K. M. Maddimsetti Srinivas and G. L. P. Ashok, "Spoken English digit classification using supervised learning," *International Journal of Research in Signal Processing, Computing & Communication System Design*, vol. 5, pp. 49–53, 2019.

[23] S. Ahamed, G. Weiler, K. Boden et al., "Deep neural network driven speech classification for relevance detection in automatic medical documentation," *Studies in Health Technology and Informatics*, vol. 281, pp. 63–67, 2021.

[24] H. A. Sánchez-Hevia, R. Gil-Pita, M. Utrilla-Manso, and M. Rosa-Zurera, "Age group classification and gender recognition from speech with temporal convolutional neural networks," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 3535–3552, 2022.

[25] P. M. Jena and S. R. Nayak, "Angular symmetric Axis constellation model for off-line Odia handwritten characters recognition," *International Journal of Advances in Applied Sciences*, vol. 7, no. 3, pp. 265–272, 2018.

[26] N. Sharma, S. Gupta, P. Mehta et al., "Offline signature verification using deep neural network with application to computer vision," *Journal of Electronic Imaging*, vol. 31, no. 4, Article ID 041210, 2022.

[27] A. Sethy, P. K. Patra, and S. R. Nayak, "A hybrid system for handwritten character recognition with high robustness," *Traitement du Signal*, vol. 39, no. 2, pp. 567–576, 2022.

[28] J. Oruh and S. Viriri, "Spectral analysis for automatic speech recognition and enhancement," in *Machine Learning for Networking. MLN 2020. Lecture Notes in Computer Science*, É. Renault, S. Boumerdassi, and P. Mühlethaler, Eds., vol. 12629, New York, NY, USA, Springer Cham, 2021.

[29] S. Krishnan, "advanced analysis of biomedical signals," in *Biomedical Signal Analysis for Connected Healthcare*, S. Krishnan, Ed., Academic Press, Cambridge, MA, USA, 2021.

[30] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*-Springer, New York, NY, USA, 2010.

[31] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[32] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, 2011.

[33] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, Coursera: Neural Networks for Machine Learning," Technical Report, University of Toronto, Toronto, Canada, 2012.

[34] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] T.-H. Lee, A. Ullah, and R. Wang, "Bootstrap aggregating and random forest," in *Macroeconomic Forecasting in the Era of Big Data*Springer, New York, NY, USA, 2020.

[36] M. Savargiv, B. Masoumi, and M. R. Keyvanpour, "A new random forest algorithm based on learning automata," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 5572781, 19 pages, 2021.

[37] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[38] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[39] M. M. Saleem, *Deep Learning for Speech Classification and Speaker Recognition*, The University of Texas at Dallas, Richardson, TX, USA, 2014.

[40] A. Zell, Simulation neuronaler netze, vol. 1, Addison-WesleyBonn, Boston, MA, USA, 1994.

[41] Pannous.Github, "Pannous/TensorFlow-speech-recognition," 2014, http://github.com/pannous/tensorflow-speech-recognition.

[42] B. McFee, M. McVicar, C. Raffel et al., "Librosa: v0.4.0.Zenodo," 2015, https://zenodo.org/record/18369#.YxurLj1BzIU.

[43] S. Afaq and S. Rao, "Significance of epochs on training a neural network," *International Journal of Scientific and Technology Research*, vol. 19, pp. 485–488, 2020.