Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.032491
Article

Tech Science Press

Implementation of Hybrid Deep Reinforcement Learning Technique for Speech Signal Classification

R. Gayathri[1,*] and K. Sheela Sobana Rani[2]

[1]Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore, 641020, India
[2]Department of Electrical and Electronics Engineering, Sri Ramakrishna Institute of Technology, Coimbatore, 641020, India
*Corresponding Author: R. Gayathri. Email: visitgayathri@gmail.com

Received: 19 May 2022; Accepted: 27 August 2022

Abstract: Classification of speech signals is a vital part of speech signal processing systems. With the advent of speech coding and synthesis, the classification of the speech signal is made accurate and faster. Conventional methods are considered inaccurate due to the uncertainty and diversity of speech signals in the case of real speech signal classification. In this paper, we use efficient speech signal classification using a series of neural network classifiers with reinforcement learning operations. Prior classification of speech signals, the study extracts the essential features from the speech signal using Cepstral Analysis. The features are extracted by converting the speech waveform to a parametric representation to obtain a relatively minimized data rate. Hence to improve the precision of classification, Generative Adversarial Networks are used and it tends to classify the speech signal after the extraction of features from the speech signal using the cepstral coefficient. The classifiers are trained with these features initially and the best classifier is chosen to perform the task of classification on new datasets. The validation of testing sets is evaluated using RL that provides feedback to Classifiers. Finally, at the user interface, the signals are played by decoding the signal after being retrieved from the classifier back based on the input query. The results are evaluated in the form of accuracy, recall, precision, f-measure, and error rate, where generative adversarial network attains an increased accuracy rate than other methods: Multi-Layer Perceptron, Recurrent Neural Networks, Deep belief Networks, and Convolutional Neural Networks.

Keywords: Neural network (NN); reinforcement learning (RL); cepstral coefficient; speech signal classification

1 Introduction

Speech recognition belongs to the class of speech processing, where the speech from individuals is recognized and translated using specific methodologies in Table 1 [1,2]. These approaches usually separate any spoken word and add a series of processing steps to obtain features that will be mapped to a particular word [3–6].

Table 1: Existing speech signal methodologies

	Shape of filter	Type of filter	Speed of computation	Reliability
Mel frequency cepstral coefficient (MFCC)	Triangular	Mel	High	High
Linear prediction cepstral coefficient (LPCC)	Linear	Linear prediction	Medium	Medium
Line spectral frequencies (LSF)	Linear	Linear prediction	Medium	Medium
Discrete wavelet transform (DWT)	—	Lowpass & highpass	High	Medium
Perceptual linear prediction (PLP)	Trapezoidal	Bark	Medium	Medium
Linear prediction coefficient (LPC)	Linear	Linear prediction	High	High

The Artificial Neural Network (ANN) is a major advancement in machine learning, which made an improving Human-machine-interface as in [7–10], through a mixture of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). Levenberg-Marquardt (LM) algorithm [11,12] is an algorithm for training the CNN and ANN. In comparison with the LM algorithm, there exist no standard procedures or algorithms to train the other Deep Neural Networks (DNN). ANN is the biochemical mechanism that enables the intensity of neuronal interactions to be changed according to the relative time of the production and spiking behavior of a single neuron. This is a training approach-based network specifically for unsupervised learning [13]. It has been deployed on many simulation and hardware platforms, including SpiNNaker. For static input signals like images, STDP has proven very effective and robust [14–21]. But in processing time-specific signals like audio samples, it is harder to enforce.

The main contribution of the paper involves the following:

- The study classifies the speech signal using a series of Generative Adversarial Networks (GAN) with feedback obtained from reinforcement learning.
- The study uses Cepstral Analysis to extract the features before the classification of speech signal classification and the extracted features obtained from the speech waveform provides a parametric representation at a relatively minimized data rate for optimal classification.
- The study uses Generative Adversarial Networks (GAN) to validate the speech signal classification and compares it with other existing deep learning classifiers namely: Deep Belief Network (DBN), Multi-Layer Perceptron (MLP) Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN).

The outline of the paper is given below: Section 2 provides the related works. Section 3 discusses the details of the proposed classification engine. Section 4 evaluates the entire work and Section 5 concludes the work with the possible direction of future scope.

2 Proposed Method

In this section, we proposed a speech classification using a series of hybrid neural network classifiers with reinforcement learning. The feature extraction uses Cepstral Analysis to extract the features prior to speech signal classification and the extracted features provide parametric representation at a relatively minimized data rate. The classification is conducted using various classifiers including GAN to validate the speech signal classification and finds the optimal classifier. Fig. 1. shows the architecture of Speech Signal Classification

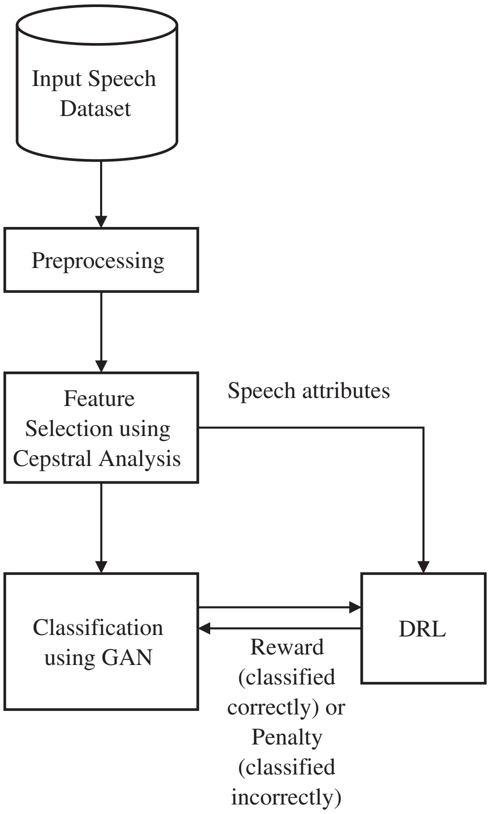
Input Speech Dataset
Preprocessing
Feature Selection using Cepstral Analysis
Speech attributes
Classification using GAN
DRL
Reward (classified correctly) or Penalty (classified incorrectly)

Figure 1: Architecture of speech signal classification

2.1 Pre-processing

Pre-processing of speech processing is performed to improve the accuracy of speech classification algorithms. This involves resampling, amplifying, and framing, and the audio recordings are sampled at 16 kHz for this analysis. The pitch of the audio recordings is normalized such that the signal spectrum of dynamics lies between −1.0 and +1.0 regardless of changes in voice intensity and microphone distance. The normalization of amplitude is reached after splitting the voice data using the absolute maximum magnitude. Speech signals are required to be analyzed in a short interval since they belong to the class of non-stationary signals. Framing is the method of splitting a signal into short frames. The resulting amplitude sampled under a 16-kHz scale is split into 256 sampled frames with 80% overlap. Thus, the process of overlapping in short frames enhances the process of classification. Formants are resonance bands within the speaking signal frequency range. The resonance bands reflect the signal significantly. In this proposed methodology, the formant extraction algorithm is carried out using Linear Prediction Coding (LPC). LPC provides a smooth estimation of the power spectrum that yields the results of the extracted features as given in Table 2. Here, 19 different samples are used for the extraction of the features.

The formant extraction is dependent on signal propagation within the frequency field. The locations of the formants are selected to balance this energy distribution. These formants are prominent bandwidth frequencies of less than 400 Hz within the spectrum. Therefore, the formants are high-energy bands with less concentration of about 400 Hz in their bandwidth.

Table 2: Result of LPC features extraction

Samples	Pitch	Intonation	Vocal cord sounds	Speech flow	Overtone intensity	Loudness
1	−1.804234	−0.394907	1.894737	1.125459	−2.343729	−2.477102
2	−1.793814	−0.574412	2.087647	1.553048	−2.712342	−3.430474
3	−2.177504	0.098012	2.940340	−0.258725	−4.875402	0.507018
4	−2.660264	1.275146	2.897111	−2.713803	−3.781905	5.737069
5	−2.310652	0.772377	2.548696	−1.894027	−3.403976	4.132140
6	−2.575752	1.105820	2.888588	−2.403643	−4.253682	5.697284
7	−2.125569	0.582516	1.830286	−0.993611	−1.831305	1.707261
8	−2.291764	0.779210	1.995675	−0.839144	−2.663706	1.497907
9	−2.451686	0.630887	3.288038	−1.490774	−5.570396	3.437595
10	−1.980194	−0.023056	2.231342	0.426740	−3.430995	−0.964538
11	−1.552383	−0.317433	1.057956	0.684869	−0.943985	−0.838674
12	−2.582596	1.125016	3.210137	−2.936713	−4.786181	6.975143
13	−2.563647	1.389807	2.654395	−3.259019	−2.910725	6.764225
14	−2.589318	1.425910	2.763993	−3.568587	−2.880212	7.461988
15	−2.401283	1.431683	1.542595	−2.340624	−0.714159	3.447326
16	−2.665586	1.608133	2.693060	−3.784050	−2.747432	8.011799
17	−2.414849	0.984690	2.755595	−2.440674	−3.603707	5.203648
18	−1.426366	−0.577824	1.323335	1.101894	−1.439636	−2.038922
19	−2.398167	0.567584	3.121445	−1.074078	−5.478509	2.353431

The coefficients obtained from LPCs are translated into polar form. The coefficient phases are derived from the spectrum as resonant bands with bandwidths below 400 Hz and a positive phase. The formants are called these constructive processes. The centroid formants are regarded as the weighted averages in the short frequency range of formants in each frame. The center formant is an indicator of how the strength of an audio signal is centralized in the frequency spectrum. For instance, the centroid formant is situated in the HF range if the remainder of the spectral resides lies in the high-frequency range. However, the centroid formants are located at a low-frequency range, if the bulk of power remains in low-frequency components.

2.2 Feature Extraction

The most widely used technique used to obtain spectral characteristics is Cepstral Coefficients (CC). CCs used to detect speech are built on a Mel Scale frequency domain that is based on the human ear and is one of the most widely known strategies for extracting features. The features considered for the study include pitch, intonation, vocal cord sounds (voiced and unvoiced), speech flow, overtone intensity, loudness, and breaks between the speeches. CCs are considered to be frequency domain features, which are much more reliable than time domain features.

The first step on the input signal is to determine the cepstral coefficient. The power spectrum is obtained using Eq. (1).

$$mel = 2595\log_{10}\left(1+\frac{x}{100}\right) \tag{1}$$

where,

x–filter bank input, and

mel–mel filter bank output.

It is important to convert at the Mel Log bank using Discrete Cosine Transform. Finally, it is called the Cepstral Coefficient that the log Mel spectrum returns into the period. By using this cepstral coefficient, the spectral properties of the signal are well reflected.

Pre-Emphasis: The sample rate of the audio signals is 16 kHz. Each word is stored in an audio file in its own right. The pre-emphasis of speech signal for signal energy at high frequencies is included in this process. The Filter discrepancy in the pre-emphasis filter is shown in Eq. (2).

$$H(z) = \frac{B(z)}{A(z)} = \frac{(b_0 + b_1 z^{-1})}{1} = 1 - 0.97z^{-1} \tag{2}$$

Framing and Windowing**:** The signal is dynamic in nature and used for stationary framing. Framing is the next stage after pre-empting: this signal is divided into smaller, overlapping frames. Windowing is used after framing to eliminate discontinuities at panel edges. The windowing system used in this study is the hamming window. The Hamming Window is given as below:

$$w(n) = \begin{cases} 0.54{-}0.46\cos\left[\frac{2\pi n}{N-1}\right] & 0 \le n \le N-1 \\ 0 & otherwise \end{cases} \tag{3}$$

where,

N-total samples present in a frame.

Fast Fourier Transform (FFT)**:** FFT estimates the discrete Fourier transform (DFT) of the speech signal. The results which transform the speech signal into its relevant frequency domain and the estimations of FFT are defined as below:

$$x[k] = \sum_{n=0}^{N-1} x(n)e^{-j2\frac{\pi}{N}kn} \tag{4}$$

where, N is the size of FFT.

Mel Filter Bank: Mel Filter Bank: The Mel filter bank transforms the frequency domain signal from Hertz to Mel Scale and the spectral power is hence converted into mel scale with triangular-shaped filter banks of overlapping ones.

Discrete Cosine Transform (DCT)**:** The DCT is applied after considering the logarithm of the Mel-filter bank output.

Delta Energy**:** The DE considers the base of 10 of the previous stage DCT output. The calculation of energy is important because the human ear response at the signal level of the acoustic speech is not linear, and human ears are not very sensitive to amplitude differences at higher amplitudes. The benefit of the logarithmic function is that the action of the human ear is usually duplicated. The estimation of energy using Eq. (5) is given as below:

$$E = \sum_{t=t1}^{t=t2} x^2(t) \tag{5}$$

The above equation tends to provide cepstral coefficients.

Classification

This section uses GAN to classify the instances obtained from feature extraction.

GAN is an alternative model of maximum likelihood technique that behaves as an unsupervised model with two neural networks acting in contrast with each other. One acts as a generator and the other act as a discriminator. The former generates the classes from features and the latter checks the correctness of the classes obtained. The process repeats until the generator output produces error-less outputs, which it is expressed as below:

$$\mathrm{min}_{\mathrm{G}}\mathrm{max}_{\mathrm{D}} V(D,\ G) = \mathrm{E(x)[log(D(x))]} + \mathrm{E(z)[log(1} - \mathrm{D(x))]}$$

2.3 Feedback from Reinforcement Learning

Reinforcement learning is considered as the process of collection of agents to assess the actions of the classifier and its error analysis for reward or penalizing the classifier's decision actions.

Algorithm 1 provides the details of how the classifier is rewarded or penalized.

Algorithm 1: DRL algorithm

```
Input:
$\lambda$ is the decay term, $\alpha$ is the learning rate, $n$ is the number of objectives, $\gamma$ is the discounting term, $a$ is the action, $r$ is the reward, $p$ is the penalty, $s$ is the state, $o$ is the observer
Initialize Population
For $s$, $a$, $o$ do
        Initialize $Q(s, a, o)$
End for
Evaluate Population
For each iteration do
    For $s$, $a$ do
            Find the error $e(s, a) = 0$
    End for
    Observe the initial state $s_t$
    Select $a$ based on exploratory policy of $Q$ $(s_t))$
    For each $a$ do
            Execute $a_t$, find $s'$and $r$ or $s$
            Select $a*$ using greedy policy of $Q(s')$
            Select $a'$using an exploratory policy of $Q(s')$
            For $o$ do
                    $\delta_o = r_o + \gamma\ Q(s_0, a*, o) - Q(s_t, a_t, o)$
```

(Continued)

Algorithm 1 (Continued)

```
        End for
        Set $e(s_t, a_t) = 1$
        For $s$, $a$ do
            For $o$ do
                set $Q(s, a, o) = Q(s, a, o) + \alpha\delta_o e(s, a)$
            End for
            If $a' = a^*$ then
                set $e(s, a) = \gamma\lambda e(s, a)$
            Else
                set $e(s, a) = 0$
            End if
        End for
        $s_t = s'$, $a_t = a'$
End for
```

The output of reinforcement learning is sent to the classifier that determines whether the classifier has correctly or incorrectly identified the speech instances on each dataset. This is carried out to reduce the classification error, where the classifier tries to reduce its error rate while classifying the speech instances.

3 Results and Discussions

This section verifies the accuracy levels of various classifiers on speech signal datasets. The experiments are conducted on various performance metrics that include accuracy, precision, recall, F-measure, and MSE, and the formula for estimating the metrics is given below:

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}} \tag{6}$$

$$\boldsymbol{F - measure = \frac{2TP}{2TP + FP + FN}} \tag{7}$$

$$\boldsymbol{Precision = \frac{TP}{TP + FN}} \tag{8}$$

$$\boldsymbol{Recall = \frac{TN}{TN + FP}} \tag{9}$$

where:

TP-true positive tweets

TN-true negative tweets

FP-false positive tweets

FN-false negative tweets

Analysis

This section provides the results of various classifiers (DBN, MLP, RNN, CNN, and proposed GAN) in terms of five different performance metrics over 9 different datasets that include RAVDESS, TED-LIUM corpus, Google Audio set, LibriSpeech ASR Corpus, CSS10, BACKBONE Pedagogic Corpus of Video-Recorded Interviews, Arabic Speech Corpus, Nijmegen Corpus of Casual French and Free-Spoken Digit Dataset. The first four datasets belong to General Voice Recognition Datasets and the remaining datasets belong to Multilingual Speech Data. The study uses 80% of the datasets for training and reaming 20% of the datasets for testing with 5-fold cross-validation. The total number of speech samples collected in each database is given in Table 1 over various datasets. Further, brief overviews of various existing classifiers are given below:

3.1 MLP

MLP is a feedforward neural network that helps in the classification of speech signals using the features extracted. The MLP has a single input, multi-hidden, and output layer. The architecture of MLP for classification is given below:

$$Y_j = \sum_{i=1}^{n} w_{ij}x_i + \theta_j \tag{10}$$

where

y_j is the parameter moved to subsequent layer

n is the amount of moving edges to node j,

x_i is the input

θ_j is the bias node.

3.2 DBN

The RBM is a building block that offers multi-layer learning and this is formed from the stacks of restricted Boltzmann machines. The restricted Boltzmann machine is a two-level model with visible layer units and hidden layers. DBN comprises multiple layered hidden units with suitable interconnections between them. In the case of classifying the speech signal, the connection links are not made between the units of each layer.

3.3 RNN

RNN uses Elman architecture, where it uses output via hidden unit layers and it is expressed as below:

$$h_t = \sigma_h(w_h x_t + u_h h_{t-1} + b_h) \tag{11}$$

$$y_t = \sigma_y(w x_t + b_y) \tag{12}$$

where

x is considered as the input vector,

h is considered as the hidden layer vectors,

y is considered as the output vectors,

b is considered as the bias vector and

w and u are considered as the weight matrices.

The process is conducted in a loop manner, which allows the data to pass in one step.

3.4 CNN

The architecture of CNN for classification is a three-layered architecture that consists of Convolution, max-pooling, and classification. The first two form the lower and middle leveled network. The max-pooling is the odd-numbered layers and convolutional layers are regarded as the even-numbered layers. For classification, the study includes convolutional and maximum pooling layers, which are considered feature mapping. Table 3 shows the combination of two or more layers in each plan enables faster computation of classes from the features of the speech signal.

Table 3: Attributes of tweet classification

Datasets	Type	Speech samples
RAVDESS	General voice recognition data	24 (12 male/12 female)
TED-LIUM corpus	General voice recognition data	2351 speech samples with 452 h of audio
Google audioset	General voice recognition data	635 audio classes and 2 million short clippings
LibriSpeech ASR Corpus	General voice recognition data	1000 h of speech
CSS10	Multilingual speech data	Samples from 10 language
BACKBONE	Multilingual speech data	Samples from 6 language
Arabic speech corpus	Multilingual speech data	Modern Standard Arabic speech (3.7 h)
Nijmegen corpus of casual French	Multilingual speech data	35 h of speech (46 French speakers)
Free spoken digit dataset	Multilingual speech data	Trimmed speech samples

Table 4 shows the results of Classifiers on speech signals with RAVDESS. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 4: Results of classifiers on speech signal with RAVDESS

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	54.98	57.07	57.33	58.69	79.48
F-measure	39.50	50.74	50.90	53.27	82.66
MSE	24.40	23.00	20.42	19.84	15.14
Precision	64.26	72.17	84.55	85.21	95.25
Recall	73.38	76.89	76.91	78.28	79.12

Table 5 shows the results of Classifiers on speech signals with TED-LIUM corpus. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 5: Results of classifiers on speech signal with TED-LIUM corpus

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	57.88	60.24	61.61	64.85	83.69
F-measure	65.80	66.69	67.81	72.85	78.41
MSE	15.72	15.63	10.77	9.44	8.29
Precision	77.92	78.01	82.85	84.19	85.34
Recall	75.28	76.22	79.38	81.38	83.47

Table 6 shows the results of Classifiers on speech signal with Google Audio set. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 6: Results of classifiers on speech signal with Google audio set

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	64.86	67.88	73.10	76.94	81.42
F-measure	68.95	69.12	69.30	73.86	79.40
MSE	63.48	56.76	38.64	35.78	33.93
Precision	70.16	70.88	72.65	72.85	79.70
Recall	71.29	74.36	79.59	80.90	81.31

Table 7 shows the results of Classifiers on speech signals with LibriSpeech ASR Corpus. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 7: Results of classifiers on speech signal with LibriSpeech ASR Corpus

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	96.18	96.21	96.29	96.30	96.43
F-measure	68.73	69.06	71.94	75.17	78.37
MSE	24.53	21.76	19.09	9.62	89.12
Precision	69.10	71.87	74.55	84.00	87.60
Recall	96.32	96.52	96.60	96.62	96.68

Table 8 shows the results of Classifiers on speech signal with CSS10. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 8: Results of classifiers on speech signal with CSS10

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	96.75	96.77	96.77	96.78	96.78
F-measure	89.63	89.78	90.29	90.50	91.14
MSE	26.03	19.27	8.29	53.64	20.02
Precision	94.58	95.68	96.26	96.54	96.55
Recall	95.78	95.78	95.78	95.78	96.42

Table 9 shows the results of Classifiers on speech signals with backbone Pedagogic Corpus of Video-Recorded Interviews. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 9: Results of classifiers on speech signals with BACKBONE pedagogic corpus of video-recorded interviews

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	93.12	93.20	93.25	93.43	93.44
F-measure	59.05	59.39	59.85	61.38	61.52
MSE	28.19	27.30	26.95	25.12	24.30
Precision	65.43	66.32	66.68	68.50	69.34
Recall	94.41	94.42	94.47	94.51	94.56

Table 10 shows the results of Classifiers on speechs signal with Arabic Speech Corpus. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 10: Results of classifiers on speech signal with Arabic speech corpus

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	94.93	94.94	95.03	95.05	95.11
F-measure	76.52	77.04	78.12	78.80	79.09
MSE	29.80	29.28	27.67	27.17	26.85
Precision	63.82	64.34	65.95	66.45	66.77
Recall	93.78	93.82	95.05	95.46	95.81

Table 11 shows the results of Classifiers on speech signals with Nijmegen Corpus of Casual French. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features.

Table 11: Results of classifiers on speech signal with Nijmegen corpus of casual French

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	96.37	96.45	96.45	96.47	96.52
F-measure	85.01	86.95	86.98	88.34	88.35
MSE	69.63	61.73	60.40	53.05	52.37
Precision	89.53	90.34	90.47	91.21	91.28
Recall	96.47	96.56	96.56	96.65	96.65

Table 12 shows the results of Classifiers on speech signal with Free Spoken Digit Dataset. The results of various performance metrics show an improved performance by GAN than other classifiers with cepstral coefficient extraction of features. Fig. 2 shows the accuracy (Training/Testing) over various datasets. Fig. 3 shows the Confusion Matrix.

Table 12: Results of classifiers on speech signal with free spoken digit dataset

Metrics	DBN	MLP	RNN	CNN	GAN
Accuracy	96.44	96.52	96.52	96.54	96.59
F-measure	85.07	87.01	87.04	88.40	88.42
MSE	69.68	61.78	60.45	53.09	52.41
Precision	89.60	90.41	90.54	91.28	91.35
Recall	96.54	96.63	96.63	96.72	96.72

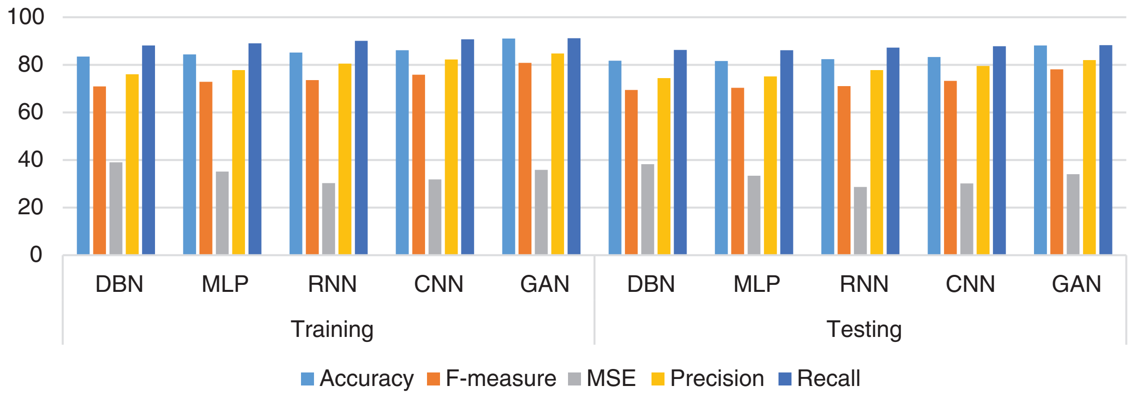

Figure 2: Accuracy (training/testing) over various datasets

True Class	0	77%	0.23%
	1	0.1%	0.99%
		0	1
Predicted Class			

Figure 3: Confusion matrix

4 Conclusions

In this paper, the Classification of the speech signal is conducted with different DNN classifiers m where reinforcement learning checks the accuracy of classification. The extraction of essential features using cepstral coefficient analysis enables accurate classification of instances by the neural network classifiers. The use of reinforcement learning as a feedback mechanism helps in correcting the errors made by a classifier. The use of different classifiers including MLP, DBN, RNN, GAN, and CNN offer improved results in terms of reduced errors. As a result of which, the deep learning classifier namely RNN, CNN, and GAN achieve reduced penalties than DBN and MLP. The results of classification further show that the proposed speech signal classification engine offers higher classification accuracy with GAN than CNN, RNN, and the other two machine learning classifiers. The use of the cepstral coefficient also has improved the performance of the classification engine due to the proper extraction of features from the pre-processed speech signal. The other performance metrics show that the GAN obtains improved precision, recall, f-measure, and MSE. In the future, the ensemble of all these classifiers can be used as a multi-modal speech signal classification engine over a large speech signal dataset.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Kannan, G. Dhiman, A. Sharma, S. N. Mohanty, M. Soni *et al.,* “Ubiquitous vehicular ad-hoc network computing using deep neural network with IoT-based bat agents for traffic management,” *Electronics*, vol. 10, no. 7, pp. 785, 2021.

[2] T. Stafylakis, M. H. Khan and G. Tzimiropoulos, “Pushing the boundaries of audiovisual word recognition using residual networks and LSTMs,” *Computer Vision and Image Understanding*, vol. 176, pp. 22–32, 2018.

[3] K. I. Iso and T. Watanabe, “Speaker-independent word recognition using a neural prediction model,” in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Albuquerque, NM, USA, pp. 441–444, 1990.

[4] S. Furui, “Speaker-independent isolated word recognition using dynamic features of speech spectrum,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[5] K. Srihari, G. Dhiman, K. Somasundaram, A. Sharma, S. Rajeskannan *et al.,* “Nature-inspired-based approach for automated cyberbullying classification on multimedia social networking,” *Mathematical Problems in Engineering*, vol. 2021, no. 6644652, pp. 12, 2021.

[6] P. McGuire, J. Fritsch, J. J. Steil, F. Rothling, G. A. Fink *et al.,* “Multi-modal human-machine communication for instructing robot grasping tasks,” in *Proc. IEEE/RSJ Int. Conf. of Intelligent Robots and Systems, 2002*, Lausanne, Switzerland, vol. 2, pp. 1082–1088, 2002.

[7] O. Russakovsky, L. J. Li and L. Fei-Fei, “Best of both worlds: Human machine collaboration for object annotation,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 2121–2131, 2015.

[8] R. Collobert, C. Puhrsch and G. Synnaeve, “Wav2letter: An endto-end convnet-based speech recognition system,” in *Proc. IEEE Conf. on Machine Learning*, New York, USA, pp. 1–12, 2016.

[9] A. V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals *et al.,* “Wavenet: A generative model for raw audio,” in *Proc. IEEE Conf. on Machine Learning*, New York, USA, pp. 34–39, 2016.

[10] M. T. Hagan and M. B. Menhaj, "Training feed forward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[11] P. U. Diehl and M. Cook, "Efficient implementation of STDP rules on SpiNNakerneuromorphic hardware," in *Proc. Int. Conf. on Neural Networks*, Beijing, China, pp. 4288–4295, 2014.

[12] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe and T. Masquelier, "STDP-based spiking deep neural networks for object recognition," in *Proc. IEEE Conf. on Machine Learning*, New York, USA, vol. 3, 2016.

[13] J. P. D. Morales, A. J. Fernandez, A. R. Navarro, E. C. Escudero, D. G. Galan *et al.,* "Multilayer spiking neural network for audio samples classification using SpiNNaker," in *Proc. Int. Conf. on Artificial Neural Networks*, Barcelona, Spain, pp. 6–9, 2016.

[14] Z. Hu, T. Wang and X. Hu, "An STDP-based supervised learning algorithm for spiking neural networks," in *Proc. Int. Conf. on Neural Information Processing*, Guangzhou, China, pp. 92–100, 2017.

[15] T. Moraitis, A. Sebastian, I. Boybat, M. L. Gallo, T. Tuma *et al.,* "Fatiguing STDP: Learning from spike-timing codes in the presence of rate codes," in *Proc. 2017 Int. Joint Conf. on Neural Networks (IJCNN)*, Anchorage, AK, USA, pp. 1823–1830, 2017.

[16] W. Sun, G. C. Zhang, X. R. Zhang, X. Zhang and N. N. Ge, "Fine-grained vehicle type classification using lightweight convolutional neural network with feature optimization and joint learning strategy," *Multimedia Tools and Applications*, vol. 80, no. 20, pp. 30803–30816, 2021.

[17] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.,* "A Multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3560, 2021.

[18] H. Sun and R. Grishman, "Lexicalized dependency paths based supervised learning for relation extraction," *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 861–870, 2022.

[19] C. R. Rathish and A. Rajaram, "Efficient path reassessment based on node probability in wireless sensor network," *International Journal of Control Theory and Applications*, vol. 34, pp. 817–832, 2016.

[20] C. R. Rathish and A. Rajaram, "Sweeping inclusive connectivity based routing in wireless sensor networks," *ARPN Journal of Engineering and Applied Sciences*, vol. 3, no. 5. pp. 1752–1760, 2018.

[21] A. Rajaram and K. Sathiyaraj, "An improved optimization technique for energy harvesting system with grid connected power for green house management," *Journal of Electrical Engineering & Technology*, vol. 2022, pp. 1–13, 2022.