ORIGINAL ARTICLE

AUTOMATIC STUTTER SPEECH RECOGNITION AND CLASSIFICATION USING HYPER-HEURISTIC SEARCH ALGORITHM

Sri Harsha Grandhi [1,*], M M Kamruzzaman [2]

[1]Intel, Folsom, California, USA Email : sriharshagrandhi@ieee.org

[2]Department of Computer Science, College of Computer and Information Sciences,

Jouf University, Sakakah, Saudi Arabia Email : mmkamruzzaman@ju.edu.sa

*Corresponding Author: Sri Harsha Grandhi , Corresponding Author Email: sriharshagrandhi@ieee.org

(**Received** 26 Aug 2024, **Accepted** 13 Dec 2025, **Published** 18 Feb 2024)

DOI: https://doi.org/10.5875/dm761m36

Abstract:

The normal flow of speech is disrupted by the repetition or delay of sounds or phrases in stuttering or stammering, a speech impairment. Speech flow becomes difficult when someone stutters. Recent advances in deep learning have changed the speech domain, while stuttering recognition has received very little attention. The detection of stuttering is a fascinating area of research that encompasses pathology, psychology, and signal processing, making it challenging to detect. Stress, delays in early development, and other anomalies are the causes of stuttering. This too seems to be complicated and perplexing. To overcome these problems, a feature of automatic Stutter speech recognition and the Stutter Audio classification with Stutter audio by deep learning using Hyper-Heuristic Search Algorithm for Stutter Audio signal analysis is employed. The study performs the sound digitalization in which the Analog signals to a digital signal by using sampling and quantization, the Mel Spectrogram algorithm is employed in the Audio classification, the Automated driving to medical devices is done in the Audio deep learning, the Hyper-parameter tuning is performed in the Feature Optimization, the Automatic Speech Recognition is implemented, in which here we use the Weighted Finite-State Transducer framework, the Perceptual Linear Prediction(PLP) and Viterbi search, the Discrimination training by deep neural networks and we implement the Hyper-heuristic Search Algorithm. Using the suggested system flow, the results analyse Automatic Speech Recognition and reaction time with the Classification accuracies of Stutter Audio signal analysis. The model shows a predetermined order of heuristics that enhances the process of fixing stuttering issues. This study evaluates Automatic Speech Recognition (ASR) and Response Time with a higher accuracy rate and more efficacy.

Keywords: Heuristic Search, Machine Learning, Speech Recognition, Stutter Classification, Stuttering

1. INTRODUCTION

Speech recognition is a multileveled pattern recognition problem that involves breaking down auditory information into a hierarchy of phonemes, words, phrases, and sentences. Each level may include extra time limitations, such as recognised word pronunciations or legal word sequences, to make up for errors or uncertainty at lower levels. The most effective way to apply this hierarchy of restrictions is to combine probabilistically all lower level judgments and make discrete decisions. In this study, Automatic Stutter speech recognition and stutter Audio classification with stutter Audio deep learning using Hyper-Heuristic Search Algorithm for Stutter Audio signal analysis are performed. The technique implemented is Hyper-parameter tuning and Hyper-Heuristic Search Algorithm, and Mel Spectrogram algorithm under the platform of Stutter Speech analysis with deep learning. Selecting a set of ideal hyper parameters for a learning system is known as hyper parameter tuning. A model argument with a hyper parameter has a value that is predetermined before the learning process even starts. A hyper-heuristic is a type of heuristic search method that attempts to automate, frequently through the use of machine learning techniques, the process of choosing, combining, creating, or adapting a number of simpler heuristics (or components of such heuristics) to effectively solve computational search problems. A Mel spectrogram algorithm is a spectrogram where the frequencies are converted to the Mel scale and can be implemented in only a couple of lines of code. In this study, the proposed system is implemented with Sound digitalization with the analog signals to a digital signal by using sampling and quantization, Audio classification by Mel Spectrogram Algorithm, Audio deep learning by automated driving to medical devices, Feature optimization with the help of hyper-parameter tuning, Automatic Speech recognition by Weighted finite-state transducer framework, Perceptual Linear Prediction(PLP) and Viterbi search and discrimination training by deep neural networks, Hyper-heuristic Search Algorithm.

The study contributions are as follows.

i. Automatic stutter speech recognition and stutter Audio classification with Stutter Audio deep learning using, Hyper-heuristic algorithm for Stutter Audio signal analysis is performed.
ii. It is implemented to convert stuttering analogue signals to digital signals using a generic sampling technique and stuttering analogue signals to digital signals using a general quantization approach.
iii. The working principle for Mel Spectrogram algorithm stutter classification with the simulation of Audio deep learning for Stutter Audio signal analysis is analysed.
iv. The hyper-heuristic search algorithm for predictions and Hyper-parameter tuning for stutter audio signal optimization is analysed.

The remainder of this article is separated into the sections that follow. Section 1 outlines the work's Introduction. The associated work of the existing methodology is discussed in Chapter 2, the experiments directly are covered in Section 3, and the analysis of the outcomes is covered in Section 4. Section 5 concludes by recapping the findings and discussing the investigation's upcoming directions.

2. LITERATURE REVIEW

[1] The speech shuttering events is analysed by tracking a speech pathologist for overtime fluency. This helps in the improvement of the recognition of the speech system. The event of the shuttering has been stated in podcasts (SEP-28K). This has many numbers in the event, such as word and sound repetition, blocks, prolongations, and interjections for the stutter interviewing in the training data detected for the dataset. Here the Dysfluency detection system has been introduced in the detection performance. This is proposed by Lea et al. (2021, June). [2] Here the author proposed that the stuttering speech recognition has automatically corrected the acoustics effect by using the speech recognition algorithm to detect the features that make the internal and the external voice notes. The noise and the speech signal will be enhanced using filters and parameters. The Mean Square Error will increase the voice note's speed in the convolutional neural network. Vaidianathan et al. (2021) suggest the deep learning technique. [3] Here, the shuttering detection is identified for the various deep learning-based algorithms. This enables the automatic speech recognition algorithm for the stuttering detection of the ASR mode to make the acoustic signal for the time delay neural network for the training parameters in this sustainability for sharing the TDNN. In this paper, automatic speech is recognized by the neural network for the trainable parameters for the stuttering data set is done. This is said by Sheikh et al. 2021. [4] The speech deficiency that is stuttering is involved in the sound repetition and the intermediate of the pathology and the interdisciplinary of the complication which is done by the development of the attention of the sound in the stuttering sound which acts as the deep learning and the statistical-based on the acoustic features. In this paper, the classification method is used in the feature classification of the voice notes detected, and it is elaborated by Sheikh et al. 2021. [5] This paper related that the feature selection will be very effective for data mining to improve the classifiers' performance by removing irrelevant features. This paper uses some techniques and algorithms to detect voice stuttering to detect the Automatic Stutter Speech Recognition and Stutter Audio classification with Stutter Audio. That some of the techniques are Feature selection is used to find the factors that attack the percentage of the different components of the hyper-heuristic based feature selection paradigm is used in the different levels of approach for the evolution of the proposed framework for the numerous experiment of the UCI machine learning method which makes the better performance of the multi-dimensional dataset this is suggested by the Ibrahim et al. 2021. [6] The classification of the stuttering voice elaborates on the standard speech recognition system to produce high-resolution speech daily. The classification of the voice has been detected. Then the voice splitter delivers as per the noise. The unwanted data can be removed, the deep learning algorithm is used in the threshold, and the amplitude model in the neural network is made into the real world. This method will eliminate the repentance and the intervals for the speech proposed by Mishra et al. 2021. [7] The reduction of the dimension makes the speech and the noisy data for the multilayer perception in the previous experiments of the voice recognition process. Here the principal component analysis is made to initiate the sound notification in the network. This will support the distance between the recognition process for classifying the multilayer work. Here the Kohen network application is used in the dynamic infusion; this is said by Świetlicka et al. 2021.

3. QUANTIZATION OF THE SHUTTERED SPEECH USING SOUND DIGITIZATION

Shuttered speech is a type of disorder that is usually the children who have this disease is classified by prolonged word sound, syllables, and repetition. Some types of shuttered speeches are developmental, neurogenic, and psychogenic stuttering [21]. The voice can be quantified using a sample audio signal, and changing the audio to a video signal will be done. The transformation of the Analog domain to the digital domain and the process of changing the digital to the Analog is the conversion that happens during sound travel. This change happens at the frequency level, and this makes a tremendous impact on the shuttered speech.

Quantization refers to mapping the continuous set of inputs into a countable set of output values. For each zone in the map, the height of the signal is calculated as,

$$D= \frac{V_{max}-V_{min}}{l} \tag{1}$$

Where, V_{min}- is the value of minimum, V_{max}- is the value of the maximum, and l- Length of the signal, The actual amplitude of the signal is known as the top value of the signal. The pulse amplitude modulation of the speech signal is,

$$PAM = \frac{A}{D} \tag{2}$$

For each signal zone, the quantization error is between the quantization value and the normalization of pulse amplitude modulation. If a sample lies on the quantization level, then the Q_{value} of the signal is half of the spacing between levels.

Algorithm for quantization

Input parameter; V_{min}, V_{max}, sampled signal, T_v -Top value, A_{actual} -actual amplitude, Q_{error}- quantized error

For each sample map

For each zone

Calculate height D

$D= \frac{V_{max}-V_{min}}{l}$

End for

$T_v \leftarrow A_{actual}$

/* calculate pulse amplitude modulation

PAM $=\frac{A}{D}$

For each zone

Calculate Q_{error}

$Q_{error} = Q_{value} + PAM_{norm}$

End for

End for

Result: quantization level at the left of the map.

3.1. Analog to digital conversion:

The flow of the sample signal to the binary values for the frequency resolution is the central aspect of the analogy to digital conversion. The modulation among the frequency related to the digital conversion is done by the patient affected by the shuttered speech [17].

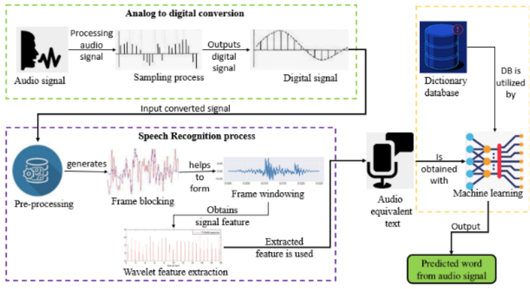

Figure. 1 Automatic Stutter Speech Recognition with deep learning

In this figure 1, the audio signal is processed to make the analog signal to the conversion of the digital conversion. This conversion is known as digital conversion. This conversion is given as the input to get the digital signal as an output [11]. This output needs the converted signal, which makes the data ns the signal pre-processed.

3.2. Speech recognition process:

While the pre-processing is happening, it generates the blocking of the frame to make the unwanted data from the voice signal [15]. This signal makes the shutter speech to hear this separation is done to frame the window for the voice signal this signal makes the wavelet extraction of the features in the voice signal this voice signal make the extraction of the feature that makes the text which is equal to the audio this audio base dictionary and this makes the machine learning process to make the voice conversion and then the clear voice will be delivered as the output as per the given shuttered voice signal [13].

Algorithm for Automatic Speech Recognition for WFST framework

Initialize; s=o, q=0 and E'=0 and T

While T≠0

$(b_1^n) \leftarrow$ head(T)

Dequeue(T)

If $(b_1^n) \neq \lambda$

If $(b_1^n) \notin$ q

q→ q ∪{(b_1^n)}

enqueue (T, (b_1^n))

end if

E'=E'∪{((b_1^n). λ -log β(b_1^n)(b_2^n))}

End if

For each w

If (w← beginning)

If (beginning ∉ q)

Q= q ∪{<state$_i$>}

f=initial state

End if

E'=E' ∪{((b_1^n). <state$_i$> log β <state$_i$> (b_1^n) <state$_i$>))}

Else If n<m-1

S_i= (b_2^n, w)

Else

s_i= (b_1^n, w)

End if

If (T≠ q)

q=q∪{ <state$_i$> }

Enqueue s_i

End if

Update E'

End if **End for End while**

H is denoted as a word sequence, and no word is represented as λ. Here, 0 indicates an initial state of WFST, and <state$_i$> means a beginning state of sentences. From the source state T, the identified sequence transition state is b_1^n And back off state isb_2^n. If n<k, the backoff state remains the state as b_2^n. In the calculation procedure of E,' the weight of the backoff coefficient is converted by a negative logarithmic function [16]. If w is the beginning of models, then each word follows the transition state as b_1^n. If w is not sentence end, then the E' is updated. If n is less than m-1, then the state S_i= (b_2^n, w) otherwise, the state is at s_i= (b_1^n, w) it means the length of the state returns m-1. Finally, the T tends to empty then the process is terminated. We can eliminate probabilities that cannot contribute to speech recognition accuracy [19].

3.3. Hyperparameter tuning in optimization of the features:

These brain images process the parameter for the data spoken by the person affected by the shuttered speech. This shuttered speech involves the formation of parameter tuning. This parameter tuning controls the values that make the central point of some of the learning processes. The frequency of the tuning in the voice recognition of the unwanted data has to be eliminated. Then the information is optimized in the features. The combination of the hyperparameter tuning makes the performance of the maximum parameter model, that is, the features of the brain images [20].

In this diagram, the hyperparameter tuning using deep learning is drawn. This involves the collection of voice data as input. This input has a voice that some person speaks. This voice input has to enter the pre-processing data section. This pre-processing data section proceeded with the feature selection; this feature selection found the voice data to optimize the voice in the data pre-processing. Then the data, which is the voice data that can deliver into the training process. This training process passes the testing and the training data to manage the progress of the voice data.

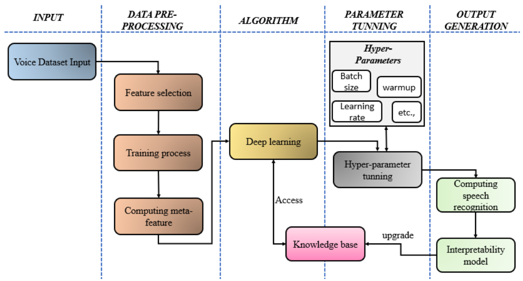

Figure. 2 Hyper-Parameter tunings using deep learning

In this figure 2, Initialize an empty hyperparameter space H_P, and fix the number of loop trials. The evaluation criteria for the Y* need to be minimized. The function is computed and gives the best solution of y. the hyperparameter id is updated based on the y and function of y. for every iteration, the m_T is updated, producing a better approximation. The Process terminates after the fixed numbers of loop trials are achieved.

Algorithm for Hyper-parameter tuning Optimization

Input parameters; hyperparameter space H_P, t, s, d, F

H_P←0

For (T→1-t)

Y*←Argmin s(Y, m_{T-1})

Evaluate the function of Y*

H_P← HU(Y*,F(Y*))

Fit a new model m_T to H

Return H

3.4. Method to access the hyper tuning in voice recognition:

Here the algorithm is used in the voice data travel in the processing of the system, and the deep learning method is used in the formation of the access of the voice data in the deep learning algorithm. This learning algorithm has been processed by the base knowledge for the exact formation of the knowledge of the voice signal transfer in the system to proceed. Then the parameter tuning will be processed. This tuning process will propagate the system requirements to follow the batch size [8]. The warmup learning rate has been got into the handling of the hyperparameter structure in the upgrade of the knowledge in the voice control to form the understanding of the transmission and the analysis for the voice transmission in the factors that affect the speech, which the model of the hyperparameter recognizes [9].

A. Parameter tuning:

This hyperparameter evolves the fraction of the system to proceed with the formation of the interpersonal factor of voice detection in deep learning processing. The interpretability model contains the fraction of the system to produce the model to make the deep learning from the voice control and the recognition of the voice data by using the voice-related signal by extracting the voice data.

Algorithm for hyper-heuristic algorithm

Input; iteration- I, training set- T_s, population size- P_s

Set; i=0, S (i) = {}, t=1, 2…nl

Initialize; q (i)

For i ε (1- I)

Evaluate S_w (i) = {y_1(i), y_2(i)…y_n(i)}

Update S_b (i)

Modify S_b (i)

Compute modified S_b (i)

Update P (i)= $\frac{\sum y_k^l(i)-y_k^l}{N}$

For (t=1, 2…n)

Update m (i) = fit (i)-worst (i)/ $\sum_{k=1}^{n} \text{fit(i)} - \text{worst(i)}$

Update q (i) = Δ fit (i)/$\sum \text{fit(i)}$

i= i+1

End for

T_r =S_{best}

End for

Return T_R

Train classifier by using T_R

The difference between the before and after using the local search is shown by the fit I to be more explicit, the local search is given equal probability in the first iteration of the suggested method. For every member of the population to use it. The population has improved the most with Fit I thanks to local search. The procedure is then repeated using the uploaded probabilities for local search selection in the following round. It should be noted that fit I avoids its deletion if fit I is less than 0.001 or the local search is not employed in an iteration.

3.5. Process of Mel spectrogram using audio signals:

The process of classifying the audio using Mel-spectrogram is a spectrum frequency that makes the range in the frequency of the visualization of the images managed in the voice data signal. This voice information is obtained from the person affected by the shuttered voice. The voice signal converts into the Mel scale [18].

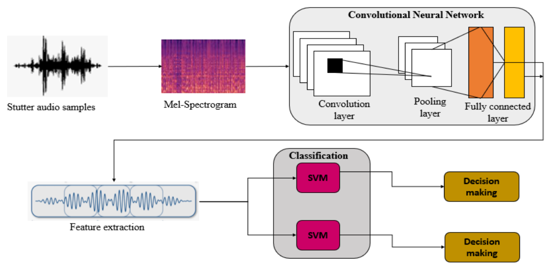

Figure. 3 Mel-spectrogram with CNN for classifying stutter audio

In this figure 3, the shuttered voice signal for the audio is sent to the Mel-Spectrogram to analyze the voice signal. This voice signal is carried to the CNN convolution neural network. This convolutional neural network makes the convolutional layer, pooling layer, and fully connected layer. These layers constantly analyze the voice signal as a classification of the voice. Also, the filtering and the interpretation of the voice signal are sent for feature extraction. This feature extraction splits the features in the voice signal [10]. This voice signal can clear all the noisy data. This noisy data is extracted, and then the classification of the original information is divided into the SVM. This division helps in the decision-making, and the output delivers the clear voice of the person affected by the shuttered voice. The deep learning technique can also hear this shutter voice using voice filtering. Here the SVM is the support vector machine, the voice is induced in this technique, and the learning models have been combined to classify the analysis in the regression. The voice will be transmitted by the voice; thus, the clear voice of the shuttered voice.

Before analyzing the speech signal, it is reduced to the determined data size. In this paper, the Mel Spectrogram method is used for stutter classification. Each speech sample is down-sampled to 16kHz and pre-emphasizes the sampled signals with a filter. The z transform of the filter is as follows,

$$h(z) = 1 - b * z^{-1} \tag{3}$$

b values from 0.9 to 1, for time-domain input and output relationship of pre, emphasized block of b value remains 0.97.for each pre-emphasize block has separated frame adjacency also has separated frame. The second frame is assigned when the first is fully created.

Algorithm for Mel Spectrogram algorithm for Stutter classification

Input parameter; 16khz sampled speech signal-x(t)

Step1; // pre-emphasized with filter

For b ε (0.9-1)

Compute z transform

$h(z) = 1 - b * z^{-1}$

T← t-(t-1)

$S'_t = S_t - bS_{t-1}$

End for

Step2; // process of windowing

For each frame

Calculate hamming window

$H_W(t) = 0.54 - 0.46 \cos(2\omega t/T-1)$

If $H_W(t)$ ε $(0 \leq t \leq T-1)$

Evaluate windowed signal

$Y(t) = x(t)\, H_W(t)$

End for

Step3; // FFT to H_W (t) and Mel scale filter

t← f

if (f< 1 kHz)

Perform logarithmic spacing

Else

Perform linear frequency

Calculate Mel frequency

Mel (f)= 2595 log(1+f/700)

Convert f→ t

If (tested y (t) = repetition speech signal) true

Classified under repetition stutter

Else If (tested y (t) = prolong ties speech signal) true

Classifiers under prolonged ties stutter

Else

Normal speech signal

End if

Each frame, the beginning, and the ending have higher and sudden fluctuation of frequency because of changes from zero to signal and signal to zero. The result of the windowed signal is the multiplication of the speech signal with the hamming window. The time domain is converted into the frequency domain. The samples are sampled to obtain the frequency resolution. The Mel scale is used to find the objective spectrum. The result of log Mel is converted into the time domain using this calculation, we can find repetitions and prolongation in stuttered speech.

a. Working process of Perceptual Linear Prediction (PLP)

The hearing of the all-pole model in the spectrum of the short time of speech is known as the perceptual linear prediction. Here the shuttered speech will analyze, and all the noisy data will be removed [14]. Then it will make the fluency of the audio signal for the continuous hearing of the audio.

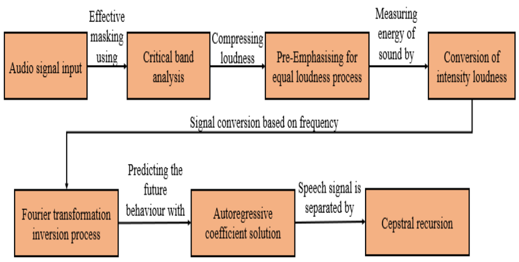

Figure. 4 Perceptual linear prediction of speech

In this figure 4, the input has been given as the audio signals. The audio used for the masking preparation in this masking is very clear other than the shuttered speech. This analyses the critical band; thus, the critical band can compress the voice signal's loudness. Here the loudness of the speech is managed by the sound energy. This sound energy convinces the inventory of the loudness in the signal conversion. This conversion of the audio signals transfers the transformation of the sound energy, the filtering, and the formulation of the noisy data, which is presented to produce the fixed decomposition of the autoregressive of the solution to make the speech signal. This speech signal separates because of the voice signal. Based on the frequency, the audio signal is analyzed by separating a clear voice other than the shuttered one. This voice signal then has the spectral reaction. This reaction will make the audio without the noisy data. This noisy data has some particular progression of the system that has to mention in the audio processing [12].

b. Weighted Finite-State Transducer framework

This transducer act as the label sequence of the finite state transducer. This makes the sequence an output label. This is also known as the Weight versing in the exemplary state transducer. This provides the natural representation of the grammars, dictionaries, and speech recognition output that make the functions that analyze the system's frequency to produce the systems.

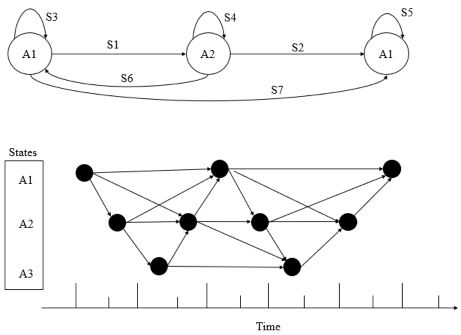

Figure 5. Weighted Finite-State Transducer framework and Viterbi search

In this figure 5, The weighted finite-state transducer is used to provide the representation of the natural phonetic models. This will represent the context of the problems in the machine. This problem has been identified, and then the node makes the interconnection of the system to form the intervention of the system to follow the exact portions of the system [13].

The Viterbi algorithm is used for finding the sequence of optimization states. It is commonly used in the speech recognition process. The input signal sequence is x={x (1),x(2)…x(n)} . Each v and u stores the uncompleted value of the particular sequence of points. The result of the tag is stored as BP. The actual value of the whole signal is not known until the whole process is completed. For each iteration, the optimal solution of the Viterbi algorithm is calculated using the Bp and returned to the optimal sequence.

Algorithm for viterbi search

Input; signal x={x(1),x(2)…x(n)}

Initialization; set Ꞷ(0,*,*)=1; Ꞷ(0,b,a)=0

For each b, a

b ≠λ, a ≠λ

For kε(1-N)

For (b, a ε i)

Ꞷ=max(Ꞷ(i-1,w,b)*p(a(c,b)*p(x(i)|a))

Bp= arg max (Ꞷ(i-1,c,b)*p(a(c,b)*p(x(i)|a))

End for

End for

End for

For (i ε (n-1)-1)

Y_i= Bp (i+2, y(i+1), y(i+2))

End for

Return, sequence y(1)…y(2)

Here the Viterbi search is the algorithm used to make the probability of the hidden states. The Markov source of information analyzes this hidden state. This transmission of the nodes is interconnected to form the exact conclusion in the nodes to transmit the signal, which is both the voice and the video signals. This signal will make the transmission of the signals into the hidden nodes.

Sampling means converting a continuous-time signal into a discrete-time signal. Speech signal was obtained from the process of finding the two vectors, namely, X=[x (1),x(2…x(n+1)] and the time-domain vector, initially, there were no changes in the speech sample return, this sample as the best sample. The current sample is down-sampled to attain the desired frequency level of speech samples. Rapid changes in the speech signal are neglected.

Algorithm1; sampling algorithm

Input parameter; X=[x (1),x(2…x(n+1)]

Initialize; N=0, j=1;

// sampling the signal X (t)

While (N<=n)

While (s_i<= t+ NΔs)

Temporary=y_j

j← j+1

End while

N=N+1

x (n) = Temporary

End while

4. RESULT AND ANALYSIS

The proposed system is Sound digitalization, Audio classification, Audio deep learning, Feature Optimization, Automatic Speech Recognition, and Hyper-heuristic Search algorithms. Hyper-parameter tuning, Hyper-heuristic Search Algorithm, and Mel Spectrogram algorithm are used for the proposed system. These factors' approximate result is 76.8% efficient.

Table 1. Analog signals to a digital signal by using sampling and quantization

Signal level	Sampled signal	Quantized signal	Voltage	Distance	Speed	Time
1	2.3	3.6	4.5	2	0.5	4
2	6.7	7.2	9.2	4	4.78	0.83682
3	11.1	10.8	13.9	6	9.06	0.662252
4	15.5	14.4	18.6	8	13.34	0.5997
5	19.9	18	23.3	10	17.62	0.567537
6	24.3	21.6	28	12	21.9	0.547945
7	28.7	25.2	32.7	14	26.18	0.534759
8	33.1	28.8	37.4	16	30.46	0.525279
9	37.5	32.4	42.1	18	34.74	0.518135
10	41.9	36	46.8	20	39.02	0.512558
11	46.3	39.6	51.5	22	43.3	0.508083
12	50.7	43.2	56.2	24	47.58	0.504414
13	55.1	46.8	60.9	26	51.86	0.50135
14	59.5	50.4	65.6	28	56.14	0.498753
15	63.9	54	70.3	30	60.42	0.496524

Time is the ongoing progression of existence and the events that take place in what appears to be an irrevocable order from the past to the present and the future. Analog signals are converted to digital signals in Table 1 via sampling and quantization. This table displays The image's digitised spatial resolution is determined by the sampling rate. The quantity of grey levels in the digitised image is instead determined by the quantization level. This table inference results first level range being 1% to 5%, and the produce range is 0.56%. The second level range is 6% to 10%, the produce range is 0.521%, the third level range is 11% to 15%, and the produced range is 0.49%.

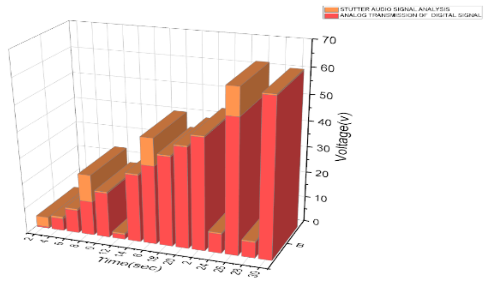

Figure 6. Analog signals to digital signal

Analog signals are converted to digital signals in Figure 6 utilizing sampling and quantization. A Signal is shown in this graphic presentation. A continuous signal that represents physical measurements is an analogue signal. Digital modulation produces discrete-time signals known as digital signals. A digital signal is created from an analogue signal.

Table 2. Automated driving to medical devices

Expectations	Efficiency	Bandwidth	Throughput	Personal care
0.1	3.6	2.6	9.36	1.26
4.3	9.8	5.6	54.88	2.2
0.23	0.56	3.4	1.904	3.14
7.8	13.6	4.7	63.92	4.08
13.5	45.8	2.4	109.92	5.02
0.8	34.67	1.8	62.406	5.96
6.7	21.89	6.8	18.852	6.9
31.4	9.11	9.2	83.812	7.84
5.78	3.67	14.8	54.316	8.78
0.98	16.45	23.4	34.93	9.72

3.56	29.23	12.8	34.144	10.66
11.78	2.01	7.8	15.678	11.6
45.89	14.79	5.6	82.824	12.54
34.78	47.57	4.3	24.551	13.48
67.8	60.35	34.6	28.11	14.42

Table 2 lists medical gadgets and automated driving. This table illustrates how high-quality hardware and software that broadly support medical practise power digital medicine solutions. This table inference results first level range being 0.1% to 0.8%, and the product range is 5.96%. The second level range is 6.7% to 3.56%, the product range is 10.66%, the third level range is 11.78% to 67.8%, and the produced range is 14.42%.

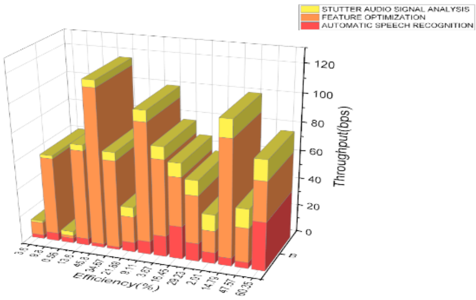

Figure 7. Automated driving to medical devices

Figure 7 shows automated driving to medical devices. This graph represents that an automated driving system on the vehicle can itself perform all driving tasks and monitor the driving environment essentially, do all the driving in certain circumstances.

Table 3. Quantization and sampling of the amplitude of the speech signal

Representation level	Sample value	Analog signal	Sampling instance
1	3.45	5.2	7.9
2	8.2	11.9	9.45
3	12.95	18.6	11
4	17.7	25.3	12.55
5	22.45	32	14.1
6	27.2	38.7	15.65
7	31.95	45.4	17.2
8	36.7	52.1	18.75
9	41.45	58.8	20.3
10	46.2	65.5	21.85
11	50.95	72.2	23.4
12	55.7	78.9	24.95
13	60.45	85.6	26.5
14	65.2	92.3	28.05
15	69.95	99	29.6

This table inference results first level range being 1% to 5%, and the product range is 14.1%. The second level range is 6% to 10%, the product range is 21.85%, the third level range is 11% to 15%, and the produced range is 29.6%.

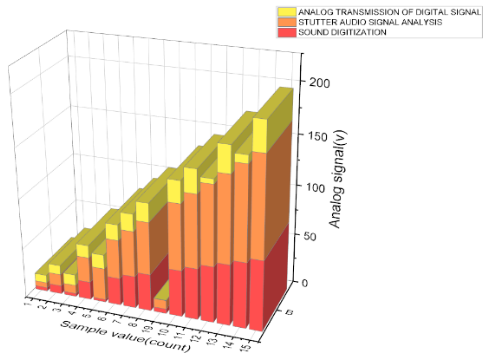

Figure 8. The amplitude of the speech signal

Table 2 lists medical equipment that can be driven automatically. This table illustrates how the technology and software that power digital medicine devices generally serve medical practise.

Table 4. Hyper-heuristic for predictions by speech, text, and Natural Language Processing (NLP)

Speech recognition	Text prediction	NLP	Heuristics selection
1.45	4.7	7.45	9.34
4.5	9.34	11.23	13.5
7.55	13.98	15.01	17.66
10.6	18.62	18.79	21.82
13.65	23.26	22.57	25.98
16.7	27.9	26.35	30.14

19.75	32.54	30.13	34.3
22.8	37.18	33.91	38.46
25.85	41.82	37.69	42.62
28.9	46.46	41.47	46.78
31.95	51.1	45.25	50.94
35	55.74	49.03	55.1
38.05	60.38	52.81	59.26
41.1	65.02	56.59	63.42
44.15	69.66	60.37	67.58

Table 4 shows the inference results first level range being 1.45% to 16.7%, and the product range is 30.14%. The second level range is 19.75% to 31.95%, the product range is 50.94%, the third level range is 35% to 44.15%, and the produced range is 67.58%.

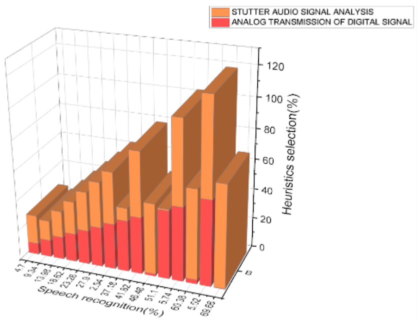

Figure 9. Hyper-heuristic predictions

Prediction hyper-heuristics are displayed in Figure 9 and Table 4. A hyper-heuristic in this graph may be thought of as a high-level approach that, given a certain issue instance or class of instances, applies to those instances. To efficiently answer the given problem, many low-level heuristics combine the supplied components in a good way automatically. The suggested system analyses stuttering audio signals, whereas the current method transmits digital signals analogly.

Table 5. Automatic stutter speech recognition and stutter audio classification for hyper-heuristics

Audio signal	Identifying stuttered	Classifiers	Automatic features
0.45	4.78	0.34	2.78
3.67	9.2	0.98	5.2
6.89	13.62	1.62	7.62
10.11	18.04	2.26	10.04
13.33	22.46	2.9	12.46
16.55	26.88	3.54	14.88
19.77	31.3	4.18	17.3
22.99	35.72	4.82	19.72
26.21	40.14	5.46	22.14
29.43	44.56	6.1	24.56
32.65	48.98	6.74	26.98
35.87	53.4	7.38	29.4
39.09	57.82	8.02	31.82
42.31	62.24	8.66	34.24
45.53	66.66	9.3	36.66

Table 5 shows automatic stutter speech recognition and stutter audio classification for hyper-heuristics. This table inference results first level range being 0.34% to 2.9%, and the product range is 12.46%. The second level range is 3.54% to 6.1%, the product range is 24.56%, the third level range is 6.74% to 9.3%, and the produced range is 36.66%.

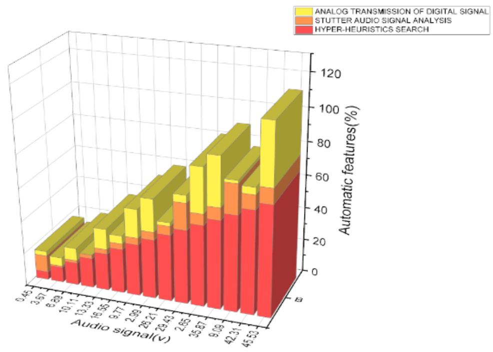

Figure 10. Automatic stutter speech recognition

The proposed system is stutter audio signal analysis and hyper-heuristics search. The existing system is Analog transmission of 't' Figure 10 shows automatic stutter speech recognition and stutter audio classification for hyper-heuristics.

Table 6. Hyper-heuristics search for stutter audio signal analysis

Amplitude	Time	Frequency	Original audio	Filtered audio
1.24	0.5	2	5.89	5.2
1.67	1	1	9.3	8.3
2.1	1.5	0.666667	12.71	11.4
2.53	2	0.5	16.12	14.5
2.96	2.5	0.4	19.53	17.6
3.39	3	0.333333	22.94	20.7
3.82	3.5	0.285714	26.35	23.8
4.25	4	0.25	29.76	26.9
4.68	4.5	0.222222	33.17	30
5.11	5	0.2	36.58	33.1
5.54	5.5	0.181818	39.99	36.2
5.97	6	0.166667	43.4	39.3
6.4	6.5	0.153846	46.81	42.4
6.83	7	0.142857	50.22	45.5
7.26	7.5	0.133333	53.63	48.6

Table 6 shows a hyper-heuristics search for stutter audio signal analysis. This table shows that Stuttering is the subject of interest of researchers from various domains like speech and signal analysis. This table inference results first level range being 5.89% to 19.53%, and the product range is 26.9%. The second level range is 22.94% to 33.17%, the product range is 30%, the third level range is 36.58% to 50.22%, and the produced range is 45.5%.

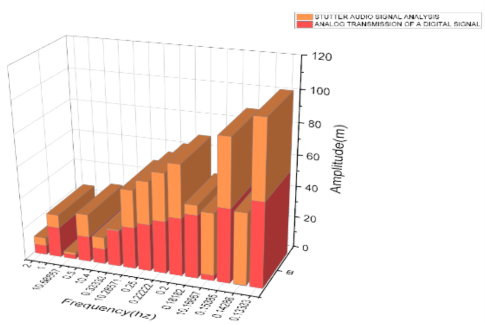

Figure 11. Hyper-heuristics search

Hyper-heuristics search is depicted in Figure 11. This graph illustrates that a hyper-heuristic is a high-level, problem-independent algorithmic framework that gives developers of heuristic optimization algorithms a collection of rules or strategies.

Table 7. Hyper-parameter tuning for feature optimization

Utility	Prediction	Validation strategy	Evaluation function	Parameter value
0.1	3.6	2.6	1.08	1.26
4.3	9.8	5.6	5.89	2.2
0.23	0.56	3.4	10.7	3.14
7.8	13.6	4.7	15.51	4.08
13.5	45.8	2.4	20.32	5.02
0.8	34.67	1.8	25.13	5.96
6.7	21.89	6.8	29.94	6.9
31.4	9.11	9.2	34.75	7.84
5.78	3.67	14.8	39.56	8.78
0.98	16.45	23.4	44.37	9.72
3.56	29.23	12.8	49.18	10.66
11.78	2.01	7.8	53.99	11.6
45.89	14.79	5.6	58.8	12.54
34.78	47.57	4.3	63.61	13.48
67.8	60.35	34.6	68.42	14.42

Table 7 shows hyper-parameter tuning for feature optimization. This table inference results first level range being 1.08% to 20.32%, and the product range is 5.02%. The second level range is 25.13% to 39.56%, the product range is 8.78%, the third level range is 44.37% to 53.99%, and the produced content is 11.6%.

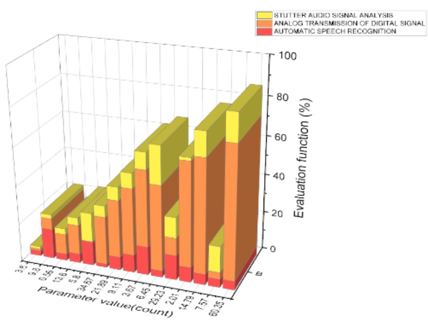

Figure 12. Hyper-parameter tunning for feature optimization

Hyper-parameter tuning for feature optimization is shown in Figure 12. The hyperparameter shown in this graph is a model input whose value is predetermined before the learning process even starts. When creating a predictive model, the method of feature optimization involves lowering the number of input variables.

Table 8. Hyperparameter tuning on binary classification

Dropout	Model training	Exact value	Approximate value	Error rate
12.4	15.78	28.18	31	10.0071
17.8	25.6	43.4	52.79	21.6359
23.2	35.42	58.62	74.58	27.2262
28.6	45.24	73.84	96.37	30.5119
34	55.06	89.06	118.16	32.6746
39.4	64.88	104.28	139.95	34.206
44.8	74.7	119.5	161.74	35.3473
50.2	84.52	134.72	183.53	36.2307
55.6	94.34	149.94	205.32	6.9348
61	104.16	165.16	227.11	37.5091
66.4	113.98	180.38	248.9	37.9865
71.8	123.8	195.6	270.69	38.3896
77.2	133.62	210.82	292.48	38.7345
82.6	143.44	226.04	314.27	39.0329
88	153.26	241.26	306.1	26.8756

$$\text{Error rate}=\frac{(exact\ value-approximate\ value)}{exact\ value}*100 \quad (4)$$

This table inference results first level range being 12.4% to 34%, and the product range is 32.67%. The second level range is 39.4% to 61%, the product range is 37.98%, the third level range is 66.4% to 88%, and the product range is 26.87%.

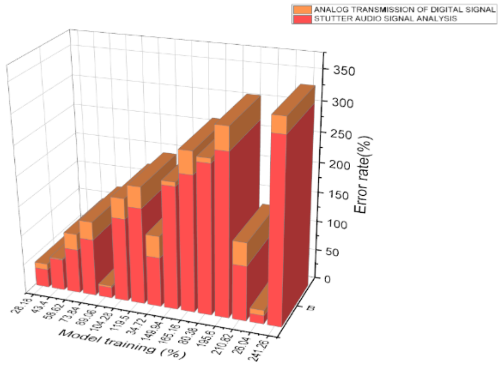

Figure 13. Hyperparameter tuning on binary classification

Hyperparameter tweaking for binary classification is shown in Figure 13. The problem of dividing a set of components into two categories in accordance with a classification rule is represented by this graph as binary classification. The part of the data science development lifecycle known as model training is where practitioners attempt to match the ideal weights and bias. The ratio of all data units that include errors to all data units transferred is known as the error rate.

Table 9. Classification accuracy of weighted finite-state transducer framework vs. Mel spectrogram

Time	Frequency	power	True positive(TP)	True negative(TN)	False positive(FP)	False negative(FN)	Classification accuracy
0.5	6.7	0.2	3	5	4.89	2.8	0.509879
1	13.4	0.56	7	12	9	6.3	0.553936
1.5	20.1	0.92	11	19	13.11	9.8	0.567001
2	26.8	1.28	15	26	17.22	13.3	0.573266
2.5	33.5	1.64	19	33	21.33	16.8	0.576944
3	40.2	2	23	40	25.44	20.3	0.579364
3.5	46.9	2.36	27	47	29.55	23.8	0.581076
4	53.6	2.72	31	54	33.66	27.3	0.582351
4.5	60.3	3.08	35	61	37.77	30.8	0.583338
5	67	3.44	39	68	41.88	34.3	0.584125

5.5	73.7	3.8	43	75	45.99	37.8	0.584766
6	80.4	4.16	47	82	50.1	41.3	0.585299
6.5	87.1	4.52	51	89	54.21	44.8	0.58575
7	93.8	4.88	55	96	58.32	48.3	0.586135
7.5	100.5	5.24	59	103	62.43	51.8	0.586468

True positive(TP)	True negative(TN)	False-positive(FP)	False-negative(FN)	Classification accuracy	Prediction rate	time	Number of samples	Response time
12	4	7	9	0.5	0.89	0.5	1	1.39
17	8	18	13	0.446429	1.22	1	2	2.22
22	12	29	17	0.425	1.55	1.5	3	3.05
27	16	40	21	0.413462	1.88	2	4	3.88
32	20	51	25	0.40625	2.21	2.5	5	4.71
37	24	62	29	0.401316	2.54	3	6	5.54
42	28	73	33	0.397727	2.87	3.5	7	6.37
47	32	84	37	0.395	3.2	4	8	7.2
52	36	95	41	0.392857	3.53	4.5	9	8.03
57	40	106	45	0.391129	3.86	5	10	8.86
62	44	117	49	0.389706	4.19	5.5	11	9.69
67	48	128	53	0.388514	4.52	6	12	10.52
72	52	139	57	0.3875	4.85	6.5	13	11.35
77	56	150	61	0.386628	5.18	7	14	12.18
82	60	161	65	0.38587	5.51	7.5	15	13.01

Table 9 shows the Classification Accuracy of Weighted Finite-State Transducer framework VS Mel Spectrogram. This table represents the A-weighted transducer putting weights on transitions in addition to the input and output symbols. This table inference results first level range being 0.2% to 1.28%, and the product range is 0.573%. The second level range is 1.64% to 3.08%, the product range is 0.58%, the third level range is 3.44% to 5.24%, and the product range is 0.586%.

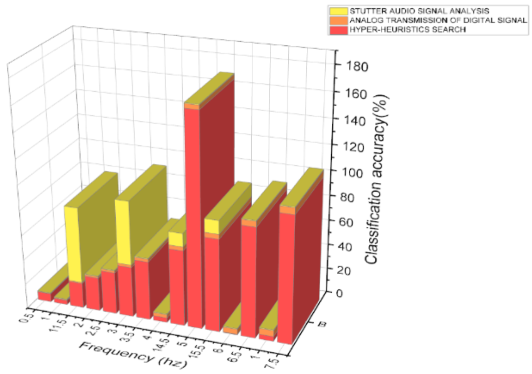

Figure 14. Classification Accuracy of Weighted Finite-State Transducer framework

Figure 14 displays the weighted finite-state transducer framework's classification accuracy. This graph shows classification accuracy, a performance indicator for classification models, as the proportion of accurate predictions to all other predictions.

Table 10. Classification Accuracy of Perceptual linear prediction (PLP) and Viterbi search VS response time

Table 10 shows the Classification Accuracy of Perceptual linear Predictions (PLP) and Viterbi search VS response time. This table represents The perceptual linear prediction analysis (PLP), a combination of spectral and linear prediction analysis. This table inference results first level range being 0.5% to 3%, and the produce range is 5.54%. The second level range is 3.5% to 5%, the produce range is 8.86%, the third level range is 5.5% to 7.5%, and the produced range is 13.01%.

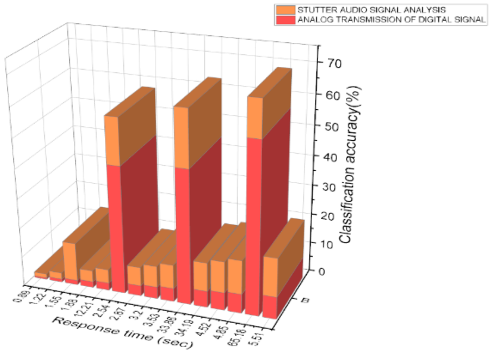

Figure 15. Classification Accuracy of Perceptual linear prediction (PLP) and Viterbi search

Figure 15 shows the Classification Accuracy of Perceptual Linear Predictions (PLP) and Viterbi search. This graph shows classification accuracy, a measure of a classification model's performance expressed as the proportion of correct predictions to all other predictions. Response time is the overall amount of time needed to address a service request.

Table 11. Classification Accuracy of Discrimination training by deep neural networks vs. Mel Spectrogram

Time	Frequency	power	True positive(TP)	True negative(TN)	False positive(FP)	False negative(FN)	Classification accuracy	Response time
0.5	2	1.67	2	3	5	7	0.294118	4.5
1	1	1.98	4	7	12	18	0.268293	9.2
1.5	0.666667	2.29	6	11	19	29	0.261538	13.9
2	0.5	2.6	8	15	26	40	0.258427	18.6
2.5	1	2.91	10	19	33	51	0.256637	23.3
3	0.333333	3.22	12	23	40	62	0.255474	28
3.5	0.285714	3.53	14	27	47	73	0.254658	32.7
4	1	3.84	16	31	54	84	0.254054	37.4
4.5	0.222222	4.15	18	35	61	95	0.253589	42.1
5	0.2	4.46	20	39	68	106	0.253219	46.8
5.5	1	4.77	22	43	75	117	0.252918	51.5
6	0.166667	5.08	24	47	82	128	0.252669	56.2
6.5	0.153846	5.39	26	51	89	139	0.252459	60.9
7	1	5.7	28	55	96	150	0.25228	65.6
7.5	0.133333	6.01	30	59	103	161	0.252125	70.3

Table 11 shows the Classification Accuracy of Discrimination training by deep neural networks VS Mel Spectrogram. This table represents Discrimination training involves reinforcing behavior in the presence of one stimulus but not others. This table inference results first level range being 0.29% to 0.256%, and the produce range is 23.3%. The second level range is 0.255% to 0.2535%, the produce range is 42.1%, the third level range is 0.2529% to 0.2521%, and the produced range is 70.3%.

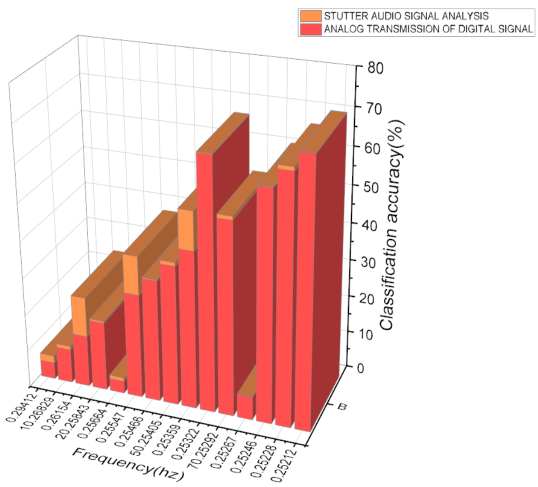

Figure 16. Classification Accuracy of Discrimination training by deep neural networks

A classification model's performance is measured by classification accuracy, which is the ratio of correct predictions to total predictions. The Classification Accuracy of Discrimination Training using Deep Neural

Networks is shown in Figure 16. An Artificial Neural Network (ANN) with numerous layers between the input and output layers is represented by the above Figure 16.

5. CONCLUSION

In this paper, we focused on Automatic stutter speech recognition and Stutter Audio classification with stutter Audio deep learning using a hyper-heuristic search Algorithm for Stutter Audio signal analysis. Our research establishes the viability of an evolutionary algorithm-driven hyper-heuristic methodology. The outcomes show that the testing examples have an impact on the performance of the model and that our theme may create generic sequences of heuristics that outperform single heuristics for certain situations. In other words, the model may be applied to instances of previously undiscovered kinds. However, the best case situation would be to use examples with identical properties to train the hyper-heuristics. Our proposed system performs Sound digitalization, audio classification, deep learning, Feature optimization, automatic speech recognition, and with the Hyper-heuristic search algorithm, and analyses automatic stutter speech recognition and stutter audio signal analysis using the proposed system flow, which obtains the Mel Spectrogram with the Classification Accuracy of 95.4%. The Automatic Speech recognition and response time concerning the classification accuracy of the weighted finite-state transducer framework is computed, and the classification accuracy of perceptual linear prediction (PLP) for the Viterbi search. The classification accuracy of discrimination training by deep neural networks. Our work underscored the greater operational efficiencies obtained from the proposed system technique. The concept of hyper-heuristics algorithms generates more efficiency and holds high relevance.

According to sampling and quantization, the product range is 23.3%, with results ranging from 0.29% to 0.256%. The produce varies from 42.1% to 70.3%, while the ranges from 0.2529% to 0.2521%. The ranges from 0.255% to 0.2535%. The produce range is 5.54%, and the autonomous driving inference ranges from 0.5% to 3%. 13.01%, 5.5% to 7.5%, 3.5% to 5%, and 8.86% of the product are generated, respectively. The product range of the voice signal is 0.573%, and its amplitude ranges from 0.2% to 1.28%. The product ranges are 0.58%, 1.64% to 3.08%, and 3.44% to 5.24%, respectively. The range of the product is 0.586%. While the product range is 32.67%, the speech result prediction range is 12.4% to 34%.

The product range for stutter speech recognition and audio categorization is 5.02% and ranges from 1.08% to 20.32%. The range is between 25.13% and 39.56%, the product is between 8.78% and 53.99%, and the content that has been developed is between 11.6% and 44.37%. The stutter audio signal analysis range ranges from 5.89% to 19.53%, 26.9%. The product range is 30%, 36.58% to 50.22%, and the range of the made is 45.5%. From 22.94% to 33.17% is the range. Results of feature optimization vary from 0.34 to 2.9%, whereas the product range is 12.46 percent. The ranges are 3.54% to 6.1%, 24.56% for the product, 6.74% to 9.3% produced

The product range is 30.14%, while the range of the binary classification results from hyper parameter tweaking is 1.45% to 16.7%. Ranges from 35% to 44.15%, produce ranges from 67.58% to 50.94% and ranges from 19.75% to 31.95% for products. The product range is 14.1%, and the categorization accuracy is 1% to 5%. The product range is between 6% and 10% and between 11% and 15%. The product range is 5.96%, and categorization accuracy results vary from 0.1% to 0.8%. 10.66% for goods, 11.78% to 67.8%, and 14.42% for output; the range was 6.7% to 3.56%. The results of a discrimination training range in classification accuracy from 1% to 5%, and the range in output is 0.56%. It varies between 6% and 10%, and 11% and 15%.

Declaration:

Funding Statement:

Authors did not receive any funding.

Data Availability Statement:

No datasets were generated or analyzed during the current study

Conflict of Interest

There is no conflict of interests between the authors.

Ethics approval:

Not applicable.

Authors' Contributions

All authors have made equal contributions to this article.

Author Disclosure Statement

The authors declare that they have no competing interests

6. References:

[1] C. Lea, V. Mitra, A. Joshi, S. Kajarekar, and J. P. Bigham, "Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter," in ***ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech***

and Signal Processing (ICASSP), June 2021, pp. 6798–6802. IEEE.

[2] B. Vaidianathan, S. Arulselvi, and B. Karthik, “Stuttered speech recognition and classification using enhanced Kamnan filter and neural network,” 2021.

[3] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, “StutterNet: Stuttering detection using time delay neural network,” in ***2021 29th European Signal Processing Conference (EUSIPCO)***, August 2021, pp. 426–430. IEEE.

[4] S. A. Sheikh, M. Sahidullah, F. Hirsch, and S. Ouni, “Machine learning for stuttering identification: Review, challenges & future directions,” ***arXiv preprint arXiv:2107.04057***, 2021.

[5] R. A. Ibrahim, M. Abd Elaziz, A. A. Ewees, M. El-Abd, and S. Lu, “New feature selection paradigm based on hyper-heuristic technique,” ***Applied Mathematical Modelling***, vol. 98, pp. 14–37, 2021.

[6] N. Mishra, A. Gupta, and D. Vathana, “Optimization of stammering in speech recognition applications,” ***International Journal of Speech Technology***, vol. 24, no. 3, pp. 679–685, 2021.

[7] I.Świetlicka, W. Kuniszyk-Jóźkowiak, and M. Świetlicki, “Artificial neural networks combined with the principal component analysis for non-fluent speech recognition,” ***Sensors***, vol. 22, no. 1, p. 321, 2022.

[8] E. D. Priya, S. Karthik, A. Sharmila, and K. G. Anand, “Design of anti-stuttering device with silence ejection speech algorithm using Arduino,” in ***2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)***, September 2021, pp. 139–142. IEEE.

[9] S. Gupta, R. S. Shukla, and R. K. Shukla, “Weighted Mel frequency cepstral coefficient based feature extraction for automatic assessment of stuttered speech using bi-directional LSTM,” ***Indian Journal of Science and Technology***, vol. 14, no. 5, pp. 457–472, 2021.

[10] P. Lorenc, P. Marek, J. Pichl, J. Konrád, and J. Šedivý, “Benchmark of public intent recognition services,” ***Language Resources and Evaluation***, pp. 1–19, 2021.

[11] Z. Hou, W. Lao, Y. Wang, and W. Lu, “Homotopy-based hyper-heuristic searching approach for reciprocal feedback inversion of groundwater contamination source and aquifer parameters,” ***Applied Soft Computing***, vol. 104, p. 107191, 2021.

[12] İ. Gölcük and F. B. Ozsoydan, “Q-learning and hyper-heuristic based algorithm recommendation for changing environments,” ***Engineering Applications of Artificial Intelligence***, vol. 102, p. 104284, 2021.

[13] C. C. Wu, D. Bai, J. H. Chen, W. C. Lin, L. Xing, J. C. Lin, and S. R. Cheng, “Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times,” ***Swarm and Evolutionary Computation***, vol. 60, p. 100765, 2021.

[14] N. Pillay and R. Qu, “Assessing hyper-heuristic performance,” ***Journal of the Operational Research Society***, vol. 72, no. 11, pp. 2503–2516, 2021.

[15] F. Zhao, S. Di, J. Cao, and J. Tang, “A novel cooperative multi-stage hyper-heuristic for combination optimization problems,” ***Complex System Modeling and Simulation***, vol. 1, no. 2, pp. 91–108, 2021.

[16] T. Yang, S. Zhang, and C. Li, “A multi-objective hyper-heuristic algorithm based on adaptive epsilon-greedy selection,” ***Complex & Intelligent Systems***, vol. 7, no. 2, pp. 765–780, 2021.

[17] B. Olgun, Ç. Koç, and F. Altıparmak, “A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery,” ***Computers & Industrial Engineering***, vol. 153, p. 107010, 2021.

[18] T. Yang, S. Zhang, and C. Li, “A multi-objective hyper-heuristic algorithm based on adaptive epsilon-greedy selection,” ***Complex & Intelligent Systems***, vol. 7, no. 2, pp. 765–780, 2021.

[19] C. C. Wu, D. Bai, J. H. Chen, W. C. Lin, L. Xing, J. C. Lin, and S. R. Cheng, “Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times,” ***Swarm and Evolutionary Computation***, vol. 60, p. 100765, 2021.

[20] W. Qin, Z. Zhuang, Z. Huang, and H. Huang, “A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem,” ***Computers & Industrial Engineering***, vol. 156, p. 107252, 2021.

[21] S. Garg, U. Mehrotra, G. Krishna, and A. K. Vuppala, “Transfer learning based disfluency detection using stuttered speech,” 2021.