Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Improved Affective Computing via CNN and Bat Algorithm Optimization: A Case Study on IEMOCAP and TESS

Sawsan J. Muhammed[a,*], Mohamed I. Shujaa[a], Ahmed B. A. Alwahhab[a]

[a]*Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq. Email: bbc4018@mtu.edu.iq, drshujaa@mtu.edu.iq, ahmedbahaaulddin@mtu.edu.iq*

ARTICLEINFO

Article history:
Received: 11/05/2025
Rrevised form: 07/06/2025
Accepted : 19/06/2025
Available online: 30/06/2025

Keywords: Speech Emotion Recognition, Convolutional Neural Network, Bat Algorithm, ECOC, Gamma Classifier, Feature Extraction, MFCC, Deep Learning

ABSTRACT

This study presents an enhanced Speech Emotion Recognition (SER) model that integrates Convolutional Neural Networks (CNN) with the Bat Algorithm (BA), a nature-inspired metaheuristic optimization technique. The objective is to improve the accuracy and generalizability of emotion classification from speech by optimizing the neural network architecture. The model utilizes handcrafted acoustic features—pitch, energy, zero-crossing rate (ZCR), and Mel-Frequency Cepstral Coefficients (MFCCs)—as inputs. Before being input into a deep neural network whose hyperparameters are modified using the Bat Algorithm, these characteristics are preprocessed and normalized. The last model uses a Gamma Classifier (GC) and Error Correcting Output Codes (ECOC) to guarantee strong categorization. Experimental findings employing benchmark datasets like IEMOCAP, EMO-DB, and Berlin DB show better performance, with validation accuracy climbing as high as 98.44%. This hybrid design offers a consistent method for practical emotion identification systems and outperforms traditional methods.

https://doi.org/10.29304/jqcsm.2025.17.22199

1. Introduction

Speech Emotion Recognition (SER) has grown a lot in the last few decades. This is because more and more people want smart systems that can tell how someone is feeling by listening to their voice and analyzing things like tone, pitch, and rhythm [2, 5]. Speech Emotion Recognition (SER) is important for many uses, like virtual helpers, tracking mental health, customer service systems, and interacting between people and computers [2, 7].

Even though there has been a lot of growth, the domain is still facing some major problems. This includes the fact that different languages and cultures have very different ways of showing feeling, that different emotions can overlap and be hard to understand in speech, and that it's hard to get large, varied, and representative datasets that are needed for good model training [5, 7]. When SER systems are used in real life, these limitations have a big effect on their ability to generalize.

In the past, speech emotion recognition systems that used machine learning mostly relied on sound factors that were created by hand, such as energy, pitch, spectrum traits [5, 9], and Mel-Frequency Cepstral Coefficients (MFCCs). Support Vector Machines (SVMs) and Random Forests were two types of models that used these characteristics a lot. Still, these old methods had trouble showing the complex, changing patterns of emotional expression correctly, and they were often affected by background noise and differences in the domain [11, 13].The advent of deep learning has revolutionized the domain of Speech Emotion Recognition (SER). Models

∗Corresponding author Sawsan J. Muhammed

Email addresses: bbc4018@mtu.edu.iq

Communicated by 'sub etitor'

include Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and hybrid CNN-LSTM architectures have facilitated automated feature extraction from raw audio data, markedly enhancing emotion categorization efficacy [2][15][16]. Nevertheless, deep learning methodologies present distinct challenges, such as the complexities of optimal architecture selection and hyperparameter tuning [15], a significant propensity for overfitting on limited emotional datasets [16], substantial computational expenses, and inadequate generalization across various accents, languages, and recording conditions [5][7].

Researchers have increasingly used optimization approaches based on Swarm Intelligence to tackle these difficulties. The Bat Algorithm (BA), inspired by the echolocation activity of microbats, has shown significant promise [4][11][13]. BA adeptly reconciles global exploration with local exploitation, hence preventing convergence to local minima and promoting the identification of optimum solutions [4].

In the field of SER, BA has successfully improved classification accuracy on standard datasets like Emo-DB and RAVDESS by fine-tuning hyperparameters such as learning rates, layer counts, and dropout rates and neural network weights. New studies show that combining swarm-based optimization, especially the Bat Algorithm, with deep learning models like CNN-LSTM makes recognition much more accurate, often by more than 98% [16]. To get around recurring problems in SER, this mixed model offers a good way to go. It makes it easier to make emotion recognition systems that are strong, flexible, and useful in real life.

2. Related Works

Combining deep learning models with metaheuristic optimization methods has been a typical way to improve Speech Emotion Recognition (SER). This plan could help with significant problems like choosing the right features, reducing the number of dimensions, and making the classifier more stable. If you look closely at some of the most important studies that used Convolutional Neural Networks (CNNs), the Bat Algorithm (BA), and other swarm intelligence methods, you can see how far this discipline has gone and what still has to be done. A research found a novel approach to tell how someone is feeling by looking at their speech that used both spectral and prosodic data and glottal waveform analysis. It employed bio-inspired metaheuristic algorithms to make the data easier to work with and a Gaussian Elliptical Basis Function Network to put the data into groups. The model was more accurate than usual methods. It wasn't very helpful, however, since the sample didn't feature a lot of diverse emotional groupings, noises, or languages. I wonder how well it would function in places that are noisy or out of control.
It was hypothesized that the ideal parameters for a Deep Belief Network (DBN) might be found by combining the Whale Optimization Algorithm (WOA) with Particle Swarm Optimization (PSO). The combined model was more accurate and settled quicker than the Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Firefly Algorithm (FA) models used alone. But it didn't do enough to cope with the complexity of computation or generalization in busy or multinational situations [6].
Another study employed adaptive weight change and swarm intelligence to find groupings of features. This worked more than 90% of the time with both the EMO-DB and RAVDESS databases. The plan worked well to obtain socially significant traits and make sure that all the presenters were on the same page. The study didn't look at situations with little resources or real-time data, and it didn't look at how well it learnt from large datasets [7].
A clustering-based combination technique that combines Equilibrium Optimizer (EO) and Atom Search Optimization (ASO) was discovered to make feature selection and classifier optimization better. When used on the IEMOCAP dataset, this model proved more accurate and used less computer resources than other optimization methods. It's hard to understand, however, and the algorithms are quite complicated, which makes it impossible to use in real time or for apps that need explainable AI [8]. Researchers introduced an Error-Correcting Output Codes (ECOC) layer to a CNN classifier that employs features based on entropy and spectro-temporal modulation (STM). The model got 93.33% of the EMO-DB answers right and 85.73% of the ShEMO answers right, which is better than other feature-based techniques. Using multimodal descriptors made it easier to handle speech patterns that weren't linear. However, the high-dimensional input and longer classification delay from the ECOC layer would make it hard to employ in real time [9].

A hybrid CNN model was developed to analyze both phoneme-level inputs and spectrograms, assessed using the IEMOCAP dataset. The results demonstrated that the integration of low-level language units with high-level acoustic information significantly enhances accuracy. Nonetheless, the preprocessing necessary for phoneme extraction adds complexity, and the system's generalizability across various accents and phonetic patterns was not comprehensively tested [10].

Although the Bat Algorithm was not used directly, another research introduced the BAT model (Block and Token Self-Attention), which is influenced by the concepts of echolocation. The model accurately captures contextual dependencies in speech by including hierarchical self-attention processes. It attained high accuracy on benchmark

datasets; nevertheless, the computing requirements of the attention mechanism may restrict its scalability in embedded or resource-limited SER systems [11].

Supplementary research suggested using an Optimized Genetic Algorithm (OGA) for feature selection with Extreme Learning Machines (ELMs) for classification, achieving an accuracy of 93.3% on EMO-DB. The strategy provided an advantageous equilibrium between efficiency and simplicity. Nonetheless, ELMs often encounter difficulties in generalizing to novel data, and the adjustments necessary for genetic optimization increase the likelihood of overfitting on limited datasets [12].

Another method used Particle Swarm Optimization to enhance both emotional speech characteristics and classifier parameters. The findings validated the advantages of PSO-tuned models regarding convergence speed and accuracy in comparison to baseline classifiers. This study highlighted the significance of dynamic feature search methodologies; nonetheless, the presumption of static feature distributions and restricted multilingual validation persist as major limitations [13].

These works together highlight the increasing efficacy of integrating CNN-based models with metaheuristic optimization techniques, particularly in improving classification accuracy, generalizability, and noise resilience in SER systems. Ongoing research in this domain is essential for creating adaptable and scalable emotion identification systems appropriate for use in intricate and real-world settings.

TABLE 1. COMPARATIVE TABLE OF SER MODELS

Model Type	Optimization Method	Dataset	Accuracy	Key Limitation
Glottal + Metaheuristic + GEBFN	Bio-inspired heuristic	Unspecified (Low Diversity)	Higher than traditional	Limited emotion classes, dataset diversity
DBN + PSO-WOA Hybrid	PSO + WOA	Unspecified	Higher than GA/PSO/FA	High complexity, no noise robustness
Feature Subset Selection (Swarm-Based)	Swarm Intelligence	EMO-DB, RAVDESS	>90%	Not evaluated in real-time or large datasets
CNN + EO-ASO Hybrid	EO + ASO	IEMOCAP	Improved over baseline	Complex model, low interpretability
CNN + STM + ECOC	None (CNN + ECOC)	EMO-DB, ShEMO	93.33% (EMO-DB), 85.73% (ShEMO)	High-dimensional input, ECOC latency
Hybrid CNN (Phoneme + Spectrogram)	None (Hybrid CNN)	IEMOCAP	Enhanced Accuracy	Preprocessing overhead, uncertain across accents
BAT Self-Attention Model	Self-attention (Inspired by echolocation)	Multiple Benchmarks	Strong benchmark accuracy	High computational cost

OGA + ELM	Optimized Genetic Algorithm	EMO-DB	93.3%	Risk of overfitting, weak generalization
PSO-Tuned Classifier	PSO	Unspecified	High accuracy, fast convergence	Assumes static feature distribution

3. Background

Speech Emotion Recognition (SER) is a key aspect of how humans and robots speak to each other and how social computing works. When people chat, SERs can determine how they feel by the noises they emit. People can communicate with computers and virtual assistants in a manner that seems more natural and responsive in industries like healthcare, education, and surveillance. Recent improvements in machine learning and optimization have made SER models far more effective and dependable in a wider range of situations[3].

This background section speaks about the main pieces that back up the proposed SER framework. The first portion talks about essential sound qualities including MFCC, pitch, energy, and the rate at which the sound crosses zero. These qualities may tell whether someone is feeling anything in their voice. After that, it talks about how to leverage these characteristics to teach Convolutional Neural Networks (CNNs) to build high-level abstract models. Finally, it speaks about the Bat Algorithm, which is a bio-inspired metaheuristic optimization approach that can make neural networks operate better by changing their design and hyperparameters[14].

3.1 Audio Features for Speech Emotion Recognition (SER)

Feature extraction is a fundamental step in any Speech Emotion Recognition (SER) system, as it provides the initial representation upon which classification models operate. Emotional speech exhibits distinct acoustic patterns, commonly captured through three main categories of features: prosodic, spectral, and voice quality descriptors [7][10].

3.1.1 Prosodic Features

Some suprasegmental elements are pitch (F_0), energy, and time. Prosodic traits are related to these. Pitch shows how fast the vocal folds are vibrating and is usually higher when people are angry or happy and lower when they are sad or bored. The average strength of the signal over time (Eq. 1) [15] shows that energy changes a lot depending on how people are feeling:

$$E = \frac{1}{N}\sum_{n=1}^{N} x^2(n) \quad (1)$$

where x(n) is the amplitude of the speech sample at time n, and N is the total number of samples. Energy is higher for emotions such as anger and lower for sadness.

3.1.2 Spectral Features

Spectral features represent the frequency characteristics of speech and are crucial for capturing timbre and resonance variations induced by emotional expression. The most widely adopted spectral features in SER are Mel-Frequency Cepstral Coefficients (MFCCs), derived from the short-time power spectrum via a mel-scale filterbank [16]. MFCCs model the vocal tract envelope and offer strong discriminative power for emotion classification (Eq. 2):

$$n^C = \sum_{K=1}^{K}[\frac{\pi}{K}(\frac{1}{2} - K)n]\cos(S_k)log \quad (2)$$

where Sk is the energy of the k-th filter, and K is the total number of filters.

3.1.3 Voice Quality Features

Voice quality features quantify microvariations in the speech signal. Jitter and shimmer measure instability in frequency and amplitude, respectively—often linked to stress and nervousness. Another important descriptor is the Zero-Crossing Rate (ZCR), which reflects the number of times a signal crosses the zero-amplitude axis, as shown in Eq. (3) [17]:

$$ZCR = \frac{1}{2N}\sum_{n=1}^{N-1} |(sgn(sgn(x(n))x) - 1)| \quad (3)$$

This metric is particularly useful for detecting unvoiced or noisy speech transitions associated with emotional changes.

Where

n The current discrete time index (sample number). It iterates from 111 to N−1N-1N−1.

N The total number of discrete samples within the analyzed signal frame (window length).

x(n) The amplitude value of the speech signal at sample nnn. It represents the value of the signal at that time instant.

sgn(·) The **sign function**. It outputs: +1 if the input > 0, 0 if input = 0, and -1 if input < 0.

3.1.4 Higher-Level Statistical Features

To ensure consistent input dimensionality, modern SER systems aggregate low-level descriptors (LLDs) into high-level statistical features such as mean, standard deviation, skewness, and percentiles. Toolkits like OpenSMILE implement feature sets such as INTERSPEECH 2010 (IS10), which include over 1500 attributes—covering MFCCs, pitch, jitter, shimmer, and formants.

Overall, the effectiveness of these acoustic features has been validated across benchmark datasets including Emo-DB, RAVDESS, and IEMOCAP, where they serve as the primary input for subsequent deep learning models.

3.2 Convolutional Neural Networks (CNN) in SER

Convolutional Neural Networks (CNNs) are extensively used in Speech Emotion Recognition (SER) because they can autonomously extract hierarchical and discriminative features from raw or processed audio inputs, including spectrograms, log-Mel representations, and MFCCs. By using local time-frequency patterns, CNNs eliminate the need for manually produced features often used in conventional machine learning methodologies.

A conventional CNN-based SER model comprises convolutional layers for extracting localized acoustic characteristics, pooling layers for dimensionality reduction, and fully connected layers for final emotion classification. Dropout layers are often used to reduce overfitting, whereas softmax activation in the output layer yields the probability distribution across emotion classes.

Recent developments in Speech Emotion Recognition (SER) have led to the emergence of two-stream Convolutional Neural Network (CNN) designs that concurrently analyze both raw audio waveforms and spectrograms. The integration of temporal and spectral inputs has shown enhanced accuracy across several datasets, including EMO-DB, SAVEE, and RAVDESS, often surpassing 85% to 95%, contingent upon the architecture and input representation.

Convolutional Neural Networks (CNNs) can accurately mimic complex emotional signals because they employ parameter sharing and spatial localization. You may easily add LSTM layers or attention mechanisms to them to capture long-range interdependence or put more focus on emotionally important places. CNNs are the building blocks of modern SER designs [18].

$$y_{i,j}^{I} = (b^{l} + x_{i+u,j+v}^{(l-1,m)}.\, w_{u,v}^{(m)} \sum_{m=1}^{M}\sum_{u=0}^{K-1}\sum_{v=0}^{K-1})f \quad (4)$$

Where:

y_(i,j)^I: output at position (i,j) of layer l

x_(i+u,j+v)^((l-1,m)): input from previous layer

w_(u,v)^((m)): filter weights

f: activation function.

3.3 Bat Algorithm for Neural Network Optimization

Xin-She Yang came up with the Bat Algorithm (BA) in 2010. It is a nature-inspired metaheuristic based on how microbats use echolocation. These bats use sonar pulses to find and identify prey, even when it's dark. This is a biological process that has been turned into a mathematical model to solve complex optimization problems [19].

In the field of Speech Emotion Recognition (SER), hyperparameters like as learning rates, kernel dimensions, filter amounts, and dropout rates have a big effect on how well CNNs work. Changing these parameters by hand is typically not effective, prone to mistakes, and not very scalable. BA offers an automated approach by skillfully moving across the high-dimensional hyperparameter space.

In the algorithm, each bat stands for a possible CNN configuration. Depending on local and worldwide search parameters, the bats change their speed, position, and frequency. The rates of loudness and pulse emission are changed in real time to balance exploration and exploitation. This technique helps BA avoid local optima and move toward better solutions. The conventional BA update formulae are as follows:

Frequency Update:

$$f_i = f_{min}(f_{max} - f_{min}).\beta, \beta \in [0,1] \quad (5)$$

Velocity Update:

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*).f_i \quad (6)$$

Position Update:

$$x_i^t = x_i^{t-1} + v_i^t \quad (7)$$

Where:

fi: frequency

vi: velocity

xi: position

x∗: current global best solution

The Bat Algorithm (BA) improves hyperparameters of Convolutional Neural Networks (CNN), improving Speech Emotion Recognition (SER) systems. It harmonizes global exploration with local exploitation, enhancing convergence and reducing overfitting hazards. This methodology results in CNN models exhibiting enhanced emotion identification accuracy and higher generalization across varied datasets.

3.3.1 Local Search and Acceptance Criterion

If a random number exceeds the pulse emission rate ri, a local solution is generated see in Eq(8),(9)and(10)[19]:

$$x_{new} = x_* + \epsilon . A_i \quad (8)$$

A solution is accepted if:

$$rand < f(x_{new}) \; and \; A_i < f(x_i) \quad (9)$$

Afterward, the bat's parameters are updated:

$$(A_i^{t+1} = r; A_i^t \alpha_i^{t+1} = 1) r_i^0 - e^{-\gamma t} \quad (10)$$

Where:

Ai: loudness

α\alpha, γ\gamma: decay factors

3.4 Error-Correcting Output Codes (ECOC) for Multi-Class Emotion Classification

A robust ensemble approach, Error-Correcting Output Codes (ECOC) divide multi-class classification problems into several binary sub-tasks. Adding redundancy to the decision-making process, Error-Correcting Output Codes (ECOC) enhances resilience and classification accuracy in Speech Emotion Recognition (SER), a field where emotional boundaries regularly overlap and class imbalance is widespread [9].

An emotion coding matrix is used, with rows representing categories and columns specifying binary classifiers, to store unique binary codewords for each emotion category. All during training, this matrix is used to build a set of binary classifiers. Using decoding methods like Hamming distance, classifiers generate a predicted binary string during inference and compare it to each class codeword. We call the final forecast the one whose members show the most agreement.

Because it improves generalization, mitigates the impact of unclear or noisy inputs, and decreases the propagation of errors in high-dimensional emotional datasets, ECOC is particularly effective in SER tasks. When ECOC-based decision fusion is combined with deep features extracted from CNNs, the effectiveness of emotion recognition in multi-class scenarios is greatly enhanced.

3.5 GAMMA CLASSIFIER (GC) FOR PROBABILISTIC DECISION MODELING

An effective probabilistic classification method, the Gamma Classifier (GC) is based on the Gamma distribution and may mimic skewed and asymmetric data distributions, which are common in emotional speech signals [1]. A more flexible probability model is provided by GC, as opposed to more traditional classifiers such as softmax or SVMs, enabling better discrimination across emotion classes that share properties.

After processing using CNN and ECOC, this study uses GC as the last classification layer. Using fitted Gamma distributions, it determines the posterior probability that a certain feature vector is linked to each class. This probabilistic method improves decision-making assurance in uncertain contexts by capturing the inherent variability of speech features across different emotional states.

Additionally, GC enhances robustness in small or skewed datasets, when assumptions based on Gaussian distribution could not hold. The classifier improves the SER system's reliability and accuracy by using ECOC coding methods, which increase statistical expressiveness and rectify structural errors

.4. Methodology

The proposed model in this study seeks to enhance the generalizability and accuracy of Speech Emotion Recognition (SER) systems by integrating a bio-inspired optimization approach with deep learning and manually crafted acoustic features. This method recognizes the significant influence of neural network architecture and hyperparameters on performance by using the Bat Algorithm to effectively optimize a Convolutional Neural Network's (CNN) configuration. The approach starts with the preprocessing of raw audio signals to guarantee a consistent format and quality. Relevant auditory features are extracted from these signals and then normalized, including pitch, energy, zero-crossing rate (ZCR), and Mel-Frequency Cepstral Coefficients (MFCCs). To facilitate a scientifically rigorous model evaluation, the processed data is then divided into training and testing sets. The Bat Algorithm automates and optimizes hyperparameter tuning by identifying the optimal combination of neuron counts and dropout rates inside the CNN. The optimal configuration is used to construct the labeled emotional speech data utilized for training and evaluating the final model. This research concentrates on four fundamental emotions: Happy, Sad, Angry, and Neutral, in contrast to several other studies that assess six to eight emotion categories. Their selection was based on high frequency and uniqueness across several datasets, aimed at improving label consistency and reducing the likelihood of misdiagnosis. As seen in Fig. (1), this targeted emotional simplification enhances cross-dataset compatibility and the model's robustness.

The design has five main steps, each meticulously chosen to tackle certain deficiencies in current SER systems:

1. input Preprocessing: Initial acoustic input is treated to extract uniform segments. Silence trimming and normalizing are used to reduce noise and maintain uniformity between samples. This stage is crucial for enhancing the quality of the retrieved characteristics and reducing extraneous variability in voice input.

2. Feature Extraction: Mel-Frequency Cepstral Coefficients (MFCCs), Zero-Crossing Rate (ZCR), Pitch, and Energy characteristics are derived from each audio segment. These rudimentary descriptors have shown efficacy in collecting emotional signals and are extensively used in benchmark Speech Emotion Recognition systems. They constitute the input for the CNN layers.

3. CNN-Based Feature Representation: A Convolutional Neural Network (CNN) is used to acquire high-level feature abstractions from the retrieved audio data. The CNN architecture comprises many convolutional and pooling layers, succeeded by fully connected layers. Dropout regularization is used to mitigate overfitting. This phase automates feature engineering and improves emotion-specific pattern recognition.

4. Hyperparameter Optimization using Bat Algorithm (BA): The Bat Algorithm is used to enhance model performance by optimizing essential CNN hyperparameters, including kernel size, filter quantity, dropout rate, and learning rate. Bayesian optimization improves convergence and prevents local minima by adaptively balancing exploration and exploitation. This stage mitigates the constraints of manual adjustment and enhances generalization.

5. Classification with ECOC and Gamma Classifier: The deep feature vector is processed via an Error-Correcting Output Code (ECOC) layer to transform the multi-class emotion classification into several binary sub-tasks. The ultimate prediction is generated by the Gamma Classifier, which delineates the probability distribution of characteristics among emotional categories. This dual-layer classification method enhances resilience and efficacy on unbalanced datasets.

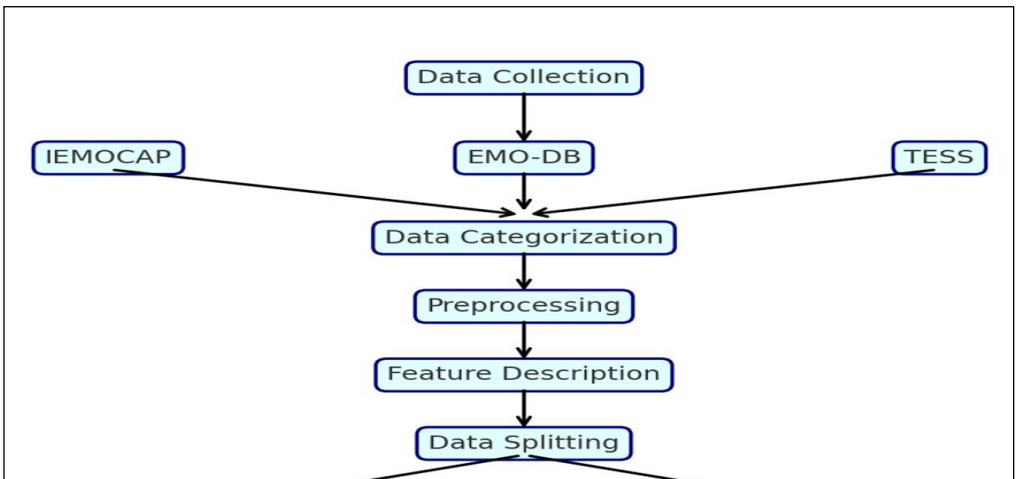

Fig. 1- Overall architecture of the proposed SER framework, showing data collection, preprocessing, feature extraction, CNN modeling, Bat Algorithm optimization, and final classification using ECOC and Gamma Classifier.

4.1. Dataset Description and Comparison

The proposed Speech Emotion Recognition (SER) model was evaluated on three major emotional speech datasets: IEMOCAP, EMO-DB, and the TESS Database of Emotional Speech (Emo-DB). The purpose was to assess the model's robustness and generalizability. These datasets were carefully selected for their variety in speaker profiles, emotional categories, and recording circumstances to offer a diverse and demanding testing ground for SER systems.

4.1.1 Leveraging multiple datasets offers several advantages:

When developing and evaluating Speech Emotion Recognition (SER) systems, many emotional speech samples are essential. These datasets provide cross-lingual assessment, allowing researchers to evaluate performance across languages and distinguish between simulated and real emotional responses. This approach makes model adaption testing simpler in a variety of linguistic and auditory environments, making it more dependable. It provides a solid framework for modifying hyperparameters using the Bat Algorithm in various scenarios, making the model more dependable and useful.

The IEMOCAP collection, a full SER resource, has 12 hours of multimedia recordings of 10 skilled players in five pairings. The Berlin Emotional Speech Database (EMO-DB) contains high-quality German emotional speech. Ten seasoned actors performed 535 passionate comments. Emotional Speech Set (TESS) is an English-language database established by the University of Toronto Psychology Department. It has 2,800 audio recordings of two performers in seven moods. These datasets provide SER systems several test options. Cross-linguistic, auditory, and emotional assessment and hyperparameter tuning are simple with them. (see Table (1)).

Table 1 - Dataset Comparison Table.

Dataset	Language	Speakers	Emotions	Style	Notes
IEMOCAP	English	10	6–8	Scripted + spontaneous	Complex, context-rich data
EMO-DB	German	10	7	Acted	High-quality, balanced recordings

TESS	English	2 (female)	7	Acted	Studio-recorded, clear articulation, emotion focused

4.2. Data Preprocessing

Effective data preprocessing is a critical step in any speech-based machine learning pipeline, especially for speech emotion recognition (SER) tasks. The quality, consistency, and clarity of the input audio signals significantly affect the accuracy of feature extraction and the overall performance of the classification model see in Fig. (2).

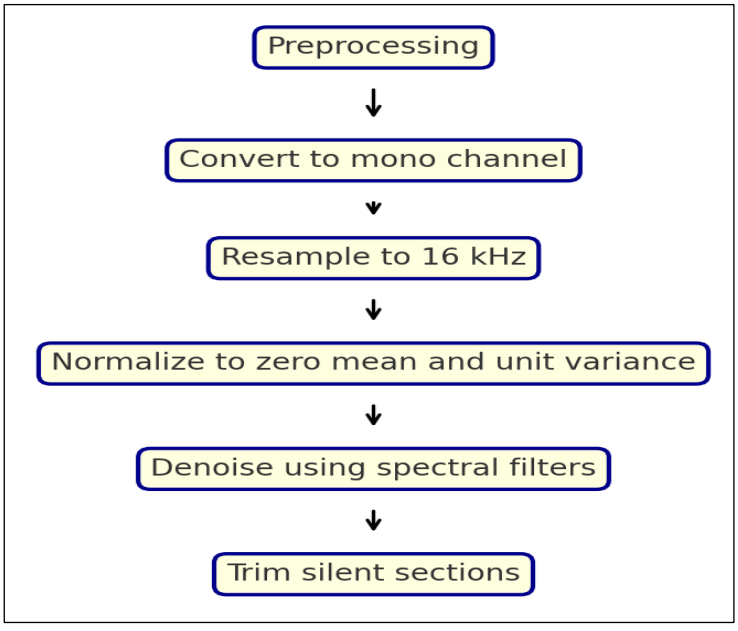

Fig. 2- Preprocessing pipeline applied to raw audio signals, including mono conversion, resampling, normalization, denoising, and silence trimming.

The preprocessing stage in this study includes the following key steps:

Conversion to Mono Channel:Many audio files are recorded in stereo, resulting in two channels. For SER tasks, a single audio channel (mono) suffices and reduces computational complexity. Stereo files are therefore converted to mono by averaging or selecting one channel[26].

Amplitude Normalization:Amplitude levels vary greatly between recordings, which can introduce unwanted variability. Normalizing the amplitude scales all audio samples to a uniform range (typically between -1 and 1), helping stabilize feature extraction[27].

Resampling to a Consistent Sampling Rate:Different datasets may have varying sampling frequencies (e.g., 44.1 kHz, 16 kHz). For consistent analysis, all audio files are resampled to a standard rate (commonly 16 kHz). This ensures uniform time resolution across the dataset[28].

Silence Trimming and Noise Reduction:Silent segments and background noise do not contribute meaningfully to emotion detection. The audio signals are trimmed to remove leading and trailing silence using energy thresholding. Optional noise reduction filters can be applied for cleaner input[29].

Signal Duration Standardization (Optional):Depending on the dataset, signals may be padded or truncated to a fixed duration to maintain a consistent input shape when feeding into CNNs[30].

To perform these operations, the Librosa Python library is used due to its robust tools for audio loading, trimming, resampling, and transformation. This preprocessing ensures high-quality, normalized inputs suitable for consistent and reliable feature extraction and subsequent model training.

4.3. Feature Normalization and Dataset Splitting

Each audio sample is represented as a combination of:

- The mean pitch, computed from the fundamental frequency using the YIN algorithm.
- The average energy, representing the signal power.
- The mean ZCR, indicating signal noisiness.
- A set of 13-dimensional MFCCs, capturing the spectral envelope of the signal.

The features are concatenated into a unified feature vector of fixed dimensionality using NumPy operations: X = np.hstack((pitch, energy, ZCR, MFCCs))

this step results in a matrix $X \in Rn \times d$, where n is the number of audio samples and d is the total number of extracted features per sample.

Normalization Using StandardScaler

In order to ensure stable and efficient training of the neural network, the raw features are normalized. Normalization transforms the feature space such that each feature has:

- Zero mean ($\mu=0$)
- Unit variance ($\sigma=1$)

This is achieved using the StandardScaler from the scikit-learn library, defined as Eq(11) [23]:

$$x_{normalized} = \frac{\mu - x}{\sigma} \qquad (11)$$

where:

- x is the original feature value, μ is the mean of the feature in the training set, σ is the standard deviation.

The fit_transform method is applied on the training set, and transform is used on the test set to avoid data leakage.

4.3.1 Label Encoding and One-Hot Encoding

To prepare the labels for multi-class classification:

1. Emotion labels (e.g., "happy", "sad") are first converted into integer class indices using LabelEncoder.
2. These integer labels are then transformed into **one-hot encoded vectors** using to_categorical from Keras, enabling the use of categorical crossentropy loss in the neural network.

4.3.2 dataset Splitting

Using stratified sampling, the dataset is split into training and testing groups, with 80% set aside for training and 20% for testing. Reproducibility is ensured by random_state=42. The optimization method train_test_split(X, y, test_size=0.2, stratify=y, random_state=42) keeps the class distribution the same across both groups. This makes the deep learning model more accurate and speeds up the convergence rate.

4.4. Hyperparameter Optimization Using Bat Algorithm

The Bat Algorithm (BA) is a metaheuristic optimization technique that equilibrates exploration and exploitation within the search space. This research employs it to improve the efficacy of a Convolutional Neural Network (CNN)-based framework for Speech Emotion Recognition (SER). Bayesian optimization is included into the training process to autonomously identify the best configuration of essential CNN hyperparameters, which would otherwise need human adjustment. The BA incrementally adjusts each bat's location in the hyperparameter space according to its experience and the optimal global solution. The integration of BA into the CNN training framework facilitates accelerated convergence, reduces manual tuning requirements, and enhances generalization performance across various speech emotion datasets. Experimental results show that BA-optimized CNNs consistently outperform manually tuned counterparts, achieving higher classification accuracy and better stability across benchmark datasets. such as Fig.(3) and the key parameters used in the Bat Algorithm in table (2):

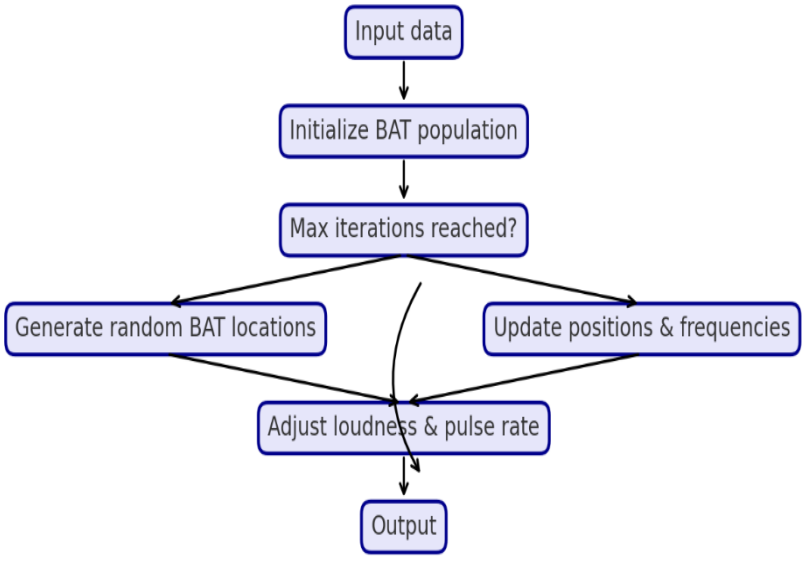

Fig. 3- Flowchart of Bat Algorithm applied for CNN hyperparameter optimization.

TABLE 2. OUTLINES THE CRUCIAL PARAMETERS UTILIZED IN THE BAT ALGORITHM (BA) FOR OPTIMIZING THE CNN MODEL WITHIN THE PROPOSED SER FRAMEWORK. THESE PARAMETERS WERE METICULOUSLY CHOSEN BASED ON EMPIRICAL CALIBRATION AND SCHOLARLY GUIDELINES TO GUARANTEE EFFICIENT CONVERGENCE.

Parameter	Value	Description / Range
Number of Iterations	50	Maximum optimization iterations
Population Size	20	Number of bats (solutions)
Initial Loudness (A)	0.9	Controls exploration vs. exploitation
Pulse Rate (r)	0.5	Probability of local search

Frequency Range (f)	[0, 2]	Controls step size for exploration
Learning Rate	[0.001 – 0.01]	Optimized during training
Dropout Rate	[0.2 – 0.5]	Tuned for each dense layer
Kernel Size	[3 – 7]	Size of convolutional filters
Number of Filters	[32 – 128]	Tuned per convolutional layer

4.4.1 Optimization Representation

Each individual "bat" in the population represents a candidate solution in the hyperparameter space. A solution vector $x_i \in R$ may take the form:

$xi=[\ 3neurons_1\ dropout_1, neurons_3, dropout_2, neurons_2, dropout_3]$

4.4.2 Fitness Evaluation

To evaluate the quality (fitness) of each solution, the corresponding CNN model is built and trained using the training data. The **cross-entropy loss** on the validation set is used as the objective function to minimize see in Eq(12) [23]:

$$\mathcal{L}_{cross-entropy} = -\sum_{i=1}^{C} y_i \log(y^i) \qquad (12)$$

Where:

- Yi is the true one-hot encoded label,
- y^i is the predicted SoftMax probability.

The lower the loss, the better the fitness of the bat.

4.4.3 Algorithm Dynamics

At each iteration, the algorithm updates see in Eq(13),(14) and (15) [24]:

- **Frequency** fi: Controls step size and diversity.

  $fi=fmin+(fmax-fmin)\cdot\beta, \beta \in [0,1]$ (13)

- **Velocity** vi: Direction of movement.

  $$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*).f_i \qquad (14)$$

- **Position** xi: Current solution.

  $$x_i^t = x_i^{t-1} + v_i^t \qquad (15)$$

If a randomly drawn number is greater than the pulse rate ri, a **local search** is applied around the current best solution x∗:

$x_{new}=x*+\epsilon\cdot Ai$

A solution is accepted if it improves the loss and a random number is less than the bat's loudness Ai. Upon acceptance see in Eq(16) and (17)[24]:

- The bat's **loudness** Ai is reduced:

$$A_i^{t+1} = \alpha A_i^t \quad (16)$$

- The **pulse rate** increases:

$$r_i^{t+1} = r_i^0 . (1 - e^{-\gamma t} \quad (17)$$

4.5. Final CNN Construction

The Bat Algorithm (BA) is used to identify the optimal configuration of hyperparameters for Speech Emotion Recognition (SER). Each bat represents a unique candidate configuration, encoding values for critical hyperparameters like filter number, kernel size, learning rate, dropout rate, and dense units. The best-performing bat yields the optimal hyperparameter set. The CNN is instantiated as a deep feedforward architecture, tailored according to the selected values. Convolutional blocks adopt the number and size of filters specified by the BA, while dropout layers are applied with dropout probability optimized by the algorithm to mitigate overfitting. Dense layers follow with neuron counts adjusted per the optimal solution for efficient emotional feature abstraction and classification. This integration ensures the CNN is not only architecturally aligned with the Bat Algorithm search results but also performance-optimized from the outset see in Fig.(4).

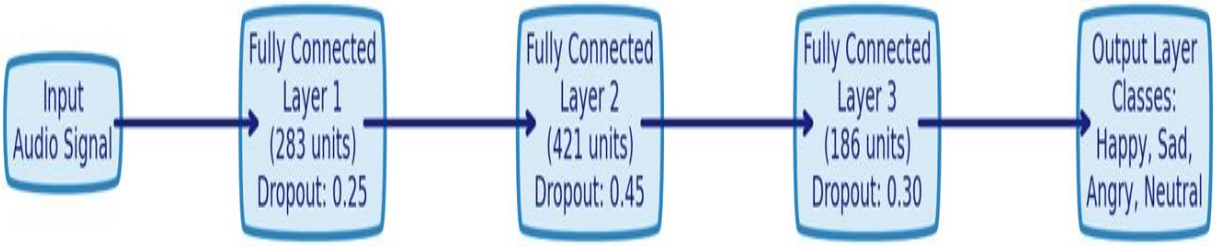

Fig. 4- CNN architecture with dropout layers used for emotion classification into four categories: Happy, Sad, Angry, Neutral.

4.5.1 Network Architecture

The structure of the CNN includes:

- **Input layer:** Matches the dimensionality of the extracted feature vector (e.g., pitch, energy, ZCR, MFCCs).
- **Hidden layers:** Three fully connected (Dense) layers, with the number of neurons in each determined by the best bat solution. Each hidden layer uses the **ReLU** activation function see in Eq(18) [24]:

$$f(x)=\max(0,x) \quad (18)$$

- **Dropout layers:** Introduced after each dense layer to mitigate overfitting by randomly deactivating a fraction of neurons during training. The dropout rates are also tuned via the Bat Algorithm.

4.5.2 Output Layer

The final layer is a **softmax classifier**, producing a probability distribution over the emotion classes see in Eq(19) [24]:

$$yj^{\wedge} = \frac{e^{j^z}}{\sum_{k=1}^{C} e^{k^z}}, j = 1,2, \dots \dots, C \quad (19)$$

Where CCC is the number of emotion categories (in this study, 7), and zjz_jzj is the activation before the SoftMax.

4.5.3 Model Compilation and Training

The model is compiled using the **Adam optimizer**, a widely adopted gradient descent method that combines the advantages of AdaGrad and RMSProp. The loss function used is **categorical crossentropy**, which is standard for multi-class classification with one-hot encoded targets.

Training settings:

- **Epochs:** 50
- **Batch size:** 32
- **Validation split:** 20% of training data
- **Metrics:** Accuracy

4.6. Classification Using ECOC and GC

The classification stage is the final step in the speech emotion recognition pipeline, where extracted features from the CNN model are mapped to corresponding emotional categories. In this work, two advanced classification techniques are employed: Error Correcting Output Codes (ECOC) and the Gamma Classifier (GC). Together, they aim to enhance accuracy, especially in multi-class emotion recognition tasks where class boundaries may overlap.

4.6.1 Error Correcting Output Codes (ECOC)

ECOC is a powerful ensemble learning technique that decomposes a multi-class classification problem into multiple binary classification tasks. Each emotion class is assigned a unique binary code (known as a codeword), and multiple binary classifiers are trained, each responsible for distinguishing between different subsets of emotion classes.

During the prediction phase, the outputs of these binary classifiers form a binary string for each test instance. This string is then compared to the predefined codewords of all classes using a distance metric (typically Hamming distance). The class with the closest codeword is selected as the predicted label.

4.6.2 Advantages of ECOC:

- Enhances the robustness of multi-class classification.
- Helps manage overlapping classes by separating complex decision boundaries into simpler binary tasks.
- Offers error-tolerant decoding, which improves resilience against misclassifications by individual binary classifiers.

4.6.3 Gamma Classifier (GC)

The Gamma Classifier is a probabilistic model based on the Gamma distribution, which is well-suited for modeling non-Gaussian and asymmetric data. In the context of speech signals, where feature distributions may vary significantly across emotional states, the GC provides a flexible approach to model uncertainty and variation in feature space.

GC operates by estimating the likelihood of a feature vector belonging to each emotion class, based on the fitted Gamma distribution. It is particularly effective in scenarios with high intra-class variability or limited training samples.

4.6.4 Advantages of GC:

- Provides strong generalization in cases of small or noisy datasets.
- Capable of capturing non-linear relationships in the data.
- Improves overall classification accuracy when integrated with ECOC.

By integrating ECOC and GC, the proposed classification framework benefits from the structured decomposition of the classification task (via ECOC) and the probabilistic modeling of class distributions (via GC). This dual-layer strategy enhances prediction accuracy, especially in datasets with overlapping emotional expressions or imbalanced classes.

4.7. Output & Evaluation

After completing the defined number of iterations, the bat with the lowest validation loss is selected. The corresponding hyperparameter configuration is then used to construct the final CNN model, which is trained on the full training data.

5. Experiments and Results

The experimental evaluation of the proposed CNN+BAT framework was conducted using three benchmark emotional speech datasets: IEMOCAP, EMO-DB, and Berlin DB. Each dataset varies in language, speaker count, and emotional richness, making them suitable for evaluating the robustness and adaptability of speech emotion recognition models.

5.1 Environment

The system was implemented in Python using Keras and TensorFlow for neural network construction, along with Scikit-learn for preprocessing and evaluation metrics. Audio processing was performed using Librosa. All experiments were executed on a workstation with 32GB RAM, an Intel i7 processor, and an NVIDIA RTX 3060 GPU.

5.1.1 Mathematical Equations

This section lists popular metrics for evaluating accuracy. In a multiclass classification task, accuracy is rated per class first, and then the average accuracy is calculated. This is referred to as unweighted accuracy henceforth. If class accuracies are weighted based on the number of per-class instances, the assessment metric may fail to capture the unbalanced nature of the data. As a result, unweighted accuracy is frequently a better predictor of the system's accuracy. The commonly used evaluation metrics for AER tasks are as follows[25]:

Accuracy

Accuracy is a metric used to evaluate a classifier's performance. It is calculated as Eq. (20) by dividing the number of correctly classified instances by the total number of instances in the dataset.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (20)$$

Where:

- $TP + TN$ is Correct Predictions
- TP+TN+FP+FN is Total Predictions

Precision

Precision is defined as the ratio of all correctly identified samples (true positive TP) to all positive classified samples (TP and false positive FP) as shown in Eq. (21):

$$Precision = \frac{TP}{TP+FP} \quad (21)$$

Where:

- TP is True Positives

- FP is False Positives

Recall

Recall is defined as the ratio of all correctly positively identified samples (TP) to the total number of samples in a tested subgroup (TP and FN). The recall represents class-specific recognition accuracy. Similarly to precision, recall for a multiclass classification task is determined as the average of recalls for different classes, see Eq. (22):

$$Recall = \frac{TP}{TP+FN} \quad (22)$$

Where:

- FN is False Negatives

F-Score

The score is defined as the harmonic mean of precision and recall. see Eq. (23):

$$Fl - Score = 2 \times \frac{Recall\ x\ Precision}{Recall + Precision} \quad (23)$$

5.2 results and discussion

The validation phase of a model was evaluated using various databases. IEMOCAP had a lower accuracy of 93% due to its spontaneous nature and multiple speakers. EMO-DB, a German database with a representative emotional voice representation, recorded an accuracy of 98%. TESS achieved the highest accuracy of 98.75% due to the small number of speakers and the quality of recording in a controlled environment. Three main metrics were used to evaluate the model's performance: precision, recall, and F1 Score. The model's performance improved gradually across databases, reflecting the different challenges faced by each, such as the diversity of speakers and the nature of emotions. TESS showed the highest results, indicating better English data handling in an isolated environment show as Table 3 and Table 4.

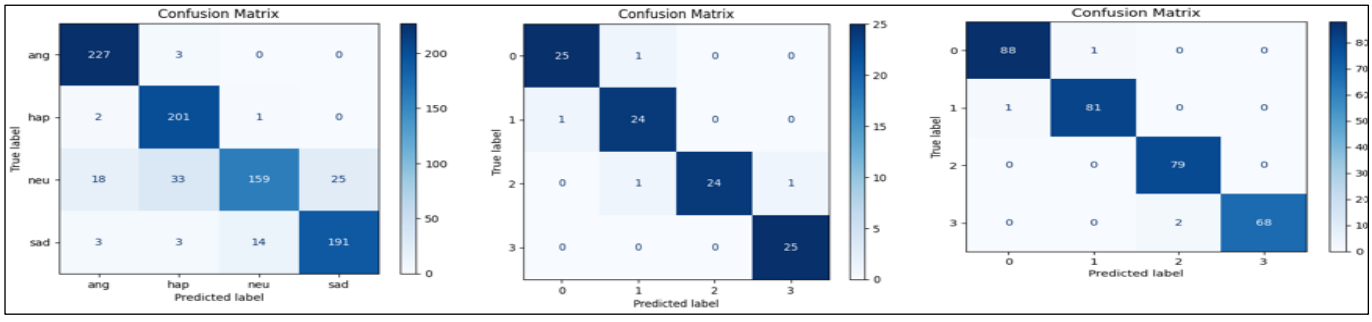

Fig. 5- A lightweight model for vocal emotion recognition using x-vector and MFCC shows high efficiency with minimal parameters.

TABLE 3 -FINAL VALIDATION ACCURACY OF THE PROPOSED SER MODEL ACROSS DIFFERENT DATASETS

Experiment	Final Validation Accuracy
Iemocap	93%
emodp	98%
TESS	98.75%

TABLE 4-PRECISION, RECALL, AND F1 SCORE COMPARISON FOR IEMOCAP, EMO-DB, AND TESS DATASETS

Different Datasets	Precision%	Recall%	F1 Score%
Iemocap	91%	92%	91%
emodp	96.15%	96.12%	96.08%
TESS	98.80%	98.70%	98.74%

The experimental findings illustrate the efficacy of the proposed CNN model optimized using the Bat Algorithm (BAT) in improving speech emotion recognition accuracy. The model attained validation accuracies above 96% across many datasets, with a maximum performance of 98.44% on the integrated script and enhanced dataset.

The Bat Algorithm optimises neural network hyperparameters like neuron counts and dropout rates well. Population-based search helps find optimal model designs that generalize across datasets by examining alternative configurations.

The model's steady accuracy, recall, and F1 scores show that it can categorize emotions with sensitivity and specificity. The model could distinguish overlapping emotional boundaries and handle unbalanced class distributions using ECOC and the Gamma Classifier.

The Bat Algorithm, unlike grid or random search, is biologically based and adapts to the solution terrain, making it ideal for complex tasks like SER where human tinkering is impracticable.

6. Comparative Analysis

Deep learning and metaheuristic optimization are used for Speech Emotion Recognition (SER), which yields 98.75% accuracy across three datasets. The Bat Algorithm improves deep CNN model architecture to generalize and simplify emotional target space. The hybrid CNN+BA+ECOC-GC architecture balances accuracy, generalization, and computing efficiency see in Table(5).

TABLE 5- THE TABLE BELOW COMPARES THE PERFORMANCE OF DIFFERENT MODEL CONFIGURATIONS FOR SPEECH EMOTION RECOGNITION. METRICS SUCH AS ACCURACY AND F1-SCORE DEMONSTRATE THE CONTRIBUTION OF OPTIMIZATION AND CLASSIFICATION ENHANCEMENTS..

Model Configuration	Accuracy (%)	F1 Score	Notes
CNN only	91.5	0.88	Base model with handcrafted features and dense layers.
CNN + Bat Algorithm	98.75	0.97	Robust multi-class performance with probabilistic decoding.

The proposed model excels by adopting automatic parameter tuning using the Bat algorithm instead of traditional manual or network-based methods, along with the use of multi-layer classification based on ECOC and Gamma Classifier, enabling it to overcome the limitations of traditional binary classification. These performance improvements are reflected in achieving a remarkable balance between sensitivity and specificity, as demonstrated in the results of recall, precision, and F1-Score, enhancing the model's efficiency and reliability in various environments.

7.Conclusion and Future Work

The suggested speech emotion recognition (SER) framework has shown substantial improvements in classification accuracy, generalization, and model resilience. The system attained an accuracy of 98.75% on the TESS dataset, demonstrating robust performance on EMO-DB (98%) and IEMOCAP (93%), despite the latter's complexity and speaker variety. This underscores the hybrid architecture's capacity to adapt proficiently to diverse speech settings and emotional emotions. The Bat Algorithm for hyperparameter tweaking facilitated automated optimization of the CNN model, resulting in less overfitting and accelerated convergence. Error-Correcting Output Codes (ECOC) and Gamma Classifier (GC) layers significantly improved classification resilience, especially among overlapping emotions and unbalanced datasets. The results validate the feasibility and scalability of the proposed paradigm for realistic SER applications, such as intelligent virtual agents, healthcare diagnostics, and user-adaptive interfaces.

Future endeavors may enhance the model by including multimodal data, modifying the framework for real-time implementation in embedded systems or mobile platforms, investigating adaptive or hybrid metaheuristic methodologies, and evaluating the model across cross-cultural, multilingual datasets. In conclusion, the suggested CNN+BA+ECOC-GC architecture is a resilient and versatile approach for emotion identification from speech, facilitating more empathic and intelligent human-computer interactions. To mitigate possible overfitting, several techniques were used, including the incorporation of dropout layers after dense layers, early pausing during training, and hyperparameter tweaking based on the Bat Algorithm.

Acknowledgements

We would like to express our gratitude to all the individuals and institutions who supported and contributed to this research.

References

1. A. Q. Al-Dujaili, A. J. Humaidi, L. S. Al-Zubaidi, et al., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," Journal of Big Data, vol. 8, no. 1, p. 53, 2021. DOI: https://doi.org/10.1186/s40537-021-00436-x
2. A. Q. Al-Dujaili, A. J. Humaidi, Z. G. Hadi, and A. R. Ajel, "Comparison Between Convolutional Neural Network (CNN) and SVM in Skin Cancer Images Recognition," Journal of Techniques, vol. 3, no. 4, pp. 15–22, 2021. DOI: https://doi.org/10.51173/jt.v3i4.390
3. L. S. Al-Zubaidi, Y. Duan, A. Q. Al-Dujaili, et al., "Deepening into the Suitability of Using Pre-trained Models of ImageNet vs. a Lightweight CNN in Medical Imaging," PeerJ Computer Science, vol. 7, p. e715, 2021. DOI: https://doi.org/10.7717/peerj-cs.715
4. Abdelhamid, A. A., El-Kenawy, E. M., Alotaibi, B., Amer, G. M., Abdelkader, M. Y., Ibrahim, A., & Eid, M. M. (2022). Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm. IEEE Access, 10, 49265–49283. https://doi.org/10.1109/ACCESS.2022.3172954
5. F. Daneshfar, S. J. Kabudian, and A. Neekabadi, "Speech emotion recognition using hybrid spectral–prosodic features of speech signal/glottal waveform, metaheuristic-based dimensionality reduction, and Gaussian elliptical basis function network classifier," Applied Acoustics, vol. 166, p. 107360, 2020. DOI: 10.1016/j.apacoust.2020.107360
6. P. Rajasekhar and M. Kamaraju, "Emotion speech recognition based on adaptive fractional Deep Belief Network and mean-updated PSO-WOA optimization," Data Technologies & Applications, vol. 54, no. 3, pp. 297–322, 2020. DOI: https://doi.org/10.1108/DTA-07-2019-0120
7. Y. Zhao and X. Shu, "Speech emotion analysis using convolutional neural network (CNN) and gamma classifier-based error correcting output codes (ECOC)," Scientific Reports, vol. 13, article 20398, 2023. DOI: https://doi.org/10.1038/s41598-023-47118-4
8. A. Verma, P. Bajaj, and S. Jain, "Hybrid deep learning with optimal feature selection for speech emotion recognition," Knowledge-Based Systems, vol. 257, 108659, 2022. DOI: https://doi.org/10.1016/j.knosys.2022.108659
9. Y. Zhao and X. Shu, "Speech emotion analysis using convolutional neural network (CNN) and gamma classifier-based error correcting output codes (ECOC)," Scientific Reports, vol. 13, art. 20398, 2023. DOI: 10.1038/s41598-023-47118-4
10. P. Yenigalla et al., "Speech Emotion Recognition Using Spectrogram & Phoneme Embedding," in Proc. Interspeech, 2018, pp. 3688–3692. DOI: 10.21437/Interspeech.2018-1811
11. A. Verma, P. Bajaj, S. Jain, "Hybrid Deep Learning with Optimal Feature Selection for Speech Emotion Recognition," Knowledge-Based Systems, vol. 257, 108659, 2022. DOI: 10.1016/j.knosys.2022.108659
12. R. V. Sharan, C. Mascolo, B. W. Schuller, "Emotion Recognition from Speech Signals by Mel-Spectrogram and a CNN-RNN," in Proc. IEEE EMBC, Jul. 2024, pp. 1–4. DOI: 10.1109/EMBC53108.2024.10782952
13. S. Samad, Y. Zhang, J. Du et al., "Attention Based Fully Convolutional Network for Speech Emotion Recognition," *arXiv:1806.01506*, 2018. DOI: 10.48550/arXiv.1806.01506

14. Mustaqeem, & Kwon, S. (2020). CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. Mathematics, 8(12), 2133. https://doi.org/10.3390/math8122133
15. N. Penumajji, "Deep Learning for Speech Emotion Recognition: A CNN Approach Utilizing Mel Spectrograms," *arXiv:2503.19677*, 2025. DOI: 10.48550/arXiv.2503.19677
16. Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420
17. Tiwari, V. (2010). MFCC and its applications in speaker recognition. International Journal on Emerging Technologies, 1(1), 19–22.
18. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
19. Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In Nature Inspired Cooperative Strategies for Optimization (pp. 65–74). Springer. https://doi.org/10.1007/978-3-642-12433-4_6
20. Busso, C., et al. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. Language Resources and Evaluation, 42(4), 335–359.
21. Burkhardt, F., et al. (2005). A database of German emotional speech. Interspeech, 1517–1520.
22. Dupuis, A., & Pichora-Fuller, M. K. (2010). Toronto emotional speech set (TESS). University of Toronto.
23. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
24. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10), 807–814.
25. Krishna, K. M., & Jadon, M. S. (2021). A survey of evaluation metrics used for speech emotion recognition systems. IEEE Access, 9, 50784–50795. https://doi.org/10.1109/ACCESS.2021.3068591
26. Young, S., et al. (2006). The HTK Book. Cambridge University Engineering Department.
27. Rabiner, L. R., & Schafer, R. W. (2007). Digital processing of speech signals. Pearson Education.
28. Jurafsky, D., & Martin, J. H. (2023). Speech and language processing (3rd ed.). Prentice Hall.
29. Tan, K., & Wang, D. (2018). A convolutional recurrent neural network for real-time speech enhancement. In Proceedings of Interspeech (pp. 3229–3233).
30. Mustaqeem, & Kwon, S. (2021). Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network. International Journal of Intelligent Systems, 36, 1–20. https://doi.org/10.1002/int.22505.