International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I2P119

Volume 74 Issue 2, 266-279, February 2026
© 2026 Seventh Sense Research Group®

Original Article

Optimizing Bullying News Detection on Social Media: An Ensemble Deep Learning Approach with Enhanced Optimization Algorithm

I. Anand Raj[1], R. Vidya[2], A. Martin[3]

[1,2]PG and Research Department of Computer Science, St. Joseph's College of Arts & Science (Autonomous), Affiliated to Annamalai University, Allpettai, Manjakuppam, Cuddalore, Tamil Nadu, India.
[3]Department of Computer Science, Central University of Tamil Nadu, Thiruvarur, Tamil Nadu, India.

[1]Corresponding Author : ammayesuraja75@gmail.com

Received: 18 March 2024 Revised: 16 May 2025 Accepted: 27 January 2026 Published: 14 February 2026

Abstract - *Cyberbullying is a growing digital threat that exploits online platforms to harm individuals, which can take place on social media, messaging environments, gaming, and mobile phones. Cyberbullying can result in deep psychiatric and emotional illnesses for those affected. Henceforth, there is a crucial necessity to develop an automated model for cyberbullying detection. Detecting cyberbullying is very challenging, due to the occurrence of complete affective content, which is also frequently sarcastic, and multimodal, such as audio, text, and image. Presently, Deep Learning (DL) techniques have attained extraordinary achievements in numerous tasks and are employed for the detection of cyberbullying for multimodal data. This paper presents the Multi-Layer Perceptron and Feature Extraction for Detecting Bullying in Multimodal Content (MLPFE-DBMMC) technique. The MLPFE-DBMMC technique aims to develop an effective multimodal cyberbullying detection framework by leveraging audio, textual, and visual inputs to recognize and identify harmful behaviour. Initially, the preprocessing stage is performed in multimodal formats such as image, audio, and text. For image preprocessing, the anisotropic diffusion method is employed for noise removal. The stationary wavelet transform-based noise removal is applied in the audio preprocessing. Moreover, the text preprocessing stage involves various levels such as lower case, tokenization, removal of stopwords, and stemming. For the feature extraction process, the MLPFE-DBMMC model implements the Contrastive Language-Image Pre-Training (CLIP) method for image, VGGish-based audio, and the generative pre-training-2 (GPT-2) technique is employed for text. After feature extraction from the multimodal data, output features of size (3672, 512) for images, (3672, 128) for audio, and (3672, 768) for text are obtained. Three multimodal features are fused via concatenation for final classification. The MLPFE-DBMMC model implements the correlation alignment loss (CORAL) multi-layer perceptron technique for the multimodal cyberbullying detection process. Finally, the improved artificial rabbit's optimization (IARO)-based hyperparameter tuning is performed to enhance the detection outcomes. A wide range of experiments of the MLPFE-DBMMC approach is performed under the MultiBully dataset. The comparison study of the MLPFE-DBMMC approach portrayed a superior accuracy value of 94.44% over existing techniques.*

Keywords - *Multimodal; Bullying Detection, Multi-Layer Perceptron, Improved Artificial Rabbits Optimization, Contrastive Language-Image Pre-Training.*

1. Introduction

Social media platforms are now deeply ingrained in daily lives. However, their rapid expansion online has unfortunately coincided with the rise in cyberbullying. While social media platforms can facilitate connections between people who share common ideas and interests, they also present risks for susceptible individuals by exposing them to harmful elements within cyberspace [1]. The misuse of social media has fuelled a surge in online violence, generating extensive user-generated data, such as text, speech, videos, images, and multimodal information. Bullying is an adverse societal issue that is escalating rapidly [2]. Generally, bullying behaviour is classified according to action, the environment, the visibility (overt and covert), its mode (direct as well as indirect), the damage caused (psychological and physical), and the environment where an event or situation takes place [3]. Multimodal cyberbullying is a social behaviour of bullying in digital environments, carried out subtly through direct or indirect ways that inflict both immediate and lasting emotional damage [4]. The continuity, reach, and rapid impact of such actions make cyberbullying more harmful than in-person bullying, leading to severe emotional health and wellness challenges to victims and leaving them feeling completely defeated [5]. Cyberbullying often leads to heightened emotional disorders, diminished self-esteem, frustration, depression, escalated rage, social isolation, and, in certain instances, the emergence of aggressive or suicidal tendencies. Technologies enable the bullies to remain unidentified, difficult to locate, and shielded from direct accountability [6].

To those targeted, cyberbullying appears intrusive and endless. Considering the emotional and psychological strain it causes, there is an urgent need to implement effective mechanisms to recognize and counter it. Scholars globally are striving to create innovative solutions to identify multimodal cyberbullying, address it, and minimize its occurrence across social media [7]. Advanced analytical techniques and computational frameworks are utilized for accurate assessment, examination, and representation in detecting multimodal cyberbullying. Over the past years, machine learning (ML) models have demonstrated excellent outcomes in identifying patterns of bullying behaviour. A few developed methods employ conventional supervised learning capabilities of logistic regression, naive Bayes, and support vector machines, among others [8]. These traditional techniques utilize statistical models to recognize patterns in the data. Simultaneously, they need to select and extract related bullying behaviour features from labelled data. To identify offensive or bullying information from the data, traditional methods leverage A Natural Language Processing (NLP) method for analyzing the text [9]. Even though it has capabilities, this method falls short in processing extensive data and extracting intricate features from it. To address these limitations, DL, a subfield of ML, is applied [10]. The DL technique enables automated feature extraction, which allows it to process extensive datasets and capture complex features from images or textual data. Compared to traditional models, DL methods function well with enhanced performances and recognition abilities of multimodal cyberbullying incidents [11].

This paper presents a Multi-Layer Perceptron and Feature Extraction for Detecting Bullying in Multimodal Content (MLPFE-DBMMC) technique. The MLPFE-DBMMC technique aims to develop an effective multimodal cyberbullying detection framework by leveraging audio, textual, and visual inputs to recognize violence accurately. Thbehaviour. The performance is performed in multimodal formats such as image, audio, and text. For image preprocessing, the anisotropic diffusion method is employed for noise removal. The stationary wavelet transform-based noise removal is applied in the audio preprocessing. Moreover, the text preprocessing stage involves various levels such as lower case, tokenization, removal of stopwords, and stemming. For the feature extraction process, the MLPFE-DBMMC model implements the Contrastive Language-Image Pre-training (CLIP) method for image, VGGish-based audio, and the Generative Pre-Training-2 (GPT-2) technique is employed for text. After feature extraction from the multimodal data, output features of size (3672, 512) for images, (3672, 128) for audio, and (3672, 768) for text are obtained. Three multimodal features are fused via concatenation for final classification. The MLPFE-DBMMC model implements the Correlation Alignment Loss (CORAL) multi-layer perceptron technique for the multimodal cyberbullying detection process. Finally, the Improved Artificial Rabbit's Optimization (IARO)-based hyperparameter tuning procedure is executed to enhance the detection outcomes. A wide range of experiments of the MLPFE-DBMMC approach is performed under the MultiBully dataset.

2. Related Works on Bullying Detection

Singh and Sharma [12] presented a technique for identifying social network cyberbullying through several processes. Data collection is the primary step, where the data is gathered using images, texts, videos, and audio. At first, text data is preprocessed through lemmatization, tokenization, spell correction, stemming, and PoS tagging techniques. Following that, preprocessing takes place, feature extraction is performed by Glove modelling and Log Term Frequency-Based Modified Inverse Class Frequency (LTF-MICF), and a Convolutional Dense Capsule Network (Conv_DCapNet) method is used for feature extraction. Mondol et al. [13] introduced an advanced Artificial Intelligence (AI)-driven approach for identifying and stopping cyberbullying among different social networks employing a multi-class classification method. Tackling the increasing intricacy and linguistic diversity of online violence, the study combines several ML and DL models. Singh and Sharma [14] proposed the hybrid model and Multimodality Decision Fusion Classifier (MMDFC) technique. Numerous necessary modalities are primarily collected from data. Then, the collected data is passed to a multimodal generation model, in which each modality is produced individually for each input. At first, the text modality is made through a hybrid Bidirectional LSTM-based Attention Hierarchical Capsule Network (BiLSTM-AHCNet).

Al-Khasawneh et al. [6] proposed a multimodal cyberbullying detection framework that integrates images, videos, comments, and temporal data using a Hierarchical Attention Network (HAN) model. Wang et al. [15] suggested a multi-stage BERT fusion architecture. It employs hierarchical embeddings, dual attention mechanisms, and additional features to enhance the recognition of cyberbullying content. This architecture integrates BERT embeddings. Kahate and Raut [16] presented a DL model to address cyberbullying incidents through social media analysis, identifying and mitigating harmful content and supporting affected individuals. Initially, this system gathers tweets posted by consumers, captures metadata, and examines language features for training an LSTM-based CNN, which helps in tweets prefiltering. The filtered tweets are passed using an NLP engine that helps identify sentiments from texts. Sentiment data and word embedding abilities are utilized for detection. Samee et al. [17] introduced the incorporation of emotional features, word embeddings, and Federated Learning (FL) models to overcome the difficulties of centralized data processing and users' privacy issues. Word embeddings extract contextual information and semantic relations, allowing for a more detailed understanding of text data. FL, a decentralized learning model, provides compelling solutions to centralize sensitive user information for training among distributed devices, maintaining privacy while utilizing collective intelligence.

3. Methodological Approach

In this manuscript, a novel MLPFE-DBMMC approach is proposed. The MLPFE-DBMMC model aims to develop an effective multimodal bullying detection framework by leveraging visual, audio, and textual inputs to identify and classify harmful behaviour precisely. For this purpose, the MLPFE-DBMMC model has input data preparation, a multimodal feature extraction layer, classification via CORAL-MLP, and model enhancement via fine-tuning. Figure 1 indicates the entire working flow of the MLPFE-DBMMC model.

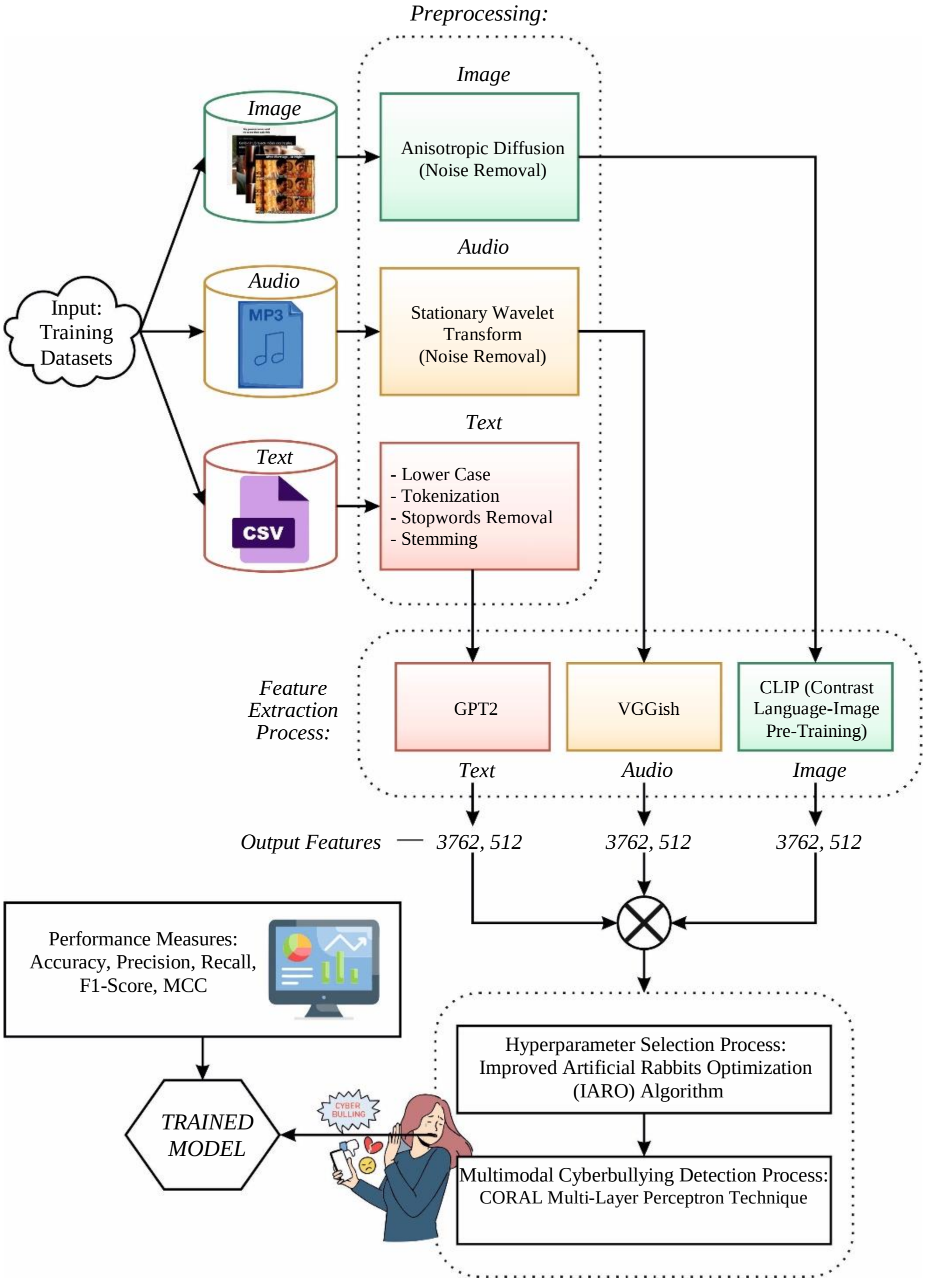
Preprocessing:
Image
Anisotropic Diffusion (Noise Removal)
Audio
Stationary Wavelet Transform (Noise Removal)
Text
- Lower Case
- Tokenization
- Stopwords Removal
- Stemming
Image
Audio
MP3
Text
CSV
Input: Training Datasets
Feature Extraction Process:
GPT2
VGGish
CLIP (Contrast Language-Image Pre-Training)
Text
Audio
Image
Output Features
3762, 512
3762, 512
3762, 512
Performance Measures: Accuracy, Precision, Recall, F1-Score, MCC
TRAINED MODEL
CYBER BULLING
Hyperparameter Selection Process: Improved Artificial Rabbits Optimization (IARO) Algorithm
Multimodal Cyberbullying Detection Process: CORAL Multi-Layer Perceptron Technique

Fig. 1 Entire workflow of MLPFE-DBMMC model

3.1. Algorithm: Multimodal Cyberbullying Detection using MLPFE-DBMMC Technique

Algorithm 1 describes the steps involved in the MLPFE-DBMMC model, capturing the preprocessing, feature extraction, fusion, classification, and optimization stages.

```
Algorithm 1: MLPFE-DBMMC technique
Input:
  • Multimodal dataset comprising image, audio, and text.
Output:
  • Predicted output ∈ {bullying, non-bullying}
Step 1: Preprocessing
1.1. Image Preprocessing
  • Apply anisotropic diffusion filtering to eliminate noise.
1.2. Audio Preprocessing
  • Apply Stationary Wavelet Transform (SWT) for denoising.
1.3.Text Preprocessing
  • Convert to lowercase → Tokenize → Remove stopwords → Apply stemming.
Step 2: Feature Extraction
2.1. Image Feature Extraction
For each preprocessed image:
  • Extract feature vector by utilizing CLIP → Output dim: (512)
2.2. Audio Feature Extraction
For each denoised audio:
  • Extract feature vector using VGGish → Output dim: (128)
2.3. Text Feature Extraction
For each preprocessed text:
  • Extract feature vector using GPT-2 → Output dim: (768)
Step 3: Feature Fusion
For each sample:
  • Concatenate features → Final shape: (1408)
  • Stack all → Form feature matrix
Step 4: Classification using CORAL-MLP
4.1. Network Architecture
  • Input layer: 1408 neurons
  • Hidden layers: Dense + ReLU + Dropout (rate = 0.3)
  • Output layer: Softmax for binary classification
  • CORAL is integrated into the loss function to align domains and improve generalization.
Step 5: Hyperparameter Optimization (IARO)
5.1. Initialize IARO Parameters
Populace size, maximum iteration, and fitness function based on validation accuracy.
5.2. Optimization Process
Repeat until convergence is reached:
  • Generate candidate solutions (network parameters)
  • Compute using the fitness function
  • Update positions of rabbits (solutions) based on IARO equations
  • Select the best candidate
Step 6: Inference
Use the optimized CORAL-MLP for predicting class labels:
Return the output
```

3.2. Stage I: Initial Processing Stage

At the primary stage, the preprocessing phase is performed on multimodal data, such as image, audio, and text. For image preprocessing, the anisotropic diffusion model is applied for noise removal. The stationary wavelet transform-based noise removal is used in audio preprocessing. The text preprocessing phase includes various levels, such as lower case, tokenization, stopwords removal, and stemming.

3.2.1. Image Denoising via Anisotropic Diffusion

An ADF filter assists in eliminating noise by maintaining boundaries and edges [18]. Unlike conventional isotropic filters, which smooth the complete image consistently, this model progressively fine-tunes the filtering procedure according to the image content.

The basic concept behind ADF is to execute diffusion along with edges. In areas with stronger gradients (edges), it is smaller, which avoids extreme smoothing and maintains the edge data. In areas with lower gradients (smoother regions), the diffusion coefficient is largest, permitting additional smoothing to decrease noise. Accurately, the process of anisotropic diffusion is defined and upgrades the pixel values through iteration.

$$\frac{\partial Im}{\partial n} = div(c(p,q,n)\nabla Imor\frac{\partial Im}{\partial n} = div(c(|\nabla mI|)\nabla Im) \quad (1)$$

Whereas Im denotes the input image, characterizing the i pixel intensities. ∇Im denotes the spatial and image gradients concerning p and q coordinates. $\frac{\partial Im}{\partial n}$ signifies the changes in image intensity in time. div denotes the divergence operator. It serves on the image gradient and estimates how the gradients are converging or diverging at every pixel. $c(Im)$ denotes a function of diffusion coefficient (∇Im).

The function $c(Im)$ is an essential feature of anisotropic diffusion. In areas with stronger gradients, the diffusion coefficient is generally smaller and aids in maintaining sharper edges. On the other hand, in areas with lower gradients (for example, smoother regions), the diffusion coefficient is greater, permitting for smoothing to lower noise.

3.2.2. Audio Signal Decomposition using SWT

SWT provides various benefits in signal processing, including computational efficiency and processing non-stationary signals [19]. It maintains the new signal length at every wavelet scale, permitting a 'sparse impulsive representation' in the wavelet domain. The time-frequency localization properties of SWT make it more robust against noise than conventional Fourier transform methods. This presents increased opportunities for the separation of noise and signals. Unlike DWT, SWT shows translation invariance of the wavelet coefficients, thus improving its performance in signal analysis. SWT decomposes the signal into lower and higher frequency bands utilizing a series of filters. To decompose the signals into detail and approximate coefficients, the SWT uses Eqs. (3) and (4).

$$A_i(t) = \sum_{k=0}^{L} A_{i-1}\,(k)\varphi_i(t-k) \qquad (2)$$

$$D_i(t) = \sum_{k=0}^{L} A_{i-1}\,(r)\psi_i(t-k) \qquad (3)$$

Whereas the terms $D_i(t)$ and $A_i(t)$ characterize the detail and approximation coefficients, respectively. Where L signifies the filter length, the terms $\varphi_i(t-k)$ and $\psi_i(t-k)$ indicate the lower pass and higher filters, respectively.

3.2.3. Text Data Handling

The preprocessing stage is implemented for processing the gathered data to enhance its value in classification [20]. In general, text data has multidimensional and unstructured patterns and requires a data cleaning procedure.

Case Folding

Case Folding is a procedure that normalizes text, including several uppercase and lowercase letters. A sample of a word before and after the case folding step: Tempat -> tempat and Tahun -> tahun.

Tokenization

It is a method where sentences are segmented into tokens or words. This stage is essential for text analysis as it transforms the unstructured text into a structured pattern that ML methods can more easily process. A sample of the tokenization procedure: 'tempat wisata yang sejuk dan menyenangkan', turn out to be 'tempat', 'wisata', 'yang', 'sejuk', 'dan', 'menyenangkan'.

Stopword Removal

At the stopword removal phase, general words that are considered to have no meaning and, more frequently than not, appear in massive quantities in a content report should be cleared. For example: 'dan,' 'atau,' 'yang,' and others.

Stemming

It is applied for converting words to their base form. Here, the Sastrawi library, which contains a dictionary of Indonesian base words, is used. For example, the term 'menyenangkan' becomes the root term 'senang'.

3.3. Stage II: Feature Representation Model

For the feature extraction process, the MLPFE-DBMMC model utilizes CLIP for images, VGGish-based audio, and the GPT-2 method for text.

3.3.1. Image Feature Encoding using CLIP

CLIP method learning visual representation utilizing language supervision. Specifically, a batch of N images is used to forecast the corresponding method by employing the input [21]. To accomplish this, the CLIP method utilizes a vision-encoded network E_i for learning a visual representation and E_t to learn a representation. Then, the CLIP method forecasts a similarity matrix $S \in \mathbb{R}^{N\times N}$, Where Every row denotes the likelihoods of matching a single image to every N. This model is enhanced to increase the similarity score of N positive pairs and decrease the similarity score of $N2-N$ negative pairs. Figure 2 specifies the architecture of the CLIP method.

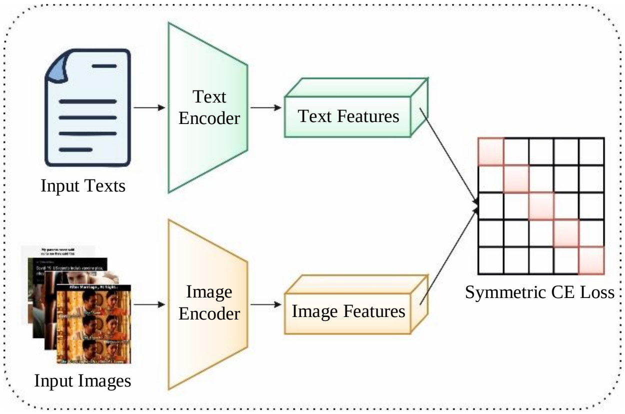

Fig. 2 Architecture of the CLIP method

The C category, $\{1,2,\dots,C\}$, CLIP employs a pre-defined prompt, for example ''an image of a [CLASS]'', to calculate the input of T. Subsequently, the semantic features of each class are created to employ the encoder network that is $F_t = E_t(T) \in R^{C\times d}$, d signifies the dimension of the feature. Notably, a batch of input images depicts $I \in \mathbb{R}^{B\times H\times W\times 3}$, where B, H, and W represent the batch size, width, and height of images, which means $F_i = E_i(I) \in \mathbb{R}^{B\times d}$. Once the classification probability matrix is attained,

$$P = softmax(F_i F_t^T/\tau) \qquad (4)$$

Now F_i and F_t refer to L2-normalised. τ indicates a learnable parameter. The Softmax layer is implemented for the size of the class as $P \in \mathbb{R}^{B\times C}$. Every row of P signifies the likelihood of allocating a single image to each probable class.

$$\hat{Y} = \arg\max(P) \qquad (5)$$

Audio Embedding via VGGish

The VGGish method is a pre-trained DL framework suitable for audio embedding elimination. It is exceptionally trained on an enormous YouTube audio dataset [22]. This pre-training enables VGGish to acquire vital features from raw audio signals that are employed for several audio classification issues. VGGish has a feature extractor based on transfer learning. Now, the pre-trained weights of VGGish are utilized to remove embeddings, meaningful representations, from the input voice instances. These embeddings shorten the intricate audio data into a lower-dimensional, fixed-length vector that encapsulates speaker-specific and discriminative features. Before feeding raw audio to VGGish, accomplish vital preprocessing stages. This assures compatibility with the structure of the model and evades inconsistency in embedding extraction. The model converts the raw audio into a spectrogram representation of size 96×64, a visual representation of audio's frequency content through time that VGGish is intended to process effectively. VGGish employs pre-trained proficiency for feature extraction; this method takes advantage of the capability of the technique to acquire speaker-specific features from audio data. This decreases the computation cost and training period to train a model. The eliminated embeddings act as input to a subsequent DL model, which means LSTM and CNN are intended explicitly for voice spoofing classification. The structure of the VGGish method contains numerous max pooling, convolution, and 3 FC layers. Figure 3 indicates the Framework of the VGGish model.

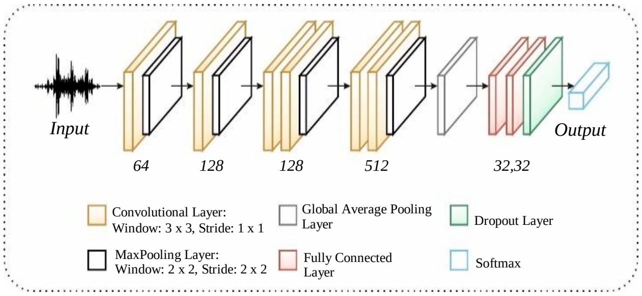
Input
Output
64
128
128
512
32,32
Convolutional Layer: Window: 3 x 3, Stride: 1 x 1
MaxPooling Layer: Window: 2 x 2, Stride: 2 x 2
Global Average Pooling Layer
Fully Connected Layer
Dropout Layer
Softmax

Fig. 3 Framework of the VGGish model

Textual Representation with GPT-2

GPT-2 is established to accomplish sophisticated outcomes like semantic similarity, natural language inference, text classification, and question answering [23]. GPT-2 is an autoregressive transformer that comprises 12, 24, 36, or 48 blocks of decoder based on the model size. Compared with BERT, it comprises encoder blocks only, a GPT-2 stacking block of the decoder. Additionally, a significant property of GPT-2 is its autoregressivity. Another key feature of GPT-2 is its capability to learn a downstream task using a zero-shot method, without any requirement for parameter tweaking or alterations to the structure of the process. GPT-2 is trained with a slightly altered language modelling objective: rather than approximating the conditional distribution $P(output|input)$, GPT-2 estimates $P(output|input, task)$. Nevertheless, distinctly modelling this at the architectural level, the task is prepended to the sequence of input.

To enhance the accuracy and robustness of the classification task, features extracted from three different modalities, image, audio, and text, are integrated using a feature-level fusion strategy. Each modality contributes unique and complementary information that, when combined, provides a richer representation of the input instance. In this work, a simple yet effective concatenation operation is employed to fuse the feature vectors obtained from each modality. Let $f_v \in R^{d_v}$: visual (image) feature vector, $f_a \in R^{d_a}$: audio feature vector, and $f_t \in R^{d_t}$: textual feature vector. The fused feature vector f_{fusion} is evaluated by using Eq. (6).

$$f_{fusion} = [f_v \parallel f_a \parallel f_t] \in R^{d_v+d_a+d_t} \qquad (6)$$

Where [∥] denotes the concatenation operation along the feature dimension.

3.4. Stage III: CORAL-Based Neural Classifier

Next, the MLPFE-DBMMC model applies the CORAL multi-layer perceptron method for multimodal cyberbullying detection. To define projected consistent rank logits (CORAL) structure for ordinal regression [24]. Notably, a training data $D = \{x_i, y_i\}_{i=1}^{N}$, a rank y_i is increased by $K-1$ dual labels $y_i^{(1)}, \dots, y_i^{(K-1)}$ like $y_i \in \{0,1\}$ signifies whether y_i surpasses rank r, for example, $y_i = 1\{y_i > r_k\}$. The indicator function 1 is 1 when the inner condition is true and 0 otherwise. According to the dual task responses, the forecast rank label for the input x_i is attained through $h(x_i) = r_q$. The rank index q is specified.

$$q = 1 + \sum_{\mathrm{k}=1}^{K-1} f_{\mathrm{le}}\,(x_i), \qquad (7)$$

Now $f_{\langle}(x_i) \in \{0,1\}$ signifies kth dual classifier in the layer of output. In $\{f_c\}_{\mathrm{k}=1}^{K-1}$ reflects the ordinal data and rank-monotonic, $f_1(x_i) \geq f_2(x_i) \geq \cdots \geq f_{K-1}(x_i)$ that safeguards consistent validation. To accomplish rank-monotonicity and ensure dual classifier consistency, the $K-1$ dual tasks share similar weighted parameters but have independent bias units.

MLP is an ANN that includes input, output layers, and a Hidden Layer (HL). The layer consists of many neurons; every neuron alters the weighted input into the output by an activation function. The neuron counts in the input layer are the counts of input features set to m, the counts of neurons in HL set to n, and the counts of neurons in the output layer are 1. In the forward propagation process, the feature vector of output $x \in R^m$ is passed to HL through the input layer.

$$z^{(1)} = W^{(1)}x + b^{(1)} \qquad (8)$$

Now $W^{(1)} \in R^{n\times m}$ indicates the weighted matrix from the input to the hidden layer, $b^{(1)}$ denotes the bias vector of HL and $z^{(1)}$ depicts the linear integration of input and output to HL.

The calculation procedure from HL to the output layer:

$$z^{(2)} = W^{(2)}a^{(1)} + b^{(2)} \qquad (9)$$

Here, $W^{(2)} \in R^{1\times n}$ denotes a weighted matrix from HL to the output layer, $a^{(1)}$ indicates the output of HL, $b^{(2)}$ refers to the offset of the output layer and $z^{(2)}$ depicts the linear integration from HL to the output layer.

3.5. Loss Function

Assume W indicates the weighted parameters of NN, excluding the bias unit of the last layer. The layer of penultimate, whose output is specified as $g(x_i, W)$, sharing a single weight with every node in the previous layer of output; $K-1$ independent biased units are later included to $g(x_i, W)$, like $\{g(x_i, W) + b_k\}_{\mathrm{k}=1}^{K-1}$ indicates the inputs to equivalent dual classifiers in the last layer. Suppose

$$\sigma(z) = \frac{1}{1+\exp(-z)} \tag{10}$$

Refers to the function of the logistic sigmoid. The forecast experiential likelihood for task k is described.

$$\hat{P}(y_i^{(\mathrm{k})} = 1) = \sigma(g(x_i, W) + b_k \tag{11}$$

The training model decreases the loss function

$$L(W,b) = -\sum_{i=1}^{N}\sum_{e=1}^{K-1}\lambda^{(\mathrm{k})}\left[\log\left(\sigma(g(x_i,W) + b_e)\right)y_i^{(\mathrm{k})} + \log\left(1-\sigma(g(x_i,W)+b_e)\right)\left(1-y_i^{(\mathrm{k})}\right)\right] \tag{12}$$

The weighted cross-entropy of $K-1$ dual classifiers.

$$f_{\mathrm{k}}(x_i) = 1\left\{\hat{P}\left(y_i^{(\mathrm{k})} = 1\right) > 0.5\right\}. \tag{13}$$

Now λ depicts the weight of loss related to the kth classifier. Specifically, $\lambda^{(k)}$ indicates the significant parameter for task k.

3.6. Stage IV: Fine-Tuned Optimisation Model

Finally, the IARO-based hyperparameter tuning model is performed. This model carries its motivation from dual survival strategies that utilize wild rabbits [25]. This approach enhances the likelihood of survival of rabbits while challenged by predators. The primary approach involves eating grass near their nests, decreasing the risk of identification.

The next approach, named "random hiding," involves transferring rabbits to several adjacent holes, thus increasing their cover-up. Furthermore, a key aspect of the model entails the transition between exploitation and exploration methods.

$$P_x(t+1) = \vec{P}_y(t) \dagger LR.MV.[\vec{P}_x(t) - \vec{P}_y(t) + round(0{,}5 + (0{,}5 + c1).s_1) \tag{14}$$

$$x, y = 1{:}\, nPop, \cdot\, x \neq y; s_1\tilde{}\, N(0{,}1)$$

$$LR = \left(e - e\left(\frac{t-1}{iteration}\right)\right)^2 \times \sin(2\pi c_2) \tag{15}$$

$$MV_{(z)} = \begin{cases} 1 if z = p(i)_{z=1,\ldots/q; i=1,\ldots[c_3,q]; p=rand(q)} \\ 0 \ \ else \end{cases} \tag{16}$$

Now $\vec{P}_y(t)$ denotes the oldest site of $the\ x^{th}$ rabbit at time t, $\vec{P}_x(t+1)$ denotes the novel upgraded location of the artificial rabbit at time $(t+1)$, q signifies the variable counts of the problem, and $Iteration$ refers to the maximal iteration counts. $n\ Pop$ signifies the total population count of artificial rabbits, s_1 represents the standard normal distribution, LR implies calculating length and deviation feeding with MV being a mapping vector, and c_1, c_2 denote an arbitrary parameter in [0,1]. Here, the x^{th} Rabbit generates the y^{th} burrow.

$$\overrightarrow{BR_{x,y}}(t) \rightarrow= \overrightarrow{P_x}(t) + h.p.\overrightarrow{P_x}(t), x = 1, \ldots, nPop, \cdot\, y = 1, \ldots, m \tag{17}$$

$$p(P) = \begin{cases} 1\ if\ P == p(i) \\ 0\ else \end{cases} \tag{18}$$

Here, h denotes the parameter of concealment, which is progressively minimized from 1 to $1/Iteration$ linearly through the process of iteration, combining an arbitrary perturbation. $\overrightarrow{BR_{x,y}}(t)$ depicts an arbitrarily selected hole for the rabbit to search for protection from danger amongst m holes. The calculation for the random hiding behaviour of the rabbit.

$$\overrightarrow{P_x}(t \rightarrow +1) = \overrightarrow{P_x}(t) \rightarrow +LR.MV.\left(c_3 \times \overrightarrow{BR_{x,a}}(t)\right) \tag{19}$$

$$\overrightarrow{BR_{x,a}}(t) = \overrightarrow{P_x}(t) + h.\rho.\overrightarrow{P_x}(t) \tag{20}$$

The variable $\overrightarrow{BR_{x,a}}(t)$ denotes the arbitrary choice of a tunnel.

$$\overrightarrow{\rho_x}(t+1) = \begin{cases} \overrightarrow{\rho_x}(t)\underline{\quad\ } f\left(\overrightarrow{\rho_x}(t)\right) \leq f\left(\overrightarrow{\rho_x}(t+1)\right) \\ \overrightarrow{\rho_x}(t+1)\underline{\quad} f\left(\overrightarrow{\rho_x}(t)\right) > f\left(\overrightarrow{\rho_x}(t+1)\right) \end{cases} \tag{21}$$

Energy degradation coefficient: the primary stage usually prioritizes exploration, although the later phase concentrates on exploitation. Nevertheless, the ARO combines a search approach motivated by the energy degradation coefficient.

$$E(t) = 4\left(1 - \frac{t}{T}\right) In \frac{1}{c_4}; c_4 \in [0, \cdot\, 1] \tag{22}$$

While ARO established efficacy to address engineering and mathematical optimization concerns, it contains certain restrictions. It presents an IARO model by integrating the searching character of grey wolves from GWO to control the exploitation or exploration change of ARO. In the proposed IARO, a novel switching mechanism is developed for grey wolves' hunting behaviour. In the forest, they circulate their target until they are inside the specific vicinity of risk. This switching mechanism is calculated:

$$\vec{U}(t) = 2 \times \vec{\theta} \times c_5 - \vec{\theta} \tag{23}$$

Now, $\vec{\theta}$ depicts a parameter subject to a slight decrease from the primary value of 2 to a terminal value of 0, and c_5 denotes the stochastic variable in [0,1]. Furthermore, enhances the accuracy of algorithmic convergence and improves the resilience of the stochastic concealment stage.

$$Behaviour = \begin{cases} Exploration : Deviation\ feeding\ if\ |u| > 1 \\ Exploitation : Contingent\ concealing\ if\ |u| < 1 \end{cases} \tag{24}$$

The IARO model originates a fitness function (FF) to improve the performance of the classifier. It governs a positive numeral to specify the higher candidate solution performance. The classifier error rate reduction is designated as FF in this study, as shown in Eq. (25).

$$fitness(x_i) = ClassifierErrorRate(x_i)$$

$$= \frac{No.of\ misclassified\ samples}{Overall\ samples} \times 100 \tag{25}$$

4. Experimental Evaluation

In this section, the performance evaluation of the MLPFE-DBMMC model is examined on the dataset [4], available at https://github.com/Jhaprince/MultiBully?tab=readme-ov-file. The dataset comprises two kinds of modalities, such as text and image. The text data in code-mixed form includes continuous transitions between languages for multilingual users. The dataset samples are annotated with bully, sentiment, emotion, and sarcasm labels gathered from open-source Twitter and Reddit platforms.

The dataset initially contained 5,999 images; of these, 3,672 images were successfully preprocessed for further analysis. Among the preprocessed images, 2,112 were categorized as "Bully", denoting content related to bullying behaviour, whereas 1,560 were labelled as "Nonbully", signifying non-bullying content. This distribution presents a solid foundation for developing models aimed at bullying detection and content moderation. Table 1 presents sample texts. Figure 4 illustrates the original and preprocessed sample images. Next, Figure 5 demonstrates the steps involved in the preprocessing of the input audio samples and preprocessed audio samples. The number of parameters is given below:

- IntegerVar (lb=256, ub=512, name="num_hidden_1")
- IntegerVar (lb=128, ub=256, name="num_hidden_2")
- FloatVar (lb=0, ub=1, name="dropout_1")
- FloatVar (lb=0, ub=1, name="dropout_2")
- IntegerVar (lb=1, ub=30, name="epochs")
- IntegerVar (lb=512, ub=1024, name="batch_size")
- FloatVar (lb=0.00001, ub=0.0001, name="learning_rate")

Fig. 4 Original and preprocessed sample images

Table 1. Sample texts

Label	Text	Preprocessed Text
Bully	"Shivam @shivamishraa Girls be named Naina and then have eyes that do not work."	Shivam Shivamishraa, girl name Naina eye work.
Bully	"If the opposite of Con is Pro, Is Congress the opposite of Progress"?	Opposition to pro-congress opposes progress.
Nonbully	"Aaloo ke paranthe is the best breakfast. Omelette is the best breakfast, BONK Paranthe Omelette Poha."	Aloo ke paranth best breakfast omelett best breakfast bonk paranth omelett poha
Nonbully	"You find a new YouTuber. He is funny. All of his videos are funny. His last video was 4 years ago."	Find a new funny YouTube video from last year.

Figure 6 displays the label distribution of the MLPFE-DBMMC approach**,** classifying instances into two groups, namely Bully and Nonbully.

The dataset comprises 2,112 bully instances and 1,560 nonbully instances, denoting a moderate class imbalance, with the bully class being more predominant. Such an imbalance can impact model performance, potentially biasing the classifier towards the majority class (bully). Figure 7 shows the confusion matrices formed by the MLPFE-DBMMC technique on training and testing.

The outputs depict that the MLPFE-DBMMC model efficiently detects and identifies each class.

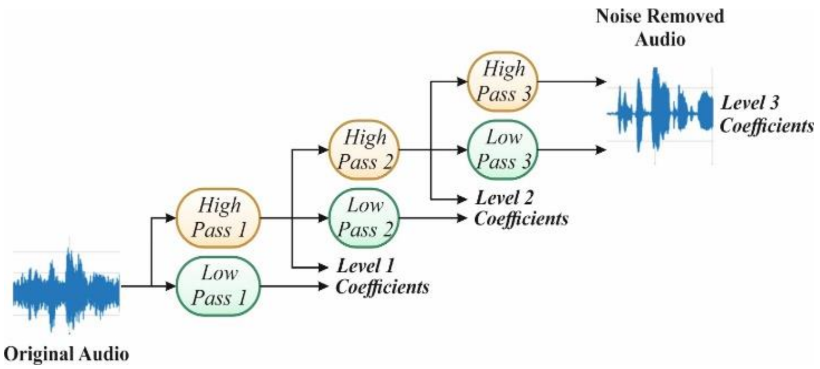

Fig. 5 Stages involved in the preprocessing of an audio sample

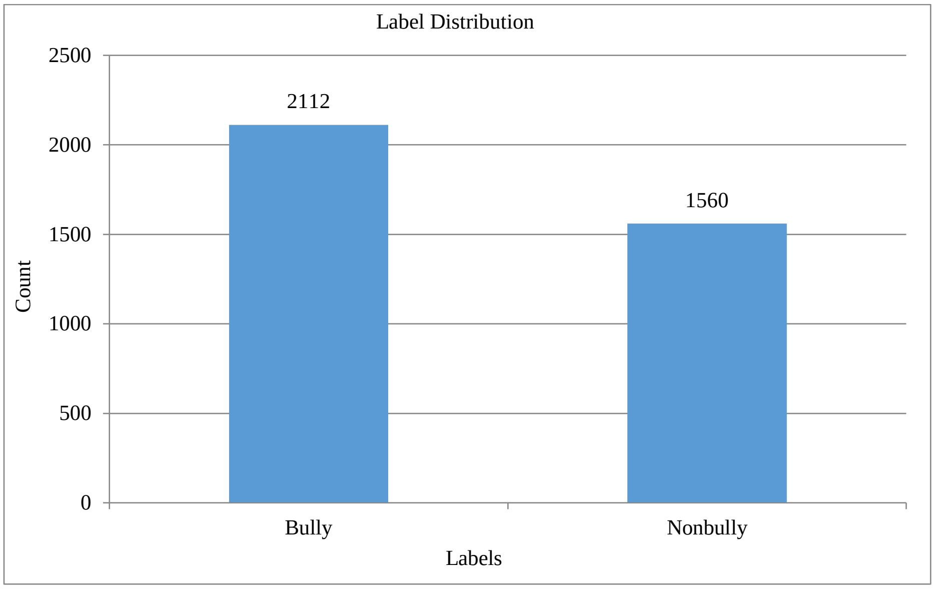

Fig. 6 Label distribution of the MLPFE-DBMMC model

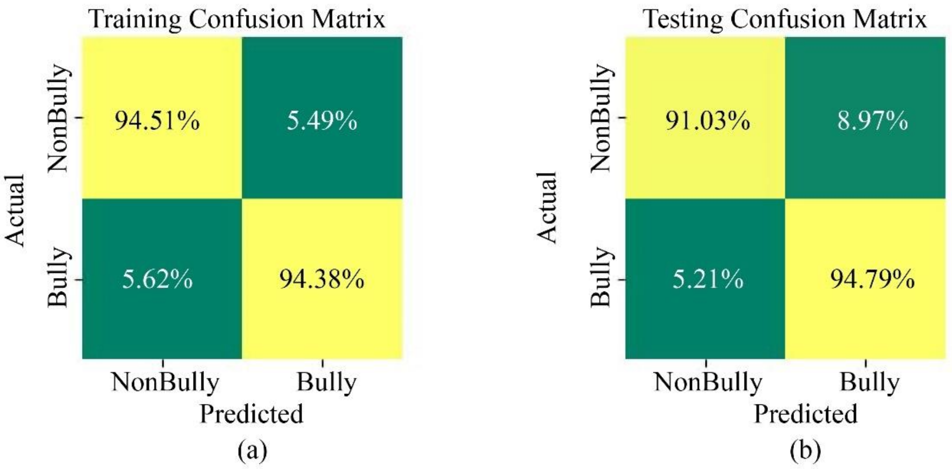

Fig. 7 Confusion matrix of MLPFE-DBMMC model (a) Training, and (b) Testing.

Table 2 and Figure 8 depict the Training Phase (TRPHE) and Testing Phase (TSPHE) of the MLPFE-DBMMC approach under various metrics. Under TRPHE, the MLPFE-DBMMC model has attained $accu_y$ of 94.44%, $prec_n$ of 95.88%, $reca_l$ of 94.38%, $F1_{score}$ of 95.12%, and MCC of 88.66%. Also, under TSPHE, the MLPFE-DBMMC model has obtained $accu_y$ of 93.19%, $prec_n$ of 93.47%, $reca_l$ of 94.79%, $F1_{score}$ of 94.13%, and MCC of 86.05%.

Table 2. TRPHE and TSPHE outcome of the MLPFE-DBMMC model

Metric	Training Phase	Testing Phase
Accuracy	94.44	93.19
Precision	95.88	93.47
Recall	94.38	94.79
F1-Score	95.12	94.13
MCC	88.66	86.05

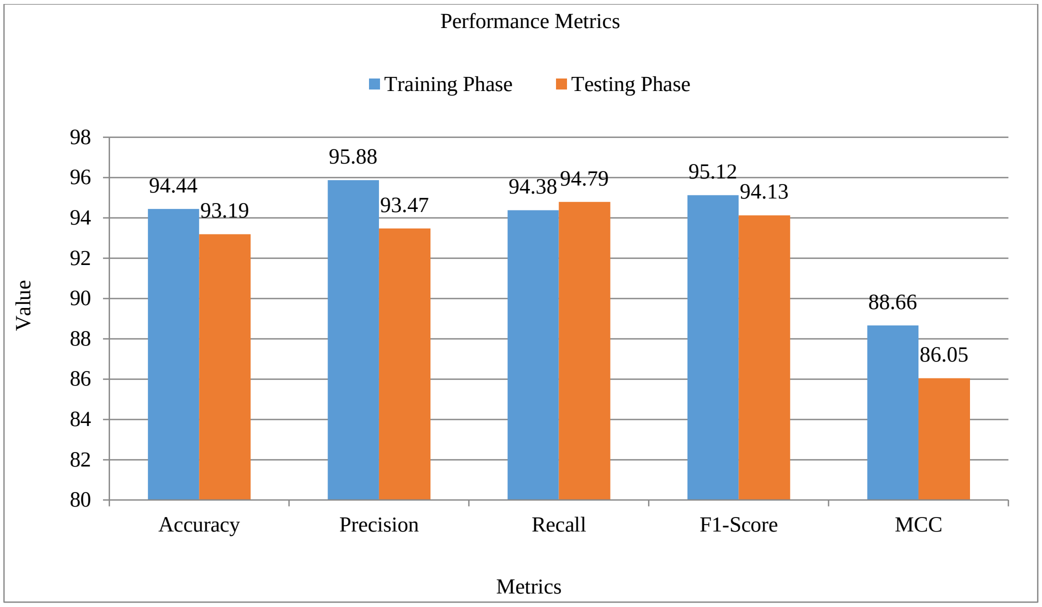

Fig. 8 TRPHE and TSPHE outcome of MLPFE-DBMMC model under various metrics

Figure 9 illustrates the Training (TRAIN) $accu_y$ and validation (VALID) $accu_y$ of an MLPFE-DBMMC method over 30 epochs. Both TRAIN and VALID $accu_y$ increase rapidly initially, illustrating effective learning. VALID slightly exceeding TRAIN shows good generalization. Their close alignment throughout training suggests robust regularization and capability.

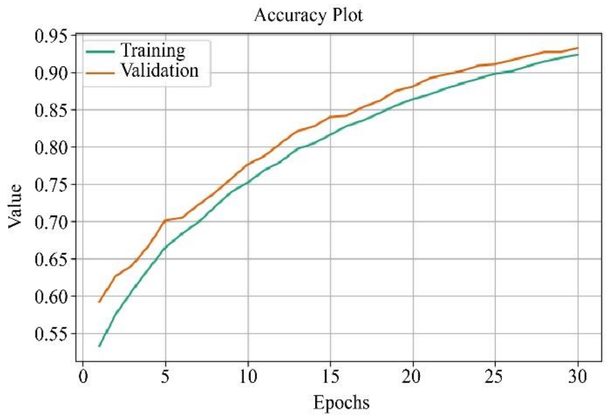

Fig. 9. $Accu_y$ plot of MLPFE-DBMMC technique

Figure 10 specifies the TRAIN and VALID losses of the MLPFE-DBMMC method over 30 epochs. Initially, TRAIN and VALID losses are high, indicating limited initial learning. As TRAIN progresses, both losses decrease steadily, and their close alignment shows effective learning with good generalization and no overfitting.

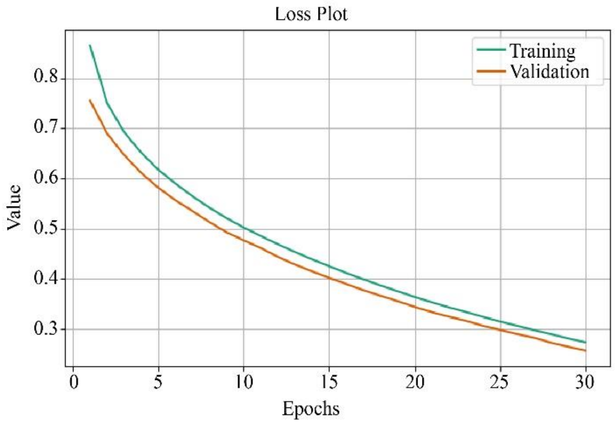

Fig. 10 Loss plot of the MLPFE-DBMMC technique

Figure 11 exhibits the best parameters over iterations of an MLPFE-DBMMC methodology. Every subplot signifies the development of a specific parameter, such as the number of hidden units (num_hidden_1, num_hidden_2), dropout rates (dropout_1), learning rate, epoch count, and batch size across tuning iterations. The plots assist in visualizing how the tuning process converges to optimal values, demonstrated by the final dots in each graph. Table 3 depicts the optimal parameter values outcome. Under Epochs, num_hidden_1, num_hidden_2, dropout_1, dropout_2, epochs, batch_size, learning_rate, Fitness, and Runtime parameters, the obtained optimal parameter values are 22, 512, 243, 0.238271, 0.113291, 30, 568, 0.0001, 0.939201, and 9.834022, respectively.

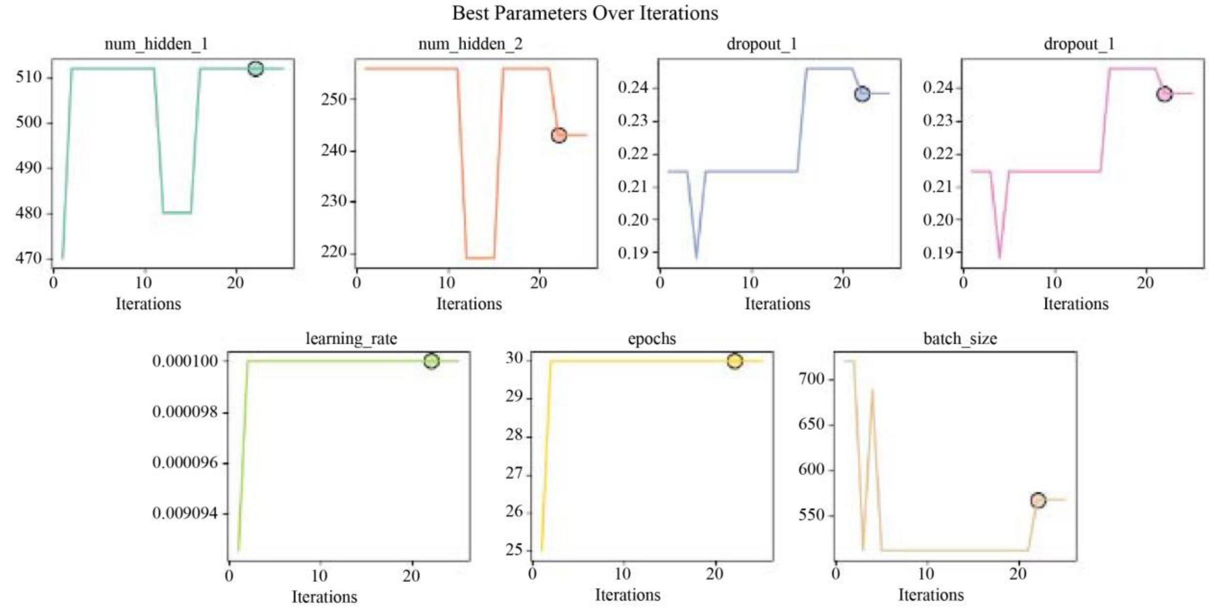

Fig. 11 Best parameters over iterations of the MLPFE-DBMMC model

Table 3. Optimal parameter values outcome

Parameter Names	Optimal Parameter Values
Epochs	22
num_hidden_1	512
num_hidden_2	243
dropout_1	0.238271
dropout_2	0.113291
epochs	30
batch_size	568
learning_rate	0.0001
Fitness	0.939201
Runtime	9.834022

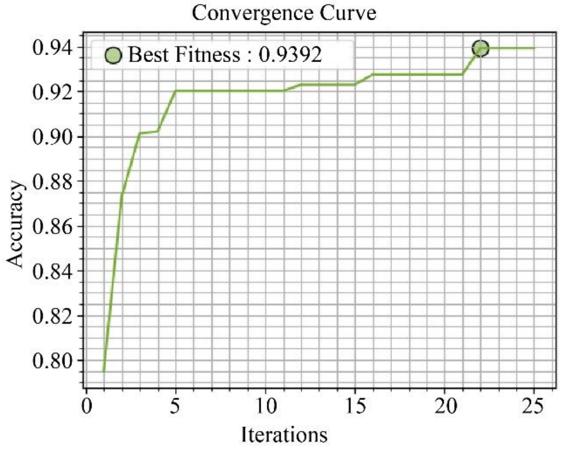

Fig. 12 Convergence curve of the MLPFE-DBMMC model under various iterations

Figure 12 portrays the convergence curve analysis of the MLPFE-DBMMC approach on various iterations. The outcomes imply that the MLPFE-DBMMC model portrays better convergence by achieving a value of 0.9392 over several iterations on the applied data.

Figure 13 exemplifies the computational power of the MLPFE-DBMMC methodology over distinct iterations. It underscores how the model's power utilization or processing capacity rises as the number of iterations increases. By examining the curve, it becomes evident that the MLPFE-DBMMC model portrays effective and constant computation, effectively balancing resource usage while obtaining optimal performance at several phases of the iterative process.

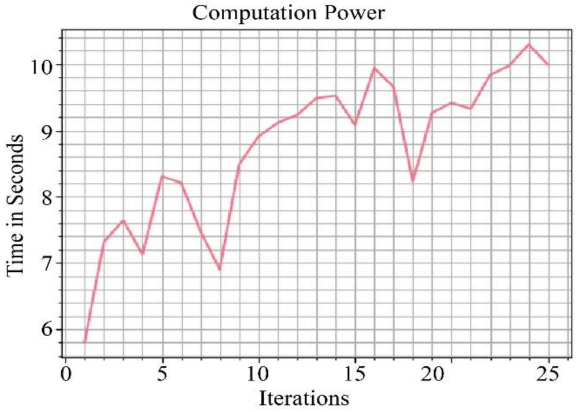

Fig. 13 Computation power of MLPFE-DBMMC model

The comparative analysis of the MLPFE-DBMMC model with the present methods is shown in Table 4 and Figure 14 [26-28]. The simulation outcome denoted that the MLPFE-DBMMC model performed better. Under $accu_y$, the MLPFE-DBMMC model attained the highest $accu_y$ of 94.44%. In contrast, the TextBERT, ResNeXt-101, Late-Fusion, Concat-BERT, LaBSE, CNN, and Mistral 7B methodologies attained the least $accu_y$ of 82.99%, 76.57%, 93.68%, 84.76%, 93.85%, 77.08%, and 82.72%, respectively. Similarly, at $prec_n$, the MLPFE-DBMMC technique attained a maximum $prec_n$ of 95.88%. In contrast, the TextBERT, ResNeXt-101, Late-Fusion, Concat-BERT, LaBSE, CNN, and Mistral 7B methodologies got a minimum $prec_n$ of 76.60%, 94.84%, 86.51%, 89.46%, 83.33%, 87.25%, and 77.30%, respectively.

Table 4. Comparative analysis of the MLPFE-DBMMC method with existing approaches

Methods	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$	MCC
TextBERT	82.99	76.60	80.24	85.29	81.08
ResNeXt-101	76.57	94.84	90.48	89.72	82.58
Late-Fusion	93.68	86.51	92.93	79.26	77.23
Concat-BERT	84.76	89.46	85.71	79.13	77.00
LaBSE	93.85	83.33	79.10	90.32	79.41
CNN	77.08	87.25	81.09	91.73	85.07
Mistral 7B	82.72	77.30	89.04	93.11	81.89
MLPFE-DBMMC	94.44	95.88	94.38	95.12	88.66

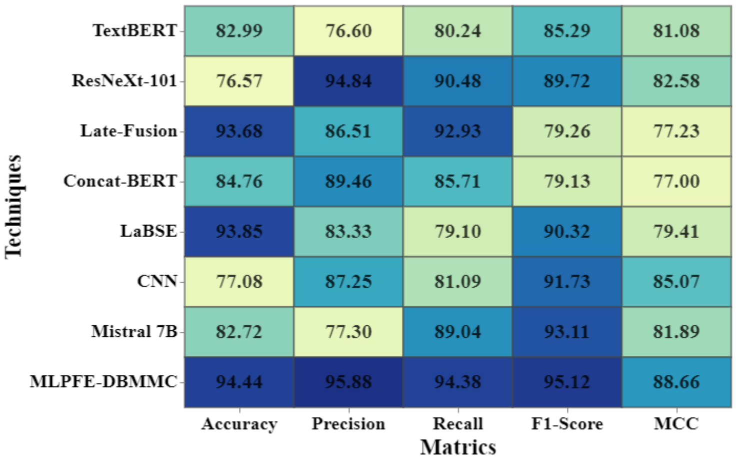

Fig. 14 Comparative analysis of the MLPFE-DBMMC method with existing approaches

Finally, on MCC, the MLPFE-DBMMC model got a higher MCC of 88.66%. At the same time, the TextBERT, ResNeXt-101, Late-Fusion, Concat-BERT, LaBSE, CNN, and Mistral 7B methodologies have obtained an MCC of 81.08%, 82.58%, 77.23%, 77.00%, 79.41%, 85.07%, and 81.89%, respectively. The comprehensive comparison study highlighted the superior performance over other models to detect cyberbullies.

5. Conclusion

In this article, the MLPFE-DBMMC model is proposed to detect cyberbullies. The objective of the MLPFE-DBMMC model is to develop a successful multimodal cyberbullying detection structure by utilizing visual, audio, and textual inputs to identify and classify harmful behavior precisely. At the primary stage, the preprocessing phase is performed on multimodal data, such as images, audio, and text. For image preprocessing, the anisotropic diffusion model is applied for noise removal.

The stationary wavelet transform-based noise removal is used in audio preprocessing. The text preprocessing phase includes various levels, such as lower case, tokenization, removal of stopwords, and stemming. For the feature extraction model, the MLPFE-DBMMC method utilizes CLIP for images, VGGish-based audio, and the GPT-2 method for text. Next, the MLPFE-DBMMC method applies the CORAL multi-layer perceptron method for the multimodal cyberbullying detection method. Finally, the IARO-based hyperparameter tuning is achieved. Experimentation using the MLPFE-DBMMC method on the MultiBully dataset depicts a superior accuracy of 94.44%, outperforming existing methods.

Data Availability Statement

The data that support the findings of this study are openly available in https://github.com/Jhaprince/MultiBully?tab=readme-ov-file, reference number [4].

References

[1] Lu Cheng et al., “XBully: Cyberbullying Detection within a Multi-Modal Context, *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, Association for Computing Machinery, New York, United States, pp. 339-347, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Kaige Wang et al., “Multimodal Cyberbullying Detection on Social Networks,” *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shuai Wang et al., “A Review of Multimodal-based Emotion Recognition Techniques for Cyberbullying Detection in Online Social Media Platforms,” *Neural Computing and Applications*, vol. 36, no. 35, pp.21923-21956, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Krishanu Maity et al., “A Multitask Framework for Sentiment, Emotion and Sarcasm Aware Cyberbullying Detection from Multi-Modal Code-Mixed Memes,” *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, United States, pp. 1739-1749, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Hong Lin et al., “Special Issue on Deep Learning Methods for Cyberbullying Detection in Multimodal Social Data,” *Multimedia Systems*, vol. 28, no. 6, pp.1873-1875, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mahmoud Ahmad Al-Khasawneh et al., “Toward Multimodal Approach for Identification and Detection of Cyberbullying in Social Networks,” *IEEE Access*, vol. 12, pp. 90158-90170, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sayanta Paul, Sriparna Saha, and Mohammed Hasanuzzaman, “Identification of Cyberbullying: A Deep Learning based Multimodal Approach,” *Multimedia Tools and Applications*, vol. 81, no. 19, pp.26989-27008, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Tingting Li et al., “Integrating GIN-based Multimodal Feature Transformation and Multi-Feature Combination Voting for Irony-Aware Cyberbullying Detection,” *Information Processing and Management*, vol. 61, no. 3, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Md. Tofael Ahmed et al., “Multimodal Cyberbullying Meme Detection from Social Media using Deep Learning Approach,” *International Journal of Computer Science and Information Technology*, vol. 15, no. 4, pp. 27-37, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Subbaraju Pericherla, and Ilavarasan Egambaram, “Cyberbullying Detection on Multimodal Data using Pre-Trained Deep Learning Architectures,” *Ingeniería Solidaria*, vol. 17, no. 3, pp.1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] C. Valliyammai et al., “Cyberbullying Detection in Social Media with Multimodal Data using Transfer Learning,” *2024 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, Chennai, India, pp. 443-447, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Neha Minder Singh, and Sanjay Kumar Sharma, “Multimodal Cyberbullying Detection with Severity Analysis using Deep-Tensor Fusion Framework,” *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 17, no. 3, pp. 144-150, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Md. Anas Mondol et al., “AI-Powered Frameworks for the Detection and Prevention of Cyberbullying Across Social Media Ecosystems,” *TechComp Innovations: Journal of Computer Science and Technology*, vol. 2, no. 1, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Neha Minder Singh, and Sanjay Kumar Sharma, “An Efficient Automated Multimodal Cyberbullying Detection using Decision Fusion Classifier on Social Media Platforms,” *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 20507-20535, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Jiani Wang et al., “Hierarchical Multi-Stage BERT Fusion Framework with Dual Attention for Enhanced Cyberbullying Detection in Social Media,” *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, Xiamen, China, pp. 86-89, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sandip A. Kahate, and Atul D. Raut, “Design of a Deep Learning Model for Cyberbullying and Cyberstalking Attack Mitigation via Online Social Media Analysis,” *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT)*, Kottayam, India, pp. 1-7, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Nagwan Abdel Samee et al., “Safeguarding Online Spaces: A Powerful Fusion of Federated Learning, Word Embeddings, and Emotional Features for Cyberbullying Detection,” *IEEE Access*, vol. 11, pp.124524-124541, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Syed Rizwana, Lenin Laitonjam, and Ranjita Das, “Adaptive Anisotropic Diffusion Filter in Unsharp Masking Scheme for Mammogram Enhancement using PLIP Operations,” *Procedia Computer Science*, vol. 258, pp. 3468-3479, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Vandana Akshath Raj, Subramanya G. Nayak, and Ananthakrishna Thalengala, “A Hybrid Framework for Muscle Artifact Removal in EEG: Combining Variational Mode Decomposition, Stationary Wavelet Transform, and Canonical Correlation Analysis,” *Cogent Engineering*, vol. 12, no. 1, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mahazam Afrad et al., “Sentiment Analysis of Visitor Reviews on Baturaden Tourist Attraction using Machine Learning Methods,” *Edu Komputika Journal*, vol. 11, no. 1, pp. 57-64, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Xiang Li et al., “RS-CLIP: Zero Shot Remote Sensing Scene Classification via Contrastive Vision-Language Supervision,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 124, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Komal Shahzad et al., “Enhancing Voice Spoofing Detection: A Hybrid Approach with VGGish-LSTM Model for Improved Security in Automatic Speaker Verification Systems,” *IEEE Access*, vol. 13, pp. 40682-40702, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mariam Bangura et al., “Automatic Generation of German Drama Texts using Fine Tuned GPT-2 Models,” *arXiv Preprint*, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka, “Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation,” *Pattern Recognition Letters*, vol. 140, pp. 325-331, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Quyet Nguyen Huu et al., “An Improved Artificial Rabbit Optimization for Structural Damage Identification,” *Latin American Journal of Solids and Structures*, vol. 21, no. 1, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Muhammad Shoib Amin et al., “Dual-Branch Neural Network for Bridging Semantic Gap in Harmful Meme Detection,” *IEEE Access*, vol. 13, pp. 125090-125100, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[27] Prashant Kapil, and Asif Ekbal, “A Transformer based Multi Task Learning Approach to Multimodal Hate Speech Detection,” *Natural Language Processing Journal*, vol. 11, pp. 1-13, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[28] A.K. Indira Kumar et al., “Multi-Task Detection of Harmful Content in Code-Mixed Meme Captions using Large Language Models with Zero-Shot, Few-Shot, and Fine-Tuning Approaches,” *Egyptian Informatics Journal*, vol. 30, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]