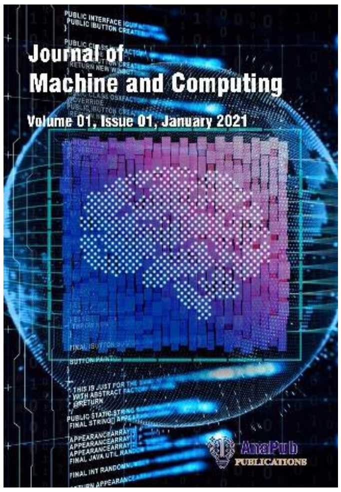
Journal Pre-proof



Enhancing Spoof Detection in Automatic Speaker Verification Using CQCC Optimization and ViT Architecture

Selin M and Preetha Mathew K

DOI: 10.53759/7669/jmc202505202

Reference: JMC202505202

Journal: Journal of Machine and Computing.

Received 23 February 2025

Revised from 02 June 2025

Accepted 08 August 2025

Please cite this article as: Selin M and Preetha Mathew K, "Enhancing Spoof Detection in Automatic Speaker Verification Using CQCC Optimization and ViT Architecture", Journal of Machine and Computing. (2025). Doi: https:// doi.org/10.53759/7669/jmc202505202.

This PDF file contains an article that has undergone certain improvements after acceptance. These enhancements include the addition of a cover page, metadata, and formatting changes aimed at enhancing readability. However, it is important to note that this version is not considered the final authoritative version of the article.

Prior to its official publication, this version will undergo further stages of refinement, such as copyediting, typesetting, and comprehensive review. These processes are implemented to ensure the article's final form is of the highest quality. The purpose of sharing this version is to offer early visibility of the article's content to readers.

Please be aware that throughout the production process, it is possible that errors or discrepancies may be identified, which could impact the content. Additionally, all legal disclaimers applicable to the journal remain in effect.

AnaPub
PUBLICATIONS

Enhancing Spoof Detection in Automatic Speaker Verification Using CQCC Optimization and ViT Architecture

M Selin[1], K Preetha Mathew[2]

[1]Department of Computer Applications, Cochin University of Science and Technology, Kerala, India..
[2]Cochin University College of Engineering, Kerala, India

selin.m.a@gmail.com, preetha.mathew.k@gmail.com

Abstract

Spoof detection is found to be essential for improving the security features of automatic speaker verification (ASV) systems, which are primarily used in authentication. The primary goal of this study is to enhance the performance and efficiency of spoof detection using speech samples taken from the ASVspoof 2019 dataset. The Constant Q Cepstral Coefficients (CQCC) extracted from these speech samples act as an important key feature. Feature optimization methods such as Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO) are used to refine these features and hence enhance the model accuracy with minimal time cost. A Vision Transformer (ViT) model is then trained using each optimized feature, and the performance is evaluated by comparing the results from different optimization methods. Time analysis shows a substantial reduction in training time per epoch when the optimized features are used. The Genetic Algorithm attained the best performance, with a test accuracy of 97% and the least training time. Equal Error Rate (EER) and the Tandem Detection Cost Function (t-DCF) are used as the evaluation metrics. This study demonstrates how feature optimization helps to enhance spoof detection accuracy while reducing processing time, hence becoming an authentic solution for real-time ASV systems.

Keywords: Feature Optimization, Vision Transformer, CQCC.

1. Introduction

Spoof detection is crucial for the safety of security and authentication systems, particularly those that use biometric data such as speech recognition [1]. For example, spoofing in voice authentication can allow unauthorized individuals to use systems by impersonating someone else's voice, paving the way for criminal and illegal activities. The importance of spoof detection is in its potential to prevent fraud and identity theft, specifically in financial transactions, where attackers may imitate people to commit unauthorized activities. The spoofing attacks, like voice conversion, speech synthesis, replay attacks, and impersonation, show the various techniques that help in creating fake or counterfeit speech. To lower the risks due to spoofing attacks, it is crucial to create a system that can effectively distinguish between fake and authentic signals..

Artificial Intelligence plays an important role in Automatic Speaker Verification (ASV) fraud detection by improving the adaptability, efficiency, and accuracy of identifying false attempts [2]. This decreases false positives and negatives, hence improving the system's total reliability. AI allows real-time processing, therefore enabling the detection of fraud attempts, a crucial feature for applications like online access security and corporate intelligence. AI, by integrating speech patterns with other technologies, combines multimodal data and thereby enhances the precision and depth of analysis. Fraud detection by AI minimizes the demand for human oversight, leads to improved system efficiency. Finally, it finds ambiguities in passwords and guards ASV systems, hence offering an effective security against AI-driven security threats such as deepfakes [3][4].

To address the challenges in ASV spoof detection, this study suggests a technique that uses Constant-Q Cepstral Coefficients (CQCC) [5] for feature extraction, followed by feature optimization and classification using a Vision Transformer (ViT). This method efficiently integrates CQCC's capability to attain essential audio features with the ViT model's strength in realizing complex patterns present in the data.

CQCC is derived from the Constant-Q Transform (CQT), which provides a logarithmic frequency resolution that aligns with human auditory perception. The CQCC [5] features capture the fine artifacts introduced by spoofing methods such as Text-to-Speech (TTS) and Voice Conversion (VC), which change the low-frequency elements. The variable resolution of CQCC finds subtle details, hence making it more robust against known as well as unknown spoofing attacks. After extracting the CQCC features, optimization is done utilizing methods like the Genetic Algorithm (GA) [6][7] and the Mayfly Optimizer (MO) [8]. These methods are utilized to process the features and thereby improving the accuracy and efficiency of the classification process.

Vision Transformer (ViT) model [9] is then trained using these optimized features. Even though initially produced for image processing, ViT's ability to attain complex model patterns and long-range dependencies makes it suitable for differentiating between spoofed and bonafide audio samples. The proposed system, using CQCC [5] for feature extraction, GA, GWO, and MO for feature optimization, and the ViT classifier altogether enhances the performance of spoof detection.

The Genetic Algorithm (GA) [6] optimizes the most suitable features from spoofed and bonafide samples, gives them to the Vision Transformer (ViT) [9] classifier for spoof detection. It does crossover and mutation functions on feature combinations to enhance classification performance. After finding the optimal set of features, they are utilized in training the Vision Transformer (ViT), enabling it to learn efficiently from the informative data [10]. This approach improves the ViT's performance in identifying spoofed samples.

Grey Wolf Optimization (GWO) [7][11] is a metaheuristic algorithm inspired by the social hierarchy and cooperative hunting behaviour of grey wolves, employed to optimize the extracted CQCC features for the ViT classifier, allowing the model to attain high accuracy in identifying spoofed inputs while reducing both false positives and false negatives.

Mayfly Optimization (MFO) [8] improves the spectral feature selection process of the Vision Transformer (ViT) classifier for spoof detection [9], thereby improving classification accuracy. Simulating mayfly behaviour, MFO improves its choice of critical features by balancing exploitation and exploration, hence concentrating on attributes that efficiently distinguish bonafide data from spoofed data. This targeted optimization leads to a more effective and precise classification process.

This work makes the following contributions:

1. Dataset Selection: The ASVspoof 2019 Logical Access (LA) dataset is used for evaluating and training the spoof detection algorithm.

2. Feature Extraction: CQCC are implemented because of their improved frequency resolution at lower frequencies, which is crucial for procuring fine differences between spoofed and Bonafide speech samples.
3. Optimization: The accuracy and efficiency of the model were improved via feature optimization strategies, like the Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO).

4. Model Selection: A Vision Transformer (ViT) was implemented as the classifier to capitalize on its capability for effective feature learning and classification.

5. Evaluation: Model efficiency was evaluated via a comprehensive performance analysis, such as an accuracy versus loss graph, confusion matrices, ROC curves, and classification reports. A time efficiency study was conducted to compare the training duration per epoch with and without optimization, showcasing the time savings attained via these optimized features.

2. Related Research Work

Mcuba et al. (2023) suggested a deepfake audio detection framework utilizing various deep learning architectures, including a custom model, Visual Geometry Group Network 16 (VGG-16), Residual Networks (ResNet), and Frequency-Gated Lightweight Convolutional Neural Networks (FG-LCNN). The assessment was done using various optimizers, including Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam), and Adadelta, along with multiple audio features such as Mel-Spectrograms, Mel Frequency Cepstral Coefficients (MFCC), Spectrograms, and Chromagrams. The highest performance was achieved by the custom model using SGD, reaching an accuracy of 83.636% on Chromagram images, while VGG-16 yielded 85.906% accuracy for MFCC features. These findings underscore the importance of context-specific architectural and feature choices in forensic audio analysis [10].

Shaaban et al. (2023) explored various methodologies for detecting audio deepfakes using both classical machine learning (ML) and deep learning (DL) models, and the performance across multiple datasets, including ASVspoof 2019, AR-DAD, and Fake-or-Real (FOR), was evaluated. Key models investigated include Convolutional Neural Networks (CNN), RES-EfficientCNN, and Siamese CNN, achieving detection accuracies of up to 99%. Notable results include RES-EfficientCNN with an F1-score of 97.61%, Deep4SNet with 98.5% accuracy, and a Support Vector Machine (SVM) model attaining 99% accuracy on the AR-DAD dataset. The study highlights the critical role of dataset selection and evaluation metrics in optimizing deepfake detection performance [4].

Kwak et al. (2023) proposed a compact model for enhancing voice spoofing detection, utilizing the ASVspoof 2019 dataset. The model integrates ResNet's skip connections with Light CNN's max feature map to produce a low Equal Error Rate (EER) of 0.30%, outperforming the top ensemble system. Additionally, by employing depth-wise separable convolutions, the model size was reduced by 84.3% while maintaining an EER of 0.36% [1].

Chen et al. (2022) introduced a GNSS spoofing detection system that integrates multiple parameters with a Support Vector Machine (SVM). Unlike traditional single-parameter approaches, which often fail against modern spoofing techniques, this method incorporates features such as composite signal quality, carrier-to-noise ratio, and PVT residuals. The multi-parameter model demonstrated significantly improved performance, achieving F1-scores of 93.[illegible]7% on the TEXBAT dataset and 97% on the OAKBAT dataset, thereby outperforming conventional single-parameter detection methods [12].

Anagha et al. (2023) proposed a deep learning-based approach for audio deepfake detection using Convolutional Neural Networks (CNNs) trained on the ASVspoof 2019 dataset. The method utilizes Mel spectrograms for feature extraction and employs the Adam optimizer for model training. The proposed system achieved an accuracy of 8[illegible]%, an AUC of 0.8[illegible], and an average precision of 0.90 [13].

Todisco et al. (2017) introduced Constant-Q Cepstral Coefficients (CQCCs) for detecting spoofing attacks in automatic speaker verification (ASV) systems. The method was evaluated across three datasets, including ASVspoof 2015, ASVspoof, and RedDots-Replayed, demonstrating CQCCs' ability to capture manipulation artifacts and outperform conventional features, with error rate improvements of up to 72%, 47%, and 64%, respectively. However, the results varied across datasets, indicating that a single feature configuration may not be universally effective. Consequently, the authors recommend exploring an ensemble of classifiers to enhance robustness against diverse spoofing scenarios [14].

Ye et al. (2019) introduced a novel approach for replay attack detection based on a normalized Constant-Q Cepstral Coefficients (CQCC) algorithm. By applying Cepstral Mean and Variance Normalization (CMVN), the method achieved the best Equal Error Rate (EER) of 15.96%. Significant improvements were observed, with EER reductions of 34.7% and 54% for CQCC and Mel-Frequency Cepstral Coefficients (MFCC), respectively. These results demonstrate the robustness of the approach in cross-device scenarios, maintaining EER values below 10% [15].

Zh[illegible] et al. (2022) proposed a Segment-Based Anti-Spoofing Detection (SASD) method for Embedded Voice Recognition systems. This novel approach leverages Constant-Q Cepstral Coefficients (CQCCs) and Zero Crossing Rate (ZCR) to enhance voice spoofing detection. By focusing on both word and silence segments, the method achieved a 33.47% improvement in anti-spoofing accuracy and a [illegible].10% reduction in detection time on embedded devices, based on evaluations conducted using the ASVspoof 2021 datasets. This demonstrates the method's effectiveness as a fast and efficient solution against voice spoofing attacks [16].

Table 1: Comparison of Related Research Work

Reference	Technique(s)	Database	Advantages	Results
Mcuba et al. (2023) [10]	Deepfake audio detection with VGG-16, ResNet, and FG-LCNN using SGD, Adam, Adadelta.	/	Context-specific models enhance audio analysis	SGD: 83.636%, VGG-16: 85.906%.

Reference	Technique(s)	Database	Advantages	Results
Shaaban et al. (2023) [4]	CNN, RES-EfficientCNN, Siamese CNN, SVM for audio detection	ASVspoof 2019, AR-DAD, FOR	High detection accuracy	SVM: 99% on AR-DAD.
Kwak et al. (2023) [1]	Compact voice spoofing model using ResNet skip connections and Light CNN.	ASVspoof 2019	84.3% size reduction, maintaining performance	EER: 0.30%, 0.36% with depthwise convolutions.
Chen et al. (2022) [12]	GNSS spoofing detection with SVM using multiple parameters.	TEXBAT, OAKBAT	Diverse features enhance detection	F1-scores: 93.97%, 97%.
Anagha et al. (2023) [13]	Deep learning method with CNNs	ASVspoof 2019	Mel spectrograms for feature extraction	85% accuracy, AUC: 0.87.
Todisco et al. (2017) [14]	CQCCs for spoofing detection in ASV	ASVspoof 2015, AVspoof, RedDots	CQCCs capture artifacts, outperform traditional methods	Error rate improvements: up to 72%, 47%, 64%.
Ye et al. (2019) [15]	Normalized CQCC for replay attack detection	/	Enhanced performance via CMVN; robust across devices	Best EER: [illegible]5.96%, reductions of [illegible]4.7% for CQCC, 54% for MFCC.
Zhan et al. (2022) [16]	Segment-Based Anti-Spoofing using CQCCs and ZCR	ASVspoof 2021	Fast solution against spoofing; improved accuracy	Accuracy improved by [illegible]3.47%; detection time reduced by 69.10%.

3. Methodology

The main intent is to design a structured spoof detection system utilizing a Vision Transformer (ViT) model [17]. A core component of this framework is feature optimization, which amplifies the models performance by detecting and sifting the most salient input features. The process initially extracts the Constant-Q Cepstral Coefficients (CQCCs) [14] features from the audio samples, which are ideal for capturing crucial spectral properties essential to differentiate bonafide speech from spoofed audio. Following feature extraction, three optimization techniques, Genetic Algorithm (GA) [18], Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO) [19], are employed to identify the optimal subset of features. By choosing the most discriminative attributes, it explores diverse feature combinations to enhance classification accuracy. A comparative analysis of the GA [20], GWO, and MO is carried out to establish which technique most constructively distinguishes subtle differences between bonafide and spoofed audio samples. This systematic feature optimization strategy magnifies the overall performance of the spoof detection system. The suggested framework is illustrated in Figure 1.

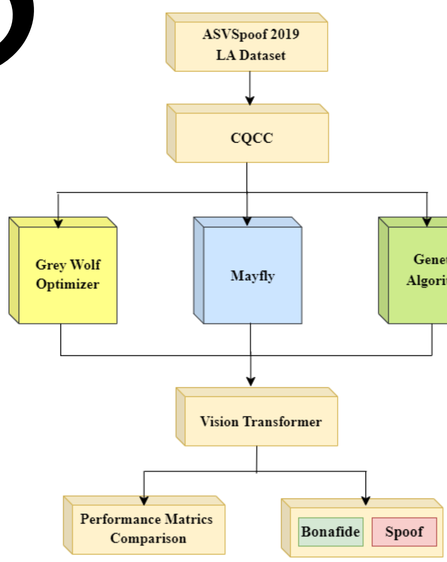


Figure 1: Proposed Workflow

3.1 Dataset

The ASVspoof 2019 dataset is a comprehensive public collection of spoofed speech samples [21] that was created specifically to aid in the study of deepfake detection and voice spoofing, as well as the creation and assessment of countermeasures against spoofing. The dataset consists of two subsets: Logical Access (LA) and Physical Access (PA), each representing distinct spoofing conditions. There are two classes in it: parody and genuine speech. The duration of each audio samples is roughly 30 seconds. Model development makes use of the Logical Access (LA) subset. This subset comprises 25,000 training samples, 9,900 validation samples, and 9,900 testing samples.

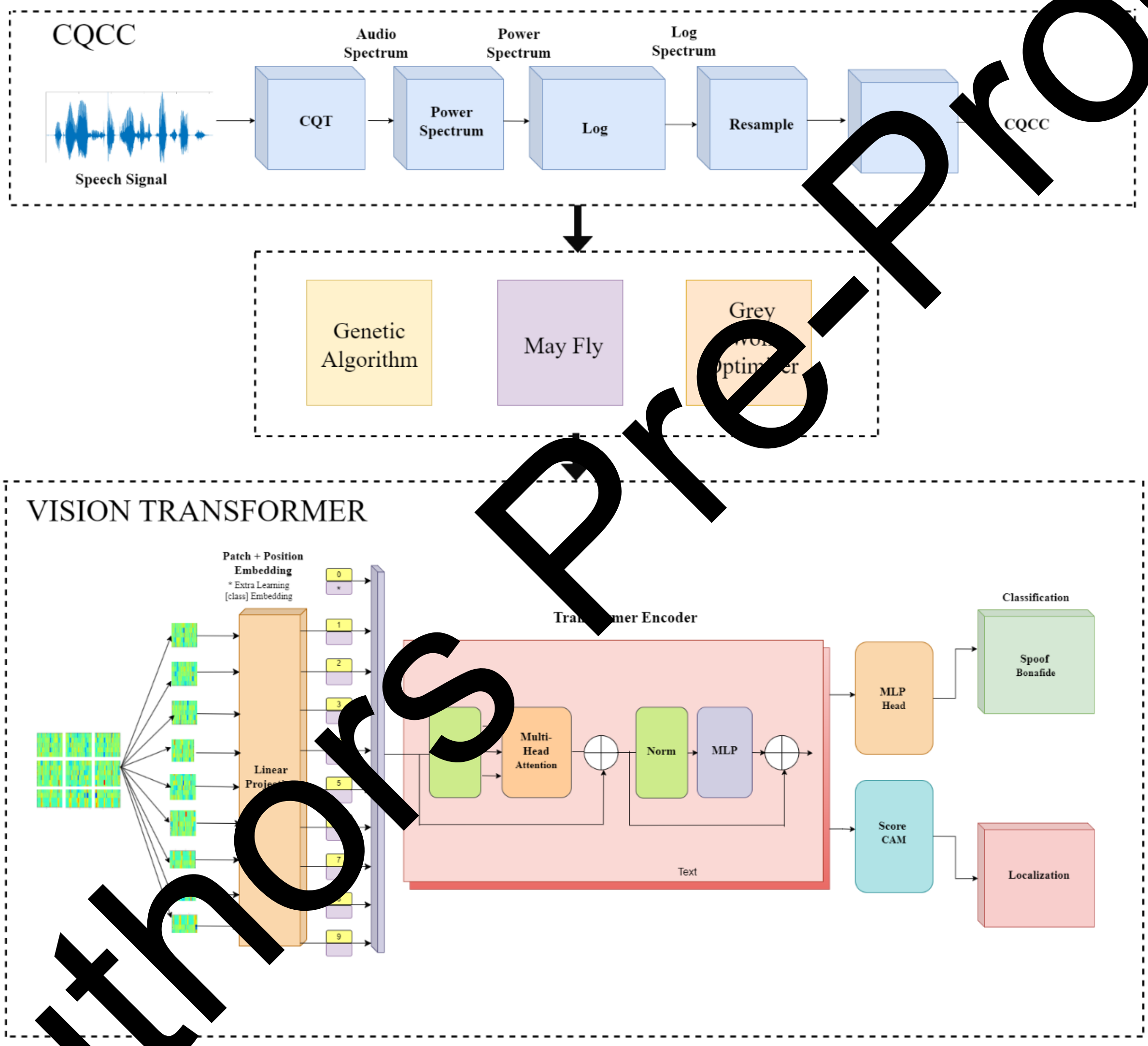


Figure 2: Proposed Model Architecture

3.2 Feature Extraction

A crucial step in spoof detection systems is feature extraction, which converts raw audio signals into informative representations that enable machine learning models to effectively distinguish real and fake audio samples. The main goal of feature extraction is to identify unique characteristics of the audio signal that highlight the subtle distinctions between bonafide and spoof samples, which are frequently imperceptible and difficult to identify.

In audio processing, Constant-Q Cepstral Coefficients (CQCC) are a well-known feature extraction method, that works especially well for tasks like speaker recognition, audio categorisation, and spoof detection [5]. Constant-Q Cepstral Coefficients (CQCCs) are derived from the Constant-Q Transform

(CQT), which generates a non-linear time-frequency representation that closely aligns with human auditory perception. By using a logarithmic frequency scale, the CQT gives a higher resolution at lower frequencies, capturing the fine spectral details, while it gives better temporal resolution at higher frequencies. This unique characteristic makes CQCCs particularly apt for capturing rich harmonic content and subtle fluctuations in audio signals [22].

The CQCC features are extracted with the application of the Constant Q Transform (CQT) on the audio signal, which is mathematically defined in Equation (1):

$$CQT(f,t) = \sum_{n=0}^{N-1} x\,[n] \cdot w[n-t] \cdot e^{-j\frac{2\pi}{Q}fn} \tag{1}$$

In this equation, $x[n]$ represents the audio signal, $w[n-t]$ denotes the window function, and $e^{-j\frac{2\pi}{Q}fn}$ applies the frequency based transformation. After computing the CQT, logarithmic scaling is applied to its magnitude of the CQT, as shown in Equation (2), to compress the dynamic range of the spectrum:

$$S(f,t) = \log|CQT(f,t)| \tag{2}$$

Next, the Discrete Cosine Transform (DCT) [23] is applied to the log-scaled spectrum to convert the data into the cepstral domain, as described in Equation (3):

$$CQCC(k) = \sum_{n=0}^{N-1} S\,(n) \cdot \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right) \quad \text{for } k = 0,1,\ldots,K-1 \tag{3}$$

The resulting CQCC features, represented by the vector in Equation (4), capture complex spectral and temporal details, making them highly effective for differentiating between genuine and spoofed audio samples:

$$CQCC = [CQCC(0), CQCC(1), \ldots, CQCC(K-1)] \tag{4}$$

CQCCs are particularly advantageous in spoof detection because they offer effectiveness against variations in pitch, speed, and recording conditions, which are often exploited in spoofing attacks such as voice conversion, synthesis, or manipulation. Their ability to capture fine-grained spectral features enhances the discriminative power of machine learning models, resulting in improved classification performance in real-world spoof detection applications.

3.3 Optimization Techniques for Performance Enhancement

Feature optimization is an essential stage in enhancing the performance of machine learning models, particularly for applications like spoof detection. During this process, the collected features (like CQCC) are sifted to upgrade their ability to capture the elementary patterns in the data. The feature optimization ensures that the model can correctly differentiate between spoofed audio samples and bonafide. Here in this study, three optimization techniques are put to function: Genetic Algorithm (GA) [24], Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO). These strategies are employed on the procured features to enhance their importance for classification tasks. Each strategy helps in conquering the usual optimization problems, such as balancing exploitation and exploration, ignoring local minima, and increasing speed of convergence. Feature refining before passing them to Vision Transformer will improve the model's overall general accuracy, along with confronting the complexity of audio input.

3.3.1 Grey wolf Optimizer

The Grey Wolf Optimizer (GWO)[7][11] was employed to improve the performance of the Vision Transformer (ViT) model used for the spoof detection task. GWO, a nature-inspired optimization algorithm, emulates the social hierarchy and hunting strategies of grey wolves[25]. In the implementation, GWO was employed for feature optimization, accurately optimizing the CQCC features extracted from the dataset for spoof detection[26]. The aim of this optimization was to refine the feature set [26], and hence increase the model's capacity to precisely distinguish between spoofed and bonafide samples.

The optimization process begins with initializing a population of "wolves," where each wolf stands for a set of feature configurations [26]. These wolves undergo periodic changes with respect to the GWO's position updating rules [11]. Through this process, the best feature configurations are identified to improve classification accuracy, as illustrated in Figure 3, which outlines the proposed integration of GWO in the model for feature optimization.

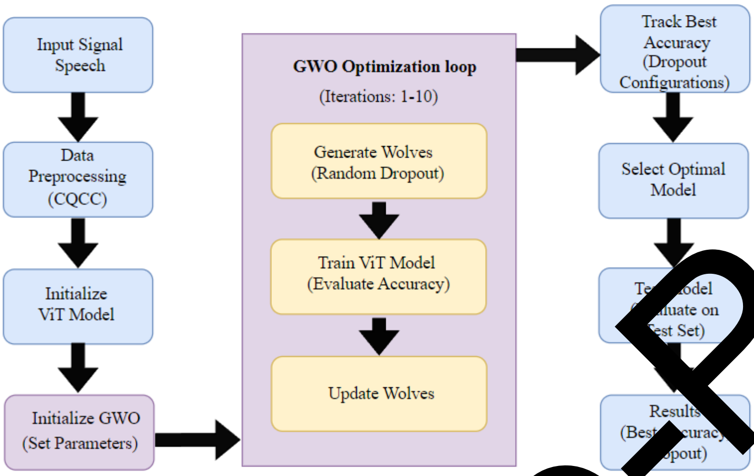


Figure 3: Proposed Integration of Grey Wolf Optimizer (GWO) in the Model for Feature Optimization

By simulating the grey wolves hunting nature, the GWO algorithm updates feature configurations, and three key equations govern the process. Initially, the distance between a wolf (current feature configuration) and the prey (optimal feature set) is calculated using Equation (5):

$$\vec{D} = |\vec{C} \cdot \vec{X}_{\text{prey}} - \vec{X}| \tag{5}$$

where:

- $\vec{C}$ is a coefficient vector, calculated as $\vec{C} = 2 \cdot \vec{r_2}$,
- $\vec{r_2}$ is a random vector in the range [0,1],
- $\vec{X}_{\text{prey}}$ represents the position of the prey (optimal feature set),
- $\vec{X}$ represents the current feature configuration.

In each iteration, the wolves update their positions depending on the three best solutions, known as the alpha, beta, and delta wolves. The modified positions are calculated using Equations (6),(7), and (8):

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \tag{6}$$
$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \tag{7}$$
$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{8}$$

Finally, the new position of each wolf (feature set) is obtained as the mean of the positions of the alpha, beta, and delta wolves, as depicted in Equation (9):

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{9}$$

This iterative process allows GWO to refine the features extracted from the audio data, leading to an enhancement in the Vision Transformer model's classification performance. The combination of

GWO optimizes the feature space by exploiting these position update equations, which leads the search towards the most relevant features. Figure 3 depicts the absolute flow of how GWO was combined in the model for optimizing the features. Table 2 displays the main parameters utilized for the Grey Wolf Optimizer (GWO) in feature optimization. These parameters, consisting of convergence control factors, population size, and fitness metrics, are critical for guiding the optimization process efficiently.

Table 2: Grey Wolf Optimizer (GWO) Parameters for Feature Optimization

Parameter	Value
Population Size	30
Maximum Iterations	3
Convergence Control Factor (a)	2 to 0
Alpha, Beta, Delta Wolves	Best Performing
Search Agents	Wolf-based Exploration
Fitness Function	Accuracy, Precision, Recall, F1 Score

3.3.2 Mayfly Optimizer

The Mayfly Optimizer (MO) was integrated in the work to optimize the features for the spoof detection task [27], used by the Vision Transformer (ViT) model. MO is simulated by the flight and mating nature of mayflies, effectively traverses for the optimal set of features extracted from the input data [19]. By choosing the finest features, this behaviour encouraged the optimization algorithm to update the performance of the model, which is given to the Vision Transformer for classification.

The optimization process begins with an initial population of mayflies, where each portrays a different layout of procured features. The MO algorithm robustly organises these features through the interaction of male and female mayflies, hence paving the way for a thorough inspection of the feature space to find the best efficient feature combinations [28][29]. The improved model performance [20] is mainly due to the mayflies nature, like attraction, repulsion, and mating, which helps in sifting chosen features repeatedly. Following the refining process of MO, the optimized feature set then goes to the Vision Transformer (ViT), leading to enhanced classification accuracy for the spoof detection task. The best accuracy achieved throughout the optimization process is 90%, demonstrating the effectiveness of MO in optimizing the features.

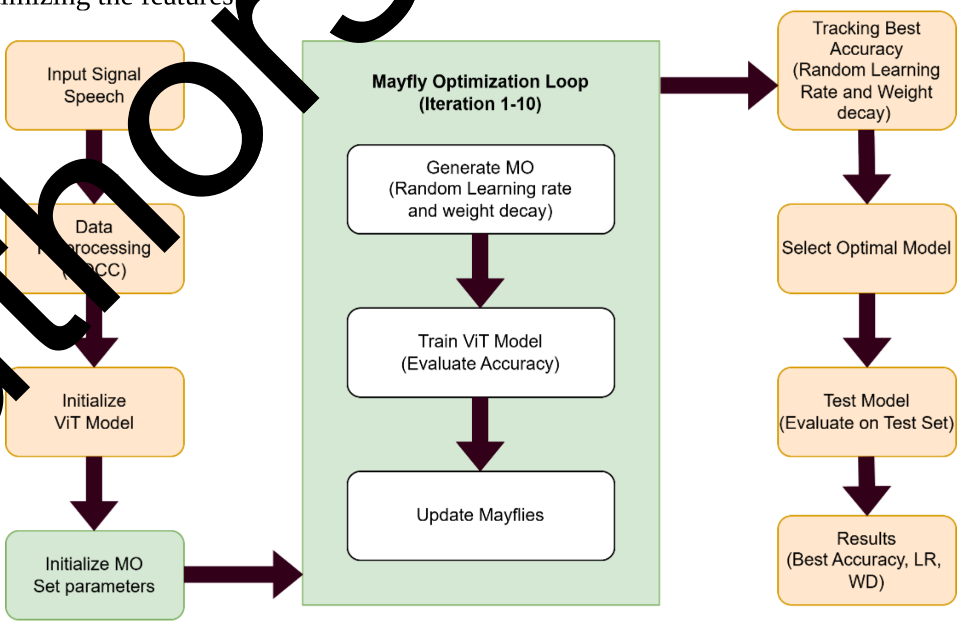


Figure 4: Mayfly Optimizer for Feature Optimization

Position Update For Male Mayflies

The velocity of the male mayfly at iteration $t+1$ is implemented using the following equations: This equation (10) describes how the velocity of a male mayfly is improved by considering both personal and global best positions, adjusted by random factors for observation.

$$\mathbf{V}_i(t+1) = w\mathbf{V}_i(t) + c_1 r_1(\mathbf{P}_i(t) - \mathbf{X}_i(t)) + c_2 r_2(\mathbf{G}(t) - \mathbf{X}_i(t)) \qquad (10)$$

Where:

- $\mathbf{V}_i(t)$ is the velocity of the i-th male at iteration t,
- w is the inertia weight, which stabilises exploration and exploitation,
- c_1 and c_2 are the cognitive and social coefficients, representing self-confidence and group influence,
- r_1 and r_2 are random values uniformly distributed between a range of $[0,1]$,
- $\mathbf{P}_i(t)$ is the personal best position of i-th male,
- $\mathbf{G}(t)$ is the global best position found by all mayflies.

Next, the position of the male mayfly is updated based on the new velocity: Here, $\mathbf{X}_i(t+1)$ represents the new position of the i^{th} male mayfly, calculated by adding the updated velocity to the current position $\mathbf{X}_i(t)$. Equation (11) guarantees that the mayfly moves in the direction determined by its velocity.

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \qquad (11)$$

Position Update For Female Mayflies

The position of the female mayfly is updated based on her attraction to the nearby male mayfly and a random disturbance, as shown below. Equation (12) governs how the position of female mayflies is influenced by their presence to male mayflies and some random movement to explore new positions.

$$\mathbf{X}_f(t+1) = \mathbf{X}_f(t) + \beta e^{-\gamma d_{fm}}(\mathbf{X}_m(t) - \mathbf{X}_f(t)) + \alpha\mathcal{N}(0,1) \qquad (12)$$

Where:

- $\mathbf{X}_f(t)$ and $\mathbf{X}_m(t)$ are the positions of the female and the nearest male, respectively, at iteration t,
- β is the attraction factor between the female and male mayflies,
- γ is the damping coefficient controlling the strength of attraction,
- d_{fm} is the distance between the female and male mayfly (calculated in equation (13)),
- α is the random walk factor adding diversity to the position update,
- $\mathcal{N}(0,1)$ is a normal distribution with mean 0 and variance 1, introducing randomness.

The distance between the female and male mayflies, d_{fm}, is calculated as follows: Here, the Euclidean distance between the positions of the female and male mayflies is calculated. This distance d_{fm} (equation (13)) is used to determine the strength of attraction between the two mayflies in equation (12).

$$d_{fm} = \| \mathbf{X}_m(t) - \mathbf{X}_f(t) \| \qquad (13)$$

Mating Process

If a male and female mayfly come any closer, they can mate, and the position of the offspring is calculated as: Equation (14) computes the position of the offspring, which is the average position of the two parent mayflies, with a small random perturbation added to maintain diversity in the population.

$$\mathbf{X}_{\text{offspring}} = \frac{1}{2}(\mathbf{X}_m(t) + \mathbf{X}_f(t)) + \delta\mathcal{N}(0,1) \qquad (14)$$

Where:

- $\mathbf{X}_m(t)$ and $\mathbf{X}_f(t)$ are the positions of the male and female mayfly,
- δ is a small random factor to introduce diversity in the offspring's position.

These optimized feature sets acquired after the MO-based feature optimization process (described in equations (10) to (14)) are later input to the Vision Transformer (ViT) model for better classification accuracy in spoof detection. Figure 4 shows the complete flow of how MO was integrated into the model for optimizing the features. Table 3 outlines the parameters utilized for the Mayfly Optimizer (MO) in feature optimization. These include key settings such as population size, crossover and mutation rates, and fitness function criteria, all of which are essential for achieving optimal performance.

Table 3: Optimizer (MO) Parameters For Feature Optimization

Parameter	Value
Population Size	30
Crossover Rate	0.7
Mutation Rate	0.1
Maximum Iterations	3[illegible]
Attraction Constant(α)	0.1
Randomness Factor(β)	0.3
Fitness Function	Accuracy, Precision, Recall, F1-Score

3.3.3 Genetic Algorithm

The Genetic Algorithm (GA) [18] was employed for feature optimization of the Vision Transformer (ViT) model deployed in the spoof detection task. GA is a nature-inspired optimization technique that simulates the process of natural selection, where the fittest individuals are right for reproduction to create the next generation [30]. In this context, each individual in the population represents a unique feature set configuration, and GA optimizes these features to revamp the model performance [31][6].

The GA process has three important steps: selection, crossover, and mutation [32]. During the selection phase, individuals (feature sets) are estimated based on their fitness, commonly measured by model performance on a validation set. The best-performing individuals are selected for reproduction, allowing them to advance their optimized features to the next generation. In the crossover phase, selected individuals exchange parts of their feature sets to generate new offspring, combining dominance from both parent configurations. Mutation introduces random changes to some individuals, continuing genetic diversity and preventing premature convergence [33]. This iterative process continues for several generations, allowing GA to refine the feature sets and enhance the model's accuracy[34]. The algorithm's ability to explore and exploit feature combinations led to the development of the spoof detection system [35] as depicted in Figure 5.

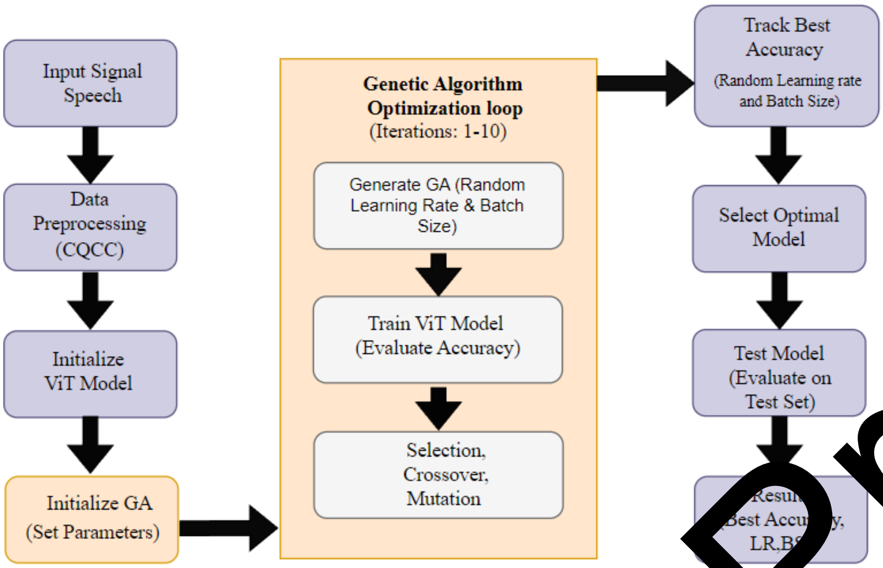


Figure 5: Genetic Algorithm for Feature Optimization

Initial Population:

The GA begins by generating an initial population of individuals (chromosomes), each illustrates a candidate solution (feature set). This is depicted as:

$$P(0) = \{C_1, C_2, \ldots, C_n\} \quad (15)$$

Equation (15) defines the initial population $P(0)$ of the GA, where C_i shows an individual chromosome, and n is the population size.

Fitness Function:

Each chromosome C_i is evaluated using a fitness function, which measures its performance:

$$F(C_i) = \text{Fitness}(C_i) \quad (16)$$

In equation (16), $F(C_i)$ denotes the fitness of a chromosome C_i, determining its effectiveness in solving the optimization problem.

Selection:

In Roulette Wheel Selection, the chance of selecting a chromosome C_i for reproduction is proportional to its fitness:

$$p(C_i) = \frac{F(C_i)}{\sum_{j=1}^{n} F(C_j)} \quad (17)$$

Equation (17) shows how the selection probability $p(C_i)$ is calculated based on the fitness values of the chromosomes, ensuring that fitter chromosomes have a higher chance of being selected.

Crossover:

For single-point crossover, two offspring are generated by combining segments of two parent chromosomes:

$$\text{Offspring}_1 = [C_1^1, C_2^1, \ldots, C_x^1, C_{x+1}^2, \ldots, C_n^2] \quad (18)$$
$$\text{Offspring}_2 = [C_1^2, C_2^2, \ldots, C_x^2, C_{x+1}^1, \ldots, C_n^1] \quad (19)$$

Equations (18) and (19) depict how the offspring are generated by combining parts of their parents' chromosomes, enabling the sharing of features and characteristics.

Mutation:

In bit-flip mutation, individual genes in a chromosome are randomly flipped, introducing diversity:

$$C_{i'} = \begin{cases} 1 - C_i & \text{if mutation occurs at position } i \\ C_i & \text{otherwise} \end{cases} \tag{20}$$

Equation (20) describes the mutation process, where a gene of a chromosome C_i may be flipped, promoting genetic diversity and preventing premature convergence.

New Population:

After selection, crossover, and mutation, a new population of individuals is generated:

$$P(t+1) = \{C_{1'}, C_{2'}, \ldots, C_{n'}\} \tag{21}$$

In equation (21), $P(t+1)$ represents the new population formed from the previous generation's individuals after undergoing selection, crossover, and mutation.

Termination:

The GA terminates if an obstructing criterion is met, such as reaching a maximum number of generations G or achieving a desired fitness level:

$$\text{Stop if: } t \geq G \quad \text{or} \quad \max(F(C_i)) \geq \text{threshold} \tag{22}$$

Equation (22) outlines the termination conditions for the GA, specifying when the optimization process should stop based on generation limits or fitness thresholds. Table 4 outlines the parameters used for the Genetic Algorithm (GA) in feature optimization. These indicate key factors like Selection Method, population size, crossover probability, and mutation probability, which are important for achieving the best outcomes.

Table 4: Genetic Algorithm (GA) Parameters For Feature Optimization

Parameter	**Value**
Population Size	30
Crossover Probability	0.8
Mutation Probability	0.2-0.5
Selection Method	Roulette Wheel or Tournament Selection
Crossover Type	Single-point
Mutation Type	Bit-Flip
Number Of Generations	3
Fitness Function	Accuracy, Precision, Recall, F1-Score

3.4 Vision Transformer for Feature Classification

The Vision Transformer (ViT) is an advanced architecture originally designed for image classification tasks, adapting the transformer model—traditionally used in natural language processing—to handle visual data proficiently [17]. The ViT processes images by dividing them into patches, flattening these patches, and inputting them into a transformer architecture that includes multi-head self-attention mechanisms and feed-forward networks. The ViTModel class is implemented using the *timm* library. Specifically, the *vit_tiny_patch16_224* model was used, architecture is employed, which features a patch size of 16x16 pixels and an input size of 224x224 pixels.

The initial input to the ViT consists of features extracted using Constant Q Cepstral Coefficients (CQCC) [22]. These CQCC features undergo resizing to align with the ViT's input dimensions and normalization to enhance training stability and convergence. Normalization typically involves scaling the input features to a standard range, such as [0, 1] or mean-centered values.

Before these CQCC features are directed into the ViT, they undergo various feature optimization techniques to refine the inputs. Three distinct optimization methods are implemented: Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO). Each of these techniques employs unique strategies to search for optimal parameter values, effectively enhancing the quality of the CQCC features.

Once optimized, the refined CQCC features [22] are then fed into the ViTModel class. In the constructor (__init__), the model is initialized using timm.create_model, specifying the desired ViT architecture and the number of output classes for classification. The forward method outlines how input data flows through the model, producing class probabilities as output. Figure 6 shows the detailed block diagram of the proposed method.

During the training phase, the model is optimized using a suitable loss function such as cross-entropy loss, effective for multi-class classification tasks. The Adam optimizer, which adapts the learning rate throughout training for better convergence. Various hyperparameters, including learning rate, batch size, and the number of epochs, are defined to guide the training process. The model is trained on the training dataset while being monitored with validation data to mitigate the risk of overfitting.

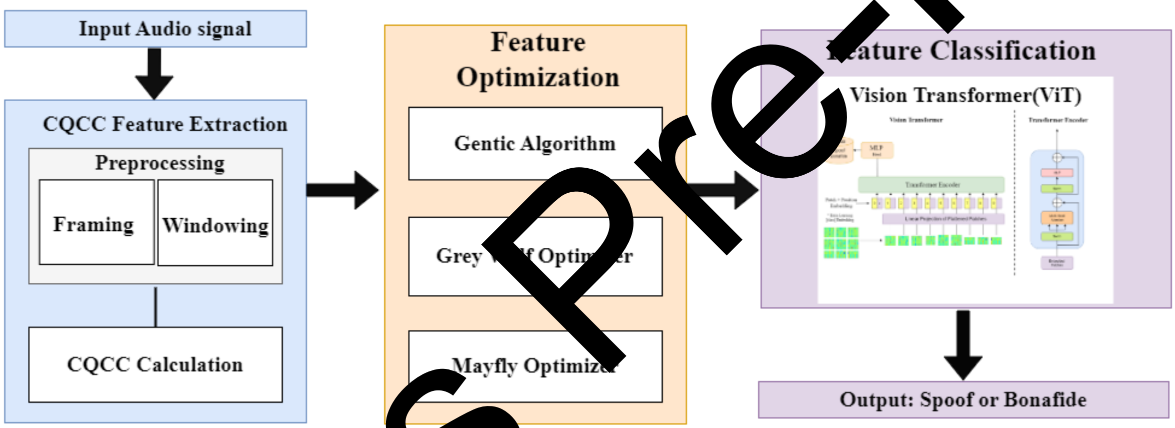


Figure 6: Detailed Block Diagram of Proposed Method

4. Results and Discussion

In the proposed model, Constant Q Cepstral Coefficients (CQCC) are combined with a Vision Transformer (ViT) to improve the accuracy of spoof detection in audio data. To improve feature optimization, three different strategies are used. Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO) are implemented. Each strategy attempted to maximize accuracy while reducing training time per epoch.

The model without optimization techniques demonstrated a relatively lower accuracy and a longer computational time. As shown in Figure 7, the model without optimization revealed a higher number of false positives and false negatives, indicating that the model struggled to accurately classify both spoofed and bonafide samples. In Figure 8, the accuracy vs. loss graph showed slower learning with instability, and Figure 9 the ROC curve reflected a lower area under the curve (AUC), showing subpar performance in distinguishing spoofed and bonafide samples.

When optimized with the Grey Wolf Optimizer (GWO), the model showed significant improvements. As depicted in Figure 10, the confusion matrix with GWO indicated a more balanced classification, with fewer false positives and false negatives. Figure 11 accuracy vs. loss graph the accuracy vs. loss graph revealed a steady increase in accuracy and a faster decrease in loss, indicating more stable and efficient training, and Figure 12 the ROC curve for GWO also demonstrated a higher AUC, highlighting the improved ability to distinguish between spoofed and bonafide samples.

The Mayfly Optimizer (MO) further enhanced the model's performance. The confusion matrix for MO in Figure 13 indicates even fewer false positives and false negatives compared to GWO, indicating better classification accuracy. The accuracy vs. loss graph in Figure 14 demonstrated superior performance, with higher accuracy and faster convergence. An outstanding performance of the model in categorizing audio samples is shown by the ROC curve for MO in Figure 15 which is even higher than GWO.

Finally, the Genetic Algorithm (GA) gave its best outcomes. The least number of false positives and false negatives was shown by the confusion matrix for the GA-optimized model in Figure 16, and hence, attaining the most classification accuracy. The accuracy vs. loss graph shown in Figure 17 specifies the greatest training precision with the fastest convergence. The ROC curve in Figure 18 demonstrated the highest AUC, depicting the model's exceptional ability to distinguish bonafide audio from the spoofed one. The impact of different optimization methods in terms of normalized minimum Tandem Detection Cost Function (tDCF) and Equal Error Rate (EER) is depicted by the experimental results displayed in Table 7. Precisely, an EER of 0.1341 and a tDCF of 0.1332 were attained by the Mayfly Optimization (MO). An EER of 0.1210 and a tDCF of 0.1204 were achieved by the Grey Wolf Optimization (GWO). Specifically, the Genetic Algorithm (GA) outshone the others with a tDCF of 0.0348 and the lowest EER of 0.0358, hence depicting its outstanding performance. Hence, understanding that the GA method functions very well in optimizing the system for improved reliability and detection accuracy.

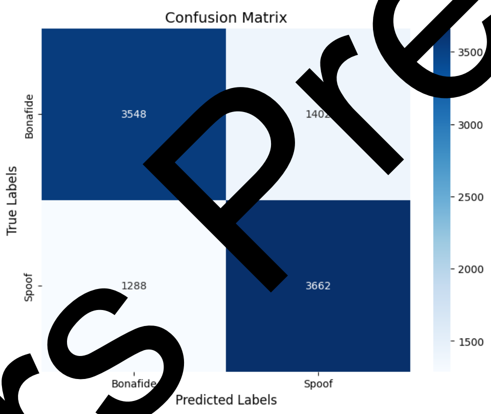


Figure 7: Confusion Matrix of the model Without Optimization

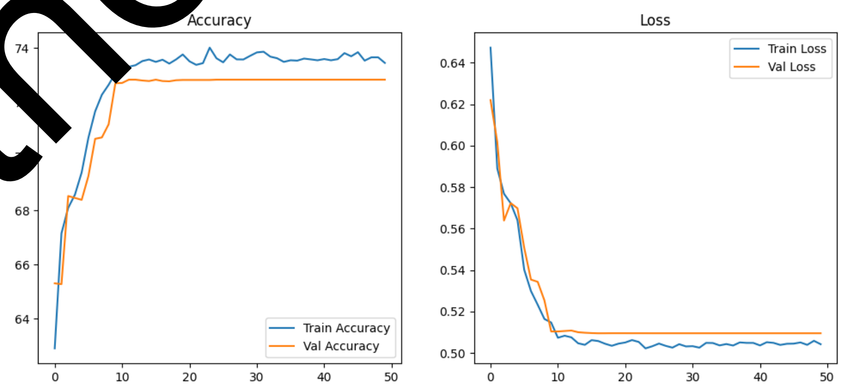


Figure 8: Accuracy vs Loss Graph of the model Without Optimization

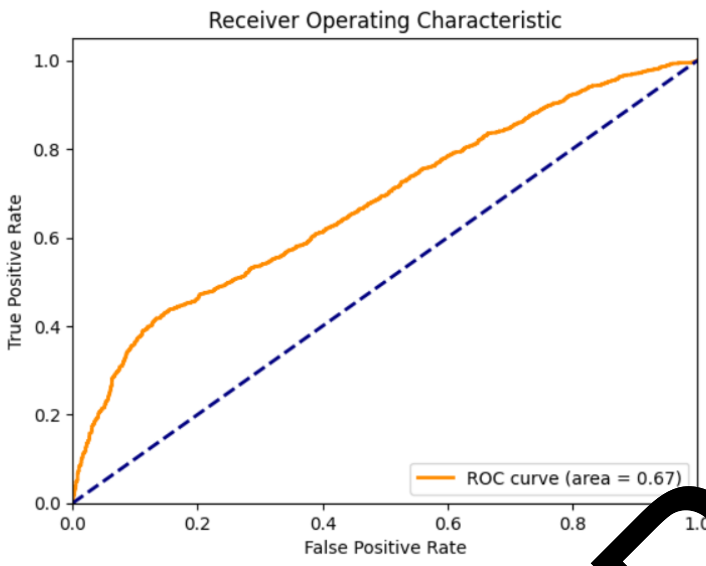


Figure 9: ROC Curve of the model Without Optimization

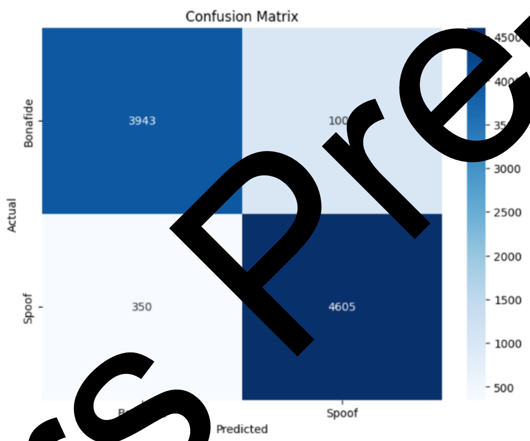


Figure 10: Confusion Matrix of the model using Grey Wolf Optimizer

Figure 11: Accuracy vs Loss Graph of the model using Grey Wolf Optimizer

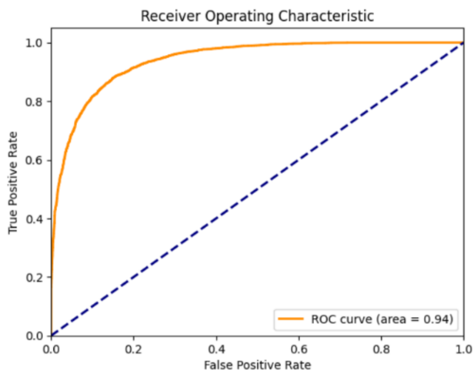


Figure 12: ROC Curve of the model using Grey Wolf Optimizer

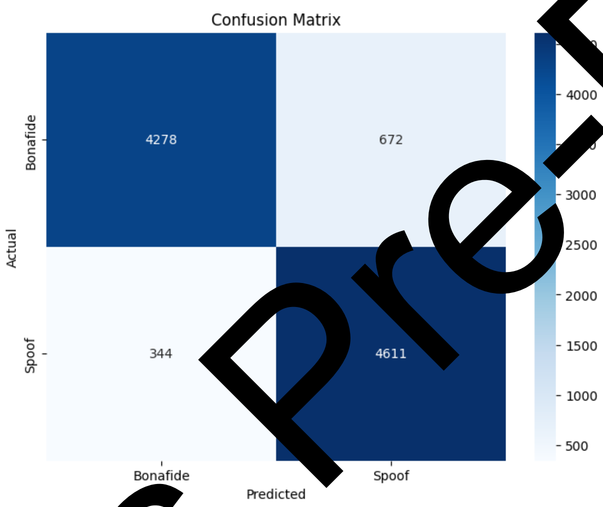


Figure 13: Confusion Matrix of the model using Mayfly Optimization

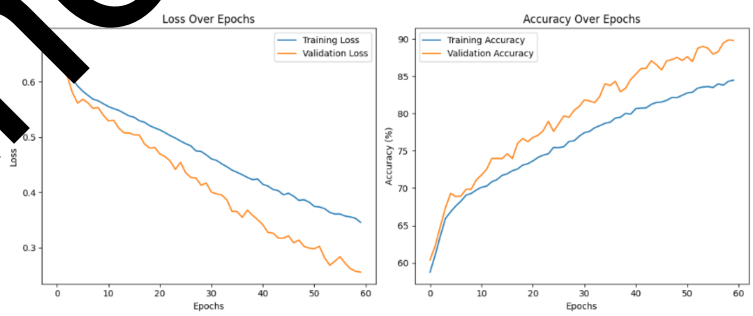


Figure 14: Accuracy vs Loss Graph of the model using Mayfly Optimization

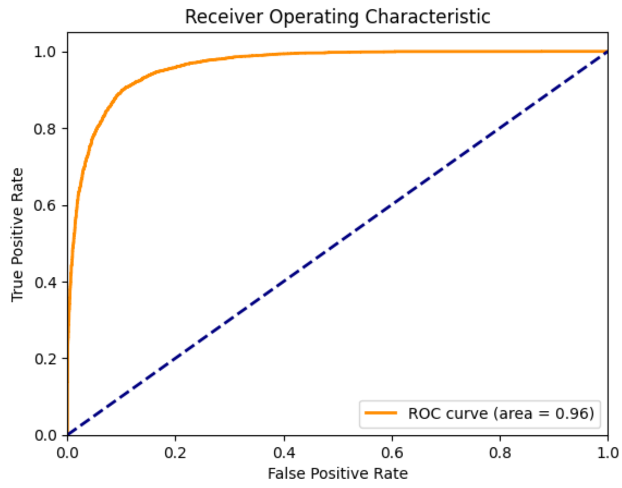


Figure 15: ROC Curve of the model using Mayfly Optimization

Figure 16: Confusion Matrix of the model using Genetic Algorithm

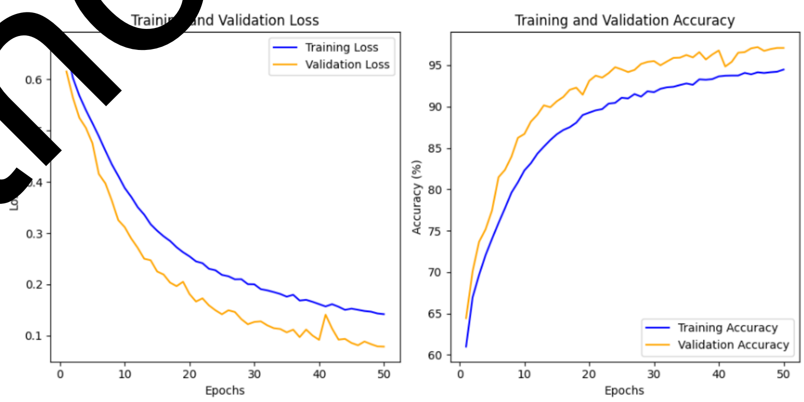


Figure 17: Accuracy vs Loss Graph of the model using Genetic Algorithm

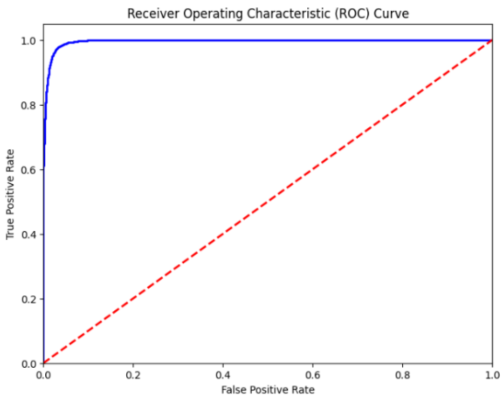


Figure 18: ROC Curve of the model using Genetic Algorithm

Table 5: Performance Metrics for Different Optimization Techniques

Techniques	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Without Optimizer (WO)	73	72	74	73
Grey Wolf Optimizer (GWO)	86	82	93	87
Mayfly Optimizer (MO)	90	87	93	90
Genetic Algorithm (GA)	97	97	98	97

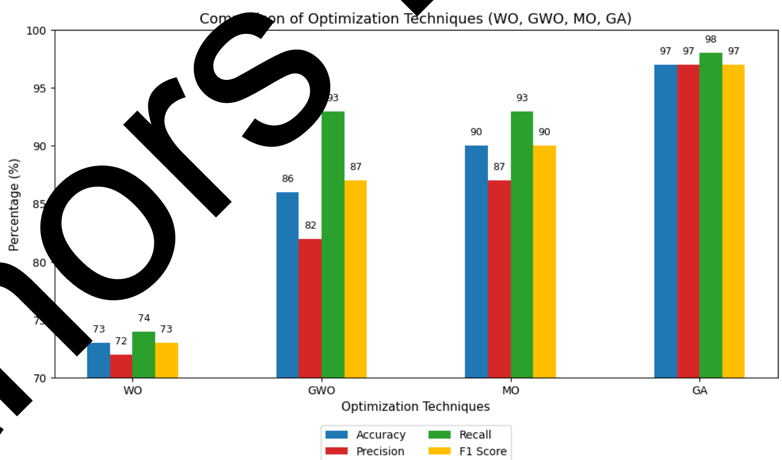


Figure 19: Comparison of Optimization Techniques (WO, GWO, MO, GA)

Table 6: Comparison of Time for Training and Testing of Model with and without Optimization

Method	Training Time(s)	Testing Time (s)
Without Optimization	4.85s	0.67s
Grey Wolf Optimization	*4.5s*	*0.45s*
May Fly Optimization (MO)	*4.3s*	*0.50s*
Genetic Algorithm (GA)	*3.64s*	*0.43s*

Table 7: Comparison of EER and tDCF

Method	EER	tDCF
May Fly Optimization (MO)	*0.1341*	*0.1332*
Grey Wolf Optimization (GWO)	*0.1210*	*0.1204*
Genetic Algorithm (GA)	*0.0358*	*0.0348*

5. Performance Comparison with Existing Models

In this study, Constant Q Cepstral Coefficients (CQCCs) are used for spoof detection in Automatic Speaker Verification (ASV) systems. CQCCs, procured from the Constant Q Transform, obtain fine spoofing artifacts along with varying spectro-temporal resolution. CQCCs, after being tested on various datasets, which also include ASV spoof 2015, surpass traditional features such as MFCCs, showcasing good performance and hence signifying the importance of customized configurations for a variety of spoofing schemes [14]. This study helps inspect deep-fake audio classification and, therefore, evaluates the currently present techniques for forensic investigation of fake audio detection. Although VGG-16 gives excellent results for MFCC features, it studies different deep learning algorithms and thereafter proves that a Custom Architecture works supremely for mel-spectrum images, spectrogram, and chromagram. This helps forensic investigators differentiate between real and fake sounds [10]. This paper proposes a segment-based anti-spoofing detection (SASD) method for embedded speech recognition, focusing on anti-spoofing features rather than speech context or voiceprints. It divides speech into word and silent segments, extracting Constant Q Cepstral Coefficients (CQCCs) for words and Zero Crossing Rate (ZCR) for silence. Combining these features with a biased decision strategy, SASD improves anti-spoofing accuracy by up to 33.47% and reduces time overhead by 69.10% compared to existing methods, as demonstrated by experiments on ASVspoof 2021 datasets[16]. The proposed method, which integrates CQCC features with a Vision Transformer and Genetic Algorithm optimization, achieved the highest accuracy of 97%, precision of 97%, recall of 98%, and an F1-score of 97%. Table 6 shows the training and testing times of models with and without optimization. The testing and training time reduced significantly while using the GA Optimization algorithm.

Table 8: Comparison of Spoof Detection Methods by Accuracy in Decreasing Order

Refrences	Methodology	Accuracy	Precision	Recall	F1-Score
Todisco et al. (2017) [14]	CQCC Evaluation, Spoof Detection on ASVspoof 2015	72.0%	72.0%	70.0%	71.0%
Mcuba et al. (2023) [10]	Feature extraction (MFCC, Mel-spectrum, Chromagram, Spectrogram), model evaluation (VGG-16, Custom Architecture) and audio datasets (ASVspoof).	85.9%	84.12%	85.90%	85.00%
Zhang et al. (2022) [16]	Segment-based Anti-Spoofing Detection (SASD), word CQCC (WCQCC), average zero crossing rate (AZCR) and ASVspoof 2021	92.0%	91.5%	89.0%	90.2%
Proposed method	CQCC, Genetic Algorithm, Vision Transformer	97%	97%	98%	97%

6. Conclusions and Future Directions

In this work, a spoof detection model was developed by integrating Constant Q Cepstral Coefficients (CQCC) with a Vision Transformer (ViT) and enhancing feature optimization through the Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), and Mayfly Optimizer (MO). The results demonstrate the effectiveness of these optimization techniques in significantly improving model performance. Among them, the Genetic Algorithm achieved the highest accuracy of 97% with superior efficiency, demonstrating the best balance across precision, recall, and F1-score.The advantage of the Genetic Algorithm lies in its ability to efficiently explore a wide search space through mechanisms like crossover and mutation, enabling the selection of relevant features while avoiding suboptimal solutions. This ensures robust optimization, leading to improved performance in spoof detection tasks. As shown in Table 5, the Genetic Algorithm outperformed MO and GWO, especially in accuracy and computational efficiency. Table 6 highlights the Genetic algorithm's capability to reduce both training and testing times, where it achieved training times as low as compared to those without optimization.

This study highlights the importance of advanced optimization methods in refining feature representation, significantly enhancing model performance in anti-spoofing tasks. The research contributes to automatic speaker verification by demonstrating how feature optimization can increase the effectiveness of deep learning models against spoofed audio. Future work could explore additional feature extraction methods like Mel-frequency cepstral coefficients (MFCC) and hybrid optimization algorithms to further improve model adaptability and accuracy. Real-time implementation, transfer learning, and robustness to noisy environments also offer promising avenues for expanding the model's application in practical voice authentication systems. These advancements would make the model more adaptable and effective in real-world machine learning scenarios.

References

[1]. Kwak, I. Y., Kwag, S., Lee, J., Jeon, Y., Hwang, J., Choi, H. J., ... & Yoon, J. W. (2023). Voice spoofing detection through residual network, max feature map, and depthwise separable convolution. *IEEE Access*, *11*, 49140-49152.

[2]. Neelima, M., & Prabha, I. S. (2024). Hybrid Feature Optimization for Voice Spoof Detection Using CNN-LSTM. *Traitement du Signal*, *41*(2), [illegible].

[3]. Yi, J., Wang, C., Tao, J., Zhang, X., Zhang, C. Y., & Zhao, Y. (2023). Audio deepfake detection: A survey. *arXiv preprint arXiv:2308.14970*.

[4]. Shaaban, O. A., Yildirim, R., & Alguttar, A. A. (2023). Audio deepfake approaches. *IEEE Access*, *11*, 132652-132682.

[5]. Chaudhari, A., & Shedge, D. K. (2022, March). Integration of cqcc and mfcc based features for replay attack detection. In *2022 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-[illegible]). IEEE.

[6]. Yang, J., & Zhang, X. (2024, March). A Modified Parameter Optimization Method of Support Vector Machine Based on Genetic Algorithm. In *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)* (pp. 878-881). IEEE.

[7]. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

[8]. Zervoudakis, K., & Tsafarakis, S. (2020). A mayfly optimization algorithm. *Computers & Industrial Engineering*, *145*, 106559.

[9]. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., ... & Tao, D. (2022). A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, *45*(1), 87-110.

[10]. Mcuba, M., Singh, A., Ikuesan, R. A., & Venter, H. (2023). The effect of deep learning methods on deepfake audio detection for digital investigation. *Procedia Computer Science*, *219*, 211-219.

[11]. Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, *30*, 413-435.

[12]. Chen, Z., Li, J., Li, J., Zhu, X., & Li, C. (2022). GNSS multiparameter spoofing detection method based on support vector machine. *IEEE Sensors Journal*, *22*(18), 17864-17874.

[13]. Anagha, R., Arya, A., Narayan, V. H., Abhishek, S., & Anjali, T. (2023, December). Audio deepfake detection using deep learning. In *2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 176-181). IEEE.

[14]. Todisco, M., Delgado, H., & Evans, N. (2017). Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, *45*, 516-535.

[15]. Ye, Y., Lao, L., Yan, D., & Lin, L. (2019, April). Detection of replay attack based on normalized constant q cepstral feature. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)* (pp. 407-411). IEEE.

[16]. Zhan, J., Pu, Z., Jiang, W., Wu, J., & Yang, Y. (2022). Detecting spoofed speeches via segment-based word CQCC and average ZCR for embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *41*(11), 3862-3873.

[17]. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., ... & Feng, J. (2021). Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*.

[18]. Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)* (pp. 380-384). IEEE.

[19]. Gao, Z. M., Zhao, J., Li, S. R., & Hu, Y. R. (2020, November). The improved mayfly optimization algorithm. In *Journal of physics: conference series* (Vol. 1684, No. 1, p. 012077). IOP Publishing.

[20]. Zhou, D., Kang, Z., Su, X., & Yang, C. (2022). An enhanced Mayfly optimization algorithm based on orthogonal learning and chaotic exploitation strategy. *International Journal of Machine Learning and Cybernetics*, *13*(11), 3625-3643.

[21]. Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., ... & Ling, Z. H. (2020). ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, *64*, 101114.

[22]. Schörkhuber, C., Klapuri, A., & Sontacchi, A. (2013). Audio pitch shifting using the constant-Q transform. *Journal of the Audio Engineering Society*, *61*(7/8), 562-572.

[23]. Ahmed, N., Natarajan, T., & Rao, K. R. (2006). Discrete cosine transform. *IEEE transactions on Computers*, *100*(1), 90-93.

[24]. Babatunde, O. H., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A genetic algorithm-based feature selection.

[25]. Shahin, I., Alomari, O. A., Nassif, A. B., Afyouni, I., Hashem, I. A., & Elnagar, A. (2023). An efficient feature selection method for arabic and english speech emotion recognition using Grey Wolf Optimizer. *Applied Acoustics*, *205*, 109279.

[26]. Ladi, S. K., Panda, G. K., Dash, R., Ladi, P. K., & Dhupar, R. (2022). A novel grey wolf optimisation based CNN classifier for hyperspectral image classification. *Multimedia Tools and Applications*, *81*(20), 28207-28230.

[27]. Bogar, M. N., Shirodkar, [illegible] D., Kulkarni, O., Jawade, S., & Kakandikar, G. (2024). Mayfly optimization algorithm: a review. *Journal of Mechatronics and Artificial Intelligence in Engineering*, *[illegible]*(1), 17-[illegible]0.

[28]. Bhattacharyya, T., Chatterjee, B., Singh, P. K., Yoon, J. H., Geem, Z. W., & Sarkar, R. (2020). Mayfly in harmony: A new hybrid meta-heuristic feature selection algorithm. *IEEE Access*, *8*, 195929-195945.

[29]. Gupta, I. K., Choubey, A., & Choubey, S. (2022). Mayfly optimization with deep learning enabled retinal fundus image classification model. *Computers and Electrical Engineering*, *102*, 108176.

[30]. Rojas, I., González, J., Pomares, H., Merelo, J. J., Castillo, P. A., & Romero, G. (2002). Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *32*(1), 31-37.

[31]. Inagaki, J., Haseyama, M., & Kitajima, H. (1999, May). A genetic algorithm for determining multiple routes and its applications. In *1999 IEEE International Symposium on Circuits and Systems (ISCAS)* (Vol. 6, pp. 137-140). IEEE.

[32]. Javadi, A. A., Farmani, R., & Tan, T. P. (2005). A hybrid intelligent genetic algorithm. *Advanced Engineering Informatics*, *19*(4), 255-262.

[33]. Li, Q., Zhu, H., Zhang, Z., Lu, R., Wang, F., & Li, H. (2019, May). Spoofing attacks on speaker verification systems based generated voice using genetic algorithm. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.

[34]. Jenkins, J., Roy, K., & Shelton, J. (2020). Using deep learning techniques and genetic-based feature extraction for presentation attack mitigation. *Array*, *7*, 100029.

[35]. Nixon, K. A., Aimale, V., & Rowe, R. K. (2008). Spoof detection schemes. In *Handbook of biometrics* (pp. 403-423). Boston, MA: Springer US.