

Research Article

DeepConvIEHO-AdaMaxNet: Secure and Robust Deepfake Detection using Optimized Deep Neural Networks

Padmavati E. Gundurti^{1,2,*} , Thangaraj Parameswaran¹ 

¹ Department of Computer Science and Engineering, School of Engineering and Technology, CMR University, Bangalore, 562149, India

² Department of Computer Science and Engineering, BVRIT Hyderabad College of Engineering for Women, Jawaharlal Nehru Technological University Hyderabad, Hyderabad, 500090, India

* Corresponding Author: Padmavati E Gundurti, E-mail: padmavatiegundurti@gmail.com

Article Info	Abstract
Article History	Advances in artificial intelligence (AI) and deep learning (DL) have enabled sophisticated Deepfake technology, which creates highly realistic fake videos by mapping one person's expressions onto another's face. While these innovations raise concerns about privacy, trust, and misuse in politics and security, they underscore the urgent need for robust detection and prevention measures to protect digital integrity. This paper addresses the challenges of deepfake detection, including high computational demands, difficulty capturing temporal inconsistencies, and susceptibility to adversarial attacks, by proposing a novel framework, DeepConvIEHO-AdaMaxNet. The framework leverages a Gaussian-Adaptive Bilateral Filter (GABF) for noise reduction and edge preservation during preprocessing, followed by a Convolutional neural network (CNN)-based classification optimized using an Improved Elephant Herding Optimization (IEHO) algorithm integrated with the AdaMax optimizer. This bio-inspired approach enhances the hyperparameter tuning process, overcoming limitations such as slow convergence and vulnerability to local minima in traditional methods. Additionally, the framework incorporates adversarial training within federated learning (FL) paradigm to improve robustness against manipulations. The suggested approach offers 96.41% of an area under the receiver operating characteristic curve (AUC), and 97.50% accuracy, as well as for class 0 achieve 100% specificity, 90% recall, 100% precision, and 95% F1-measure and class 1 achieves 93.53% specificity, 90.50% sensitivity, 100% recall, 90% precision, and 95% F1-measure using the Celeb-DF dataset. Evaluated through comprehensive metrics, the proposed model demonstrates superior performance compared to conventional CNNs, showcasing its potential in reliable deepfake detection.
Received Apr 05, 2025	
Revised Jul 21, 2025	
Accepted Aug 16, 2025	
Keywords	
Fake Video Detection	
Deep Learning	
Deepfake	
Elephant Herding Optimization	
Adamax	
CNN	



Copyright: © 2025 Padmavati E. Gundurti, and Thangaraj Parameswaran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY 4.0) license.

1. Introduction

Technology has never been more vital in today's environment. Due to the increasing dependence on digitization and the Internet of Things (IoT), security incidents such as unauthorized access, malware

attacks, zero-day attacks, data breaches, denial-of-service (DoS) attacks, social engineering or phishing, and cyberbullying have surged exponentially in recent years [1]. The term "cybersecurity" describes a collection of security procedures that may be applied to protect user activity and the digital environment from intrusions and unauthorized access [2]. According to device connectivity, the IoT's idea has improved accessibility, authenticity, availability, flexibility, security, and compatibility [3]. Although IoTs are vulnerable to assaults due to their numerous areas for attack, their recentness, and the resulting lack of security norms and laws. The variety of assaults that attackers can employ against IoTs depends on the specific system component they are attempting to penetrate and the result they hope to achieve. As a result, several studies on cybersecurity related to IoT have been conducted. This includes using AI methods to protect IoT devices from intrusions, often by detecting unusual activity that may indicate an imminent attack [4]. AI is the development of sophisticated computer programmes that take on human mentalities and can carry out tasks like speech recognition and language processing. It is an all-encompassing scientific approach that combines philosophy, computer science, and arithmetic to create intelligent systems with intelligence traits. This is frequently used to describe devices that mimic human perceptual abilities, such as swotting and problem-solving [5].

An essential tool in the battle against cybercrime is AI, which enables automated reaction actions, real-time threat detection and prevention, and assists organizations in staying ahead of developing threats [6]. Conventional security methods are insufficient as cybercrime continues to increase. AI-based cybersecurity solutions analyze massive quantities of information, spot trends, and anticipate potential threats using machine learning (ML) and DL algorithms [7]. When identifying novel and emerging threats that conventional signature-based techniques can miss, these solutions are beneficial. AI can swiftly analyze massive amounts of information, identify patterns, and learn from past experiences to predict potential threats. AI's automation of threat identification and response reduces the workload for human security analysts, which is its primary benefit in cybersecurity [8]. The rapid advancement of artificial neural networks has enabled realistic face swapping in multimedia, leading to the rise of "Deepfake" technology. This involves creating convincing fake videos that can spread misinformation, incite political unrest, blackmail individuals, or simulate terrorist events, posing significant concerns [9].

People, society [10], and the credibility of digital media [11] are all seriously threatened by deepfakes. They may result in inaccurate information, swaying of public opinion, and a decline in confidence in media outlets. Additionally, they may be used to manipulate politics, affecting elections and undermining public confidence in the political system [12, 13]. Additionally, deepfakes may produce convincing impersonations for monetary or personal benefit or non-consensual explicit content to harass and harm one's reputation. A problem of authenticity and confidence in digital evidence utilized in criminal justice and legal processes is also brought on by the advent of deepfakes. These effects underscore the significant

threat posed by deepfakes, particularly in the sensitive international political climate and the high stakes associated with global conflicts. There might be disastrous outcomes if deepfakes and fake news are weaponized in the current media war [14]. To keep up with the rapid advancement of generative AI, deepfake detection methods must be continuously enhanced [15]. For academics and experts to create strong countermeasure techniques and to provide the proper foundation for detecting and mitigating deepfakes, literature reviews are necessary to stay up to date with the rapidly evolving fields of AI and deepfakes. The key contribution of the suggested article is given as:

- The research introduces the DeepConvIEHO-AdaMaxNet framework, which combines IEHO with the AdaMax optimizer, to optimize CNN hyperparameters, addressing challenges such as slow convergence, local optima, and inefficient hyperparameter tuning in traditional CNNs.
- A GABF is utilized to improve image quality by reducing noise while preserving edges, overcoming limitations in conventional noise reduction methods, and enhancing feature extraction accuracy for deepfake detection.
- The framework integrates adversarial training within an FL approach, increasing the model's resilience to adversarial manipulations and ensuring more reliable performance in detecting manipulated content.
- The proposed method addresses the computational challenges of processing large-scale video datasets by optimizing CNNs with IEHO-AdaMax, thereby achieving a better balance between exploration and exploitation, which leads to improved detection accuracy and efficiency.

2. Literature Review

Bansal, et al. [16] analyzed that the insufficiency of effective models and the high processing costs make it difficult to recognize fake videos. The combination of mobNet, separable convolution, max-pooling layers, the Swish activation function, and the XGBoost classifier has been presented as a novel model architecture called DFN (Deep Fake Network). The DFN method demonstrates the highest level of precision and accuracy. Also, the model can manage any face modification, increasing its strength, generalizability, and lightweight nature. Despite its high accuracy and ability to handle various face modifications, the DFN model may face limitations in scalability and computational efficiency, making it challenging to deploy in real-time or resource-constrained environments.

Khormali and Yuan [17] presented an attention-based DeepFake detection (ADD) method that uses fine-grained and spatial locality attributes of artificially synthesized videos for enhanced detection. To improve the CNN classifier, ADD employs face close-up and face shut-off data augmentation techniques, concentrating on modified input images and forgery areas for decision-making. The outcomes show that ADD considerably improves the presentation of all four standard classifiers, particularly with the ResNet backbone, outperforming existing approaches. However, the model may suffer from increased

computational complexity and processing time, potentially hindering real-time application and deployment in resource-limited settings.

Ismail, et al. [18] analyzed how face-swapping deepfake techniques are increasing the realism of fake videos, posing a substantial danger to privacy and necessitating a basic understanding of the distinction between actual and deepfake content. This article proposes a novel technique called You Only Look Once (YOLO-CNN-XGBoost), which combines convolutional neural networks with extreme gradient boosting. Face features are extracted using the InceptionResNetV2 CNN, and face area is extracted from video frames by the YOLO face detector. XGBoost, a high-level CNN network recognizer, uses this information. However, the model may suffer from a limited capability to handle the complex temporal and spatial relationships inherent in video data, compared to deep neural network approaches, and may also face potential difficulties in real-time implementation due to its complex architecture.

Xu, et al. [19] created a new approach to identify DeepFake videos to counter this issue. The process entails building texture characteristics using the wavelet transform of the face region, the gradient domain, the standard deviation, and the grey-level co-occurrence matrix. These characteristics are used with a feature selection technique to create a discriminant feature vector. Support Vector Machine (SVM) is then utilized to categorize the videos at the frame level. The suggested method's perfect performance is demonstrated by experimental findings on popular DeepFake datasets, demonstrating its efficacy in identifying DeepFake videos. As a result, creating efficient detection techniques was crucial to stopping the spread of misinformation. However, it may not capture temporal inconsistencies due to a lack of temporal information, requires frequent updates, lacks robustness against adversarial attacks, and struggles with high-dimensional feature spaces or large-scale datasets.

Kandasamy, et al. [20] presented a unique approach in which the initial phase used a sparse autoencoder with a graph long short-term memory (SAE-GLSTM) approach during training to extract feature frames from fake films. The analyzed findings demonstrated the efficacy of the suggested approach. However, it faces computational complexity, time-consuming training, overfitting, feature extraction risks, graph construction challenges, adversarial attacks, and limited effectiveness in specific datasets or real-world scenarios.

The existing DFN model may face scalability and computational efficiency limitations, making deploying in real-time or resource-constrained environments difficult. Its complex architecture may limit its ability to handle temporal and spatial relationships in video data, making real-time implementation difficult. The existing models may not capture temporal inconsistencies, require frequent updates, lack robustness against adversarial attacks, and struggle with high-dimensional feature spaces or large-scale datasets. It also faces challenges, including computational complexity, time-consuming training, overfitting, risks associated with feature extraction, difficulties in graph construction, and limited

effectiveness in specific datasets or real-world scenarios. To overcome the issues, this paper proposes a novel model for deepfake detection.

3. Proposed Methodology

Deepfake detection involves identifying and mitigating digitally manipulated media, typically created using artificial intelligence, to ensure the authenticity of videos and images and protect against misinformation and fraud. However, deepfake detection using videos involves high computational requirements, difficulty in capturing subtle temporal inconsistencies, vulnerability to adversarial attacks, and challenges in processing large-scale datasets in real-time. To overcome this, the paper proposes a novel framework, namely DeepConvIEHO-AdaMaxNet (Deep Convolutional Improved Elephant Herd Optimisation with Adamax Network), for deepfake detection by tuning and updating the hyperparameters of CNN, as shown in Figure 1. Initially, the preprocessing step involves converting the input video into corresponding frames. However, video frames face drawbacks such as difficulty in reducing noise without blurring edges, challenges in preserving crucial details, and issues with maintaining high image quality, which can negatively impact the accuracy of feature extraction. To address the inherent problem with the Gaussian range kernel when applied to noise-filtering inputs and its impact on the edge-preserving image smoothing process, this paper employs a GABF that enhances image quality by reducing noise while preserving edges, thereby improving the efficiency of models in detecting deepfakes. After preprocessing, the feature extraction and classification occur using an optimized deep-learning model. However, the traditional convolutional model faces drawbacks including inefficient hyperparameter search, slow convergence, vulnerability to local minima, and difficulty in dynamically adapting learning rates. CNNs use hyper-parameters to provide high-accuracy outputs, which involves more computing time and effort. To overcome this, the paper suggested the advanced version of CNN using a bioinspired metaheuristic approach that depends on an improved form of EHO. However, due to the stochastic nature of EHO and its improper balance between exploration and exploitation, it is often limited to local optima, resulting in limited analytical capability and slower convergence speeds. This research proposes an IEHO-AdaMax to address the issue of low convergence variability in EHO, aiming to achieve a better balance between the exploration and exploitation stages. The process is applied to a CNN classifier for deepfake detection as fake or real, leveraging spatial-spectral features and considering neighbourhood dependencies to enhance classification accuracy. However, the deepfake detection models are vulnerable to adversarial attacks where subtle manipulations can evade detection algorithms, compromising their reliability. To overcome the adversarial attack in deepfake detection using FL methods within the CNN model involves several strategies to enhance model robustness and resilience. The adversarial training within FL frameworks helps increase the security and reliability of deepfake detection models by preparing them to withstand potential

adversarial manipulations and ensuring more dependable performance in detecting manipulated content. Therefore, our suggested model will perform better than the conventional convolutional neural networks.

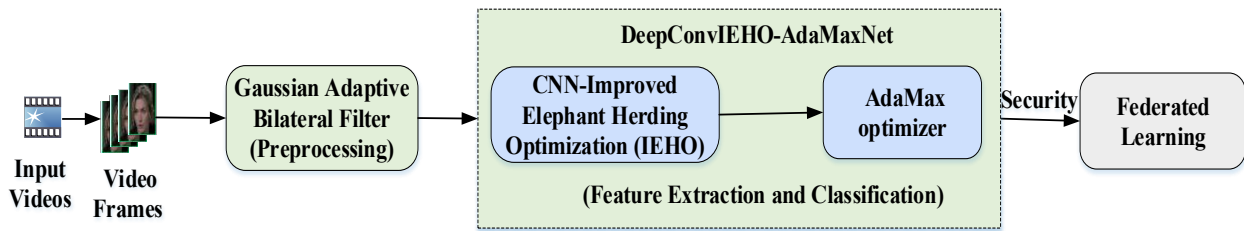


Figure 1. Block diagram of the proposed Deep ConvIEHO-AdaMaxNet framework.

3.1. Preprocessing using GABF

When the guidance g and noise filtering input I are similar, the BF considerably deteriorates. For instance, the range kernel is affected by additive noise, resulting in limited smoothing for the bilateral kernel. Therefore, a vital feature of the GABF is that g and I are non-identical. This means that rigorous protection of the image contours is required as soon as a low-pass filtering procedure is carried out to provide a guide for the range kernel.

First, the Gaussian blur approach, which is the same as low-pass filtering, is applied to a provided image I to generate a low-pass guidance image. A weighted average of the pixels in the surrounding locations is used to achieve this, with the weight diminishing from the centre position i . An illustration of this procedure is as follows:

$$f(i) = \sum_j W_{i,j}^g(g) I_j, \quad (1)$$

where the filter kernel is specified by

$$W_{i,j}^g(g) = \frac{1}{K_i} \exp\left(-\frac{\|i-j\|^2}{\sigma_s^2}\right), \quad (2)$$

To filter noisy inputs, we ensure that g and I are distinct. Specifically, the Gaussian spatial kernel is applied to the filtering input I , while the Gaussian range kernel determines the influence of pixels based on the low-pass guidance $\bar{g}_j$, derived from a Gaussian blur procedure on I_i . As a result, the Gaussian-adaptive bilateral kernel is described as:

$$W_{i,j}^{gabf}(I, \bar{g}) = \frac{1}{K_i} \exp\left(-\frac{\|i-j\|^2}{\sigma_s^2}\right) \exp\left(-\frac{\|I_i - \bar{g}_j\|^2}{\sigma_r^2}\right), \quad (3)$$

where Eq. (3) may be used to derive $\bar{g}$, which is the low-pass guidance. Even if the filtering input I is biased by numerous noise compositions, the Gaussian spatial kernel on I joined with the Gaussian range kernel from $\bar{g}$ confirms that the edges and contours of the image are strictly preserved while smoothing the surfaces of objects in the filtering output $f(i)$. Therefore, the GABF's filtering output $f(i)$ may be given as

$$f(i) = \sum_j W_{i,j}^{gabf}(I, \bar{g}) I_j. \quad (4)$$

3.2. Proposed DeepConvIEHO-AdaMaxNet Framework

Deepfake detection is a critical challenge due to the increasing sophistication of manipulated content and the limitations of traditional CNNs. Existing CNN-based models often encounter issues such as inefficient hyperparameter tuning, slow convergence, and vulnerability to adversarial attacks. This research introduces an advanced CNN model optimized with an Improved Elephant Herding Optimization algorithm and the AdaMax optimizer (IEHO-AdaMax), aiming to enhance classification accuracy, robustness, and computational efficiency while addressing adversarial attack vulnerabilities through FL strategies.

3.2.1. Fundamentals of EHO

Elephants are highly social animals that live in matriarchal groups consisting of females and their offspring. These groups, known as elephant clans, are led by a matriarch. While male leaders often desire to stay outdoors and eventually become completely independent from their families, female members tend to want to live with their families. The subsequent presumptions are taken into account in EHO:

- The population of elephants is limited to a certain number of elephants in each clan.
- A certain number of male elephants from the chief group will depart from their related family and live by themselves in a secluded area at the end of each generation.
- A matriarch is in charge of the elephants in each clan.

3.2.1.1. Operator for Clan-Updating

According to elephant tradition, the matriarch of each clan is in charge of the elephants. Consequently, the matriarch decides each elephant's new location. Equation (5) illustrates how to determine an elephant m 's location within the clan C_n :

$$p_{new,C_n,m} = p_{C_n,m} + s \times (p_{best,C_n} - p_{C_n,m}) \times f \quad (5)$$

In clan C_n , elephant m 's new and old locations are denoted by $p_{new,C_n,m}$ and $p_{C_n,m}$, correspondingly. The clan's matriarch p_{best,C_n} , is the best and fittest elephant in the group. The optimum elephant position is $f[0, 1]$, whereas the new location $s[0, 1]$ is a scale factor defining the matriarch's impact. Equation (6) determines which elephant in the clan is the best and fittest.

$$p_{new,C_n,m} = \omega \times p_{center,C_n} \quad (6)$$

where $\omega [0, 1]$ is a factor that influences C_n on the best fit elephant $p_{new,C_n,m}$ and the elephant's new location at p_{center,C_n} . Equation (7) is used to determine the clan centre individual p_{center,C_n} .

$$p_{center,C_n,z} = \frac{1}{gC_n} \times \sum_{m=1}^{gC_n} p_{C_n,m,z} \quad (7)$$

where $p_{C_n,m,z}$ represents the individual elephant $p_{C_n,m}$ in the z -dimension, and $1 \leq z \leq Z$ and gC_n indicate the number of elephants in clan C_n . Equation (7) is used to update the optimum location of an elephant in clan C_n , which is now $p_{center,C_n,z}$.

3.2.1.2. Separating Operator

The separating procedure that male elephants use to leave their family group may be used as a separation operator to solve optimization problems. The elephant member having the worst performance in every generation applies the separation operator, as shown in Equation (8).

$$p_{worst,C_n} = p_{min} + (p_{max} - p_{min} + 1) \times Rand \quad (8)$$

where each individual elephant's location within the family is indicated by the lower bound (p_{min}) and the upper bound (p_{max}). The worst member of clan C_n is indicated by p_{worst,C_n} . A stochastic distribution having values between 0 and 1 is called $Rand[0, 1]$. The fundamental structure of the EHO algorithm can be outlined by focusing on the processes of clan upgrading and separating operators. Algorithm 1 of EHO is defined as follows:

Algorithm 1. EHO algorithm

```

Begin
Initialise. Select Gen Max for maximum generation and elephant count; choose  $P$  for population initialization; and set  $E = 1$ 
for the number of repetitions.
While searching, do
Arranging each elephant's true fitness during the search is used to categorize the population.
For every clan, create a count.
For elephant  $j$  in the family clan
Determine  $p_{new,C_n,m}$ , then use Equation (5) to upgrade  $p_{C_n,m}$ .
Arrange the population based on each individual's level of fitness.
For each clan's ci do
For each elephant  $m$  in the clan  $C_n$  do
Construct  $p_{new,C_n,m}$ , then use Equation (5) to upgrade  $p_{C_n,m}$ .
If  $p_{C_n,m} = p_{best,C_n}$ , then
Generate  $p_{new,C_n,m}$  and use Equation (6) to upgrade  $p_{C_n,m}$ .
End if
End for
End for
For all current clans'  $C_n$  do
Using Equation (8), swap out the worst individual elephant  $C_n$ .
End for
Assign a new position to each individual elephant in the clan.
Increasing the number of generations by  $K = K + 1$ .
End while
Output: the optimal finest elephant location
End.
```

3.3. Improved EHO with Hyper-Parameter Updating with AdaMax Optimiser

According to studies on elephant behavioural biases, Wang, et al. [21] developed the EHO technique,

a generalized stochastic search algorithm. The EHO technique, which optimizes neural networks, significantly improves deepfake detection performance. The EHO method has shortcomings, including:

- There is a lesser impact on future extension when there is unreasonable convergence to the upgraded operator at the origin.
- Uneven primary allocation of elephant positions.

For the reasons stated above, this work suggests an improved EHO using an AdaMax optimizer. The AdaMax optimizer is used to update the hyperparameter.

3.3.1. AdaMax Optimizer

The AdaMax optimizer is used to adjust the hyperparameters, enabling the training of the suggested CNN model with improved EHO. By utilizing the infinite norm for parameter updates, the AdaMax optimizer offers a more reliable and effective convergence procedure, its primary benefit. This aids in achieving strong optimization results, particularly when dealing with high-dimensional data or sparse gradients. An enhancement of the Adam optimizer, AdaMax ensures robust convergence by adjusting the step size according to the highest absolute gradient. The following formulas are used to update the AdaMax optimizer parameters:

$$h_t = \frac{\partial Q_i(\omega)}{\partial \omega} \quad (9)$$

$$n_t = \beta_1 n_{t-1} + (1 - \beta_1) h_t \quad (10)$$

$$u_t = \max(\beta_2 u_{t-1}, |h_t|) \quad (11)$$

$$\eta_t = \frac{\alpha}{u_t} \quad (12)$$

where h_t is the gradient at step t , n_t is the exponentially moving average of gradients, u_t represents the moving maximum of gradients using the ∞ -norm, η_t is the learning rate at step t , calculated based on u_t . The momentum parameters β_1 and β_2 are typically set to 0.9 and 0.999, respectively. The learning rate η_t is scaled adaptively to ensure smooth convergence across varying optimization conditions. Furthermore, the following is an explanation of the parameter update.

$$w_{t+1} = w_t - \eta_t \cdot n_t \quad (13)$$

AdaMax is used in EHO with updated hyper-parameters to further improve it because of this benefit. Algorithm 2 introduces the AdaMax optimizer to IEHO. In the IEHO approach, the hyper-parameters α and β are taken into account, and their initial values are chosen at random from 0 to 1. While β_1 and β_2 have less of an effect on classification accuracy, learning η_t is the main factor influencing the algorithm's convergence rate. Therefore, by improving the EHO algorithm and changing the hyper-parameters, this is the factor that will increase the classification performance. With a significant number of iterations, the AdaMax coefficients are set as $\eta_t = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. Additionally, the

lowest error rate is noted with these updated hypermeters; for this reason, these values are referred to as ideal hyper-parameters. Figure 2 is an illustration of the IEHO–AdaMax flow chart.

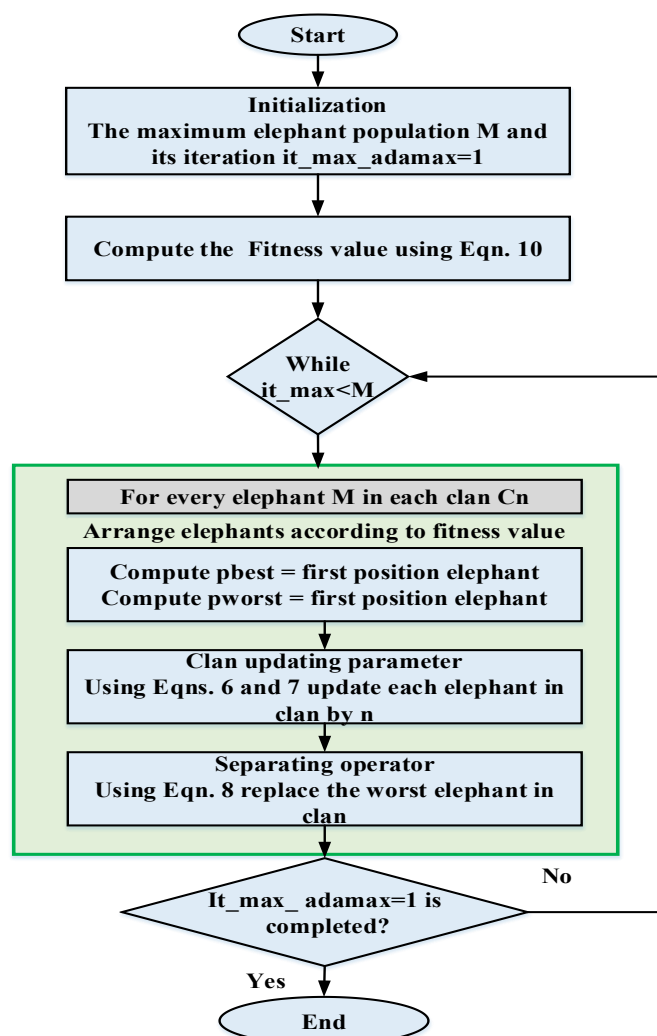


Figure 2. Flow chart of improved EHO with AdaMax optimizer.

3.4. The IEHO–CNN Method

An essential component of deepfake detection from deepfake frames is classifier design. CNN is a classifier that can handle challenging nonlinear problems and has significant self-learning and self-adaptive capabilities due to advancements in ML. CNN has grown in significant popularity in the field of image categorization. This section explains how the deepfake images are classified using a CNN based on the IEHO method with an AdaMax optimizer. A CNN is shown in Figure 3, where every node in the network instantaneously and adaptively chooses the distinguishing feature and simultaneously extracts all of the important feature parameters, guaranteeing that the order in which they are employed does not restrict the accuracy of image processing. This research uses a CNN classifier using a three-layer network structure. The no. of input and output images defines the no. of nodes in the input and output layers. Equation (14) demonstrates how to calculate the number of nodes in the suggested method's hidden layers:

Algorithm 2. IEHO with AdaMax method for updating hyper-parameters

```

# it_max indicates the EHO algorithm's maximum number of iterations
# it_max_adamax indicates the AdaMax method's maximum number of iterations
# C_it_adamax indicates the AdaMax algorithm's current iteration number
# C_it_EHO indicates the EHO algorithm's current iteration number
# Begin Initializations
Set the initial values for the following input feature parameters:  $\eta_t, \beta_1, \beta_2, \alpha, it\_max$ , and  $it\_max\_adamax$ .
Set the elephants' current location
For  $C\_it\_adamax = 1: it\_max\_adamax$ 
# Develop IEHO
Determine each elephant's fitness function within the clan
Compute and update the clan operator to determine the best and worst new elephant positions
For  $C\_it\_EHO = 1: it\_max$ 
The new positions of each elephant are updated using Equation (5).
Determine the elephants' central location using Equation (7).
Utilising Equation (6), update the elephant's position's new optimal value.
To update the elephant's location, use Equation (8) to get the worst value.
For every elephant, repeat and assess the new fitness function.
The best and worst elephant positions are identified.
End for
Upgrade the locations of each elephant
# End of EHO
Determine the learning and loss error  $\omega_{t+1}$  by applying Equation (13)
Utilising Equations (10)–(13), update the hyper-parameters with the new values
End for

```

$$\hat{\rho} = \sqrt{\hat{r} + \hat{s}} + k \quad (14)$$

where $\hat{r}$ represents the amount of input features, $\hat{s}$ represents the output features, and k [1,100] is constant, CNN's hidden layer nodes fall within the integer range $[\sqrt{\hat{r} + \hat{s}} + 1, \sqrt{\hat{r} + \hat{s}} + 100]$.

Furthermore, the CNN's preliminary thresholds and weights vary from -1 to 1, affecting training duration. This impacts CNN's convergence findings and outcomes when it lacks adequate robustness. The CNN's performance will thus be significantly enhanced by selecting the optimal starting weights and thresholds. In this work, the initial threshold values and CNN weights are optimized using the IEHO with the AdaMax optimizer.

The fitness function's goal is to reduce the mean absolute error (MAE) between the CNN output layer and its results. The CNN is trained using the input feature set to anticipate the system output. The ideal answer is described as follows:

$$f(\hat{\vartheta}) = \frac{1}{\hat{n}} \sum_{i=1}^{\hat{n}} |\hat{Y}_i - P_i| \quad (15)$$

where the feature vector $\hat{\vartheta} = [\hat{\vartheta}_1, \hat{\vartheta}_2, \dots, \hat{\vartheta}_D]$ is the result of merging CNN's initial weights and threshold

values. The weights are set as $\hat{\vartheta}_1 = [\hat{\vartheta}_1, \hat{\vartheta}_2, \dots, \hat{\vartheta}_{d_1}]$, and the threshold value is supplied by $\hat{t}_1 = [\hat{\vartheta}_{d_1+1}, \hat{\vartheta}_{d_2+2}, \dots, \hat{\vartheta}_{d_2}]$. The features between the input layer and the hidden layers are determined by initial weights, represented as $\hat{\vartheta}_2 = [\hat{\vartheta}_{d_2+1}, \hat{\vartheta}_{d_2+2}, \dots, \hat{\vartheta}_{d_3}]$, and a threshold value defined as $\hat{t}_2 = [\hat{\vartheta}_{d_3+1}, \hat{\vartheta}_{d_3+2}, \dots, \hat{\vartheta}_D]$. Here, $d_1 = \hat{n} \times \hat{\rho}$, $d_2 = \hat{n} \times \hat{\rho} + \hat{\rho}$, $d_3 = \hat{n} \times \hat{\rho} + \hat{\rho} \times \hat{r}$, and D signifies the total amount of nodes in the CNN, expressed as $D = \hat{n} \times \hat{\rho} + \hat{\rho} + \hat{\rho} \times \hat{r} + \hat{r}$. The expected feature output is denoted as $\hat{Y} = [\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_n]$, while the predicted output is given by $P = [P_1, P_2, \dots, P_n]$. Figure 3 displays the CNN based on IEHO workflow with the AdaMax optimizer for deepfake detection that is described in the suggested methodology.

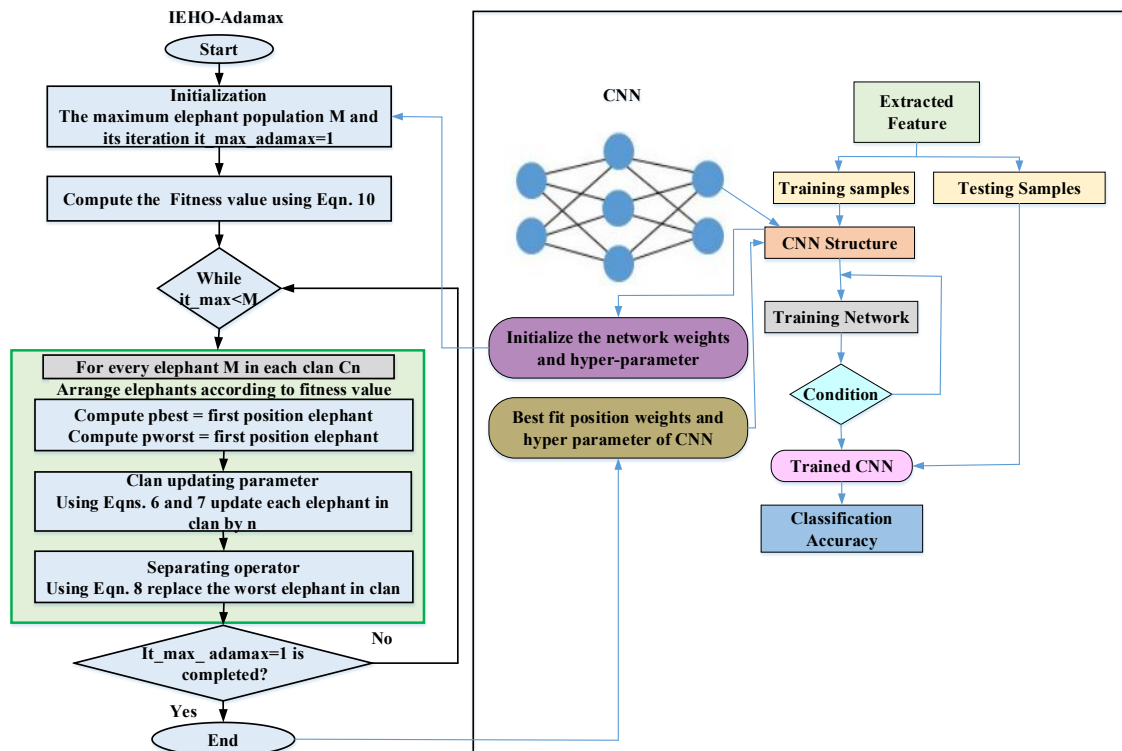


Figure 3. The Deep ConvIEHO-AdaMaxNet flowchart for Deepfake detection.

The following is a procedural description of the suggested process:

Step 1: Establish the parameters. $Z = M \times n$; the maximum number of group iterations is $GenMax$; the total number of current elephant groups is Z ; the number of elephants is M ; and the number of elephants in each clan is C_n . Take into account the maximum number of iterations Gen , the effect factor ε , and the qubit mutation probabilities $q1$ and $q2$. Determine the elephant's initial location in the domain at random.

Step 2: Map the optimal location to the current location using Equation (10), then use Equation (11) to compute every elephant's actual fitness value, $f(\hat{\vartheta})$. Put the elephants in ascending order based on the given estimated fitness value. Together with the elephant's new location value, $\vartheta^t g$ is the global fitness value.

Step 3: Divide each group of elephants into clan C . Then, determine which elephants in the C_n clan are the most and least suited.

Step 4: Utilising Equations (5)–(7), update the clan operator with each elephant's location.

Step 5: Using Equations (8) and (13), assess the separation operator to substitute the individual elephant with its worst-case fitness in C_n .

Step 6: Integrate the elephants from each clan; calculate each elephant's fitness value $f(\hat{\vartheta})$, using Equation (13). Arrange the elephants in ascending order of fitness count to determine the new site with the global ideal fitness value.

Step 7: Follow step 3 again until the final elephant has the location; if not, calculate the global location $\hat{\vartheta} = [\hat{\vartheta}_1, \hat{\vartheta}_2, \dots, \hat{\vartheta}_D]$ and terminate the algorithm.

Step 8: The trained CNN models attain deepfake detection accuracy once the network has been trained using the optimal starting threshold values and weights.

3.5. Federated Learning

The growth of distributed networks and devices generates enormous volumes of diverse data every day for deepfake detection. However, the growing power and capabilities of these technologies present questions regarding data privacy during transmission. To address these challenges, emerging technologies focus on minimizing the need for sensitive data transfer by leveraging local storage and computation capabilities. This shift is particularly advantageous for deepfake detection, where user-generated content can remain on local devices, ensuring greater privacy. FL has become a potential solution for this scenario. It enables the training of models directly on remote devices without transferring raw data to a central server, thereby decreasing the risk of data leakage. By distributing the learning process among multiple devices and aggregating results centrally, FL supports robust model training while maintaining data privacy. This strategy has been effectively applied in several fields, including medical imaging and IoT device settings. In the context of deepfake detection, FL can play a transformative role. It enables the development of advanced models that can identify manipulated content without exposing sensitive user data. As this technology evolves, it has the potential to enhance the privacy and security of systems designed to combat the growing threats posed by deepfakes in modern digital environments. In the context of deepfake detection, FL can be divided into two key types depending on their networking structure:

- *Centralized FL*: This is the commonly adopted FL method. It operates with a central server that coordinates training with a network of clients. Neural network models are trained in parallel by clients utilizing their locally accessible datasets throughout each training cycle. The process involves each client sending their model parameters to a central server for aggregation, often using weighted averaging. The server generates a global model, which is then redistributed to clients who refine it using

their data. Once the training process is complete, each client has both a global model and a personalized version fine-tuned to their local data.

- *Decentralized FL*: In contrast to centralized FL, decentralized FL operates in a peer-to-peer fashion. In this model, clients communicate directly with each other rather than relying on a central server. Each client trains a local model on its dataset in each communication round. Afterwards, clients exchange their model parameters with neighboring clients, updating their local models through peer-to-peer communication until a consensus is reached on a global model. This approach eliminates or reduces the need for a central server and can function even if direct communication with the server is unavailable, making it more resilient in some scenarios.
- *Privacy and Deepfake Detection*: FL enhances privacy by keeping data local on the devices where it is collected, preventing the need for sensitive data to be transmitted. Only the model updates, not the raw data, are shared, which is particularly useful for deepfake detection. This ensures that personal or sensitive content is never transferred, maintaining privacy while enabling robust collaborative model training to identify deepfakes.
- *Parallel FL*: Parallel FL is a variation of centralized FL, where local neural network models are trained on each client's edge device, then averaged and tested on a global model in an aggregator. The global model's weights are returned to clients, who continue training on their local datasets, repeating over multiple rounds until the desired number of training rounds is completed. This method enables efficient model training across distributed devices while minimizing the need for centralized data handling, thereby further enhancing data privacy in tasks such as deepfake detection.

3.5.1. Federated Learning in Proposed Model

Parallel FL was implemented in the proposed approach of Deepfake detection. In this method, we simulated an FL setup to enable distributed task processing, establishing a decentralized network of clients on a single machine, with each client having access to its own dataset. Initially, the key step was to define the number of training rounds, which indicated how many times the complete FL cycle would be executed. Figure 4 illustrates the cycle of parallel FL, consisting of three client nodes connected by edges.

To enhance the robustness of the Deepfake detection system against adversarial attacks, adversarial learning is incorporated into the FL framework depicted in Figure 4. Both the server and clients maintain the same model architecture, with the server initializing the global model and defining the number of learning rounds. At the start of each round, starting at time t , the server distributes the current global model (i.e., weight vector X_t) to all clients. Each client then initializes its local model ($X_{i,t} = X_t$) and generates adversarial examples from its local dataset using techniques such as the Fast Gradient Sign Method (FGSM). Local training is performed for one epoch using a combination of clean data and adversarial examples,

allowing the model to learn both general and adversarial features. After completing the local training, the clients update their models (i.e., weight vectors $X_{i,t+1}$) and send them back to the server. The server aggregates these local models to create an updated global model (X_{t+1}), incorporating the adversarially trained knowledge from the clients. This procedure is repeated for a predefined number of rounds. By combining federated and adversarial learning, the system becomes more resilient to adversarial manipulations, improving its capability to identify even the most subtle Deepfake content.

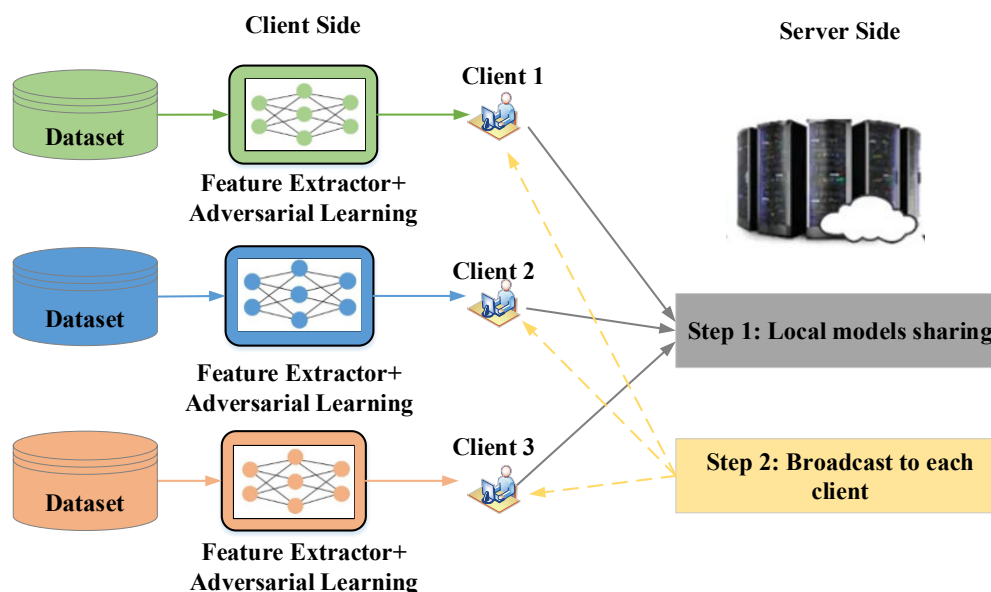


Figure 4. The proposed deepfake detection parallel FL with three edge-clients.

4. Result and Discussion

The results of the proposed DeepConvIEHO-AdaMaxNet framework for accurate deepfake detection are presented in this sector. The suggested technique was implemented in this study using Python 3. The model implementation was carried out using TensorFlow, an open-source, high-level deep-learning framework in Python. The training and validation processes were performed on a 64-bit Windows 10 Pro system with a 12th-generation IC CPU and 16 GB of RAM. The suggested model's performance on the Celeb-DF dataset was evaluated using the performance measures listed below. The Celeb-DF dataset is used to train the proposed deepfake detection algorithm, and the Celeb-DF test set is used to access it. The dataset is randomly partitioned into two sets, with 80% for training and 20% for validation. Several of the most prevalent metrics are used to test and assess the performance of the Celeb-DF dataset.

4.1. Dataset Description

It is essential to test any digital video authentication system against outstanding DeepFake datasets to have a significant influence on the actual world. To preserve its excellent visual quality, the dataset should be incredibly realistic, cover a wider variety of real-world scenarios, and have few visual artefacts. The Celeb-DF (V2) is a highly demanding video dataset comprising 590 authentic celebrity videos and 5,639 superior DeepFake films generated using an enhanced synthesis technique, totalling more than 2

million frames. The false videos are made by switching faces for each pair of subjects, while the real ones are gathered from YouTube videos that are accessible to the general public.

4.2. Evaluation Results

The effectiveness of the suggested deepfake video detection technique is assessed using the AUC, a widely used assessment statistic. It assesses a binary classifier's performance using a single scalar value. The AUC is a reliable metric since it is calculated using the whole ROC curve across all categorizing levels. It quantifies the whole 2D area between (0, 0) and (1, 1) that is beneath the ROC curve. True positives (TP) and false positives (FP) are traded off in the ROC curve. It was constructed by graphing FP and TP rates on the x and y axes, respectively. The model's ability to assess the authenticity of the video is shown by its AUC value. The presentation of the suggested method is further assessed using five additional metrics: Accuracy, Specificity, Recall, Precision, and F1-score, as given below

4.2.1. Accuracy

The percentage of correctly identified samples relative to the total number of samples is known as accuracy. It serves as a broad gauge of the model's effectiveness.

$$\text{Accuracy} = \frac{\text{Number of True Negatives} + \text{Number of True Positives}}{\text{Total Number of Samples}} \quad (16)$$

4.2.2. Specificity

The model's ability to accurately detect negative cases is measured by specificity, commonly referred to as the true negative rate. It focuses on minimizing FPs.

$$\text{Specificity} = \frac{\text{Number of True Negatives}}{\text{Number of True Negatives} + \text{Number of False Positives}} \quad (17)$$

4.2.3. Sensitivity (or Recall)

Sensitivity, also referred to as recall or the true positive rate, evaluates the model's accuracy in identifying positive cases. It emphasizes minimizing false negatives.

$$\text{Sensitivity(Recall)} = \frac{\text{Number of True Positives}}{\text{Number of False Negatives} + \text{Number of True Positives}} \quad (18)$$

4.2.4. Precision

Precision measures the proportion of predicted positive cases that are true positives. It focuses on minimizing false positives.

$$\text{Precision} = \frac{\text{Number of True Positives}}{\text{Number of False Positives} + \text{Number of True Positives}} \quad (19)$$

4.2.5. F1-Score

The F1-Score is the harmonic mean of recall and precision, providing a single metric that balances the trade-off between precision and recall. It is mainly useful when the data is imbalanced.

$$F1 - \text{Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (20)$$

Table 1. Performance of the for DeepConvIEHO-AdaMaxNet.

Metrics	Performance (%)	
	Class 0	Class 1
Precision	100	90
Recall	90	100
F1-score	95	95
Specificity	100	90

Table 1 presents performance metrics for a binary classification task, where Class 0 represents "fake" and Class 1 represents "real." For Class 0 (fake), the model achieves a perfect precision of 100%, meaning all instances predicted as fake are indeed fake. The recall for Class 0 is 90%, representing that the model correctly classifies 90% of all actual fake cases, with a few being misclassified. The F1 score for Class 0 is 95%, reflecting a balanced trade-off between precision and recall. The specificity for Class 0 is 100%, signifying that the model perfectly distinguishes fake instances from real ones when the ground truth is real. For Class 1 (real), the model achieves a precision of 90%, meaning 90% of the predictions for real instances are accurate. The recall is 100%, indicating the model identifies all real instances without missing any. The F1-score for Class 1 is 95%, showing effective performance in recognizing real instances. The specificity for Class 1 is 90%, suggesting that 10% of actual fake instances are misclassified as real. Overall, the model demonstrates strong predictive ability for both classes, with a slight trade-off in specificity for the real class.

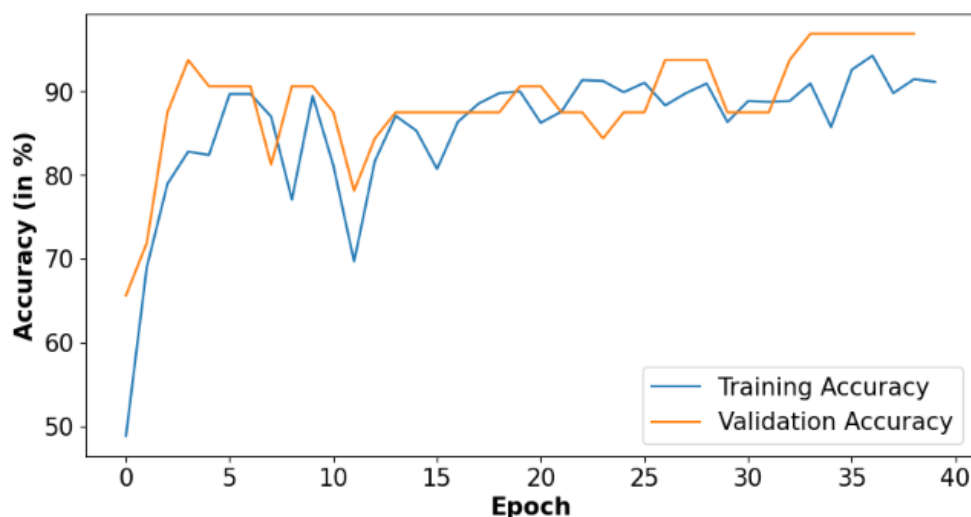
**Figure 5.** Training and validation accuracy for DeepConvIEHO-AdaMaxNet.

Figure 5 depicts a line graph showing the training and validation accuracy of a model over 40 epochs. The blue line signifies the training accuracy, illustrating how well the model fits the training data, while the orange line signifies the validation accuracy, demonstrating how well the model generalizes to unknown data. Initially, both accuracies improve rapidly during the first few epochs. After approximately 5–10 epochs, the accuracies begin to fluctuate but gradually stabilize. Notably, the validation accuracy often matches or exceeds the training accuracy, suggesting effective regularization or potential slight underfitting.

This graph provides useful information on the model's functionality, helping to evaluate its training behaviour and generalisation capabilities.

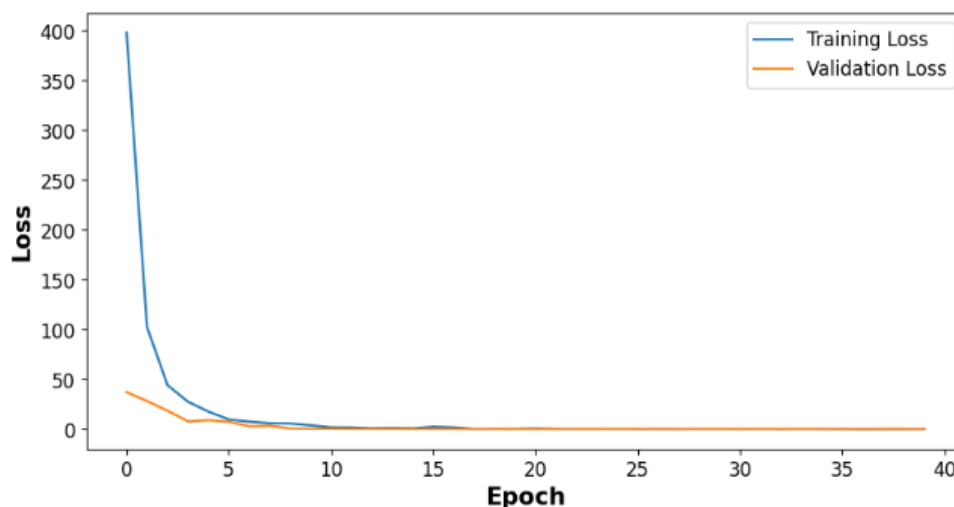


Figure 6. Training and validation loss for DeepConvIEHO-AdaMaxNet.

Figure 6 illustrates a line graph showing the training and validation loss of a model over 40 epochs. The blue line corresponds to the training loss, representing the model's error on the training dataset, and the orange line corresponds to the validation loss, which indicates the model's error on unseen validation data. Initially, both the training and validation losses are high, with the training loss starting significantly above 400 and the validation loss beginning at a lower value. However, both losses decrease sharply within the first few epochs, reflecting rapid improvement in the model's performance during early training. After around 5–10 epochs, the losses stabilize and converge to relatively low values, with minimal fluctuations observed in both curves. The model appears to generalize well without experiencing severe overfitting or underfitting, as indicated by the proximity of the loss curves. This graph provides insights into the model's learning process, demonstrating effective convergence and stable performance on both the training and validation datasets. It is a useful tool for evaluating the model's optimization and generalization capabilities.

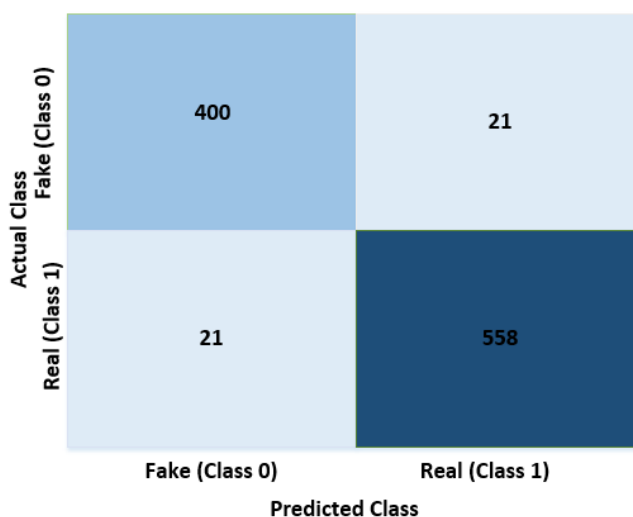


Figure 7. Confusion matrix of the proposed model.

As shown in Figure 7, the confusion matrix demonstrates the performance of the DeepConvIEHO-AdaMaxNet model on a balanced dataset of 1000 samples. The model correctly identified 400 fake videos (true positives) and 558 real videos (true negatives). It misclassified 21 fake videos as real (false negatives) and 21 real videos as fake (false positives). This results in a high precision and recall of approximately 95%, indicating balanced and accurate detection. The matrix reflects the model's strong ability to differentiate between authentic and manipulated content with minimal misclassifications.

4.3. Comparison Result

This segment compares the accuracy and AUC of the suggested system with those of several existing classification methods for deepfake detection using the Celeb-DF (V2) dataset. Table 2 compares the presentation of various methods in terms of accuracy for a specific task. It highlights various approaches, including ProtoExplorer [22], NoiseDF [23], MPC-CA [24], and PUDD [25], as well as a proposed method. Among these, the proposed method demonstrates superior performance, surpassing the others in accuracy. Table 3 summarizes the AUC performance of various methods for a specific task. It includes techniques such as Capsule [26], Xception [26], Add-Net [26], Multi-Att [26], PEL [26], and FedPR [26], with the proposed method also listed. Among these, the proposed method achieves the highest AUC, significantly outperforming the other approaches and demonstrating its effectiveness in the given context.

Table 2: Comparison of accuracy with existing methods.

Methods	Accuracy (%)
ProtoExplorer	92.5
NoiseDF	70.1
MPC-CA	79.5
PUDD	95.1
Proposed	97.50

Table 3: Comparison of AUC with existing methods.

Methods	AUC (%)
Capsule	70.18
Xception	77.91
Add-Net	62.12
Multi-Att	76.95
PEL	82.94
FedPR	83.95
Proposed	96.41

Figure 8 is a line plot comparing the accuracy (%) of different methods for a specific task. The x-axis represents the methods evaluated, labelled as "NoiseDF," "MPC-CA," "ProtoExplorer," "PUDD," and "Proposed". The y-axis displays the accuracy as a percentage, with values spanning from 70% to 100%.

The plot shows an increasing trend, where the accuracy improves progressively from "NoiseDF" to the "Proposed" system. "NoiseDF" has the lowest accuracy, slightly above 70%, while the "Proposed" system achieves the highest accuracy, exceeding 95%. The blue line connecting the data points highlights the improvement in performance across methods, demonstrating the higher presentation of the "Proposed" system compared to others.

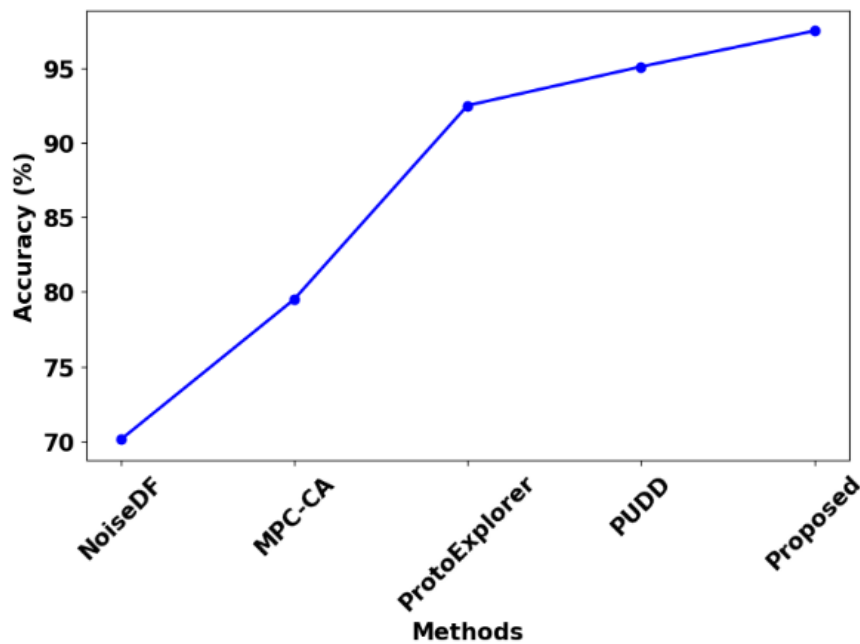


Figure 8. Comparison of accuracy with existing methods using Celeb-DF (V2) dataset.

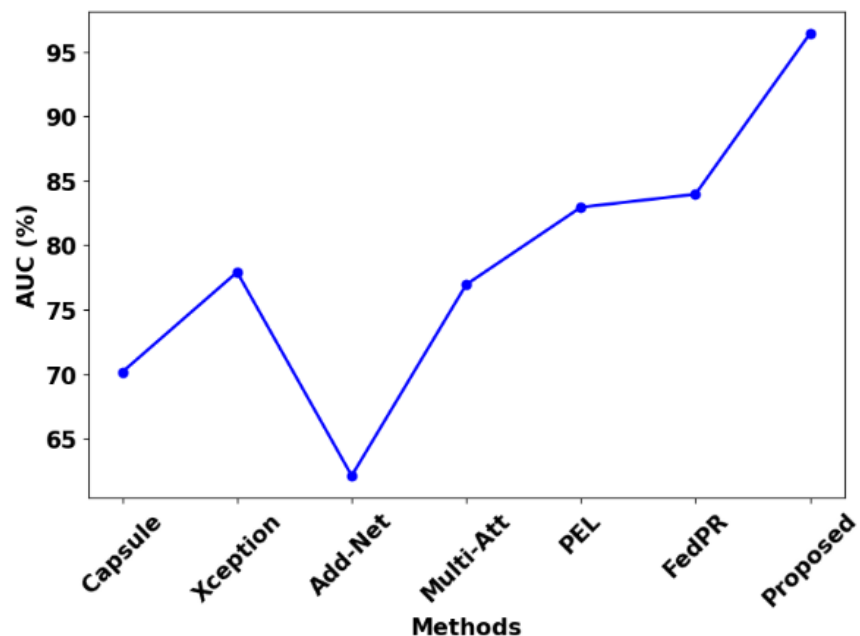


Figure 9. Comparison of AUC with existing methods using Celeb-DF (V2) dataset.

Figure 9 illustrates the accuracy performance of various methods for a specific task, with accuracy shown in percentage and the methods compared. The methods include NoiseDF, MPC-CA, ProtoExplorer, PUDD, and the Proposed method. The graph demonstrates a clear upward trend, indicating progressive

improvement in accuracy across the methods. NoiseDF achieves the lowest accuracy, approximately 70%, followed by MPC-CA, which achieves about 80%. ProtoExplorer shows a significant improvement, reaching 90%, while PUDD slightly surpasses it. The Proposed method achieves the highest accuracy, exceeding 95%, highlighting its superior performance compared to the other approaches.

Table 4. Different performance comparison of deepfake detection techniques on fake and real images.

Techniques	Fake images			Real images		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Mesonet [27]	67.50%	74.60%	70.91%	71.65%	64.20%	67.72%
Mesonet with Optimized Preprocessing Layer [27]	91.26%	94.00%	92.61%	93.81%	91.00%	92.39%
GenConViT	89.26%	93.20%	91.62%	93.82%	94.50%	94.60%
YoloV8-Resnet50 [27]	83.90%	95.00%	89.20%	95.20%	80.20%	87.02%
Proposed	100 %	90 %	95 %	90%	100 %	95 %

Table 4 presents a comparative analysis of deepfake detection models based on precision, recall, and F1-score for both fake and real image classification. The proposed model achieves the best F1-scores for both classes, with perfect precision for fake images and perfect recall for real ones. Compared to GenConViT and YoloV8-Resnet50, the proposed method exhibits a more balanced and superior performance. This demonstrates its robustness in accurately identifying both real and manipulated media content.

5. Conclusion

In conclusion, the proposed DeepConvIEHO-AdaMaxNet framework successfully tackles the multifaceted challenges of deepfake detection by combining innovative techniques for preprocessing, optimization, and robust learning. The framework employs a GABF to enhance preprocessing by effectively reducing noise and preserving crucial edge details, thereby improving input quality for the detection model. It further integrates a CNN-based classification mechanism optimized through the IEHO algorithm coupled with the AdaMax optimizer. This bio-inspired optimization strategy overcomes the constraints of conventional techniques, including slow convergence and vulnerability to local minima, ensuring a more efficient and effective hyperparameter tuning process. Additionally, the inclusion of adversarial training within a FL paradigm improves the model's robustness towards adversarial assaults and manipulations, making it well-suited for real-world applications. The model's performance, evaluated on the Celeb-DF dataset, is noteworthy, achieving a 97.50% accuracy and a 96.41% AUC. It further demonstrates exceptional class-specific metrics, including 100% specificity, precision, and F1-measure for class 0, as well as 93.53% specificity, 100% recall, and 95% F1-measure for class 1. These results highlight its ability to detect deepfake manipulations with remarkable precision and reliability. Compared to

conventional CNNs, the proposed framework showcases superior performance, proving its efficacy in addressing critical concerns in digital integrity and trustworthiness. This study underscores the importance of combining adaptive preprocessing, bio-inspired optimization, and FL to advance cutting-edge deepfake detection systems, contributing to the development of secure and ethical AI applications. Future work could explore integrating transformer-based architectures with the proposed DeepConvIEHO-AdaMaxNet framework to enhance its capability to capture long-term dependences and contextual features in deepfake detection. Additionally, expanding the framework to handle multimodal deepfake content, such as audio-visual and text-based manipulations, would further bolster its applicability in combating increasingly sophisticated deepfake technologies.

Acknowledgements: The authors would like to thank the Deanship of BVRIT Hyderabad College of Engineering for Women for supporting this work.

Data Availability Statement: The Celeb-DF (v2) dataset used in this study is publicly available. No new data were generated.

Funding: The authors declare that no funding was received for this research.

Declaration of Competing Interest: The authors declare they have no known competing interests.

Research Involving Human/Animal Participants: This study involved human participants; informed consent was obtained from all participants. No animals were involved in this research.

References

- [1] I. H. Sarker, "Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview," *Security and Privacy*, vol. 6, no. 5, p. e295, 2023, doi: <https://doi.org/10.1002/spy2.295>.
- [2] M. A. Manjramkar and K. C. Jondhale, "Cyber Security Using Machine Learning Techniques," in *Proceedings of the International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022)*, 2023: Atlantis Press, pp. 680-701, doi: https://doi.org/10.2991/978-94-6463-136-4_59.
- [3] Y. Lu and L. D. Xu, "Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103-2115, 2019, doi: <https://doi.org/10.1109/JIOT.2018.2869847>.
- [4] M. Kuzlu, C. Fair, and O. Guler, "Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity," *Discover Internet of Things*, vol. 1, no. 1, p. 7, 2021, doi: <https://doi.org/10.1007/s43926-020-00001-4>.
- [5] F. Tao, M. S. Akhtar, and Z. Jiayuan, "The future of artificial intelligence in cybersecurity: A comprehensive survey," *EAI Endorsed Transactions on Creative Technologies*, vol. 8, no. 28, pp. 1 - 15, 2021, doi: <https://doi.org/10.4108/eai.7-7-2021.170285>.
- [6] S. Srivastava and B. Benny, "Artificial intelligence (ai) and it's application in cyber security," *EasyChair Preprint*, 2021, doi: <https://easychair.org/publications/preprint/rCvr>.
- [7] K. R. Bhatele, H. Shrivastava, and N. Kumari, "The Role of Artificial Intelligence in Cyber Security," in *Research Anthology on Artificial Intelligence Applications in Security*, I. R. Management Association Ed. Hershey, PA, USA: IGI Global Scientific Publishing, 2021, pp. 1806-1823. <https://doi.org/10.4018/978-1-7998-7705-9.ch079>

- [8] M. Rizvi, "Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention," *International Journal of Advanced Engineering Research and Science*, vol. 10, no. 5, pp. 55 - 60, 2023, doi: <https://dx.doi.org/10.22161/ijaers.105.8>.
- [9] M. Solaiman and M. S. Rana, "Enhancing global security: A robust CNN model for deepfake video detection," in *2024 7th International Conference on Information and Computer Technologies (ICICT)*, 2024: IEEE, pp. 238-243, doi: <https://doi.org/10.1109/iciict62343.2024.00044>.
- [10] J. T. Hancock and J. N. Bailenson, "The Social Impact of Deepfakes," *Cyberpsychology, Behavior, and Social Networking*, vol. 24, no. 3, pp. 149-152, 2021, doi: <https://doi.org/10.1089/cyber.2021.29208.jth>.
- [11] N. Giansiracusa, *How algorithms create and prevent fake news*. Apress Berkeley, CA: Springer, 2021, doi: <https://doi.org/10.1007/978-1-4842-7155-1>
- [12] S. Karnouskos, "Artificial Intelligence in Digital Media: The Era of Deepfakes," *IEEE Transactions on Technology and Society*, vol. 1, no. 3, pp. 138-147, 2020, doi: <https://doi.org/10.1109/TTS.2020.3001312>.
- [13] D. Fallis, "The Epistemic Threat of Deepfakes," *Philosophy & Technology*, vol. 34, no. 4, pp. 623-643, 2021, doi: <https://doi.org/10.1007/s13347-020-00419-2>.
- [14] M. Ridouani, S. Benazzouza, F. Salahdine, and A. Hayar, "A novel secure cooperative cognitive radio network based on Chebyshev map," *Digital Signal Processing*, vol. 126, p. 103482, 2022, doi: <https://doi.org/10.1016/j.dsp.2022.103482>.
- [15] G. S. Nariman and H. K. Hamarashid, "Enhancing Communication Efficiency in Federated Learning Through Client Contribution Selection Based Normalized L2 Norm with Cosine Similarity and L2 Norm Filtering," *Journal of Studies in Science and Engineering*, vol. 5, no. 1, pp. 1-23, 2025, doi: <https://doi.org/10.53898/josse2025511>.
- [16] N. Bansal *et al.*, "Real-Time Advanced Computational Intelligence for Deep Fake Video Detection," *Applied Sciences*, vol. 13, no. 5, p. 3095, 2023, doi: <https://doi.org/10.3390/app13053095>.
- [17] A. Khormali and J.-S. Yuan, "ADD: Attention-Based DeepFake Detection Approach," *Big Data and Cognitive Computing*, vol. 5, no. 4, p. 49, 2021, doi: <https://doi.org/10.3390/bdcc5040049>.
- [18] A. Ismail, M. Elpeltagy, M. S. Zaki, and K. Eldahshan, "A New Deep Learning-Based Methodology for Video Deepfake Detection Using XGBoost," *Sensors*, vol. 21, no. 16, p. 5413, 2021, doi: <https://doi.org/10.3390/s21165413>.
- [19] B. Xu, J. Liu, J. Liang, W. Lu, and Y. Zhang, "DeepFake Videos Detection Based on Texture Features," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1375-1388, 2021, doi: <https://doi.org/10.32604/cmc.2021.016760>.
- [20] V. Kandasamy, Š. Hubálovský, and P. Trojovský, "Deep fake detection using a sparse auto encoder with a graph capsule dual graph CNN," *PeerJ Computer Science*, vol. 8, p. e953, 2022, doi: <https://doi.org/10.7717/peerj-cs.953>.
- [21] G.-G. Wang, S. Deb, X.-Z. Gao, and L. D. S. Coelho, "A new metaheuristic optimisation algorithm motivated by elephant herding behaviour," *International Journal of Bio-Inspired Computation*, vol. 8, no. 6, pp. 394-409, 2016, doi: <https://doi.org/10.1504/ijbic.2016.081335>.
- [22] M. d. L. d. Bouter, J. L. Pardo, Z. Geradts, and M. Worring, "ProtoExplorer: Interpretable forensic analysis of deepfake videos using prototype exploration and refinement," *Information Visualization*, vol. 23, no. 3, pp. 239-257, 2024, doi: <https://doi.org/10.1177/14738716241238476>.
- [23] T. Wang and K. P. Chow, "Noise based deepfake detection via multi-head relative-interaction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, vol. 37, no. 12, pp. 14548-14556, doi: <https://doi.org/10.1609/aaai.v37i12.26701>.
- [24] F. Zhang, S. Tian, L. Yu, and Q. Yang, "Multi-channels Prototype Contrastive Learning with Condition Adversarial Attacks for Few-shot Event Detection," *Neural Processing Letters*, vol. 56, no. 1, p. 31, 2024, doi: <https://doi.org/10.1007/s11063-024-11515-1>.
- [25] A. Lopez Pellicer, Y. Li, and P. Angelov, "PUDD: Towards Robust Multi-modal Prototype-based Deepfake Detection," *arXiv e-prints*, p. arXiv: 2406.15921, 2024, doi: <https://doi.org/10.48550/arXiv.2406.15921>.

- [26] D. Liu, Z. Dang, C. Peng, N. Wang, R. Hu, and X. Gao, "Federated face forgery detection learning with personalized representation," *arXiv preprint*, p. arXiv:2406.11145, 2024, doi: <https://doi.org/10.48550/arXiv.2406.11145>.
- [27] S. Sharma and A. Selwal, "Improved Deepfake Detection with Optimized Preprocessing for Low-Quality Images," *Procedia Computer Science*, vol. 258, pp. 507-516, 2025, doi: <https://doi.org/10.1016/j.procs.2025.04.286>.