Journal of Artificial Intelligence and Technology, (Ahead of Print)
https://doi.org/10.37965/jait.2026.1310

ISTP
RESEARCH ARTICLE

Enhancing deepfake detection with dual-strategy Puma optimization and stacked long short-term memory models

Padmavati E. Gundgurti[1,2] **and Parameswaran Thangaraj**[1]

[1]Department of Computer Science and Engineering, School of Engineering and Technology (SOET), CMR University-Lakeside Campus, Bengaluru, India

[2]BVRIT HYDERABAD College of Engineering for Women, Nizampet, Hyderabad, India

(Received 04 April 2026; Revised 08 May 2026; Accepted 16 June 2026; Published online 13 July 2026)

Abstract: A deepfake is a synthetic media model used to modify, manipulate, and generate videos, images, and audio recordings. However, the existing fake detection algorithm struggles to differentiate real from fake content due to redundant information in the extracted features, which degrades its performance. To overcome this limitation, a dual-strategy-based Puma optimization (DSPO) algorithm feature selection model is proposed to identify significant features that effectively distinguish normal from fake images. These features are then processed by a stacked long short-term memory (SLSTM)-based detection model to differentiate between real and fake videos. Initially, videos are acquired and preprocessed into image frames, and relevant features are extracted using transfer learning-based approaches. Next, the proposed DSPO algorithm employs a dual strategy, involving a heavy-tailed distribution and adaptive parameter tuning, to address the algorithm's limitations and select an optimal feature set. Finally, the SLSTM model uses a sech activation function to identify fake videos and improve deepfake detection accuracy. Experimental results demonstrate that the DSPO-SLSTM model achieved accuracies of 99.54%, 99.75%, and 99.54% on the Forensic Face (FF++), Celeb-DF, and Celeb-DF-V2 datasets, respectively, which is greater than existing detection models such as discrete wavelet transform-vision transformer (DWT-ViT).

Keywords: deepfake detection; EfficientNet-B0; Inceptionnet-V3; Puma optimization algorithm; stacked long short-term memory

I. INTRODUCTION

Deepfake is a facial video forgery technique that continually evolves, using artificial intelligence to generate fake videos. Due to the deepfake model, it produces videos of people doing things that do not happen, making it hard to distinguish the real from the fake with the naked eye [1]. As a result, these deepfake models can be exploited for malicious purposes, including spreading false information and violating others' privacy, leading to serious security and privacy issues [2]. Malicious actors create deepfakes based on false profiles to spread disinformation on social media. Therefore, it is vital to develop effective detection methods to prevent the misuse of deepfake techniques [3]. Deep learning (DL) approaches, namely convolutional neural networks (CNNs) and generative adversarial networks (GANs), are commonly used for both creating and detecting fake videos. Most existing deepfake detection models rely on temporal and spatial inconsistencies, such as unnatural blink patterns and misaligned facial movements for accurate detection [4].

Existing detection methods become less effective and face a major challenge: they lack generalization capability due to their evolving nature [5]. Most existing studies focus on detecting deepfake images or videos using DL techniques, which are categorized into CNN-based models and region CNN (RCNN)-based approaches. A CNN-based technique uses facial images from video frames to train and predict at the image level. In deepfake videos, these algorithms utilize only the spatial information from a single frame [6]. However, previous deepfake detection models only use spatial data, making it difficult to distinguish between real and fake face images leading to lower detection performance [7]. To address this issue, transfer learning models, namely the long short-term memory (LSTM) model and its variants, are most appropriate to address the temporal aspects of video data [8]. The key challenge in deepfake detection is that existing DL models struggle to differentiate between authentic and manipulated facial images while processing video frames. Despite advancements, these models often fail to detect subtle artifacts in high-quality deepfakes, leading to misclassification. Variations such as lighting, compression, and facial expressions further complicate detection and decrease accuracy. Additionally, models trained on specific datasets may not generalize well to new manipulation techniques, which limits their effectiveness in real-world applications.

Several methods are developed for deepfake detection using different types of datasets that involve conventional machine learning models and DL architectures. The advantages and limitations of the existing deepfake detection models are discussed as follows: Xiong *et al.* [8] introduced a bidirectional LSTM (Bi-LSTM) and multi-head self-attention network (BMNet) method for deepfake detection. Deep features were extracted from 68 facial landmarks effectively, and MHSA was enabled to capture detailed and localized features from different regions of the face. BiLSTM was used to model temporal dependencies, which effectively

Corresponding author: Padmavati E. Gundgurti (e-mail: padmavati.21cphd@cmr.edu.in).

classified whether a video was real or forged. However, it struggled with traditional models for classification because of its limited sensitivity to temporal features. El-Gayar *et al.* [9] developed a detection model for deepfake videos based on graph neural network (GNN). The batch normalization technique was used to normalize the input by adjusting and speeding up training. Mini-batch GNN extracted structural features effectively, and a block CNN stream was used to extract visual features from the images. Fusion networks were enabled to combine the outputs from mini-GNN and CNN streams, and then dense layers were used for classification. However, some deepfake types, such as swapped faces, were presented less in the dataset, which introduced a class imbalance and made it difficult for the model to learn and detect those classes effectively.

Prathibha *et al.* [10] introduced a DL-based framework for deepfake video detection, which was based on a self-attention residual block. Deepfake detection involved a normalization technique that was used to adjust pixel values to prepare data for the model; self-attention residual module helped to extract detailed facial features by focusing on essential parts. Xception-Net was well suited for image-based tasks such as video frame classification, which adapted new synthetic data and improved generalization. However, storing and using videos or images of people caused privacy issues, which required a privacy preservation mechanism to protect sensitive data. Ramadhani *et al.* [11] introduced a video vision transformer (ViViT) method for deepfake detection. A Depth-wise Separable Convolution (DSC) block was integrated with Convolution Block Attention Module (CBAM) to capture detailed and localized features from different regions of the face for accurate deepfake detection. The introduced ViViT was used to model temporal dependencies, which essentially classified whether a video is real or forged. However, it struggled with traditional models for classification because of its limited sensitivity to temporal features. Boongasame *et al.* [12] developed a detection method named LSTM model with Visual Graph Geometry-16 (LSTM-VGG-16) for deepfake videos. The batch normalization technique was used to normalize the input by adjusting and speeding up training. The VGG-16 extracted structural features effectively, and the CNN stream was used to extract visual features from the images. After the feature was extracted from the VGG-16 model, the LSTM model was used for classification. However, class imbalance occurred due to some deepfake types, such as swapped faces, which were represented less in the dataset. This made the model learn more effectively and detect those less-represented classes accurately.

Rathore *et al.* [13] presented a multi-layer-based GAN model for deepfake video detection. To enhance deepfake detection, the presented model utilized a three-layer GAN to accurately categorize the input video into fake and real. An advantage of this introduced model was that multiple layers in the classification framework of GAN identified the fake images and enhanced generalizability. However, the model struggled to distinguish between real and fake images that led to manipulation of sensitive data. Agarwal and Haneef [14] presented a Convolutional GAN (Con-GAN) method for deepfake detection through eye-blinking feature processing. Initially, frames were preprocessed using the Sequential Adaptive Bilateral Wiener (SABiW) filtering for noise removal, while face detection was performed via a two-dimensional (2D) Haar discrete wavelet transform (DWT). A transfer learning approach was used to extract the features, namely a depth-wise separable residual network. However, the presented Con-GAN model struggled with conventional models for classification because of its limited sensitivity to temporal features. Gupta *et al.* [15] developed an improved transformer model along with a dual-order frequency attention module based on FreqFaceNet for deepfake detection. The developed FreqFaceNet model extracted the frequency-based features to ensure minimal interference of noise, which occurred due to image compression or low resolution. An advantage of the FreqFaceNet model was that it utilized two attention mechanisms, such as wavelet attention and Fourier attention, to improve deepfake detection. However, it struggled to differentiate between real and fake images in deepfake types, namely swapped faces.

Uddin *et al.* [16] suggested a ViT model framework for deepfake video detection. The suggested model involved a face tampering detection framework that integrated a multi-level DWT and a ViT to identify fake images. The ViT extracted multi-level frequency features using DWT and captured global inconsistencies across local face areas. However, storing and using videos or images of people caused privacy concerns, which required a privacy preservation mechanism to protect sensitive data. Ziyuan Cheng *et al.* [17] developed a detection model based on the Multi-Scale Interactive Dual Stream network (MSIDSnet) to detect deepfake in images. In the network, spatial and frequency streams were separated, and a multi-scale fusion module captured both global and fine-grained facial features. Subsequently, MSIDSnet was used to detect and manipulate images in the spatial domain and acquired fine-grained information about high-frequency noise. However, the developed MSIDSnet model struggled to differentiate between real and fake images because of certain deepfake types, such as swapped faces.

Existing detection methods often struggle with redundant data and limited sensitivity to long-term temporal dependencies. Recent DL-based detection models often show reduced effectiveness when identifying complex deepfake types due to gradient saturation. The GNN and CNN architectures failed to effectively integrate temporal dependencies, which limits their capacity to identify subtle motion inconsistencies, such as irregular blinking. This directly impacts accurate detection of fake content and impacts overall model accuracy. To address these limitations, a dual strategy-based Puma optimization algorithm (POA) with a stacked long short-term memory (DSPO-SLSTM) model by utilizing a dual-strategy optimization algorithm is proposed to filter redundant data and isolate highly discriminative features. The integration of a sech-activated stacked LSTM further mitigates gradient saturation, enabling the system to capture subtle temporal inconsistencies across video sequences. This combination ensures precise identification of complex manipulations while maintaining high detection accuracy across diverse datasets.

The key contributions of this research are as follows: Initially, the video data is preprocessed through frame-by-frame extraction and image sharpening, followed by face localization using a cascaded object detection model to enhance the deepfake detection model. The most relevant features are extracted from the transfer learning-based approaches, namely InceptionNet-V3 and EfficientNet-B0. The proposed DSPO algorithm utilizes a new mechanism for changing phase between exploration and exploitation with a dual strategy, namely heavy-tailed (HT) distribution walks and an adaptive parameter tuning to address the algorithm's limitations for selecting the best set of features efficiently. For detection, an SLSTM with a sech activation function is used for capturing and learning the complex information that identifies the fake images precisely.

This research paper is structured as follows: Section II explains the methodology used for detecting real and fake images.

Section III represents simulation experiments and discussion, and Section IV describes the conclusion of this research.

II. METHODOLOGY

The proposed deepfake detection framework consists of four main stages: data acquisition from benchmark datasets, preprocessing, feature extraction, the proposed feature selection method, and finally, the deepfake detection process. Figure 1 depicts a conceptual diagram of the proposed deepfake detection model. First, raw images are collected from three different benchmark datasets and preprocessed to transform the raw data into an enhanced version for further analysis. Next, two transfer learning-based models are utilized for extracting relevant features, which are then integrated to obtain the most important feature set. An optimization algorithm is then proposed to efficiently select the most suitable features based on their natural behavior. Finally, using the selected features, an improved DL model is developed to differentiate between real and fake images for accurate detection.

A. DATASET ACQUISITION

To improve deepfake detection, three distinct datasets are used in this research: Forensic Face ++ (FF++), Celeb-DF, and Celeb-DF-V2.

FF++: This dataset [18] contains a collection of forgery videos, categorized into four distinct types: DeepFakes (DF), FaceSwap (FS), Face2Face (F2F), and NeuralTextures (NT). Each category includes 1,000 videos selected from YouTube. The collection features a total of 1,000 original videos alongside 4,000 manipulated videos. The frame sizes vary, with raw videos in 1080p resolution, medium compressed videos in 720p, and highly compressed videos in 480p. Figure 2 illustrates sample real and fake images of FF++ dataset.

Celeb-DF: The Celeb-DF dataset [19] includes both genuine and deepfake-generated videos, all with visual quality comparable

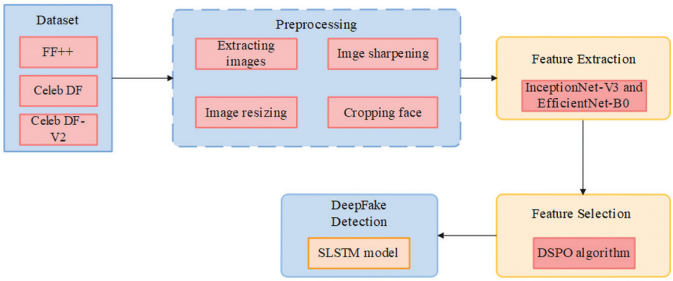

Fig. 1. Proposed deepfake detection framework using the DSPO-SLSTM model based on three different datasets.

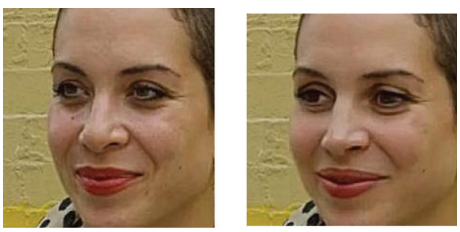

Fig. 2. Real and fake sample images of FF++ dataset.

Fig. 3. Real and fake sample images of Celeb-DF dataset.

to those commonly found on the internet. This dataset contains video of 52 celebrities, considering factors such as age, ethnic group, and gender. It includes 5,639 deepfake videos and 590 original videos from YouTube. Figure 3 represents real and fake sample images from the Celeb-DF dataset.

Celeb-DF-V2: This dataset [19] consists of both real and deepfake videos, totaling 6,229 videos. Similar to the Celeb-DF dataset, the data here is categorized into 5,639 manipulated videos and 590 real videos of celebrities, all downloaded from YouTube. These video frames are then fed into the preprocessing phase for enhanced deepfake detection.

B. PREPROCESSING

After data acquisition, videos are transformed into image frames [20] to facilitate efficient identification of fake images. The images are then sharpened and resized to 224×224 using a resizing technique, and faces are cropped from the frames with the vision cascade object detection method. This helps localize facial features effectively, thereby improving the accuracy of deepfake detection.

C. FEATURE EXTRACTION

Following preprocessing, images are passed to the feature extraction phase to identify and select meaningful patterns and characteristics. The features extracted assist in distinguishing between real and manipulated (deepfake) images efficiently. In this research, transfer learning-based models such as InceptionNet-V3 and EfficientNet-B0 are used to extract deepfake features for detecting real and fake images. These models generate deep features from preprocessed images, aiding the detection model in differentiating between normal and deepfake images. The architecture and processes of both InceptionNet-V3 and EfficientNet-B0 models are described below: InceptionNet-V3 [21] is a transfer learning approach that combines convolutional layers, max pooling layers, and inception modules, which involve multiple parallel convolutional layers with varying filter sizes. Its architecture enables effective feature extraction at multiple spatial scales, enhancing the model's ability to accurately detect fake images. The output is then passed through subsequent CNN layers, such as a Global Average Pooling (GAP) layer to reduce spatial dimensions, followed by a dropout layer and a batch_normalization layer.

The EfficientNet-B0 model [22] offers a balance among all dimensions, including width, depth, and resolution, using a set of fixed scaling coefficients that satisfy specific constraints. The EfficientNet model has different versions, with the simplest being EfficientNet-B0. Other EfficientNet configurations are derived from EfficientNet-B0 with different scaling values. The EfficientNet-B0 architecture includes 18 convolution layers, each utilizing either $k = (3 \times 3)$ or $k = (5 \times 5)$ kernels to extract deep features efficiently. After extracting features from the pretrained models, the feature concatenation technique is employed to combine multiple

features into a single long vector by stacking them sequentially. Features are merged along a shared dimension, allowing the model to learn from all features collectively without losing their individual properties.

This feature extraction architecture employs a two-stage training strategy, utilizing frozen convolutional backbones from InceptionNet-V3 and EfficientNet-B0 to preserve general spatial features. Simultaneously, custom GAP and dense layers are fine-tuned on deepfake datasets to capture specific forgery patterns. Feature fusion occurs through the concatenation of vector outputs, as shown in Equation (1):

$$z_{fused} = [V_{inc} \oplus V_{eff}] \tag{1}$$

where z_{fused} denotes the high-dimensional fused feature vector that serves as input for subsequent feature selection and temporal modeling phases; V_{inc} indicates the feature vector extracted from InceptionNet-V3 with a dimensionality of 2,048, while V_{eff} represents the feature vector extracted from EfficientNet-B0 with a dimensionality of 1,280, and $\oplus$ indicates the concatenation operator, which stacks the two individual vectors sequentially to form a single integrated vector with a combined dimensionality of 3,328. To facilitate temporal modeling, these features are extracted across a sequence of $S_T = 30$ frames, forming an input tensor of shape (16, 30, 3,328) for the SLSTM. This high-dimensional representation provides the DSPO algorithm with a diverse pool of multi-scale features, ensuring a robust selection process during optimization. These features are then fed into the proposed feature selection model.

D. PROPOSED OPTIMIZATION-BASED FEATURE SELECTION

After feature extraction, the combined feature vector denoted as z_{fused} features is forwarded to the proposed feature selection approach using an optimization algorithm to identify the most relevant features. In deepfake detection, the feature selection process is crucial because it helps the detection model focus on the most informative and discriminative features to accurately distinguish between real and fake images. However, standard feature selection techniques, such as filter-based and wrapper methods, often rely on greedy or local search strategies that yield suboptimal feature sets. Therefore, in recent years, optimization algorithms, namely genetic algorithm (GA), particle swarm optimization (PSO), and other swarm intelligence or metaheuristic algorithms, are widely used for feature selection due to their efficient search capabilities across the feature space. In this research, a POA [23], a recently developed metaheuristic algorithm, is used for selecting the most important features from the extracted set.

Most of the metaheuristic optimizers are inspired by natural behaviors to perform optimization by generating random solutions. Parameters are changed for exploration factors within each optimization algorithm. However, the proposed POA employs a new mechanism for switching between exploration and exploitation phases. Additionally, both phases utilize two distinct mechanisms to carry out optimization processes. In this algorithm, the most relevant feature, known as the best solution, is represented as the male puma within the entire search space, which is referred to as the puma's territory. Other solutions, considered irrelevant features, are represented as female pumas. In this context, the best solution, the full search space, and other solutions are denoted as male puma, puma's territory, and female puma, respectively. In POA, the phase change mechanism ensures that all solutions pass into either the exploitation or exploration phases during each iteration. The experience phase and the unexperienced phase are the main stages of this change mechanism, helping to balance exploration and exploitation in the POA algorithm [23]. Furthermore, the primary phases, such as initialization, exploration, and exploitation, are described below.

1). INITIALIZATION. The population of pumas in POA is initialized in the search space to find optimal parameters among the total population. This process is mathematically expressed in Equation (2):

$$X_{i,j} = lb_D + r \cdot (ub - lb) \ where \ D = 1,2,\ldots,m \tag{2}$$

where $X_{i,j}$ represents candidate solutions and their positions, lb and Ub indicate lower and upper bounds, respectively, r stands for a random vector, D indicates the dimension index, and m represents the total dimensionality. Each puma represents a candidate solution, which is used to select the best set of features.

2). EXPLORATION PHASE. During the exploration stage, the best feature sets are selected by utilizing the puma's foraging behavior to search for food. In this phase, pumas search randomly for food within their territory (search space) or move randomly toward other pumas to utilize their prey. First, the puma population is initialized and organized in ascending order. Then, the positions of the puma are updated during the exploration phase, as mathematically formulated in Equation (3):

$$F_{1\,explore} = PF_1 \left(\frac{S^1_{v\,explore}}{S_T} \right) \tag{3}$$

where F_1 represents the exploration force used to determine the search intensity of pumas during the exploration phase, S_V denotes exploration speed or velocity, which indicates the step size of the first search vector, and S_T is the total speed constant, respectively.

3). EXPLOITATION PHASE. In the exploitation stage, the POA employs two distinct operators to enhance solutions within the search space. To obtain an optimal set of features, two mechanisms are used in this phase, based on puma behaviors such as hunting with ambush and sprinting. Pumas typically hide in bushes, trees, or on rocks to surprise their prey, and this ambush strategy is reflected in the exploitation phase to select the best features. The mathematical expression of the exploitation phase is given in Equation (4), and the optimal fitness function is evaluated in the exploration and exploitation phases using Equations (5) to (8):

$$F_{2\,exploit} = PF_2 \left(\frac{S^1_{v\,exploit}}{S_T} \right) \tag{4}$$

$$F_t^{explore} = PF_1 \cdot \frac{C_i^{explore} - C_{i+1}^{explore}}{T_t^{explore}} \tag{5}$$

$$F_t^{exploit} = PF_2 \cdot \frac{C_i^{exploit} - C_{i+1}^{exploit}}{T_t^{exploit}} \tag{6}$$

where

$$PF_1 = PF_1\_initial \times \left(1 - \frac{iter}{Max\ iteration} \right) \tag{7}$$

$$PF_2 = PF_2\_initial \times \left(1 - \frac{iter}{Max\ iteration}\right) \tag{8}$$

where F_2 represents a secondary exploitation force used to refine the position toward the best solution, and $F_t^{explore}$ and $F_t^{exploit}$ denote the total force generated at the current iteration t for exploration and exploitation phases, respectively. $S^1_{v\,exploit}$ indicates initial exploitation velocity, S_T represents the total territory constant, $T_t^{exploit}$ denotes the time step, and C_i specifies the current position of the i^{th} puma. C_{i+1} represents the subsequent position or target point used to calculate the directional vector of movement, PF_1 and PF_2 indicate primary factors for exploration and exploitation, respectively, which are dynamic weights controlling the influence of each phase at a given time. $PF_1\_initial$ and $PF_2\_initial$ denote the initial user-defined values, the current iteration is denoted by *iter*, and $1 - \frac{iter}{Max\ iteration}$ specifies a liner decay function. This is a linear decay function that ensures that, as the algorithm proceeds, the forces (PF_1 and PF_2) gradually decrease. It allows the model to make large jumps early on (Exploration) and very small, precise adjustments toward the end (Exploitation). However, the proposed POA is promising but sensitive to parameter settings, with risks of premature convergence and high computational cost impacting the selection of the best feature set, thus reducing accuracy and degrading deepfake detection efficiency. Therefore, a dual strategy is employed in this research to address the optimization challenge and enhance deepfake detection. Techniques such as HT distribution walk and adaptive weight tuning are the dual strategies used to overcome the algorithm's limitations and efficiently select the best feature set.

4). HT DISTRIBUTION WALK STRATEGY. To improve the search capability of puma in the search space for finding the optimal set of features during exploration, HT perturbation is integrated, as shown in Equation (9):

$$F_t^{explore} = PF_1 \cdot \frac{C_i^{explore} - C_{i+1}^{explore}}{T_t^{explore}} + \lambda \cdot HT \cdot (\alpha) \tag{9}$$

where $F_t^{explore}$ represents the optimal exploration value; λ denotes an adaptive scaling parameter for the exploration step size; $HT \cdot (\alpha) = \eta \cdot \frac{x}{|y|^{1/\alpha}}$ refers to the HT perturbation term, where α indicates the stability index; η denotes the amplitude controller that regulates the height of exploration jumps; x and y represent samples from a normal distribution; $1/\alpha$ denotes the exponential power that determines the probability of making large jumps. A smaller α results in "heavier tails" and more frequent long-range jumps.

By including random jumps of varying sizes, the POA escapes the issue of getting stuck in local optima. This HT perturbation mechanism prevents the algorithm from converging too early on redundant feature subsets that seem optimal within a limited search area. Furthermore, this HT perturbation-based exploration strategy promotes a more diverse and informative selection of features. During initial iterations, by analyzing broader regions of the search space, the POA algorithm ensures a comprehensive evaluation of potential feature combinations before making the final decision. Additionally, this HT perturbation strategy effectively addresses limitations of POA such as premature convergence and enhances computational efficiency by selecting a more concise feature set, which allows for faster processing.

The POA algorithm avoids redundant feature subsets that only identify obvious artifacts, such as prominent pixel-level inconsistencies, by escaping local optima. Moreover, the HT perturbation mechanism enables a wide search to detect subtle "hidden" weak signals. This approach ensures a robust selection of critical deepfake indicators, including abnormal eye-blinking rates and inconsistent lip synchronization.

5). ADAPTIVE PARAMETER TUNING STRATEGY. Next, an adaptive weight updating strategy for parameter tuning is introduced in POA to enhance feature selection, as described in Equations (10–12):

$$C_i^{t+1} = C_i^t + a_t^{exploit} \cdot (C_{best}^t - C_i^t) \tag{10}$$

$$a_t^{exploit} = 2 - 2 \cdot E_0 \cdot \omega_t \cdot \left(\frac{t}{T}\right) \tag{11}$$

$$\omega_t = r \cdot \left[\omega_{init}(\omega_{init} - \omega_{end}) \cdot \left(\frac{1}{e-1} \cdot \left(e^{\left(\frac{1}{7}\right)} - 1\right)\right)\right] \tag{12}$$

where C_i^t denotes current position i^{th} puma, C_{best}^t represents best puma (best solution), E_0 indicates a random value range between [−1,1], and r stands for a random weight adjustment factor. ω_{init} and ω_{end} represent initial and final weight values, respectively, t indicates current iteration, and T denotes maximum iteration, respectively. In POA, the adaptive control parameter in Equation (11) regulates the puma's movement toward the best solution during exploitation. Here, the initial value 2 enables stronger convergence in early stages, while the decreasing term ensures fine-tuned local search as iterations progress. The adaptive weight strategy in Equation (12) dynamically adjusts search pressure based on iteration feedback.

The proposed adaptive weight tuning strategy minimizes manual parameter calibration and reduces sensitivity to human bias. By dynamically balancing exploration and exploitation, it ensures the algorithm remains robust across diverse forgery types and changing environmental conditions. This approach effectively filters out unreliable signals, resulting in a more precise and efficient selection of high-quality deepfake indicators. These features are passed to the final detection phase using the SLSTM model.

E. DEEPFAKE DETECTION

The selected features are fed as input to the SLSTM model to learn facial information and detect fake images, improving deepfake detection. In this research, an LSTM model is used for identifying fake images, as it can efficiently capture sequential dependence and refined temporal patterns. Although the images are static, they are part of video frames, where the LSTM model [24] identifies hidden inconsistencies across the frames as well as within the frames. However, the neural network struggles to capture this information, which can affect accurate detection of fake images. Therefore, an improved LSTM is employed to distinguish real from fake images more efficiently. The LSTM is an enhanced recurrent neural network that uses memory blocks instead of neurons. Each memory block contains three gates: input, output, and forget gate as well as a memory cell to process the data from input to output. Figure 4 represents the architecture of the LSTM model. The memory cell in the LSTM helps remember long-term dependencies and detects minute temporal information efficiently.

However, the conventional LSTM model effectively captures basic temporal relationships but overlooks deeper hierarchical patterns across different levels of abstraction. Additionally, traditional LSTM models tend to overfit and struggle to learn

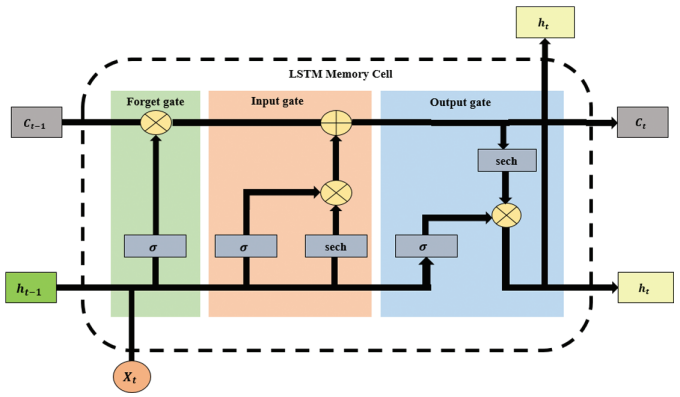

Fig. 4. Architecture of the LSTM cell.

generalized features, which are essential for detecting unseen deepfakes in images. Therefore, this research proposes a stacked LSTM to improve deepfake detection performance, as presented in Fig. 5. By integrating two LSTM models, increasing the model's depth allows it to learn complex information from selected features, enhancing generalization and making the detection more robust against unseen fake images. The proposed SLSTM extends the traditional LSTM model by stacking multiple hidden LSTM layers, each comprising several memory cells. These stacked LSTM hidden layers make the model deeper and more effective at distinguishing between real and fake images facilitating efficient deepfake detection. The three gates and memory cell in SLSTM are mathematically described in Equations (13–18):

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{13}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{14}$$

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{15}$$

$$\widehat{C}_t = sech(W_c * [h_{t-1}, x_t] + b_c) \tag{16}$$

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{17}$$

$$h_t = o_t \times sech(C_t) \tag{18}$$

where i_t, o_t, f_t, and c_t denote input, output, forget, and memory cell, respectively. W and b denote the weight and bias of the three gates and the memory cell, respectively, and h_t represents the hidden state; and *sech* denotes the hyperbolic secant activation function.

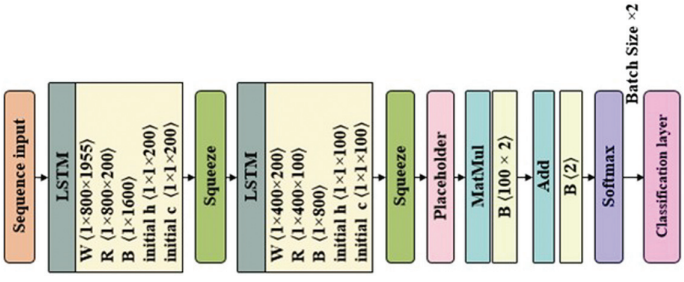

Fig. 5. Proposed SLSTM architecture for deepfake detection.

The sech activation function provides a smooth and bounded output in [0, 1], which helps stabilize the update of a memory cell in the SLSTM model. Unlike conventional activation functions that can cause values to explode or vanish, *sech* prevents numerical instability and reduces the risk of gradient saturation. By offering a sharper, more focused peak than the sigmoid function, it delivers a more effective gradient that alleviates vanishing gradient problems. The *sech* function enhances the stability and learning capacity of deep (stacked) architectures by maintaining proper memory cell updates. This allows the SLSTM to effectively capture long-term temporal dependencies and subtle artifacts in deepfake videos that conventional models often fail to recognize. The mathematical representation of this *sech* activation is expressed in Equation (19):

$$f(x) = \frac{2}{e^x + e^{-x}} \tag{19}$$

where e^x and e^{-x} denote exponential functions. Using *sech* in stacked LSTMs improves deepfake detection by mitigating the vanishing gradient issue common with sigmoid, enabling deeper layers to learn subtle temporal inconsistencies in manipulated videos. Its bounded, strictly positive nature prevents the exploding activations seen with rectified linear unit (ReLU), ensuring numerical stability during long-sequence processing, such as frame-by-frame analysis. This stability results in faster convergence and more accurate weight updates, directly enhancing the model's ability to distinguish natural facial movements from synthetic artifacts in deepfake videos.

To ensure evaluation integrity and prevent data leakage, dataset partitioning takes place at the video source level rather than the frame level. This protocol ensures that frames from the same video or specific identities do not appear in both training and testing phases, providing a true measure of generalization to unseen faces. By using 80:10:10 split, the validation set facilitates early stopping to prevent overfitting, while the testing set remains entirely isolated until the final assessment. During preprocessing, face crops are resized to 224×224 pixels to preserve critical high-frequency textures, such as skin pores, while maintaining computational efficiency. Subsequently, 30 consecutive frames per segment are extracted and normalized to a [0, 1] range to provide a consistent temporal window for analysis. The SLSTM identifies irregular biometric rhythms, such as inconsistent lip synchronization or eye-blinking, which often remain undetectable in static images. By analyzing motion continuity, the architecture detects temporal discontinuities, including subtle flickering at facial boundaries resulting from the generative process. This focus on temporal sensitivity enables the model to distinguish synthetic artifacts from natural movements accurately.

III. RESULTS AND DISCUSSION

The experimental analysis of the DSPO-SLSTM-based detection model in deepfake detection utilizing three distinct datasets, namely FF++, Celeb-DF, and Celeb-DF-V2, is depicted in this section. The DSPO-SLSTM and comparison models are implemented using MATLAB 2020b on a system with 16 GB RAM, an i7 processor, and Windows 10 OS.

Experimental Setup:

Tables I and II show the parameter settings for the proposed DSPO-SLSTM models used in deepfake detection through video clips. These parameters for the DSPO algorithm and SLSTM network are determined through a systematic selection process instead of random assignment. Initial values for population size and

Table I. Parameter settings of the proposed DSPO algorithm along with the conventional POA

Parameters	Values	
	POA	Proposed DSPO algorithm
Maximum iteration	100	100
Lower bound	0.00	0.00
Upper bound	1.00	1.00
Search agent	30	30
PF1	0.50	0.50
PF2	0.50	0.50
PF3	0.30	0.30

Table II. Parameter settings of the proposed SLSTM method for three benchmark datasets

Parameters	Values		
	FF++ dataset	Celeb-DF dataset	Celeb-DF-V2 dataset
Initial learning rate	0.0015	0.001	0.0004
Gradient decay factor	0.6	0.75	0.8
Epochs	50	50	50
Batch size	32	32	32

learning rates are based on standard benchmarks for the POA and LSTM architectures to establish a reliable baseline. Hyperparameters such as learning rate and batch size are refined via grid search to ensure efficient convergence. Specifically, the learning rate of 0.001 maintains a balance between training speed and update accuracy, preventing instability in the loss curve. The batch size of 16 enables necessary randomness to help the model escape local optima in complex video data while maintaining stable gradient updates. Finally, dataset-specific adjustments include setting maximum iterations to $T = 50$, providing sufficient time for the feature selection process to reach a steady state across the complex FF++, Celeb-DF, and Celeb-DF-V2 datasets.

To ensure a rigorous evaluation, the datasets FF++, Celeb-DF, and Celeb-DF-V2 are split using an 80:10:10 ratio for training, validation, and testing. The training set is used to optimize SLSTM weights and the DSPO feature selection, while the validation set helps tune hyperparameters and implement early stopping to prevent overfitting. To verify the statistical significance of the results, K-fold cross-validation is employed, rotating through multiple iterations for consistent performance across different subsets. Furthermore, the DSPO-SLSTM maintains a high area under the curve (AUC) by identifying general biological inconsistencies rather than dataset-specific artifacts. Standard performance metrics such as accuracy, precision, recall, and the Matthews correlation coefficient (MCC) are used for a comprehensive assessment of the model's reliability, beyond just simple classification rates.

A. PERFORMANCE METRICS

Various performance measures are utilized in this research to assess the effectiveness of the DSPO-SLSTM method in deepfake detection. Evaluation metrics include precision, sensitivity, accuracy, F1-score, specificity, and MCC, respectively. The mathematical formulas for these metrics are given in Equations (20–25):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100 \tag{20}$$

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F1 - score = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity} \tag{23}$$

$$Classification\ Error = (100 - Accuracy) \tag{24}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{25}$$

where TP and TN denote true positive and true negative, respectively; FP and FN represent false positive and false negative, respectively.

B. PERFORMANCE ANALYSIS

A quantitative and qualitative evaluation of the DSPO-SLSTM deepfake detection model is conducted, comparing it with state-of-the-art methods utilizing FF++, Celeb-DF, and Celeb-DF-V2 datasets. This section presents the analysis of the proposed feature selection, detection model, and their performance. The performance of the proposed model including its feature extraction, selection, and detection components is evaluated with existing state-of-the-art approaches using various metrics.

1). PROPOSED FEATURE SELECTION. Figure 6 demonstrates the performance of the DSPO algorithm for deepfake detection using FF++ dataset. Figure 7 illustrates the feature selection performance of DSPO combined with different optimization algorithms for deepfake detection on the Celeb-DF dataset. Figure 8 presents the performance analysis of the DSPO-based feature selection model with various optimization algorithms on benchmark datasets including Celeb-DF-V2. Optimization algorithms such as Grey Wolf Optimization (GWO), GA, Cheetah

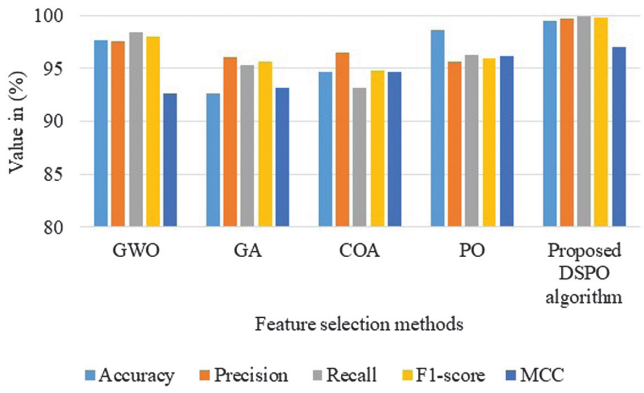

Fig. 6. Evaluation results of the proposed DSPO algorithm with different optimization algorithms for feature selection using FF++ dataset.

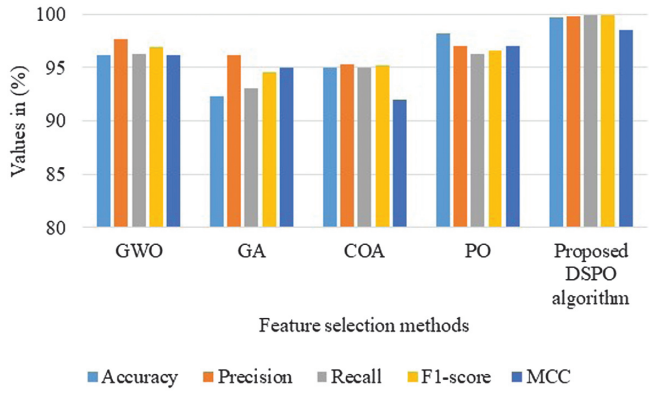

Fig. 7. Performance evaluation of the proposed DSPO algorithm with different optimization algorithms for feature selection using Celeb-DF dataset.

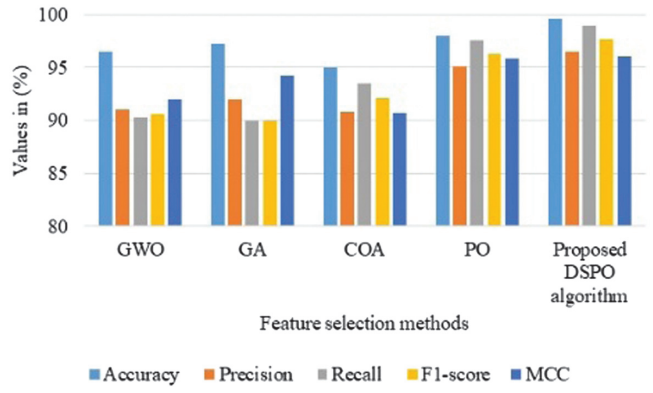

Fig. 8. Performance analysis of different optimization algorithms with the proposed DSPO algorithm used for feature selection based on Celeb-DF-V2 dataset.

Optimization Algorithm (COA), and conventional PO are used in comparison with the proposed DSPO algorithm.

2). DETECTION MODEL. A quantitative and qualitative analysis of the DSPO-SLSTM detection model is carried out, comparing it with various DL methods for identifying deepfake images. Tables III–V display the performance evaluation of the SLSTM-based detection model using FF++, Celeb-DF, and Celeb-DF-V2 datasets. DL methods such as CNN, traditional LSTM, and Bi-LSTM are evaluated, both with and without feature selection.

A comparative performance analysis of the sech activation functions and existing activation functions using the SLSTM architecture is shown in Table VI. The results indicate that the proposed SLSTM with sech function enhances deepfake detection by providing a focused gradient that mitigates the vanishing gradient problem. This allows the model to better capture subtle temporal inconsistencies across video sequences. The bell-shaped curve acts as a soft-thresholding filter, emphasizing high-quality biometric signals and reducing noise from compression or lighting conditions. Additionally, its smooth, symmetric activation ensures numerical stability in stacked architectures, leading to more accurate detection of synthetic artifacts.

The statistical reliability of the DSPO-SLSTM model is evaluated using the measures presented in Table VII. A Wilcoxon signed-rank test is conducted to compare the proposed architecture against the baseline, yielding a p-value of less than 0.05. This result confirms that the observed performance improvements stem from the model design rather than random variation. Moreover, a minimal standard deviation of ± 0.04, along with narrow 95% confidence intervals across all datasets, demonstrates consistent performance. These statistical findings provide objective evidence that the accuracy gains are significant, validating the effectiveness of the dual-strategy optimization and sech-activated temporal modeling.

3). GRAPHICAL REPRESENTATION. The graphical representation involves visual tools used to present and explain the data and comparison between models. This visual representation makes complex results easier to understand, highlighting patterns and allowing visual model comparison. In this research, the AUC curve, standard deviation, and confusion matrix are used to illustrate the performance of the proposed DSPO-SLSTM model for deepfake detection. Figure 9 (a) confusion matrix, (b) ROC, and (c) standard deviation show the performance of the proposed deepfake detection model using the FF+ dataset. Figure 10 (a) confusion matrix, (b) ROC, and (c) standard deviation display the performance of the DSPO-LSTM model on the Celeb-DF dataset. Figure 11 (a) confusion matrix, (b) ROC, and (c) standard deviation depict the performance of the DSPO-LSTM model on the Celeb-DF-V2 dataset.

In Fig. 9 (a), the confusion matrix shows that the DSPO-SLSTM model correctly identifies 397 TN, 398 TP, and only five misclassifications (two FN and three FP). This graphical data indicates that the proposed method can accurately distinguish between real and fake images, demonstrating the model's strong discriminative ability. Figure 9 (b) clearly shows that both real and fake classes achieve an AUC of 1.00, indicating that the model is highly effective at distinguishing between classes across all thresholds, with zero overlap between class distributions.

Table III. Performance evaluation of the SLSTM model based on feature selection using FF++ dataset

Methods	Feature selection	Accuracy (%)	Recall (%)	MCC (%)	Precision (%)	F1-score (%)
CNN	No	94.076	94.109	90.705	96.109	95.098
LSTM		92.909	95.298	92.109	92.1	93.671
Bi-LSTM		95.91	93.035	92.106	95.066	94.039
SLSTM		98.991	94.401	96.497	99.211	99.306
Proposed DSPO-CNN	Yes	94.625	64.658	91.254	96.658	95.647
Proposed DSPO-LSTM		93.458	95.847	92.658	92.648	94.220
Proposed DSPO-Bi-LSTM		96.458	93.584	92.654	95.614	94.588
Proposed DSPO-SLSTM		99.54	99.95	97.045	99.76	99.854

Table IV. Performance evaluation of SLSTM model based on feature selection using Celeb-DF dataset

Methods	Feature selection	Accuracy (%)	Recall (%)	MCC (%)	Precision (%)	F1-score (%)
CNN	No	95.191	93.39	92.888	94.888	94.133
LSTM		91.888	93.089	91.796	95.434	94.247
Bi-LSTM		95.558	97.885	95.855	96.916	97.398
SLSTM		99.184	99.714	97.978	99.294	99.503
Proposed DSPO-CNN	Yes	95.457	93.655	93.154	95.154	94.399
Proposed DSPO-LSTM		92.154	93.658	92.364	96	94.814
Proposed DSPO-Bi-LSTM		96.124	98.154	96.124	97.485	97.818
Proposed DSPO-SLSTM		99.75	99.98	98.5467	99.86	99.92

Table V. Performance evaluation of SLSTM model based on feature selection using Celeb-DF-V2 dataset

Methods	Feature selection	Accuracy (%)	Recall (%)	MCC (%)	Precision (%)	F1-score (%)
CNN	No	91.755	92.589	90.770	90.734	91.652
LSTM		93.188	93.779	92.302	93.366	93.572
Bi-LSTM		95.755	96.019	93.740	94.734	95.372
SLSTM		98.274	98.071	94.755	95.211	96.620
Proposed DSPO-CNN	Yes	93.021	93.457	92.036	92.000	92.723
Proposed DSPO-LSTM		94.056	95.044	93.568	94.632	94.838
Proposed DSPO-Bi-LSTM		97.021	96.887	95.005	96.74	96.441
Proposed DSPO-SLSTM		99.54	98.94	96.021	96.48	97.694

Table VI. Performance evaluation of sech activation function in SLSTM model with existing activation functions

Configurations	Dataset	Accuracy (%)	Recall (%)	MCC (%)	Precision (%)	F1-score (%)
SLSTM-sigmoid	FF++	97.42	97.55	94.85	97.30	97.42
SLSTM-ReLU		98.15	98.30	95.90	98.05	98.17
SLSTM-Tanh		98.64	98.50	96.22	98.70	98.60
SLSTM-Leaky ReLU		98.92	99.10	96.75	98.85	98.97
SLSTM-sech		**99.54**	**99.95**	**97.045**	**99.76**	**99.854**
SLSTM-sigmoid	Celeb-DF	97.80	98.05	95.60	97.65	97.85
SLSTM-ReLU		98.35	98.50	96.70	98.22	98.36
SLSTM-Tanh		98.90	99.12	97.45	98.80	98.96
SLSTM-Leaky ReLU		99.25	99.40	98.10	99.20	99.30
SLSTM-sech		**99.75**	**99.98**	**98.5467**	**99.86**	**99.92**
SLSTM-sigmoid	Celeb-DF-V2	96.10	95.85	92.20	94.15	94.99
SLSTM-ReLU		97.55	96.90	93.80	95.05	95.97
SLSTM-Tanh		98.20	97.75	94.55	95.80	96.77
SLSTM-Leaky ReLU		98.95	98.40	95.40	96.20	97.29
SLSTM-sech		**99.54**	**98.94**	**96.021**	**96.48**	**97.694**

Table VII. Statistical stability and significance analysis of the DSPO-SLSTM model in deepfake detection

Dataset	Metric	Mean value (%)	Std. deviation (±%)	95% confidence interval (CI)	*p*-Value (vs. baseline)
FF++	Accuracy	99.54	±0.04	[99.42, 99.66]	0.014
Celeb-DF		99.75	±0.05	[99.63, 99.87]	0.013
Celeb-DF V2		99.54	±0.06	[99.38, 99.70]	0.012

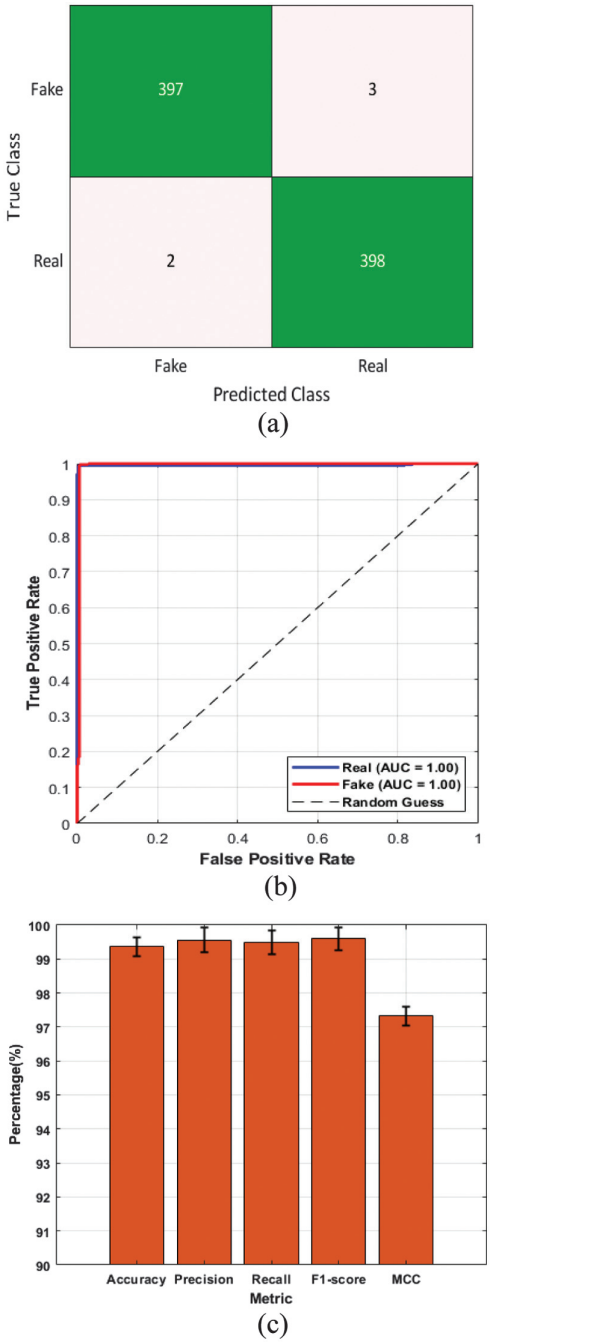

Fig. 9. Graphical representation of proposed DSPO-SLSTM model in terms of the (a) confusion matrix, (b) AUC curve, and (c) standard deviation for deepfake detection.

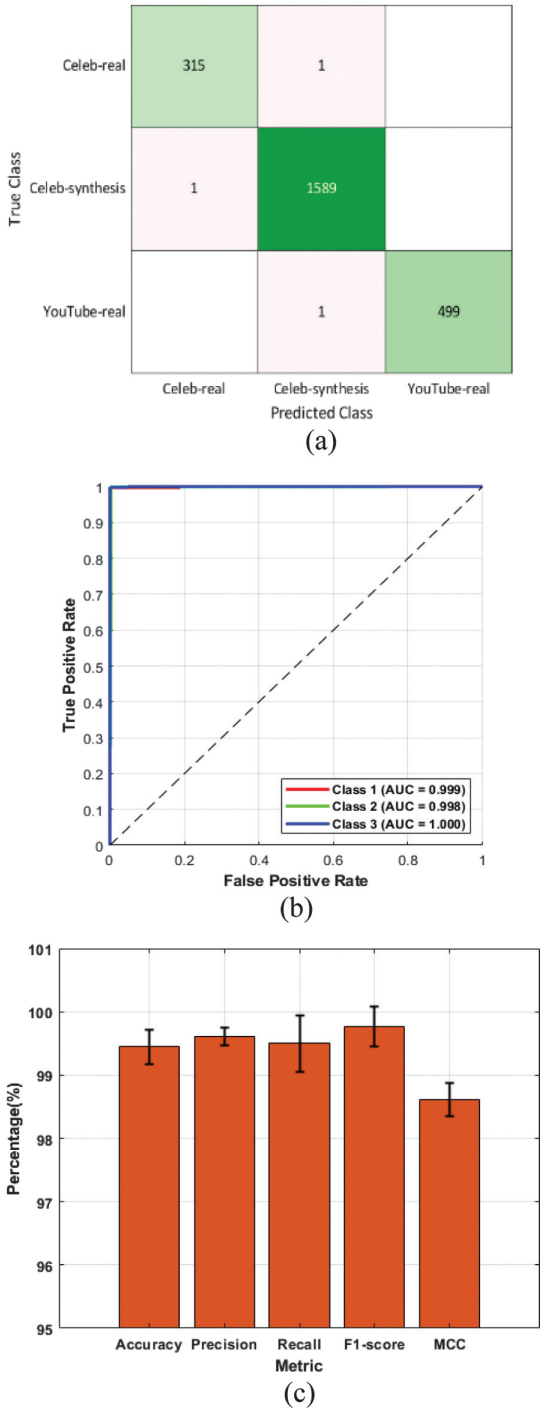

Fig. 10. Graphical representation of the proposed DSPO-SLSTM model in terms of (a) confusion matrix, (b) AUC curve, and (c) standard deviation for deepfake detection.

- **For Celeb-DF-dataset:**

From Fig. 10 (a–c), showing the confusion matrix, ROC curve, and standard deviation, the figure demonstrates exceptional deepfake detection performance of the DSPO-SLSTM model on the Celeb-DF dataset. The near-perfect confusion matrix, with only one misclassification per class, and high AUC scores confirm excellent class separability. The consistently high values across accuracy, precision, recall, F1-score, and MCC along with minimal standard deviation reflect the model's robustness. These results highlight how DSPO optimizes critical features and SLSTM's strength in capturing temporal dependencies.

- **For Celeb-DF-V2 dataset:**

Similarly, Fig. 11 (a–c) illustrate the confusion matrix, ROC curve, and standard deviation for the DSPO-SLSTM model testing on the Celeb-DF-V2 dataset, showing outstanding performance.

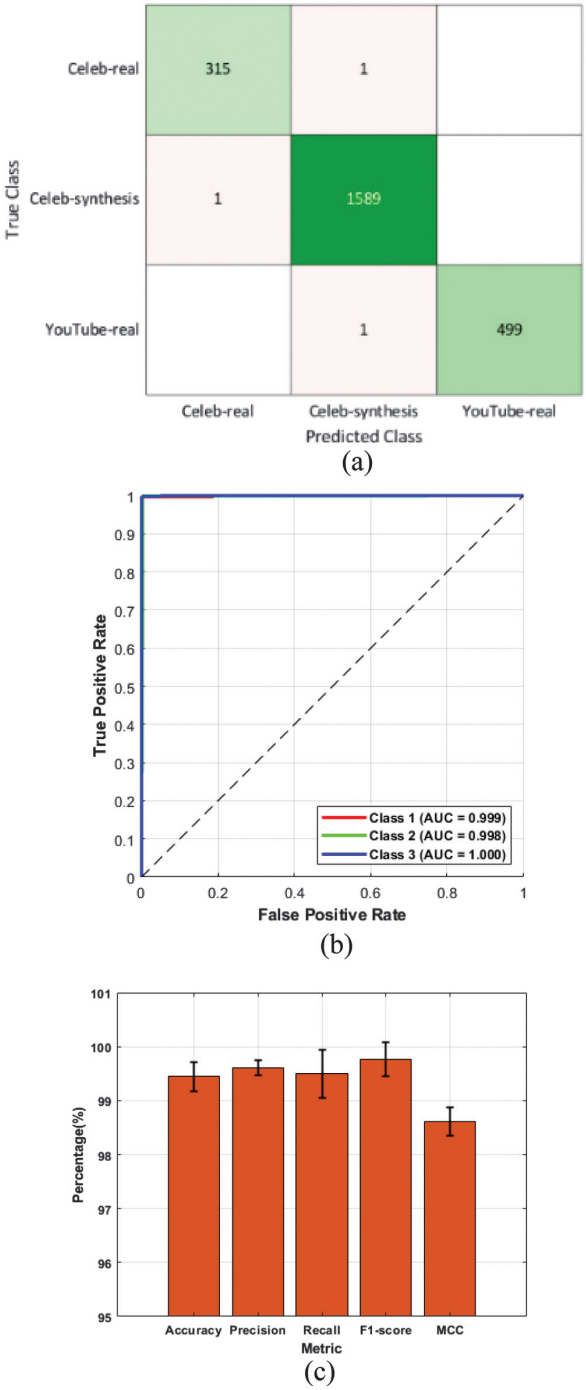

Fig. 11. Graphical representation of the proposed DSPO-SLSTM model in terms of (a) confusion matrix, (b) AUC curve, and (c) standard deviation for deepfake detection.

The confusion matrix with only one misclassification per class and AUC scores of 0.999, 0.998, and 1.000 indicate near-perfect class separation. The evaluation metrics such as accuracy, precision, recall, F1-score, and MCC also achieve improved results with low standard deviations, demonstrating the model's stability. These findings confirm the effectiveness of the DSPO in extracting optimized temporal features and classifying deepfakes reliably.

C. CROSS-VALIDATION WITH K-FOLDS

Table VIII illustrates the k-fold cross-validation results for the proposed DSPO-SLSTM model across three benchmark datasets. The optimal performance at $k = 5$ reflects the best balance between training data size and model stability. At lower k values such as 2 and 3, the smaller training sets limit the model's ability to learn complex temporal dependencies and biometric irregularities typical of deepfakes, leading to higher bias. The transition to $k = 5$ provides sufficient data for the DSPO algorithm to learn generalized features rather than dataset-specific artifacts, yielding a significant performance boost. At $k = 7$, performance declines because the model reaches its maximum informational capacity, and the smaller validation sets introduce higher variance without providing additional insights for deepfake detection.

Ablation Study:

Table IX represents the ablation study results of the DSPO-SLSTM deepfake detection approach across three publicly available datasets. The baseline model utilizing single-stream feature extraction demonstrates that individual CNN architectures lack the diversity needed to capture all types of forgeries in deepfake detection. In contrast, integrating InceptionNet-V3 and EfficientNet-B0 offers multi-scale spatial features that greatly enhance detection accuracy over the baseline. The absence of the DSPO feature selection results in lower accuracy due to processing redundant and irrelevant spatial data, indicating that without an optimization mechanism, the inclusion of non-discriminative features introduces noise, impairing the system's ability to spot critical deepfake indicators. Omitting the stacked LSTM causes a significant decline in performance, highlighting its crucial role in detecting temporal inconsistencies and motion artifacts that static CNNs fail to identify. Replacing the sech activation function with traditional layer Tanh leads to lower MCC scores. This demonstrates that the sech activation function helps prevent gradient saturation and ensures numerical stability during training of deep architectures. Additionally, the sech activation function improves deepfake detection by enabling the SLSTM to better capture long-term temporal inconsistencies such as irregular eye-blinking and lip synchronization.

D. SENSITIVITY ANALYSIS

To assess the stability and robustness of the proposed architecture, a sensitivity analysis is performed on three main variables: population size (N), the scaling parameter (α), and the learning rate. For the DSPO algorithm, increasing the population size from 10 to 50 improves feature diversity; however, accuracy gains become marginal beyond N = 30, while computational costs increase linearly. This indicates that the algorithm is resilient to changes in population size. Regarding the scaling parameter (α) in the HT walk, values between 1.2 and 1.8 offer optimal convergence speeds. Extreme values cause erratic behavior or local optima, but the adaptive tuning strategy effectively manages these fluctuations to maintain consistent final accuracy.

For the SLSTM network, a learning rate of 0.001 achieves an ideal balance between training speed and gradient stability. Rates exceeding 0.01 lead to instability in the sech activation gradients, while rates below 0.0001 cause premature performance. Across diverse datasets, the proposed model remains highly robust at the 0.001 level, consistently achieving an MCC above 0.98. Overall, this analysis confirms that the DSPO-SLSTM model is highly stable. The adaptive parameter tuning acts as a self-correcting mechanism,

Table VIII. Performance of the DSPO-SLSTM across varying K-fold cross-validation configurations

k-folds	Accuracy (%)	Recall (%)	MCC (%)	Precision (%)	F1-score (%)
FF++ dataset					
$k=2$	97.85	98.12	95.20	97.90	98.01
$k=3$	98.60	98.85	96.15	98.72	98.78
$k=5$	**99.54**	**99.95**	**97.045**	**99.76**	**99.854**
$k=7$	99.15	99.40	96.80	99.25	99.32
Celeb-DF dataset					
$k=2$	98.10	98.35	96.05	98.20	98.27
$k=3$	98.95	99.10	97.40	99.05	99.07
$k=5$	**99.75**	**99.98**	**98.5467**	**99.86**	**99.92**
$k=7$	99.40	99.65	98.15	99.50	99.57
Celeb-DF-V2 dataset					
$k=2$	97.20	96.50	93.85	94.10	95.28
$k=3$	98.45	97.80	94.90	95.35	96.56
$k=5$	**99.54**	**98.94**	**96.021**	**96.48**	**97.694**
$k=7$	99.05	98.55	95.65	96.10	97.31

Table IX. Ablation study of the DSPO-SLSTM architecture evaluating the impact of individual components on detection accuracy

Variant	Components removed	Deepfake datasets		
		FF++ (%)	Celeb-DF (%)	Celeb-DF-V2 (%)
Baseline	No feature extraction, No DSPO, No SLSTM (static CNN only) (single CNN only)	92.15	91.40	90.12
w/o InceptionNet-V3 and EfficientNet-B0	No combined feature extraction (single stream only)	94.30	93.85	92.45
w/o DSPO	All extracted features used (no optimization)	96.85	96.40	95.10
w/o SLSTM	Single frame analysis (no temporal modeling)	95.15	94.80	93.90
w/o sech activation function	Replaced sech with the traditional activation function Tanh	98.64	98.90	98.20
Proposed	**Full DSPO-SLSTM (with combined feature extraction and sech activation function)**	**99.54**	**99.75**	**99.54**

allowing the system to adjust dynamically during the iteration process to maintain peak performance despite varying initial conditions.

1). COMPARATIVE STUDY. Comparative analysis of the proposed deepfake detection model is compared with the existing research methods using the FF++, Celeb-DF, and Celeb-DF-V2 benchmark datasets. Table X presents a comparative study of the DSPO-SLSTM model against previous deepfake detection algorithms on these datasets. The approaches in [16–19,21–25] are used for comparison with the DSPO-SLSTM model. To ensure fair and consistent evaluation, performance metrics are standardized across all comparative analyses. The accuracy values from previous studies, originally reported as decimals, are converted to percentages, while AUC remains in decimal format. This uniform scaling allows for a clearer comparison of the proposed DSPO-SLSTM with existing benchmarks.

The DSPO-SLSTM method achieved better performance than all existing CNN, GAN, and ViT-based approaches in accuracy and AUC across all benchmark datasets. This improvement results from integrating multi-scale spatial feature fusion with the DSPO optimization algorithm, which isolates the most discriminative features while filtering redundant data. The sech-activated SLSTM captures long-term temporal inconsistencies more effectively than traditional sequential models, ensuring robust detection across both compressed and high-quality deepfake videos.

IV. CONCLUSION

This research proposed the DSPO-SLSTM architecture to improve deepfake detection by addressing feature redundancy. By integrating multi-scale spatial features from InceptionNet-V3 and EfficientNet-B0, the model established a diverse set of foundational features. The proposed DSPO algorithm balanced exploration and exploitation through an HT distribution walk and adaptive parameter tuning to select highly discriminative features. These optimized features were processed by an SLSTM utilizing a sech activation function to capture complex temporal inconsistencies. By mitigating gradient saturation, this configuration allowed the model to learn subtle, long-term motion anomalies across video sequences, ensuring high precision in distinguishing authentic frames from manipulated content. Ablation results confirmed that each component was vital, as removing the DSPO algorithm,

Table X. Comparative evaluation of the DSPO-SLSTM model for deepfake detection using three benchmark datasets with various detection methods

Detection methods	Datasets	Accuracy (%)	AUC
BMNet [16]	FF++	95.54	98.60%
	Celeb-DF	80.20	75.72
Multi-fusion between GNN and CNN [17]	FF++	95.09	N/A
	Celeb-DF	98.9	0.98
DSC-CBAM-ViViT [18]	FF++	90.27	N/A
	Celeb-DF-V2	87.17	N/A
LSTM-VGG-16 [19]	Celeb-DF	97.01	N/A
GAN [21]	FF++	94.89	99.90
	Celeb-DF-V2	92.34	95.40
Con-GAN [22]	FF++	98.91	98.92
	Celeb-DF-V2	98.89	N/A
FreqFaceNet [23]	FF++	0.939	0.981
	Celeb-DF	0.983	0.998
DWT-ViT [24]	FF++	98.86	0.9995
	Celeb-DF	99.72	0.9996
MSIDNet [25]	FF++	95.51	95.34
	Celeb-DF-V2	99.30	99.11
Proposed DSPO-SLSTM method	**FF++**	**99.54**	**99.899**
	Celeb-DF	**99.75**	**99.96**
	Celeb-DF-V2	**99.54**	**99.26**

SLSTM, and sech activation significantly impacted detection accuracy and robustness. Additionally, the comparative analysis demonstrated that the proposed method attained better performance than state-of-the-art approaches, including BMNet and DWT-ViT, by identifying subtle artifacts such as facial flickering and irregular motion transitions. Experimental results of DSPO-SLSTM showed that the model achieved high detection accuracy across three benchmark datasets such as FF++, Celeb-DF, and Celeb-DF-V2.

Future Work: This research mainly focuses on visual, spatial and temporal artifacts, which excludes the analysis of textual and audio-based forgeries. Furthermore, the computational demands of the dual-stream feature extraction pose challenges for deployment on resource-constrained devices. To address these constraints, future work will explore transformer-based architectures to enhance detection capabilities across both visual and textual data. This expansion will enable a more comprehensive approach for identifying complex synthetic media by integrating multimodal analysis.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] X. Qiu *et al.*, "D2Fusion: Dual-domain fusion with feature superposition for deepfake detection," *Inf. Fusion*, vol. 120, p. 103087, 2025.

[2] Z. Lai *et al.*, "Leveraging high-frequency diversified augmentation for general deepfake detection," *J. Inf. Secur. Appl.*, vol. 89, p. 103994, 2025.

[3] M. A. Raza, K. M. Malik, and I. U. Haq, "Holisticdfd: Infusing spatiotemporal transformer embeddings for deepfake detection," *Inf. Fusion*, vol. 645, p. 119352, 2023.

[4] B. E. Katı, E. U. Küçüksille, and G. Sarıman, "Enhancing deepfake detection through quantum transfer learning and class-attention vision transformer architecture," *Appl. Sci.*, vol. 15, no. 2, p. 525, 2025.

[5] B. Xiang *et al.*, "Locality sensitive hashing-based deepfake image recognition for athletic celebrities," *Int. J. Intell. Syst.*, vol. 2025, no. 1, p. 1313970, 2025.

[6] A. Heidari *et al.*, "Deepfake detection using deep learning methods: A systematic and comprehensive review," *Wiley Interdiscip. Rev.: Data Min. Knowl. Discov.*, vol. 14, no. 2, p. e1520, 2023.

[7] B. Wang *et al.*, "Frequency domain filtered residual network for deepfake detection," *Mathematics*, vol. 11, no. 4, p. 816, 2023.

[8] D. Xiong *et al.*, "BMNet: Enhancing deepfake detection through BiLSTM and multi-head self-attention mechanism," *IEEE Access*, vol. 13, pp. 21547–21556, 2025.

[9] M. M. El-Gayar *et al.*, "A novel approach for detecting deepfake videos using graph neural network," *J. Big Data*, vol. 11, no. 1, p. 22, 2024.

[10] P. G. Prathibha *et al.*, "SARB-DF: A continual learning aided framework for deepfake video detection using self-attention residual block," *IEEE Access*, vol. 12, pp. 189088–189101, 2024.

[11] K. N. Ramadhani, R. Munir, and N. P. Utama, "Improving video vision transformer for deepfake video detection using facial landmark, depthwise separable convolution and self-attention," *IEEE Access*, vol. 12, pp. 8932–8939, 2024.

[12] L. Boongasame *et al.*, "Design and implement deepfake video detection using VGG-16 and long short-term memory," *Appl. Comput. Intell. Soft Comput.*, vol. 2024, no. 1, p. 8729440, 2024.

[13] N. Rathoure *et al.*, "Combating deepfakes: A comprehensive multilayer deepfake video detection framework," *Multimed. Tools Appl.*, vol. 83, pp. 85619–85636, 2024.

[14] D. R. Agrawal and F. Haneef, "Eye blinking feature processing using convolutional generative adversarial network for deep fake video detection," *Trans. Emerg. Telecommun. Technol.*, vol. 36, no. 3, p. e70083, 2025.

[15] V. Gupta *et al.*, "FreqFaceNet: An enhanced transformer architecture with dual-order frequency attention for deepfake detection," *Appl. Intell.*, vol. 55, no. 6, p. 477, 2025.

[16] M. Uddin, Z. Fu, and X. Zhang, "Deepfake face detection via multi-level discrete wavelet transform and vision transformer," *The Visual Computer*, Jan. 2025.

[17] Z. Cheng *et al.*, "Deepfake detection method based on multi-scale interactive dual-stream network," *J. Vis. Commun. Image Represent.*, vol. 104, p. 104263, 2024.

[18] A. Rossler *et al.*, "Faceforensics++: Learning to detect manipulated facial images," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, 2019.

[19] Y. Li *et al.*, "Celeb-df: A large-scale challenging dataset for deepfake forensics," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3207–3216, 2020.

[20] S. Muppalla, S. Jia, and S. Lyu, "Integrating audio-visual features for multimodal deepfake detection," in *2023 IEEE MIT Undergraduate Research Technology Conference (URTC)*. Cambridge, MA, USA: IEEE, 2023, pp. 1–5.

[21] G. Meena, K. K. Mohbey, and S. Kumar, "Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach," *Int. J. Inf. Manage. Data Insights*, vol. 3, no. 1, p. 100174, 2023.

[22] S. U. Amin *et al.*, "Enhanced anomaly detection in pandemic surveillance videos: An attention approach with EfficientNet-B0 and CBAM integration," *IEEE Access*, vol. 12, p. 162697–162712, 2024.

[23] B. Abdollahzadeh *et al.*, "Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning," *Cluster Comput.*, vol. 27, no. 4, pp. 5235–5283, 2024.

[24] Y. K. Saheed, A. I. Omole, and M. O. Sabit, "GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data," *Sens. Int.*, vol. 6, p. 100297, 2025.