DOI: 10.53608/estudambilisim.1804920

(Geliş Tarihi / Received Date: 17.10.2025, Kabul Tarihi/ Accepted Date: 22.01.2026)

Research Article

Heuristic Optimization of RNN and LSTM Models for Detecting DeepFake Voice

Tolun KELEŞ[1], Emrah ATILGAN[2], İdris DAĞ[3]

[1]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 26480, Eskişehir, ORCID No : http://orcid.org/0009-0009-9992-9584
[2]Eskişehir Osmangazi Üniversitesi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 26480, Eskişehir, ORCID No : http://orcid.org/0000-0002-0395-9976
[3]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 26480, Eskişehir, ORCID No : http://orcid.org/0000-0002-2056-4968

Keywords:
Deepfake Voice,
Deep Learning,
Heuristic Algorithms

Abstract: Deepfake voice synthesis, which enables the artificial replication of human speech through deep learning and natural language processing, poses increasing risks to information security and digital trust. Detecting such synthetic voices remains a challenging task due to the high realism and variability of generated speech. This study proposes an enhanced Deepfake voice detection framework that integrates heuristic optimization algorithms with deep learning architectures to improve model performance. Specifically, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models were optimized using nine heuristic and meta-heuristic algorithms, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Jaya. The ADD 2022 dataset—comprising 11,536 samples (8,275 fake and 3,260 real)—was preprocessed into spectrogram-based numerical features for training and testing. Each heuristic algorithm performed hyperparameter optimization over learning rate, dropout rate, input size, and batch size to maximize detection accuracy. Experimental results demonstrate substantial performance gains, with RNN–PSO and LSTM–DE combinations achieving 99.0% and 97.2% accuracy, respectively—an improvement of over 20% compared to the baseline model. These findings indicate that heuristic optimization significantly enhances generalization and convergence efficiency in Deepfake voice detection. The proposed approach contributes a scalable optimization framework adaptable to other deep learning-based media authentication tasks.

Araştırma Makalesi

DeepFake Ses Tespiti İçin RNN ve LSTM Modellerinin Sezgisel Optimizasyonu

Anahtar Kelimeler:
Deepfake Ses,
Derin Öğrenme,
Sezgisel Algoritmalar

Özet: Derin öğrenme ve doğal dil işleme yöntemleriyle insan sesinin yapay olarak taklit edilmesi, bilgi güvenliği ve dijital güven açısından ciddi riskler oluşturmaktadır. Üretilen Deepfake seslerin gerçek seslerle yüksek benzerlik göstermesi, tespit süreçlerini karmaşık hale getirmektedir. Bu çalışmada, Deepfake ses tespitinde model performansını artırmak amacıyla derin öğrenme mimarileri ile sezgisel optimizasyon algoritmaları birleştirilmiştir. Özellikle, Yinelenen Sinir Ağı (RNN) ve Uzun-Kısa Süreli Bellek (LSTM) modelleri, Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO), Diferansiyel Evrim (DE) ve Jaya gibi toplam dokuz sezgisel ve meta-sezgisel algoritma ile optimize edilmiştir. 8.275 sahte ve 3.260 gerçek ses örneğinden oluşan ADD 2022 veri kümesi, spektrogram tabanlı sayısal özniteliklere dönüştürülerek

*Sorumlu Yazar/Corresponding Author:kelestolun08@gmail.com

eğitilmiş ve test edilmiştir. Sezgisel algoritmalar, öğrenme oranı, dropout oranı, giriş boyutu ve batch boyutu gibi hiperparametreleri optimize ederek doğruluk oranını maksimize etmiştir. Deneysel sonuçlara göre, RNN-PSO ve LSTM-DE kombinasyonları sırasıyla %99,0 ve %97,2 doğruluk değerlerine ulaşmış, bu da başlangıç modeline kıyasla %20'nin üzerinde bir iyileşme sağlamıştır. Sonuçlar, sezgisel optimizasyonun Deepfake ses tespitinde genelleme kabiliyetini ve yakınsama hızını önemli ölçüde artırdığını göstermektedir. Önerilen yaklaşım, diğer derin öğrenme tabanlı medya doğrulama görevlerine de uyarlanabilir ölçeklenebilir bir optimizasyon çerçevesi sunmaktadır.

1. INTRODUCTION

In recent years, *Deepfake* technologies have rapidly evolved, enabling the generation of highly realistic synthetic media that imitates human appearance, behavior, and voice (Almutairi & Elgibreen, 2022; Westerlund, 2019). Among its various forms—video, image, text, and audio—Deepfake voice has gained particular attention due to its ability to mimic the tone, prosody, and rhythm of a target speaker with near-human precision (Amezaga & Hajek, 2022; Khanjani et al., 2021; Mcuba et al., 2023). By training deep learning models on speech recordings, attackers can produce synthetic voices indistinguishable from genuine human speech (Khochare et al., 2022). While this innovation has legitimate uses in entertainment, assistive technologies, and virtual assistants, it also introduces severe threats related to impersonation, misinformation, and identity fraud (Dixit et al., 2023; Frank & Schönherr, 2021; Tariq et al., 2022; Yi, Tao, et al., 2023).

The increasing use of voice-based authentication systems and audio communication platforms magnifies the potential risk of such synthetic speech (Salih et al., 2025). Deepfake voice can facilitate *telephone scams, political misinformation, and social engineering attacks*, posing ethical and security challenges (Ahmad et al., 2024) (Shaaban et al., 2023). Unlike visual Deepfakes, which can often be detected by facial artifacts or inconsistencies, audio Deepfakes lack clear perceptual cues, making detection substantially more difficult (Mubarak et al., 2023; Yi, Wang, et al., 2023). Identifying subtle manipulations in speech signals requires sophisticated analysis of frequency, phase, and temporal characteristics—areas where current detection methods still fall short (Chintha et al., 2020; Martin-Donas & Alvarez, 2022).

Traditional Deepfake voice detection approaches primarily rely on Convolutional Neural Networks (CNNs) (Jain & Singh, 2024) or Recurrent Neural Networks (RNNs) (Al-Dhabi & Zhang, 2021) trained on handcrafted features such as Mel-Frequency Cepstral Coefficients (MFCCs), spectrograms, or chromograms (Asuai et al., 2025). While these models have achieved moderate success, their performance is highly sensitive to hyperparameter configuration (e.g., learning rate, dropout rate, layer size) (Chowdhury et al., 2022; Liao et al., 2022). Improper parameter tuning often leads to overfitting, slow convergence, or suboptimal detection accuracy (Kadhim et al., 2023). Moreover, CNN-based architectures tend to focus on local features rather than capturing the long-term temporal dependencies intrinsic to speech, limiting their effectiveness against highly coherent synthetic audio (Kim et al., 2019).

To overcome these limitations, hyperparameter optimization has emerged as a critical step in improving model robustness and generalization (Iqbal et al., 2022). Conventional grid or random search methods, however, are computationally expensive and inefficient in high-dimensional search spaces typical of deep learning (Bergstra & Bengio, 2012). Recently, heuristic and meta-heuristic algorithms—such as Genetic Algorithm (GA) (Holland, 1975), Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Harris Hawks Optimization (HHO) (Heidari et al., 2019), Differential Evolution (DE) (Storn & Price, 1997), and Jaya (Pandey, 2016)—have demonstrated remarkable capability in exploring large, nonlinear parameter spaces efficiently (Jabbari Arfaee et al., 2011). These algorithms imitate natural or social phenomena (e.g., evolution, swarm behavior, pollination) to converge toward near-optimal solutions without exhaustive search (Beheshti & Shamsuddin, 2013). When integrated with deep learning architectures, they can automatically discover hyperparameter configurations that maximize detection accuracy and stability (Xiao et al., 2020).

Several prior studies have explored Deepfake voice detection using deep learning methods. Almutairi and Elgibreen (2022) reviewed modern detection methods and reported CNN-based accuracies between 85.99% and 94.33% across datasets such as ASVspoof 2019 and AR-DAD (Almutairi & Elgibreen, 2022). Dixit et al. (2023) similarly evaluated CNN and BiLSTM architectures, achieving 94.33% and 91.00% accuracy on Arabic Diversity and H-Voice datasets, respectively (Dixit et al., 2023). Although these works established a foundation, few studies have systematically examined the effect of heuristic optimization on Deepfake detection performance (Cunha et al., 2024). Thus, a methodological gap exists between algorithmic optimization and application-specific accuracy improvement.

This study addresses that gap by integrating heuristic and meta-heuristic optimization algorithms with RNN and LSTM architectures to enhance Deepfake voice detection. Using the ADD 2022 dataset, the proposed approach performs hyperparameter optimization—covering learning rate, dropout, input size, and batch size—through iterative heuristic search. The primary objective is to evaluate the influence of these algorithms on model convergence speed and detection accuracy. By comparing the optimized and non-optimized models, the study provides empirical evidence that heuristic-based optimization significantly improves detection reliability while maintaining computational efficiency.

The remainder of this paper is organized as follows. Section 2 presents the materials, dataset characteristics, and applied methodologies. Section 3 details the experimental setup and performance results. Section 4 discusses the findings in light of existing literature and outlines directions for future research.

2. MATERIALS AND METHODS

2.1. Dataset and Preprocessing

The experiments were conducted using the ADD 2022 dataset, a widely used benchmark for voice Deepfake detection. The dataset contains 11,535 samples, including 8,275 synthetic (fake) and 3,260 genuine (real) speech recordings. Each audio file was converted into numerical feature representations through spectrogram and chromogram analysis, which capture both frequency and temporal characteristics of the signal. The resulting dataset was stored in CSV format for efficient computation and further preprocessing.

Before model training, data normalization was applied to ensure that all features had comparable scales and to accelerate convergence. No additional augmentation was performed to preserve the authenticity of Deepfake-to-real sample ratios. The dataset was split into 80% training and 20% testing subsets, and stratified sampling was used to maintain class balance. Each sample was labeled as 1 (FAKE) or 0 (REAL).

2.2 Deep Learning Models

Two deep learning architectures were implemented for the detection task: Recurrent Neural Network (RNN) (Williams & Zipser, 1989) and Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997). Both models were selected for their ability to process sequential data and capture temporal dependencies inherent in speech signals.

2.2.1 Recurrent Neural Network (RNN)

RNNs are designed to retain contextual information through feedback connections that allow information from previous time steps to influence current computations. This makes them suitable for modeling time-dependent and sequential data, such as speech, where contextual continuity is essential. However, conventional RNNs often struggle with vanishing or exploding gradients when learning long-term dependencies, which can limit their representational capacity (Zucchet & Orvieto, 2024).

2.2.2 Long Short-Term Memory (LSTM)

LSTM networks were introduced to overcome the limitations of standard RNNs by incorporating gating mechanisms that control information flow (Van Houdt et al., 2020). As illustrated in Figure 1, the LSTM cell includes a cell state that stores long-term information, and three gates—forget, input, and output—that regulate how information is stored, updated, and passed to subsequent layers. This structure enables LSTM networks to effectively model both short-term and long-term dependencies in sequential data, making them particularly robust for speech-based Deepfake detection (Muruganandham et al., 2025).

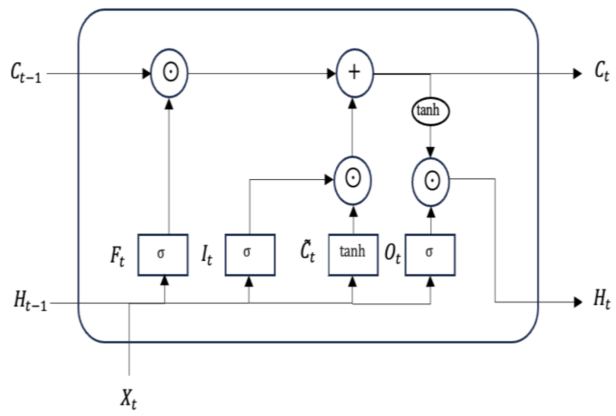

Figure 1. LSTM Structure

The models were implemented using Python 3.12 and Keras, and trained with the categorical cross-entropy loss function.

2.3 Baseline RNN and LSTM architectures

In the baseline configuration (i.e., without heuristic optimization), both the RNN and LSTM models shared the same overall architecture. The input layer received the spectrogram-based feature vectors extracted from the ADD 2022 dataset. This was followed by two recurrent layers: in the RNN model, these were standard RNN layers, whereas in the LSTM model, they were LSTM layers. Each recurrent layer contained a fixed number of neurons (128 units in the first layer and 64 units in the second layer), using the *tanh* activation function. The recurrent stack was followed by a fully connected dense layer, a dropout layer and, finally, a Softmax output layer with two neurons corresponding to the “REAL” and “FAKE” classes.

For both architectures, the baseline hyperparameters were kept fixed and manually selected: the learning rate was set to 0.0001, the dropout rate to 0.3–0.5, the input size to 32–64 features per time step, the batch size to 32, and the number of training epochs to 20, as summarized in Table 2. The Adam optimizer and categorical cross-entropy loss function were used throughout. This configuration serves as the non-optimized reference point against which the heuristically optimized models are compared in Section 3.

2.3 Heuristic Algorithms and Optimization Strategy

Heuristic algorithms are adaptive problem-solving methods inspired by natural or social processes. They provide approximate but efficient solutions to optimization problems where exhaustive search is computationally infeasible (Desale et al., 2015). In this study, heuristic and meta-heuristic algorithms were employed to optimize hyperparameters of deep learning models to achieve maximum classification accuracy.

Nine heuristic algorithms were examined, including:

- Evolutionary algorithms: Genetic Algorithm (GA) (Holland, 1992), Differential Evolution (DE) (Storn & Price, 1997)

- Swarm intelligence algorithms: Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995), Ant Lion Optimizer (ALO) (Mirjalili, 2015), Harris Hawks Optimization (HHO) (Heidari et al., 2019), Flower Pollination Algorithm (FPA) (Yang, 2012)

- Arithmetic and mathematical heuristics: Arithmetic Optimization Algorithm (AOA) (Abualigah et al., 2021), Jaya (Pandey, 2016), Cross-Entropy Method (CEM) (R. Rubinstein, 1999; R. Y. Rubinstein, 1997)

In this study, the objective of all heuristic algorithms is to maximize the classification accuracy of the Deepfake detection models. To enable the algorithms to perform minimization—which is the default operation in many meta-heuristics—the fitness value was defined as the complement of accuracy:

$$fitness = 1 - accuracy \qquad (1)$$

Because accuracy is computed as the proportion of correctly classified samples and therefore always lies within the range $[0,1]$, the corresponding fitness value also naturally falls within the interval $[0,1]$. A perfect classifier yields an accuracy of 1.0, resulting in a fitness value of 0 (the optimal minimum), whereas an accuracy of 0 corresponds to a fitness of 1 (the worst possible value). During optimization, each candidate hyperparameter set produces an accuracy value on the validation set, which is then converted into a fitness value using the above relation. The meta-heuristic algorithms iteratively update their populations to minimize this fitness value, thereby indirectly maximizing model accuracy.

The hyperparameter ranges in Table 1 were selected based on commonly used values in RNN/LSTM training and practical considerations from prior Deepfake and speech-processing studies. The learning rate interval (0.0001–0.1) covers the typical range required for stable recurrent network optimization. The dropout range (0.1–0.5) prevents both under- and over-regularization. Input sizes between 32–256 reflect standard spectrogram feature dimensions, balancing computational cost and information retention. Batch sizes (16–128) were chosen to ensure stable gradient updates without exceeding memory limits. These ranges provide a sufficiently broad but meaningful search space for the heuristic algorithms.

Each metaheuristic algorithm was executed 30 independent runs, and the best-performing configuration was reported. For each heuristic run, a population size of 50 and 100 iterations were used. The optimization objective was to maximize the accuracy of the Deepfake detection model on validation data after each iteration.

2.4. Integration Framework and Experimental Setup

The integration of heuristic optimization into deep learning training was conducted through a multi-stage workflow (Figure 2):

1. Initialization:
   The heuristic algorithm initializes a population of candidate hyperparameter sets.
2. Evaluation:
   Each candidate configuration is used to train the RNN or LSTM model for a fixed number of epochs (20). The validation accuracy is computed.
3. Selection and Update:
   The heuristic algorithm evaluates the fitness of each candidate and updates the population using its specific evolutionary or social rules (e.g., mutation, crossover, attraction, or cooperation).
4. Termination:
   The process repeats until the maximum number of iterations (100) is reached or convergence is achieved.

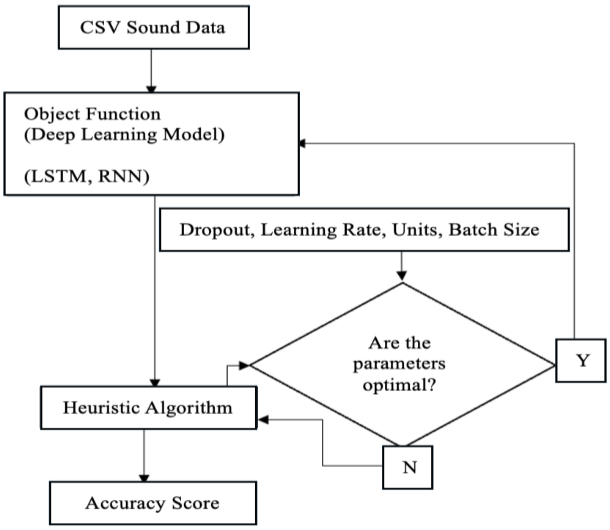

Figure 2. Workflow of this study

This optimization process dynamically adjusts model parameters at each iteration, resulting in improved convergence behavior and reduced overfitting. The framework was implemented in PyCharm IDE on a MacBook Pro (Apple M1 Pro, 16 GB RAM) using the pyMetaheuristic library for optimization routines.

2.5. Evaluation Metrics

The performance of all models was evaluated using accuracy as the primary metric, defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

where *TP*, *TN*, *FP*, and *FN* represent true positives, true negatives, false positives, and false negatives, respectively. In addition to accuracy, fitness values, convergence curves, and iteration-based performance plots were analyzed to assess the stability and efficiency of each heuristic method. To ensure fair comparison, all models were trained with identical data splits and computational resources. The best-performing model configurations were then reported in Section 3.

3. RESULTS

This section presents the experimental results obtained by integrating heuristic and meta-heuristic algorithms with the RNN and LSTM architectures for Deepfake voice detection. The optimization was conducted over four primary hyperparameters—learning rate, dropout rate, input size, and batch size—using 9 different heuristic algorithms. Each experiment consisted of 50 population members over 100 iterations, and model performance was primarily evaluated using validation accuracy.

The baseline Deepfake detection model, trained with fixed and manually selected hyperparameters, achieved an average accuracy of 77.1%. After applying heuristic optimization, all models exhibited substantial performance improvements, with several configurations exceeding 99% accuracy.

The relatively low baseline accuracy can be attributed to the non-optimized architectural and training parameters. Specifically, the fixed learning rate of 0.0001 and relatively high dropout values (0.3–0.5) limited convergence efficiency and increased the likelihood of suboptimal local minima. In addition, the small input sizes (32–64) imposed excessive dimensionality reduction on spectrogram features, reducing the model's ability to capture fine-grained temporal–frequency structures necessary for distinguishing genuine from Deepfake speech. The class imbalance in the ADD 2022 dataset (8,275 fake vs. 3,260 real) further contributed to reduced generalization capability, particularly in the minority class. By contrast, the heuristic optimization procedures were able to dynamically adjust key hyperparameters, allowing the models to converge more effectively and overcome the limitations present in the baseline configuration. The resulting performance outcomes are summarized in Table 2.

3.1 Optimization Results

Table 1 summarizes the range of hyperparameters explored during optimization.

Table 1. Hyperparameter Values

Hyperparameter	Search Range
Learning rate	0.0001 – 0.1
Dropout rate	0.1 – 0.5
Input size	32 – 256
Batch size	16 – 128

The best parameter sets identified by the heuristic algorithms are shown in Table 2, along with the corresponding accuracy and iteration number at which convergence occurred.

3.2 Performance Comparison

The results reveal a clear improvement in accuracy following heuristic optimization. Across all configurations:

- RNN models consistently outperformed LSTM models under the same optimization conditions, suggesting that recurrent state propagation provides a more stable basis for learning temporal dependencies in speech data.
- Among all tested algorithms, Differential Evolution (DE) yielded the highest accuracy for the LSTM model (97.16%), while Particle Swarm Optimization (PSO) achieved the best performance for the RNN model (99.03%).

Table 2. Optimized hyperparameters and performance result

Heuristic Algorithm	Model	Learning Rate	Dropout Rate	Input Size	Batch Size	Test Accuracy	Precision	AUC
DE	LSTM	0.0031	0.1178	182	113	0.9716	0.973	0.986
	RNN	0.0001	0.3481	256	49	0,9798	0.982	0.991
CEM	LSTM	0.0007	0.2290	161	79	0.8737	0.875	0.910
	RNN	0.0186	0.2919	148	87	0.7544	0.751	0.801
AOA	LSTM	0.0001	0.1	256	16	0.9176	0.911	0.944
	RNN	0.0001	0.1	256	16	0.9873	0.988	0.995
ALO	LSTM	0.0161	0.1	42	128	0.8356	0.822	0.882
	RNN	0.0089	0.1	39	124	0.8202	0.809	0.869
GA	LSTM	0.0011	0.3812	213	16	0.9766	0.974	0.988
	RNN	0.0001	0.1962	255	93	0.9750	0.971	0.987
Jaya	LSTM	0.0031	0.1349	70	117	0.9499	0.945	0.968
	RNN	0.0001	0.1	256	16	0,9858	0.986	0.994
FPA	LSTM	0.0098	0.3938	110	96	0.8765	0.866	0.912
	RNN	0.0001	0.5	256	64	0.9651	0.958	0.982
HHO	LSTM	0.0081	0.2220	116	128	0.8623	0.853	0.899
	RNN	0.1	0.5	122	64	0.7356	0.718	0.780
PSO	LSTM	0.0009	0.5	256	128	0.9669	0.961	0.983
	RNN	0.0001	0.1	256	16	0.9903	0.991	0.997
Baseline	LSTM	0.0001	0.3, 0.5	32, 64	32	0.7715	0.764	0.845
	RNN	0.0001	0.3, 0.5	32, 64	32	0.7503	0.734	0.771

- Optimization significantly reduced the variance in training accuracy across iterations, indicating improved convergence stability.
- Certain algorithms such as Jaya and GA achieved near-saturated accuracies within fewer than 15 iterations, demonstrating faster convergence than stochastic methods like CEM or ALO.

Overall, the integration of heuristic algorithms led to an average accuracy improvement of 18–22 percentage points relative to the baseline model. The results confirm that heuristic-based hyperparameter tuning can substantially enhance model generalization without increasing computational cost excessively.

3.3 Visual Analysis

The convergence behavior of each heuristic algorithm was visualized using iteration–accuracy plots (Figures 3). Each figure illustrates the model's progression toward the optimal solution, where lower fitness values correspond to higher accuracy. Across all visual analyses, algorithms such as DE, GA, and PSO displayed smooth, monotonic improvement, indicating effective exploration–exploitation balance. In contrast, methods like ALO and HHO showed higher oscillation, reflecting sensitivity to initial population diversity. Across all visual analyses, algorithms such as DE, GA, and PSO displayed smooth, monotonic improvement, indicating effective exploration–exploitation balance. In contrast, methods like ALO and HHO showed higher oscillation, reflecting sensitivity to initial population diversity.

The results demonstrate that combining deep learning with heuristic optimization yields substantial performance gains in Deepfake voice detection. Heuristic algorithms were able to automatically identify optimal learning configurations that would be infeasible to determine through manual tuning or grid search.

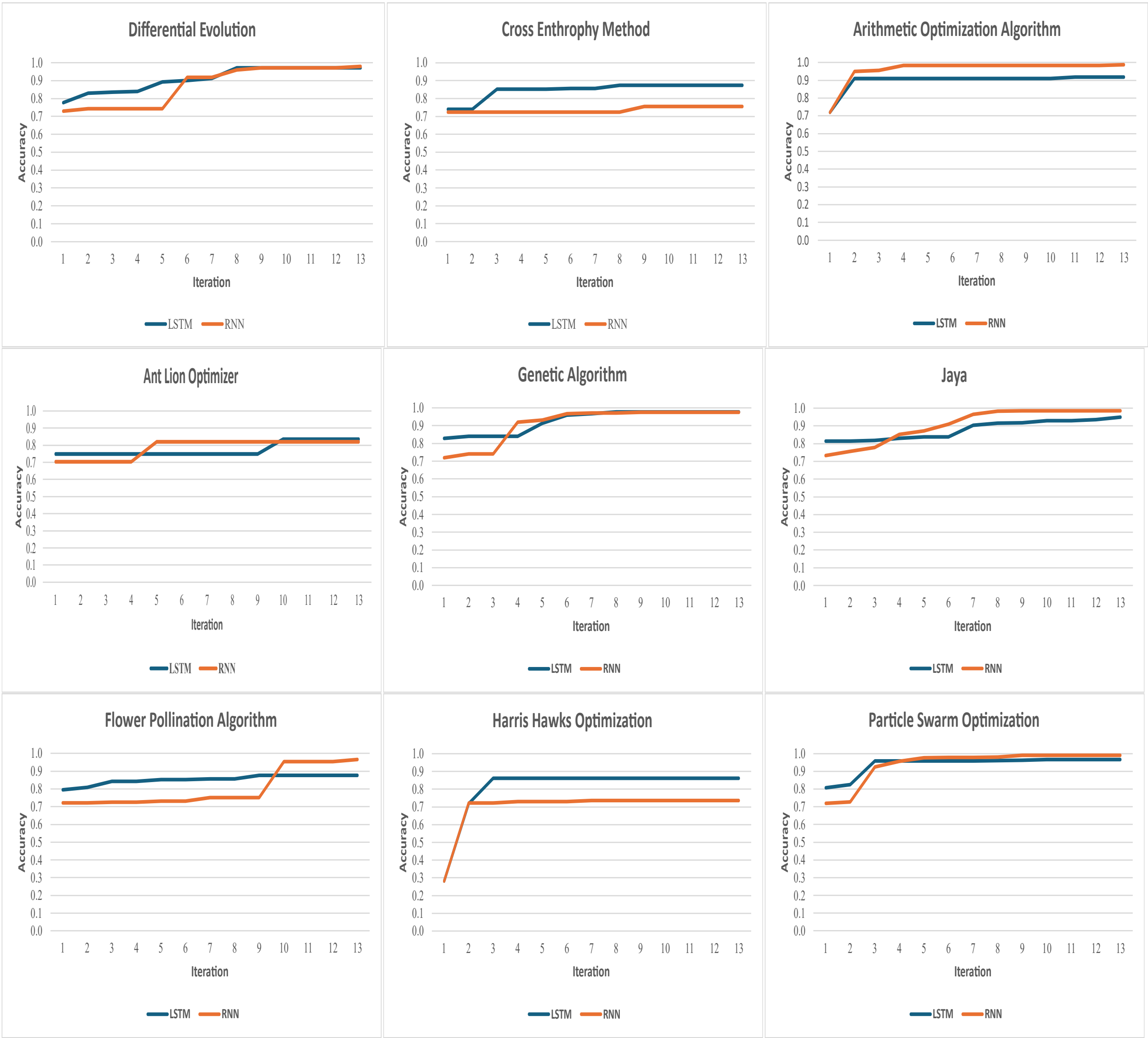

Figure 3. The convergence of each heuristic algorithm by LSTM and RNN

Specifically:

- **PSO** showed strong adaptability and convergence speed for RNNs due to its cooperative swarm dynamics, which efficiently adjusted learning rates and dropout ratios.
- **DE** proved particularly effective for LSTMs, likely because of its mutation–recombination mechanism that balances exploration and fine-tuning of sensitive hyperparameters such as learning rate.
- **GA** and **Jaya**, though slightly less accurate, required fewer iterations to stabilize, indicating computational efficiency suitable for large-scale deployment.

The consistent improvement across different algorithms confirms that heuristic-driven optimization can generalize across model architectures, enhancing both detection accuracy and training efficiency. Compared to previous studies that achieved accuracies between 85% and 94% using CNN or BiLSTM networks (Almutairi &

Elgibreen, 2022; Dixit et al., 2023), the optimized RNN–PSO configuration in this study reached 99.0% accuracy on the ADD 2022 dataset (Martin-Donas & Alvarez, 2022; Yi, Tao, et al., 2023). This marks one of the highest reported results for audio Deepfake detection in similar experimental conditions. In summary, the application of heuristic algorithms for hyperparameter optimization significantly enhanced the performance of RNN and LSTM models.

Key observations include:

- Optimization improved baseline accuracy from 77.1% to up to 99.0%.
- Differential Evolution and Particle Swarm Optimization achieved the most stable and accurate results.
- Convergence was typically achieved within the first 10–15 iterations for most algorithms.
- The integration framework proved computationally feasible and adaptable to multiple model architectures.

The next section discusses these outcomes in the broader context of Deepfake detection research, highlighting implications, limitations, and directions for future work.

4. DISCUSSION AND CONCLUSION

The findings of this study demonstrate that integrating heuristic and meta-heuristic optimization algorithms with deep learning architectures can substantially enhance the accuracy and reliability of Deepfake voice detection. Through systematic optimization of key hyperparameters—learning rate, dropout rate, input size, and batch size—the proposed framework achieved an average improvement of more than 20 percentage points over the baseline model, raising accuracy from 77.1 % to as high as 99.0 %. These results confirm the hypothesis that heuristic-driven parameter tuning provides a powerful alternative to manual or grid-based search methods, which are often computationally expensive and less adaptive.

The comparative results reveal clear distinctions among the tested optimization algorithms. Particle Swarm Optimization (PSO) achieved the highest performance with the RNN model (99.03 %), whereas Differential Evolution (DE) yielded the best accuracy for the LSTM architecture (97.16 %). This outcome can be interpreted through the inherent strengths of each algorithm: PSO’s social learning mechanism allows efficient exploration and exploitation of the parameter space, while DE’s mutation–recombination strategy effectively balances global and local search, leading to smoother convergence. Both algorithms successfully minimized fitness fluctuations, which is critical for achieving generalizable performance.

By contrast, algorithms such as Harris Hawks Optimization (HHO) and Ant Lion Optimizer (ALO) exhibited irregular convergence patterns and lower accuracy, suggesting sensitivity to population initialization and parameter control. The Genetic Algorithm (GA) and Jaya, however, demonstrated computational efficiency by converging rapidly with only marginally lower accuracy, which could be advantageous for time-constrained or resource-limited applications.

The observed performance improvements surpass those reported in earlier Deepfake voice detection studies that relied solely on deep learning architectures. Almutairi and Elgibreen (2022) evaluated multiple CNN-based frameworks on the ASVspoof 2019 (Asuai et al., 2025) and AR-DAD (Lataifeh & Elnagar, 2020) datasets and achieved accuracies between 85.99 % and 94.33 %. Similarly, Dixit et al. (2023) tested CNN and BiLSTM models on several benchmark datasets, reporting 91–99 % accuracy on Arabic Diversity and H-Voice data but with notable variability across conditions (Ballesteros et al., 2020). In contrast, the present study consistently achieved accuracies above 97 % for both RNN and LSTM models across all heuristic configurations, with the RNN–PSO combination reaching 99 %. This highlights the contribution of optimization rather than architectural change as the principal factor behind the improvement.

From a theoretical perspective, this study provides empirical evidence that meta-heuristic optimization can effectively adapt deep learning models to high-dimensional, nonlinear problems such as audio Deepfake detection. The integration of heuristic search enables automatic tuning of hyperparameters that govern model complexity and learning dynamics, yielding a better bias–variance balance. This reinforces the growing view that deep learning performance is as dependent on optimization strategy as on network architecture.

Practically, the results suggest that heuristic optimization can be implemented as a scalable, architecture-agnostic enhancement for existing Deepfake detection pipelines. Because the optimization process is model-independent, the same framework could be extended to CNN, Transformer, or hybrid attention-based architectures. The relatively low computational cost observed in this study—convergence typically within 10–15 iterations—demonstrates its feasibility for real-time or embedded detection systems.

Despite these encouraging outcomes, several limitations should be acknowledged. First, the study employed a single dataset (ADD 2022); cross-dataset validation is required to confirm generalizability to other languages, speakers, and recording conditions. Second, the heuristic algorithms were executed sequentially rather than in a hybrid or ensemble configuration. Combining multiple algorithms could further enhance convergence diversity and performance stability.

Future research should focus on extending the proposed optimization framework in several directions:

1. Cross-domain validation on multilingual and noisy datasets to evaluate robustness under real-world audio conditions.
2. Hybrid optimization approaches, combining, for example, DE-PSO or GA-Jaya hybrids, to leverage complementary exploration strengths.
3. Integration with advanced architectures such as Conformer or Wav2Vec-based models to capture higher-level speech representations.
4. Model interpretability and adversarial robustness analysis, investigating how optimized models respond to unseen synthetic patterns or adversarial manipulations.

In conclusion, this study introduced a heuristic-driven hyperparameter optimization framework for Deepfake voice detection using RNN and LSTM architectures. By systematically exploring parameter spaces through 9 heuristic algorithms, the framework achieved state-of-the-art accuracy while maintaining computational efficiency. The results affirm that optimization plays a decisive role in maximizing the potential of deep learning models for detecting synthetic speech. Beyond audio Deepfakes, the proposed methodology offers a generalizable path for performance enhancement across other media authentication and signal-forensics applications.

Funding

This study was conducted without financial support from any funding agency, commercial entity, or non-profit organization.

Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this research article.

REFERENCES

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113609. https://doi.org/10.1016/j.cma.2020.113609

Ahmad, J., Khan, H., Salman, W., Amin, M., Ali, Z., & Shokat, S. (2024). A Survey on Enhanced Approaches for Cyber Security Challenges Based on Deep Fake Technology in Computing Networks. *Spectrum of Engineering Sciences*, *2*(3), 133–149.

Al-Dhabi, Y., & Zhang, S. (2021). Deepfake Video Detection by Combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*, 236–241. https://doi.org/10.1109/CSAIEE54046.2021.9543264

Almutairi, Z., & Elgibreen, H. (2022). A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions. *Algorithms*, *15*(5), 155. https://doi.org/10.3390/a15050155

Amezaga, N., & Hajek, J. (2022). Availability of Voice Deepfake Technology and its Impact for Good and Evil. *Proceedings of the 23rd Annual Conference on Information Technology Education*, 23–28. https://doi.org/10.1145/3537674.3554742

Asuai, C., Arinomor, A., Atumah, C., Kowhoro, I., & Ogheneochuko, D. (2025). Hybrid CNN-LSTM Architectures for Deepfake Audio Detection Using Mel Frequency Cepstral Coefficients and Spectogram Analysis. *American Journal of Mathematical and Computer Modelling*, *10*(3), 98–109. https://doi.org/10.11648/j.ajmcm.20251003.12

Ballesteros, D. M., Rodriguez, Y., & Renza, D. (2020). A dataset of histograms of original and fake voice recordings (H-Voice). *Data in Brief*, *29*, 105331. https://doi.org/10.1016/j.dib.2020.105331

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A Review of Population-based Meta-Heuristic Algorithm. *International Journal of Advances in Soft Computing and Its Applications*, *5*(1), 1–35.

Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, *13*, 281–305. https://doi.org/10.1162/153244303322533223

Chintha, A., Thai, B., Sohrawardi, S. J., Bhatt, K., Hickerson, A., Wright, M., & Ptucha, R. (2020). Recurrent Convolutional Structures for Audio Spoof and Video Deepfake Detection. *IEEE Journal of Selected Topics in Signal Processing*, *14*(5), 1024–1037. https://doi.org/10.1109/JSTSP.2020.2999185

Chowdhury, A. A., Das, A., Hoque, K. K. S., & Karmaker, D. (2022). A Comparative Study of Hyperparameter Optimization Techniques for Deep Learning. *Proceedings of International Joint Conference on Advances in Computational Intelligence*, 509–521. https://doi.org/10.1007/978-981-19-0332-8_38

Cunha, L., Zhang, L., Sowan, B., Lim, C. P., & Kong, Y. (2024). Video deepfake detection using Particle Swarm Optimization improved deep neural networks. *Neural Computing and Applications*, *36*(15), 8417–8453. https://doi.org/10.1007/s00521-024-09536-x

Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey. *International Journal of Computer Engineering in Research Trends*, *2*(5), 296–304. http://www.ijcert.org

Dixit, A., Kaur, N., & Kingra, S. (2023). Review of audio deepfake detection techniques: Issues and prospects. *Expert Systems*, *40*(8), 1–19. https://doi.org/10.1111/exsy.13322

Frank, J., & Schönherr, L. (2021). *WaveFake: A Data Set to Facilitate Audio Deepfake Detection*. *NeurIPS*. http://arxiv.org/abs/2111.02813

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology, control, and artificial intelligence.* https://doi.org/10.1086/418447

Holland, J. H. (1992). Genetic Alghorithms. *Scientific American*, *267*(1), 66–73.

Iqbal, S., Qureshi, A. N., Ullah, A., Li, J., & Mahmood, T. (2022). Improving the Robustness and Quality of Biomedical CNN Models through Adaptive Hyperparameter Tuning. *Applied Sciences*, *12*(22), 11870. https://doi.org/10.3390/app122211870

Jabbari Arfaee, S., Zilles, S., & Holte, R. C. (2011). Learning heuristic functions for large state spaces. *Artificial Intelligence*, *175*(16–17), 2075–2098. https://doi.org/10.1016/j.artint.2011.08.001

Jain, E., & Singh, A. (2024). Deepfake Voice Detection Using Convolutional Neural Networks: A Comprehensive Approach to Identifying Synthetic Audio. *2024 International Conference on Communication, Control, and Intelligent Systems (CCIS)*, 1–5. https://doi.org/10.1109/CCIS63231.2024.10931997

Kadhim, Z. S., Abdullah, H. S., & Ghathwan, K. I. (2023). Automatically Avoiding Overfitting in Deep Neural Networks by Using Hyper-Parameters Optimization Methods. *International Journal of Online and Biomedical Engineering (IJOE)*, *19*(05), 146–162. https://doi.org/10.3991/ijoe.v19i05.38153

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*, 1942–1948. https://doi.org/10.1007/978-3-319-46173-1_2

Khanjani, Z., Watson, G., & Janeja, V. P. (2021). *How Deep Are the Fakes? Focusing on Audio Deepfake: A Survey*. 1–27. http://arxiv.org/abs/2111.14203

Khochare, J., Joshi, C., Yenarkar, B., Suratkar, S., & Kazi, F. (2022). A Deep Learning Framework for Audio Deepfake Detection. *Arabian Journal for Science and Engineering*, *47*(3), 3447–3458. https://doi.org/10.1007/s13369-021-06297-w

Kim, T., Lee, J., & Nam, J. (2019). Comparison and Analysis of SampleCNN Architectures for Audio Classification. *IEEE Journal of Selected Topics in Signal Processing*, *13*(2), 285–297. https://doi.org/10.1109/JSTSP.2019.2909479

Lataifeh, M., & Elnagar, A. (2020). Ar-DAD: Arabic diversified audio dataset. *Data in Brief*, *33*, 106503. https://doi.org/10.1016/j.dib.2020.106503

Liao, L., Li, H., Shang, W., & Ma, L. (2022). An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks. *ACM Transactions on Software Engineering and Methodology*, *31*(3), 1–40. https://doi.org/10.1145/3506695

Martin-Donas, J. M., & Alvarez, A. (2022). The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, *2022-May*, 9241–9245. https://doi.org/10.1109/ICASSP43922.2022.9747768

Mcuba, M., Singh, A., Ikuesan, R. A., & Venter, H. (2023). The Effect of Deep Learning Methods on Deepfake Audio Detection for Digital Investigation. *Procedia Computer Science*, *219*(2022), 211–219. https://doi.org/10.1016/j.procs.2023.01.283

Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, *83*, 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010

Mubarak, R., Alsboui, T., Alshaikh, O., Inuwa-Dutse, I., Khan, S., & Parkinson, S. (2023). A Survey on the Detection and Impacts of Deepfakes in Visual, Audio, and Textual Formats. *IEEE Access*, *11*(December), 144497–144529. https://doi.org/10.1109/ACCESS.2023.3344653

Muruganandham, P., Thangasamy, G. R., Jayaraman, S., & Dharmarajan, R. (2025). LSTM autoencoder based parallel architecture for deepfake audio detection with dynamic residual encoding and feature fusion. *Scientific Reports*, *15*(1), 23514. https://doi.org/10.1038/s41598-025-08198-6

Pandey, H. M. (2016). Jaya a novel optimization algorithm: What, how and why? *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, 728–730. https://doi.org/10.1109/CONFLUENCE.2016.7508215

Rubinstein, R. (1999). The Cross-Entropy Method for Combinatorial and Continuous Optimization. *Methodology And Computing In Applied Probability*, *1*(2), 127–190. https://doi.org/10.1023/A:1010091220143

Rubinstein, R. Y. (1997). Optimization of computer simulation models with rare events. *European*

Journal of Operational Research, *99*(1), 89–112. https://doi.org/10.1016/S0377-2217(96)00385-2

Salih, A. O. M., Emam, A. H. M., Ahmed, A. B. G. E., Khalifa, M., Suliman, A., & Babiker, N. B. M. (2025). Deepfake Audio Detection in Voice Authentication: A Spectral and CNN-Based Comprehensive Review. *Engineering, Technology & Applied Science Research*, *15*(6), 29824–29832.

Shaaban, O. A., Yildirim, R., & Alguttar, A. A. (2023). Audio Deepfake Approaches. *IEEE Access*, *11*(October), 132652–132682. https://doi.org/10.1109/ACCESS.2023.3333866

Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Tariq, S., Jeon, S., & Woo, S. S. (2022). Am I a Real or Fake Celebrity ? Evaluating Face Recognition and Verification APIs under Deepfake Impersonation Attack. *Proceedings of the ACM Web Conference 2022*, 512–523. https://doi.org/10.1145/3485447.3512212

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Westerlund, M. (2019). The Emergence of Deepfake Technology: A Review. *Technology Innovation Management Review*, *9*(11), 39–52.

Williams, R. J., & Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, *1*(2), 270–280. https://doi.org/10.1162/neco.1989.1.2.270

Xiao, X., Yan, M., Basodi, S., Ji, C., & Pan, Y. (2020). Efficient Hyperparameter Optimization in Deep Learning Using a Variable Length Genetic Algorithm. *ArXiv:2006.12703v1*. http://arxiv.org/abs/2006.12703

Yang, X.-S. (2012). Flower Pollination Algorithm for Global Optimization. In *Unconventional Computation and Natural Computation: Vol. 7445 LNCS* (pp. 240–249). https://doi.org/10.1007/978-3-642-32894-7_27

Yi, J., Tao, J., Fu, R., Yan, X., Wang, C., Wang, T., Zhang, C. Y., Zhang, X., Zhao, Y., Ren, Y., Xu, L., Zhou, J., Gu, H., Wen, Z., Liang, S., Lian, Z., Nie, S., & Li, H. (2023). ADD 2023: the Second Audio Deepfake Detection Challenge. *CEUR Workshop Proceedings*, *3597*, 125–130. http://arxiv.org/abs/2305.13774

Yi, J., Wang, C., Tao, J., Zhang, X., Zhang, C. Y., & Zhao, Y. (2023). *Audio Deepfake Detection: A Survey*. *14*(8), 1–20. http://arxiv.org/abs/2308.14970

Zucchet, N., & Orvieto, A. (2024). Recurrent neural networks: vanishing and exploding gradients are not the end of the story. *Advances in Neural Information Processing Systems*, *37*, 139402–139443.