Optimal CNN Model for Obstructive Sleep Apnea Detection using Particle Swarm Optimization

Thanh-Huong Tran
VNU University of Engineering and Technology, Hanoi, Vietnam
thanhhuong792000@gmail.com

Phuong Anh Nguyen
Ecole de Technologie Superieure, Montreal, QC, Canada
phuong-anh.nguyen@etsmtl.ca

Le Anh Ngoc
Swinburne Vietnam, FPT University, Hanoi, Vietnam
ngocla2@fe.edu.vn

Duc-Tan Tran
Faculty of Electrical and Electronic Engineering, Phenikaa University, Hanoi, Vietnam
tan.tranduc@phenikaa-uni.edu.vn

Minh Trien Pham
VNU University of Engineering and Technology, Hanoi, Vietnam
trienpm@vnu.edu.vn (corresponding author)

Received: 2 October 2024 | Revised: 12 November 2024 | Accepted: 20 November 2024

 DOI: https://doi.org/10.48084/etasr.9154

ABSTRACT

Obstructive Sleep Apnea (OSA) is a prevalent sleep disorder with significant health risks. It is characterized by the narrowing of the upper airway during sleep, leading to vibrations in the airway structures and the production of snoring sounds. Recently, Convolutional Neural Networks (CNNs) have been leveraged to extract meaningful features from snoring sound data, enabling early and accurate detection of OSA. The effectiveness of these neural network optimizations depends on the starting values of the model, the gradient algorithm used, and the complexity of the problem. This study introduces an improved Particle Swarm Optimization (PSO) strategy that linearly adjusts the learning rate coefficient to enhance accuracy and convergence speed. Our approach was evaluated on a collected and pre-processed dataset based on the PSG-Audio database. Experimental results demonstrate that our method significantly outperforms the conventional optimization algorithm and existing PSO techniques, achieving a remarkable accuracy of 99.1%. These findings confirm the potential of our optimized model for OSA detection.

Keywords-OSA detection; CNN; hyperparameter optimization; PSO

I. INTRODUCTION

Obstructive Sleep Apnea (OSA) is a sleep disorder that can lead to several significant medical problems. In OSA patients, the upper airway repeatedly collapses during sleep, resulting in either a complete (apnea) or partial (hypopnea) interruption of airflow. These obstructions can lead to oxygen desaturation in the blood and frequent sleep arousals [1]. Population-based studies estimate that approximately 200 million people suffer from this disorder [2]. However, up to 80% of the patients might be unaware that they possess OSA. This implies that early OSA diagnosis and its associated risks is a critical health issue today.

Polysomnography (PSG) remains the gold standard for objectively diagnosing and monitoring OSA, necessitating overnight sleep laboratory observation under specialist supervision [3]. The complex and uncomfortable nature of PSG has limited its widespread clinical use. To address these limitations, researchers have explored single-channel signal approaches, such as analyzing respiratory sounds [4], SaQ2 scores [5], and ECG data [6], for OSA diagnosis. These methods aim to reduce cost and improve accessibility.

Previous studies have recommended the use of a single-channel signal for the diagnosis of OSA. Snoring is the most common nocturnal symptom of OSA and is almost always present in OSA cases among the above symptoms [7]. It occurs due to vibrations in the upper airway structures caused by obstructed breathing during sleep. These vibrations are influenced by the physical and geometric properties of the upper airway, which are determined by physiological and anatomical factors. Consequently, snoring sounds are likely to contain crucial information about the dynamic characteristics of the upper airway. Therefore, snoring was chosen as the input for OSA diagnosis in this study.

Numerous studies have investigated the detection of OSA based on acoustic aspects of snoring signals. Recent studies [8-10] suggest that Convolutional Neural Networks (CNNs) outperform traditional machine-learning models as they can easily map nonlinear features. While several CNNs have been developed for OSA detection, inefficient network structures and complicated manual hyperparameter tuning hinder their practical application. Traditional optimization methods such as grid search [11] are computationally intensive and may miss optimal hyperparameter combinations. Random search [12] improves on this, but can suffer from decreased performance as the number of variables grows. Bayesian optimization [13] offers efficiency by using probabilistic models to strategically select hyperparameters, reducing iterations needed to find the global optimum. Evolutionary algorithms [14], which mimic natural evolutionary processes, effectively handle complex and high-dimensional optimization challenges. Due to their ability to progressively enhance model performance through mechanisms like mutation and crossover, evolutionary algorithms have become a leading research trend in CNN hyperparameter optimization [15].

In this study, we propose a hyperparameter tuning method using Particle Swarm Optimization (PSO) to optimize the architecture of CNN. PSO is an algorithm designed to emulate social behavior. It is a population-based heuristic method that models and represents the movements of organisms within bird flocks or fish schools [16]. PSO achieves robust and rapid global optimization by utilizing straightforward heuristic operations, which reduce computational costs and efficiently conserve computing resources. In this paper, we optimize several hyperparameters of a base CNN model inspired by LeNet5 for OSA detection, including the number of kernels, kernel size, activation function for the Conv2D layer, pooling method, number of dense units, and the activation function of the fully connected layer. Utilizing PSO, the optimized model demonstrates a 4% improvement over the base model. While this is not the first application of PSO for such optimization, it showcases promising performance. The dataset employed in our experiments is pre-processed from the PSG-Audio dataset [17] from the Science Databank.

II. THE PROPOSED METHOD

A. Overview

Figure 1 shows the overall workflow of the proposed method for OSA detection using the PSO-based CNN model. The model learns from feature-extracted data pre-processed by the Mel Frequency Cepstral Coefficients (MFCC) [18] or the Discrete Wavelet Transform (DWT) [19] methods. MFCC analyzes the Mel frequency components, while DWT examines signal details at various scales. Next, the hyperparameters that form the CNNs model are optimized using the PSO algorithm. We pre-define the search space, including the range values of CNNs hyperparameters to be optimized. Thus, a CNN is applied to each particle, and then the position, velocity, personal best (p_{best}), and global best (g_{best}) are updated according to the fitness value achieved. This procedure is repeated a certain number of times. Finally, the parameters of the global best particle are output as the optimal parameters. The loss function used to train the CNN is categorical cross-entropy, and the evaluation metric is accuracy.

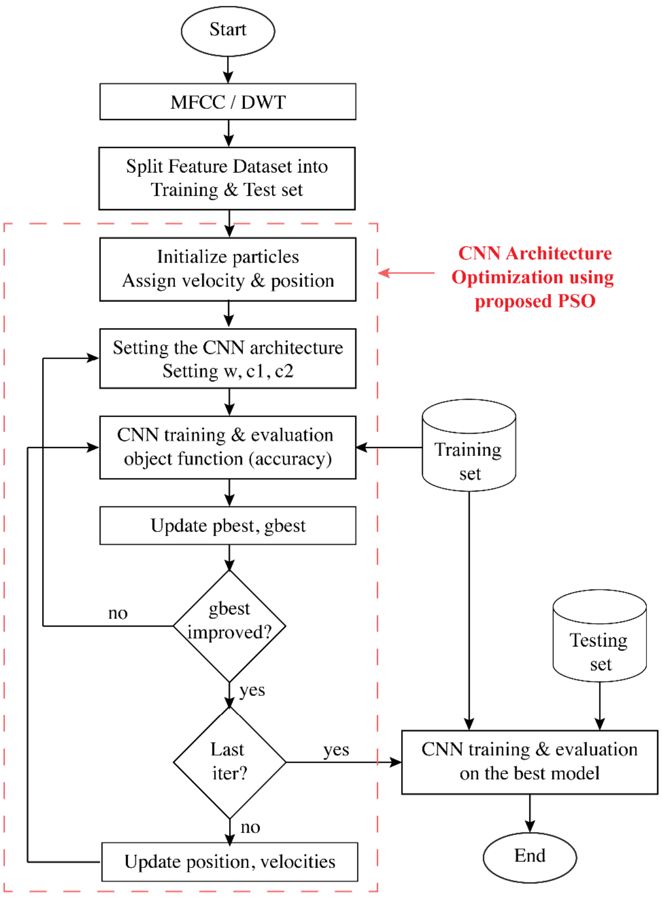

Fig. 1. Overall flowchart of the proposed method.

B. CNN Networks for OSA Detection

In this study, a CNN architecture was designed as the base network for OSA detection. The model was inspired by LeNet-5 [20], the most basic architecture to capture the difference between audio features like MFCC or DWT. The main structure of the base model is shown in Figure 2 below.

The input layer, also the first layer in the network, receives features extracted from raw audio using methods such as MFCC or DWT. Accordingly, the size of the input layer corresponds to the size of the extracted feature data. The configuration then starts with two convolutional layers, each with 32 filters of size 3×3. "Same" padding preserves image size in the initial layer. The next layer uses "valid" padding, refining feature extraction. This dual approach balances dimension retention and detail enhancement in the network's architecture.

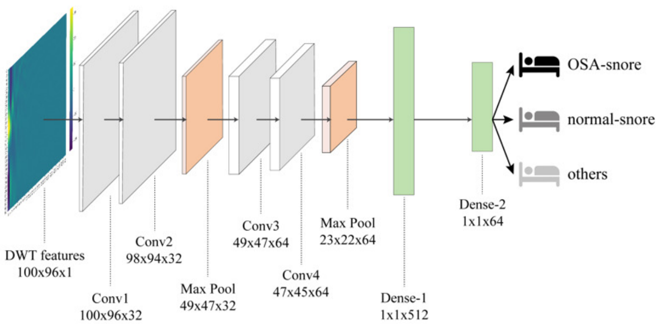

Fig. 2. The base CNN architecture network.

Max-pooling layers are applied to reduce the image size and add translational invariance. Dropout layers with probabilities of 0.25 and 0.50 mitigate overfitting by design. After these pre-processing steps, the network converts the data into a one-dimensional vector. This vector is then input into fully connected layers. The model includes two dense layers of 512 and 64 units, respectively, using ReLU activation to introduce non-linearity. The architecture culminates with a single output unit that employs sigmoid activation, producing a value between 0 and 1. This setup is ideal for multi-label classification tasks where each input may simultaneously belong to multiple classes.

C. Particle Swarm Optimization for CNN Hyperparameter Optimization

Instead of manually selecting the required hyperparameters, this study utilized the PSO algorithm to determine the optimal hyperparameters and fine-tune parameters for the OSA detection task. PSO is a computational optimization method modeled after the social behavior exhibited by flocks of birds during foraging. In this framework, each bird is represented as a "particle" (i.e., a potential solution) that navigates a multi-dimensional search space, adjusting its position and velocity based on the collective experiences of the entire swarm [21]. The procedure is repeated iteratively until the global optimal solution is achieved or the termination criteria are satisfied. The algorithm updates the particles' positions using (1) and adjusts their velocities according to (2). In (1), $x_i(t)$ denotes the position of particle x at time t during the search process. In the subsequent iteration, the particle's position is updated by adding the velocity $v_i(t+1)$.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (1)$$

In (2), v_i denotes the velocity of the particle i. The inertia weight w regulates the particle's velocity at time t, c_1 and c_2 are the social and cognitive coefficients that control the refinement of the particle's search and that of the entire swarm. The position p_{best} is the particle's optimal position, whereas g_{best} is the best position identified by the entire swarm at time t. The update rule defined by the equation ensures that the swarm's search space progressively converges toward the optimal values.

$$v_i(t+1) = wv_i(t) + c_1r_1\big(p_{best}(t) - x_i(t)\big) + c_2r_2\big(g_{best} - x_i(t)\big) \quad (2)$$

Each particle within the swarm is assessed using the fitness function. In this study, the fitness function employed to assess each particle's fitness is the model's accuracy in screening OSA. Accuracy is the metric measured by the proportion of correct predictions relative to the total number of predictions made. The overall process of the optimization implementation is depicted in Figure 1 above. Firstly, each particle is denoted in an order set x, where each element is denoted $x_i$representing a different configuration of hyperparameters that need to be tuned as shown in Table I, which are randomly initialized using a uniform distribution.

TABLE I. CUSTOM MODEL HYPERPARAMETERS NEED TO BE OPTIMIZED.

Dimension	Hyperparameters	Search Range	Base Mode
x_1	Number of first-layer convolution kernels	[1-128]	32
x_2	Size of first layer convolution kernels	[3×3, 5×5, 7×7]	3×3
x_3	Type of first layer activation function	[Sigmoid, ReLU, Tanh]	ReLU
x_4	Number of second layer convolution kernels	[1-128]	32
x_5	Size of second layer convolution kernels	[3×3, 5×5, 7×7]	3×3
x_6	Type of second layer activation function	[Sigmoid, ReLU, Tanh]	ReLU
x_7	Type of third pooling layer	[max.pooling, avg. pooling]	max.pooling
x_8	Number of fourth layer convolution kernels	[1-128]	64
x_9	Size of fourth layer convolution kernels	[3×3, 5×5, 7×7]	3×3
x_{10}	Type of fourth layer activation function	[Sigmoid, ReLU, Tanh]	ReLU
x_{11}	Number of fifth layer convolution kernels	[1-128]	64
x_{12}	Size of fifth layer convolution kernels	[3×3, 5×5, 7×7]	3×3
x_{13}	Type of fifth layer activation function	[Sigmoid, ReLU, Tanh]	ReLU
x_{14}	Type of sixth pooling layer	[max.pooling, avg. pooling]	max.pooling
x_{15}	Number of neurons in the seventh layer	[1-128]	128
x_{16}	Type of seventh layer activation function	[Sigmoid, ReLU, Tanh]	ReLU
x_{17}	Number of neurons in the eighth layer	[1-128]	64
x_{18}	Type of eighth layer activation function	[Sigmoid, ReLU, Tanh]	ReLU

As shown in Table I, the hyperparameters critical to our CNN architecture include several key aspects: the parameters of the convolutional layers, such as the number and size of filters, and the parameters of the fully connected layers, specifically the number of neurons. The accompanying table details the search ranges and base model settings for these parameters, demonstrating a structured approach to their optimization. For example, the first convolutional layer's filters range from 1 to 128, with a base model starting at 32, and the activation functions considered include Sigmoid, ReLU, and Tanh across various layers.

Next, for each particle in the swarm, the fitness value is computed by training and validating the CNN model using the hyperparameter represented by its position x. If the current position of a particle has a better fitness than its personal best position $p_{best}(t)$, update $p_{best}(t)$ with the current position. If any personal best position has a better fitness than the current global best position g_{best}, update g_{best} with $p_{best}(t)$. Update its velocity and position using (1) and (2). Finally, the process is completed when the maximum iteration is reached as an end condition, returning the g_{best} as the best set of hyperparameters. Several parameters were kept constant throughout the model training and validation stages of the optimization process.

```
Algorithm 1: PSO-CNN-Optimization
Input:
  Training set (X_train, y_train)
  Test set (X_test, y_test)
Output:
  Optimized CNN architecture and its
  accuracy on test set
Initialize PSO parameters:
  Set population size
  Set maximum iterations
  Set inertia weight w
  Set cognitive coefficient c_1
  Set social coefficient c_2
  Define accuracy as the objective
  function
Initialize particles:
  For each particle p in swarm
    Assign a random initial position
    p.position from the search space
    Set initial velocity p.velocity ← 0
    Set personal best position p.pbest ←
    p.position
    Set personal best score p.pbest_score ←
    −1000
  End For
  Set global best gbest ← None
  Set global best score gbest_score ← 0
PSO Optimization Loop:
  For each iteration i from 1 to max_it:
    For each particle p in the swarm:
      Train CNN with the model
      architecture defined by p.position
      Evaluate CNN on the validation set
      to obtain accuracy
      If accuracy > p.pbest_score:
        Update p.pbest ← p.position
        Update p.pbest_score ← accuracy
      End If
      If accuracy > gbest_score:
        Update gbest ← p.position
      Update gbest_score ← accuracy
      End If
    End For
    If gbest has not improved in the last
    iteration:
      Adjust w using a decay formula
    End If
    For each particle p in the swarm:
      Generate random factors r_1,r_2 ∈ [0,1]
      Update velocity using the PSO
      formula
      Update position: p.position ←
      p.position + p.velocity
    End For
  End For
Final Evaluation:
  Train the CNN model defined by gbest on
  Training set
  Evaluate the CNN on the Test set
Output:
  Return the optimized CNN architecture
  gbest and its test accuracy
```

During the training phase of the CNN model, the Stochastic Gradient Descent (SGD) optimizer is employed with a learning rate of 1e-4 to minimize the risk of overfitting. The swarm size is fixed at 30, with a maximum of 100 iterations. The inertia weight is updated within the range of 0.9 to 0.4, but only if there is no improvement in the fitness function value at the global best position. The inertia weight, denoted by the symbol w , plays a critical role in improving the accuracy and convergence speed of the PSO algorithm. Typically, standard PSO employs a linearly decreasing inertia weight strategy. However, in practical applications, finding optimal solutions frequently demonstrates nonlinear behavior [22]. Therefore, in this paper, an adaptive decreasing inertia weight strategy is proposed to update the inertia weights. The mathematical expression of the update condition is as follows:

$$w(t)=\begin{cases} w(t-1),\ if\ f\left(g_{best}(t)\right)\geq f\left(g_{best}(t-1)\right) \\ w(t-1)\cdot\frac{max_it-it}{max_it},\ othewise \end{cases} \quad (3)$$

where the variable max_it denotes the maximum number of iterations for the evolutionary process, while it represents the current iteration number. Besides the inertia weights, the learning factors c_1 and c_2 are crucial parameters influencing the performance of PSO. These two parameters correspond to the weights assigned to the individual cognition and the social cognition of the particles, respectively. Exploitation and exploration are key phases in the evolutionary process of PSO.

In the initial stages of evolution, it is critical to increase the population's diversity, while in the later stages, the focus should shift to improving the particles' exploration capabilities. Thus, in this study, these acceleration constants are adjusted dynamically; c_1 decreases from 2 to 0.5, while c_2 increases from 0.5 to 2. Finally, this optimization process will be complete when the maximum iteration is reached as an end condition.

III. EXPERIMENTS AND RESULTS

A. Dataset and Experimental Setting

In this paper, the PSO-based CNN model was evaluated using the dataset PSG-Audio in the Science Databank [17]. The collected sounds were then meticulously processed and segmented to create uniform 10-second clips, matching the format used for the other classes. This careful selection and processing of ambient sounds was essential in developing a diverse and realistic dataset, allowing the model to effectively distinguish between snoring sounds and unrelated background noise when classifying OSA-related events. This approach ensures that the dataset accurately represents sounds encountered in real-world sleep monitoring, enhancing the model's robustness and practical utility. A total of 15,000 sound audios, including 5,000 audio segments in each class, were chosen as the input dataset in our experiment. Then, the dataset was randomly divided into training sets, validation sets, and testing sets with proportions of 70%, 20%, and 10% of the total dataset, respectively, and was used for feature extraction for classification and optimization.

The experiments were conducted using Python 3.9 and TensorFlow, leveraging the robust ecosystem provided by TensorFlow for machine learning tasks. The proposed model utilized the PSO algorithm alongside the Hyperactive library to optimize the CNN architecture. This setup enabled efficient hyperparameter tuning and model training to maximize classification accuracy. The computational resources included an Intel® Xeon® CPU E5-2680 v4 and an NVIDIA TU102 (GeForce RTX 2080 Ti Rev. A) GPU, running on Ubuntu 22.04.4 LTS, ensuring a high-performance environment for extensive experiments. The PSO optimization was performed every 100 epochs, and the CNN was trained using the optimized hyperparameters to achieve the best performance. The performance of the optimized CNN model was evaluated by comparing it against the base model and other optimization methods, ensuring consistent experimental conditions to isolate the effects of the applied optimization techniques. The optimized model demonstrated significant improvements in accuracy, reduced overfitting, and better computational efficiency, validating the effectiveness of the PSO-based hyperparameter optimization approach.

B. CNN Model Evaluation Metrics

The accuracy metrics are calculated to evaluate the performance of the proposed method. The metric formula is shown in (3). Other metrics such as F1-score, precision, and recall are also calculated to ensure the optimized model's performance is robust and generalizable. Precision measures the ratio of TP to the sum of TP and FP, indicating how well the model identifies positive cases without false alarms. Recall calculates the ratio of TP to the sum of TP and FN, reflecting the model's ability to capture all actual positive cases. The F1-score, defined as the harmonic mean of precision and recall, provides a balanced assessment, especially in cases of class imbalance. These metrics' mathematical expressions are as follows:

$$\text{Accuracy} = \frac{\text{TP+ TN}}{\text{TP+ TN+ FP+ FN}} \quad (4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (6)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \quad (7)$$

where TP, TN, FP, and FN above are True Positive, True Negative, False Positive, and False Negative, respectively. In addition, performance graphs display accuracy and loss trends throughout training, visually depicting the model's learning progression. These tools offer crucial insights into a model's strengths and weaknesses.

C. PSO Performance Evaluation

This subsection examines the impact of the proposed method on PSO performance for optimizing CNN hyperparameters. The hyperparameters were fine-tuned every 100 epochs, and the network was trained based on these optimized settings. This study proposes a dynamic adjustment of inertia weight throughout the iterations to improve optimization, as shown in Figure 3. This approach allows flexible control over particle velocity, balancing global exploration and local exploitation, which helps avoid local optima and enhances convergence to the global optimum. The results indicate that the enhanced inertia weight significantly boosts optimization accuracy. Its adaptive nature allows for finer adjustments of model parameters, resulting in a robust architecture with superior performance across different test cases. This improvement highlights the value of dynamic parameter tuning in achieving optimal results. After running the PSO algorithm for approximately 8 hours, we obtained the optimal parameter table shown in Table II.

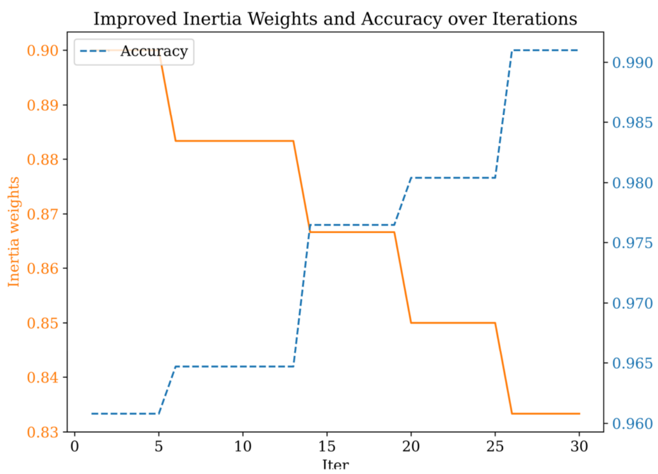

Fig. 3. Impact of improved inertia weight on diagnosis accuracy.

TABLE II. THE OPTIMAL HYPERPARAMETERS OF THE PROPOSED MODEL

Dimensions	The optimal value	Dimensions	The optimal value
x_1	28	x_{10}	ReLU
x_2	3×3	x_{11}	53
x_3	ReLU	x_{12}	7×7
x_4	28	x_{13}	ReLU
x_5	3×3	x_{14}	max.pooling
x_6	ReLU	x_{15}	4
x_7	max.pooling	x_{16}	ReLU
x_8	4	x_{17}	14
x_9	3×3	x_{18}	ReLU

Compared to the base model, the obtained optimal architecture reveals several key differences in its hyperparameters, reflecting distinct architectural choices that could enhance the model's performance.

D. Performances of PSO-based CNN Model for OSA Detection

To evaluate the performance of the optimized architecture model, we calculated four metrics as described in Section II. The results in Table III below demonstrate the enhancements achieved through optimization and provide a comparative analysis against the base model's performance.

TABLE III. PERFORMANCE OF OPTIMIZED CNN ARCHITECTURE AGAINST BASE CNN MODEL

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CNN + DWT	95.56	95.67	95.35	95.46
CNN + MFCC	96.41	96.54	96.48	96.47
Proposed PSO-CNN + DWT	97.77	98.02	97.87	97.92
Proposed PSO-CNN + MFCC	99.10	99.08	99.17	99.11

Table III shows that the optimized CNN architectures, particularly the PSO-CNN models using DWT and MFCC, achieve high performance across all metrics. The optimized CNN with the MFCC feature extraction achieves the best results, with an accuracy of 99.1%, precision of 99.08%, recall of 99.17%, and an F1-score of 99.11%. Meanwhile, the optimized CNN with the DWT method also performs well but with slightly lower scores: 97.77% accuracy, 98.02% precision, 97.87% recall, and an F1-score of 97.92%.

These results suggest that the MFCC-based model is more effective for optimizing the CNN's performance in this task. Therefore, in the following comparison and discussion section, we will focus on features extracted by MFCC to evaluate the proposed algorithm against existing studies. The two figures below show the accuracy and loss of the optimized model using the MFCC method, demonstrating the model's superior detection capability as concluded above. Figure 4 demonstrates the model's accuracy over training epochs, showing a rapid increase and stabilization around 0.9, indicating effective learning. Meanwhile, Figure 5 illustrates the loss, which decreases sharply and then fluctuates at a low value, confirming the model's convergence and generalization. Overall, these results demonstrate the model's effective detection capabilities.

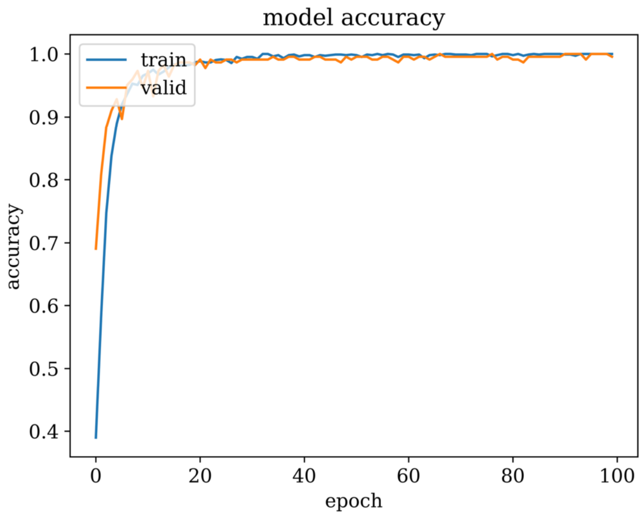

Fig. 4. The accuracy diagram of the proposed method.

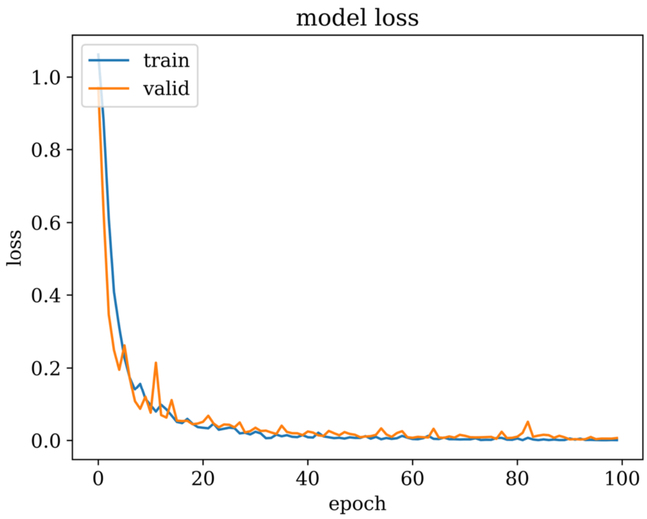

Fig. 5. The loss diagram of the proposed method.

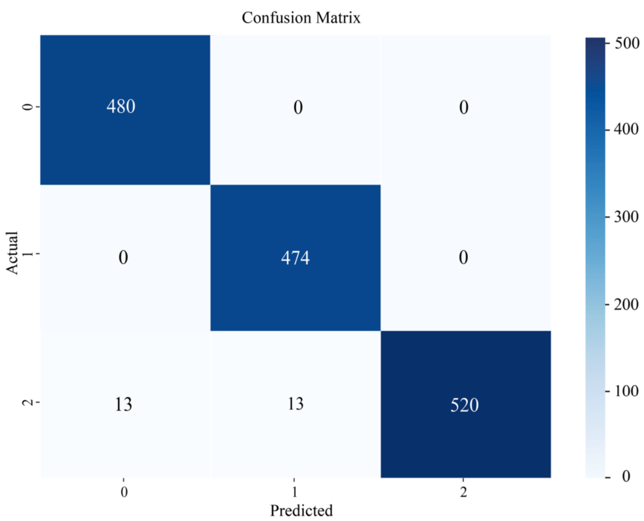

Fig. 6. Confusion matrix of the optimized model with the MFCC method.

Figure 6 presents the confusion matrix, showing the model's strong performance in classifying classes 0 and 1, with few errors in class 2. Overall, these results demonstrate the model's effective detection capabilities.

E. Performance Comparison with Other Methods

Accurately predicting the presence of OSA from specific snoring segments is crucial as it forms a solid basis for identifying patients potentially suffering from OSA. Therefore, the developed method was compared with other optimization methods that are commonly employed for detecting optimized CNN architecture. The overall performance of the collected dataset was assessed, and the accuracy of each method is reported in Table IV below.

TABLE IV. COMPARISON OF PERFORMANCES WITH OTHER METHOD

Method	Strategy	Complexity	Accuracy
Random Search CNN	start_point = base CNN, n_iter = 50	$O(n)$	96.87%
Bayes Opt CNN	start_point = base CNN, n_iter = 10	$O(n^3)$	96.67%
Base CNN	SGD optimizer	-	96.41%
PSO-CNN [23]	$w = 0.9$, $c_1 = c_2 = 2$	$O(pn)$	80.67%
LDWPSO-CNN [24]	w: 0.9 – 0.4, $c_1 = c_2 = 2$	$O(pn)$	98.67%
GA-CNN [25]	mutation = 0.1, crossover = 0.9	$O(pg)$	97.67%
Proposed	w_{max}= 0.9, w_{min} = 0.4, w updates as in (3) c_1: 2 - 0.5, c_2: 0.5 - 2	$O(pn)$	99.10%

Comparing the results with other methods reveals significant differences in algorithm performance. The Base CNN using the SGD optimizer achieves 95% accuracy without any parameter optimization strategy. The Random Search CNN, starting from the Base CNN and iterating 50 times, has a complexity of $O(n)$, and achieves an accuracy of 96.87%. The Bayes Opt CNN, with the same starting point and using just 10 iterations, has a higher complexity of $O(n^3)$ but achieves a similar accuracy of 96.67%.

In contrast, methods such as PSO-CNN, LDWPSO-CNN, and GA-CNN, which optimize using particle swarm strategies or genetic algorithms, have a complexity of $O(pn)$ or $O(pg)$ where p is the number of particles or swarm size, n is the number of iterations, and g is the number of generations. Despite the higher complexity, LDWPSO-CNN and GA-CNN achieve impressive accuracies of 98.67% and 97.67%, respectively, indicating better optimization than PSO-CNN, which has a lower accuracy of 80.67%. The proposed CNN algorithm, using the weight update in (3) and dynamic adjustments to $c1$ and $c2$ based on performance, achieves 99.1% accuracy. These results indicate that the proposed algorithm is superior in optimizing and improving model accuracy compared to other methods, which is crucial for ongoing research and projects in deep learning applications for healthcare purposes.

IV. CONCLUSION

This study introduced a novel methodology for optimizing CNN hyperparameters using an improved PSO algorithm with a dynamic inertia weight updating strategy. Applied to the detection of OSA based solely on snoring signals, our approach was evaluated on a comprehensive, collected, and pre-processed dataset. The optimized PSO-CNN model achieved a notable accuracy of 99.1%, outperforming the original CNN and other PSO-based methods with different parameter settings. This significant improvement underscores the effectiveness of our method in hyperparameter optimization and its potential for practical applications in OSA detection. While the results are promising, future research should focus on enhancing feature extraction methods and exploring additional optimization algorithms to further improve model performance. Validating the optimized model on diverse datasets and various biomedical signals will also be crucial in assessing its generalizability and practical utility for diagnosing OSA.

REFERENCES

[1] M. J. Sateia, "International classification of sleep disorders-third edition: highlights and modifications," *Chest*, vol. 146, no. 5, pp. 1387–1394, Nov. 2014, https://doi.org/10.1378/chest.14-0970.

[2] T. Young, M. Palta, J. Dempsey, J. Skatrud, S. Weber, and S. Badr, "The occurrence of sleep-disordered breathing among middle-aged adults," *The New England Journal of Medicine*, vol. 328, no. 17, pp. 1230–1235, Apr. 1993, https://doi.org/10.1056/NEJM199304293281704.

[3] C. Song, K. Liu, X. Zhang, L. Chen, and X. Xian, "An Obstructive Sleep Apnea Detection Approach Using a Discriminative Hidden Markov Model From ECG Signals," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1532–1542, Jul. 2016, https://doi.org/10.1109/TBME.2015.2498199.

[4] A. Azarbarzin and Z. M. K. Moussavi, "Automatic and Unsupervised Snore Sound Extraction From Respiratory Sound Signals," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 5, pp. 1156–1162, May 2011, https://doi.org/10.1109/TBME.2010.2061846.

[5] R. Hornero, D. Alvarez, D. Abasolo, F. del Campo, and C. Zamarron, "Utility of approximate entropy from overnight pulse oximetry data in the diagnosis of the obstructive sleep apnea syndrome," *IEEE Transactions on Bio-medical Engineering*, vol. 54, no. 1, pp. 107–113, Jan. 2007, https://doi.org/10.1109/TBME.2006.883821.

[6] N. Pombo, B. M. C. Silva, A. M. Pinho, and N. Garcia, "Classifier Precision Analysis for Sleep Apnea Detection Using ECG Signals," *IEEE Access*, vol. 8, pp. 200477–200485, 2020, https://doi.org/10.1109/ACCESS.2020.3036024.

[7] R. L. Owens, D. J. Eckert, S. Y. Yeh, and A. Malhotra, "Upper airway function in the pathogenesis of obstructive sleep apnea: a review of the current literature," *Current Opinion in Pulmonary Medicine*, vol. 14, no. 6, pp. 519–524, Nov. 2008, https://doi.org/10.1097/MCP.0b013e3283130f66.

[8] J. Zhang *et al.*, "Automatic Detection of Obstructive Sleep Apnea Events Using a Deep CNN-LSTM Model," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, 2021, Art. no. 5594733, https://doi.org/10.1155/2021/5594733.

[9] R. Hyder, S. Ghaffarzadegan, Z. Feng, J. H. L. Hansen, and T. Hasan, "Acoustic Scene Classification Using a CNN-SuperVector System Trained with Auditory and Spectrogram Image Features," presented at the *Interspeech 2017*, Stockholm, Sweden, Aug. 20–24, 2017, pp. 3073–3077, https://doi.org/10.21437/Interspeech.2017-431.

[10] A. K. Abasi, M. Aloqaily, and M. Guizani, "Optimization of CNN using modified Honey Badger Algorithm for Sleep Apnea detection," *Expert Systems with Applications*, vol. 229, Nov. 2023, Art. no. 120484, https://doi.org/10.1016/j.eswa.2023.120484.

[11] I. Priyadarshini and C. Cotton, "A novel LSTM–CNN–grid search-based deep neural network for sentiment analysis," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 13911–13932, Dec. 2021, https://doi.org/10.1007/s11227-021-03838-w.

[12] R. Andonie and A.-C. Florea, "Weighted Random Search for CNN Hyperparameter Optimization," *International Journal of Computers Communications & Control*, vol. 15, no. 2, Mar. 2020, https://doi.org/10.15837/ijccc.2020.2.3868.

[13] J. Gu *et al.*, "Bayesian Optimized 1-Bit CNNs," in *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 2019, pp. 4908–4916, https://doi.org/10.1109/ICCV.2019.00501.

[14] J. Fregoso, C. I. Gonzalez, and G. E. Martinez, "Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition," *Axioms*, vol. 10, no. 3, Sep. 2021, Art. no. 139, https://doi.org/10.3390/axioms10030139.

[15] A. A. K. Taher and S. M. Kadhim, "Improvement of genetic algorithm using artificial bee colony," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 2125–2133, Oct. 2020, https://doi.org/10.11591/eei.v9i5.2233.

[16] S. P. Singh and S. C. Sharma, "A Novel Energy Efficient Clustering Algorithm for Wireless Sensor Networks," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1775–1780, Aug. 2017, https://doi.org/10.48084/etasr.1277.

[17] G. Korompili *et al.*, "PSG-Audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies," *Scientific Data*, vol. 8, Aug. 2021, Art. no. 197, https://doi.org/10.1038/s41597-021-00977-w.

[18] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980, https://doi.org/10.1109/TASSP.1980.1163420.

[19] K. N. V. P. S. Rajesh, R. Dhuli, and T. S. Kumar, "Obstructive sleep apnea detection using discrete wavelet transform-based statistical features," *Computers in Biology and Medicine*, vol. 130, Mar. 2021, Art. no. 104199, https://doi.org/10.1016/j.compbiomed.2020.104199.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015, Art. no. 7553, https://doi.org/10.1038/nature14539.

[21] J. Kennedy, "Swarm Intelligence," in *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*, A. Y. Zomaya, Ed. Boston, MA, USA: Springer US, 2006, pp. 187–219, https://doi.org/10.1007/0-387-27705-6_6.

[22] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Transactions*, vol. 97, pp. 415–430, Feb. 2020, https://doi.org/10.1016/j.isatra.2019.08.018.

[23] T. Yamasaki, T. Honma, and K. Aizawa, "Efficient Optimization of Convolutional Neural Networks Using Particle Swarm Optimization," in *2017 IEEE Third International Conference on Multimedia Big Data*, Laguna Hills, CA, USA, 2017, pp. 70–73, https://doi.org/10.1109/BigMM.2017.69.

[24] T. Serizawa and H. Fujita, "Optimization of Convolutional Neural Network Using the Linearly Decreasing Weight Particle Swarm Optimization," in *36th Annual Conference of the Japanese Society for Artificial Intelligence*, Kyoto, Japan, 2022, https://doi.org/10.11517/pjsai.JSAI2022.0_2S4IS2b03.

[25] S. Lee, J. Kim, H. Kang, D. Y. Kang, and J. Park, "Genetic Algorithm Based Deep Learning Neural Network Structure and Hyperparameter Optimization," *Applied Sciences*, vol. 11, no. 2, Jan. 2021, Art. no. 744, https://doi.org/10.3390/app11020744.