Tech Science Press

WACPN: A Neural Network for Pneumonia Diagnosis

Shui-Hua Wang[1], Muhammad Attique Khan[2], Ziquan Zhu[1] and Yu-Dong Zhang[1],*

[1]School of Computing and Mathematical Sciences, University of Leicester, Leicester, LE1 7RH, UK
[2]Department of Computer Science, HITEC University Taxila, Taxila, Pakistan
*Corresponding Author: Yu-Dong Zhang. Email: yudongzhang@ieee.org

Abstract: Community-acquired pneumonia (CAP) is considered a sort of pneumonia developed outside hospitals and clinics. To diagnose community-acquired pneumonia (CAP) more efficiently, we proposed a novel neural network model. We introduce the 2-dimensional wavelet entropy (2d-WE) layer and an adaptive chaotic particle swarm optimization (ACP) algorithm to train the feed-forward neural network. The ACP uses adaptive inertia weight factor (AIWF) and Rossler attractor (RA) to improve the performance of standard particle swarm optimization. The final combined model is named WE-layer ACP-based network (WACPN), which attains a sensitivity of $91.87 \pm 1.37\%$, a specificity of $90.70 \pm 1.19\%$, a precision of $91.01 \pm 1.12\%$, an accuracy of $91.29 \pm 1.09\%$, F1 score of $91.43 \pm 1.09\%$, an MCC of $82.59 \pm 2.19\%$, and an FMI of $91.44 \pm 1.09\%$. The AUC of this WACPN model is 0.9577. We find that the maximum deposition level chosen as four can obtain the best result. Experiments demonstrate the effectiveness of both AIWF and RA. Finally, this proposed WACPN is efficient in diagnosing CAP and superior to six state-of-the-art models. Our model will be distributed to the cloud computing environment.

Keywords: Wavelet entropy; community-acquired pneumonia; neural network; adaptive inertia weight factor; rossler attractor; particle swarm optimization

1. Introduction

Community-acquired pneumonia (CAP) is considered a sort of pneumonia [1] developed outside hospitals, and clinics, along with infirmaries [2]. CAP may affect people of any age, but it is more prevalent in very young and elderly groups, which may need hospital treatment if they develop CAP [3]. Chest computed tomography (CCT) is a crucial way to help radiologists/physicians to diagnose CAP patients. Recently, automatic diagnosis models based on artificial intelligence (AI) have gained promising performances and attracted researchers' attention. For example, Heckerling, et al. [4] employed the genetic algorithm for neural networks to foresee CAP. This approach is shortened to the genetic algorithm for pneumonia (GAN). Afterward, Liu, et al. [5] proposed a computer-aided detection (CADe) model to uncover lung nodules in the CCT slides. Strehlitz, et al. [6] presented several prediction systems by means of support vector machines (SVMs) together with Monte Carlo cross-validation. Dong, et al. [7] proposed an improved quantum neural network (IQNN) for pneumonia image recognition. Ishimaru,

et al. [8] proposed a decision tree (DT) model to foresee the atypical pathogens of CAP. Zhou [9] introduced the cat swarm optimization (CSO) method to recognize CAP. Wang, et al. [10] proposed an advanced deep residual dense network for the image super-resolution problem. Wang, et al. [11] proposed a CFW-Net for X-ray based COVID-19 detection.

However, the above methods still have room to improve. Their recognition performances, for example, the accuracies, are no more than or barely above 91.0%. We analyze their models and believe the reason is their training algorithms. After comparing recent global optimization algorithms, we find that particle swarm optimization (PSO) is one of the most successful optimization algorithms, compared to otheroptimization algorithms such as artificial bee colony [12] and bat algorithm [13]. Hence, we use the framework in Zhou [9] but replace CSO with an improved PSO. In addition, we introduce the two-dimensional wavelet-entropy (2d-WE) layer, introduce an improved PSO method—adaptive chaotic PSO (ACP) [14], and combine it with a feed-forward neural network. The final combined model is named WE-layer ACP-based network (WACPN). The experiments show the effectiveness of this proposed WACPN model. In all, we exhibit three contributions:

(a) The 2d-WE layer is managed as the feature extractor.
(b) ACP is utilized for training the neural network to gain a robust classifier.
(c) The proposed WACPN is proven to give better results than six state-of-the-art models.

2 Dataset and Preprocessing

The dataset is described in Zhou [9], where we have 305 CAP images and 298 healthy control (HC) images. The detailed demographical information can be found in Ref. [9]. Assume the raw CCT dataset is signified as F_A, within which each image be signified as f_a, and the number of entire images of both classes is $|F| = 603$, we get $F_A = \{f_a(i), \ i = 1, \ 2, \ \cdots, \ |F|\}$. The size of each image can be obtained as:

$$h_{size}[f_a(i)] = W_0 \times H_0 \times 3, \tag{1}$$

where $(W_0, \ H_0)$ connotes the width and height of the image set F_A and $h_{size}(x)$ outputs the size of x. Here $W_0 = H_0 = 1024$. Figs. 1a and 1b depicts the schematic for preprocessing, which aims to grayscale the raw images, enhance their contrasts, cut the margins and texts, and resize the images.
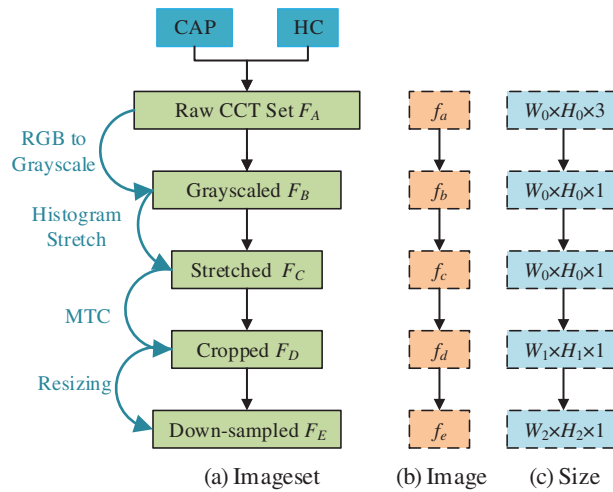

Figure 1: Diagram of preprocessing

Initially, the color CCT image set F_A is transformed into grayscale images by holding the luminance channel. The grayscaled CCT image set is symbolized as $F_B = \{f_b(i), \ i = 1, \ 2, \ \cdots, \ |F|\}$.

Second, we use histogram stretching (HS) on all images $F_B = \{f_b(i)\}$ to enhance the contrast. Take the i-th image $f_b(i)$ as a case, its image-wise minimum, and maximum grayscale value $f_b^l(i)$ and $f_b^h(i)$ are calculated as:

$$\begin{cases} f_b^l(i) = \min_{p_w=1}^{W_0} \min_{p_h=1}^{H_0} f_b(i|p_w, \ p_h) \\ f_b^h(i) = \max_{p_w=1}^{W_0} \max_{p_h=1}^{H_0} f_b(i|p_w, \ p_h) \end{cases}, \tag{2}$$

where $(p_w, \ p_h)$ are temporary variables signifying the index of width and height along with the image $f_b(i)$, respectively. The HSed image set $F_C = \{f_c(i), \ i = 1, \ \cdots, \ |F|\}$ can be determined as:

$$f_c(i) = \frac{f_b(i) - f_b^l(i)}{f_b^h(i) - f_b^l(i)} \tag{3}$$

Third, margin & text cropping (MTC) is implemented to eradicate (a) the checkup bed at the bottom zone, (b) the privacy-related scripts at the margin or corner zones, and (c) the ruler adjacent to the right-side and bottom zones. The MTCed image set $F_D = \{f_d(i), \ i = 1, \ \cdots, \ |F|\}$ can be determined as $f_d(i) = f_c(i; \ p_w, \ p_h), \ p_w \in [p_1 + 1, \ W_0 - p_2], \ p_h \in [p_3 + 1, \ H_0 - p_4]$, where $(p_1, \ p_2, \ p_3, \ p_4)$ stand for pixels to be cut from four directions (left, right, top, and bottom) with the unit of pixels. Note here the size of $f_d(i)$ is $h_{size}[f_d(i)] = W_1 \times H_1$. By means of straightforward maths calculation, we reckon that

$$\begin{cases} W_1 = W_0 - p_1 - p_2 \\ H_1 = H_0 - p_3 - p_4 \end{cases} \tag{4}$$

Lastly, each image in F_D is resized to the extent of $[W_2, \ H_2]$, acquiring the resized image set $F_E = \{f_e(i), \ i = 1, \ \cdots, \ |F|\}$ as $f_e(i) = h_{resize}[f_d(i); \ (W_2, \ H_2)]$, where h_{resize} signifies the resizing function.

Fig. 1c shows the extent of every raw image in F_A is $W_0 \times H_0 \times 3$, and that of the final preprocessed image in F_E is reduced to $W_2 \times H_2$. In addition, the value of data-compression ratio (DCR) z_1 is obtained as $z_1 = W_0 \times H_0 \times 3/(W_2 \times H_2) = 48$. The value of space-saving ratio (SSR) z_2 is calculated as $z_2 = 1 - W_2 \times H_2/(W_0 \times H_0 \times 3) = 97.92\%$. Fig. 2 shows two examples of the preprocessed image set. We use 10-fold cross-validation in our experiment.
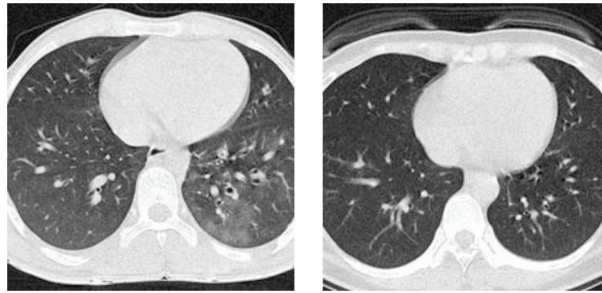

Figure 2: Examples of the preprocessed image set

3 Methodology of WACPN

3.1 Discrete Wavelet Transform

Tab. 1 enumerates all abbreviations and their associated meanings. The advantage of wavelet transform (WT) is that it holds both time/spatial and frequency information of the given signal/image. Nevertheless, the

discrete wavelet transform (DWT) is chosen to convert the raw signal $r(t)$ into the wavelet coefficient domain [15] in reality. Suppose the signal $r(t)$ is one-dimension, first, we define the continuous wavelet transform (CWT) $E_\gamma(s_a, s_t)$ of $r(t)$ as:

$$E_\gamma(s_a, s_t) = \int_{-\infty}^{\infty} r(t) \times \gamma(t|s_a, s_t)\mathrm{d}t, \tag{5}$$

in which E stands for the wavelet coefficient, γ the mother wavelet. $\gamma(t|s_a, s_t)$ is defined as:

$$\gamma(t|s_a, s_t) = \frac{1}{\sqrt{s_a}}\gamma\left(\frac{t - s_t}{s_a}\right), \ s_a > 0, \ s_t > 0, \tag{6}$$

where the s_a signifies the scale factor (SF) and s_t the translation factor (TF).

Table 1: Abbreviation and meaning

Abbreviation	Meaning	Abbreviation	Meaning
2d-DWT	two-dimensional DWT	HS	histogram stretch
2d-WE	two-dimensional wavelet entropy	IW	inertia weight
ACP	adaptive chaotic PSO	MCC	Matthews correlation coefficient
AF	activation function	MDL	maximum decomposition level
AI	artificial intelligence	MSD	mean and standard deviation
AIWF	adaptive IW factor	MSE	mean-squared error
AUC	area under the curve	MTC	margin & text cropping
BP	best position	OFNN	one-hidden-layer FNN
CAP	community-acquired pneumonia	PoP	position of particle
CC	cloud computing	PMF	probability mass function
CV	cross-validation	PSO	particle swarm optimization
CWT	continuous WT	RA	Rossler attractor
DCR	data-compression ratio	SB	subband
DWT	discrete WT	SF	scale factor
DV	discrete variable	SSR	space-saving ratio
EB	error bar	TF	translation factor
FMI	Fowlkes–Mallows index	WACPN	WE-layer ACP-based network
FNN	feed-forward neural network	WB	weight and bias
GAN	genetic algorithm for pneumonia	WT	wavelet transform
HC	healthy control	VoP	velocity of particle
HL	hidden layer		

Now, we deduct the definition of DWT from CWT. The Eq. (5) is discretized by substituting s_a and s_t with two discrete variables (DVs) c and v,

$$\begin{cases} s_a = 2^c \\ s_t = v \times 2^c \end{cases} \tag{7}$$

where c signifies the DV of the SF s_a, and v the DV of the TF s_t [16]. Moreover, the original signal $r(t)$ is a DV to $r(q)$, of which q signifies the DV of t. Like this, two subbands (SBs) can be calculated. The approximation SB $E^A(q|c, v)$ is determined as:

$$E^A(q|c, v) = S_D \left[\sum_o r(q) \times f_A^* \left(\frac{q - 2^c v}{2^c} \right) \right], \tag{8}$$

where $f_A(q)$ signifies the low-pass filter. S_D is the down-sampling operation. The detail SB $E^D(q|c, v)$ is determined as:

$$E^D(q|c, v) = S_D \left[\sum_o r(q) \times f_D^* \left(\frac{q - 2^c v}{2^c} \right) \right]. \tag{9}$$

where $f_D(q)$ signifies the high-pass filter.

3.2 2d-WE Layer

Suppose we handle a two-dimensional (2d) image Q; the 2d-DWT [17] is worked out by processing row-wise and column-wise 1d-DWT in succession [15]. Initially, the 2d-DWT operates on the original image Q. Later, four SBs (Z_1, O_1, F_1, A_1) are generated, where the subscript i means i-th level decomposition. Tab. 2 itemizes the description of four SBs. Note here MDL means the maximum decomposition level.

Table 2: Definition of four SBs

Symbol	Meaning	Symbol	Meaning
Q	Original image	F	Diagonal quadrant
Z	Horizontal quadrant	A	Approximate component quadrant
O	Vertical quadrant	M	MDL

Assuming h^{2d-DWT} signifies a 2D-DWT decomposition operation, we deduce

$$\begin{pmatrix} A_1 & Z_1 \\ O_1 & F_1 \end{pmatrix} = h^{2d-DWT}(Q). \tag{10}$$

The subsequent decompositions run as:

$$\begin{pmatrix} A_m & Z_m \\ O_m & F_m \end{pmatrix} = h^{2d-DWT}(A_{m-1}), \ m = 2, \ldots M, \tag{11}$$

where M is the MDL and m the current decomposition level [18].

The subband A_1 is further decomposed into four SBs (A_2, Z_2, O_2, F_2) at the 2nd level. The SB A_2 is later decomposed to (A_3, Z_3, O_3, F_3), and then SB A_3 is decomposed accordingly. Fig. 3 portrays a diagram of 5-level 2d-DWT, whose pseudocode is represented in Algorithm 1. This study chooses a M-level decomposition. The optimal value of M is found via trial-and-error approach [19] and related in Section 4.1.
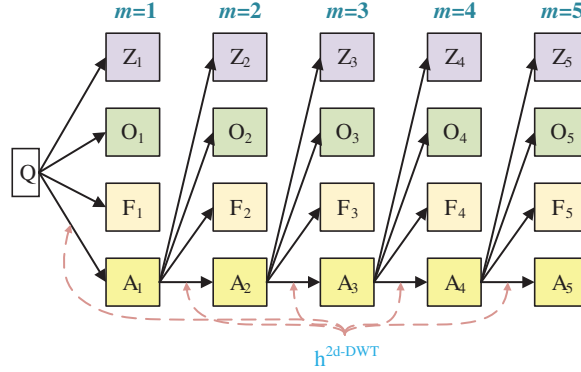
Figure 3: Diagram of a 2d-DWT ($M = 5$)

Algorithm 1: Pseudocode of 2d-DWT

Input	Image Q
Step 1	Decompose the image Q into four subbands $\begin{pmatrix} A_1 & Z_1 \\ O_1 & F_1 \end{pmatrix}$.
	for $m = 2\!:\!M$
Step 2	The approximation subband A_{m-1} is decomposed into four subbands. $$A_{m-1} \overset{h^{2d-DWT}}{\mapsto} \begin{pmatrix} A_m & Z_m \\ O_m & F_m \end{pmatrix}$$
	end
Output	Output the $(3M + 1)$ SBs $(A_M,\, Z_M,\, O_M,\, F_M,\, Z_{M-1},\, O_{M-1},\, F_{M-1},\, \ldots,\, Z_1,\, O_1,\, F_1)$.

The $(3M + 1)$ SBs $(A_M,\, Z_M,\, O_M,\, F_M,\, Z_{M-1},\, O_{M-1},\, F_{M-1},\, \ldots,\, Z_1,\, O_1,\, F_1)$ may contain redundant features. Here we use the db4 wavelet. To decrease the number of features, we employ two-dimensional wavelet entropy (2d-WE) layer. The pseudocode of 2d-WE is illustrated in Algorithm 2. For each SB s in the generated $(3M + 1)$ SBs, we imagine s to be a random DV S with H quantization values $(s_1,\, s_2,\, \ldots,\, s_h,\, \ldots,\, s_H)$. In the beginning, we gauge the matching probability mass function (PMF) $\boldsymbol{p}(s) = \{p_h(s)\}$.

$$p_h(s) = h_{\mathrm{Pr}}(\mathrm{S} == s_h),\ h = 1,\, 2,\, \cdots H, \tag{12}$$

where h_{Pr} signifies the probability function.

Second, the entropy of the PMF $\boldsymbol{p}(s)$ is calculated as $f_e(s)$:

$$f_e(s) = -\sum\nolimits_{h=1}^{H} p_h(s) \times \log p_h(s), \tag{13}$$

where f_e is the entropy function.

Lastly, the entropy values of the whole SBs are concatenated to grow a feature vector I.

$$I = \begin{bmatrix} f_e(A_M) & f_e(Z_M) & f_e(O_M) & f_e(F_M) \\ & f_e(Z_{M-1}) & f_e(O_{M-1}) & f_e(F_{M-1}) \\ & \cdots & \cdots & \cdots \\ & f_e(Z_1) & f_e(O_1) & f_e(F_1) \end{bmatrix}, \tag{14}$$

where the number of the features in I is $N_I = (3M + 1)$, which equals the number of SBs.

Algorithm 2: Pseudocode of 2d-WE

Input: $(3M + 1)$ SBs: $(A_M, Z_M, O_M, F_M, Z_{M-1}, O_{M-1}, F_{M-1}, \ldots, Z_1, O_1, F_1)$
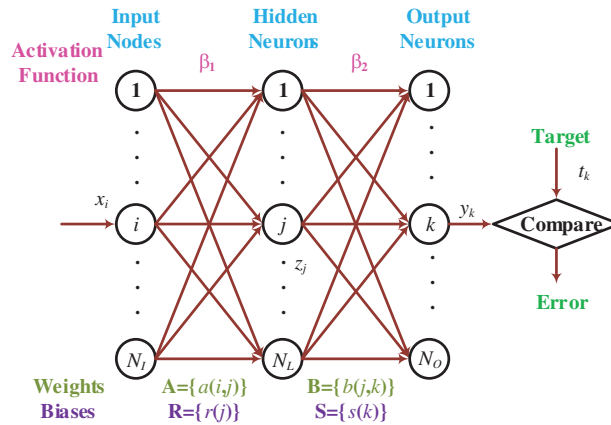
for $m = 1:3M + 1$

     | Choose the m-th SB s.

     | Compute PMF $p(s) = \{p_h(s)\}$. See Eq. (12).

     | Reckon entropy $f_e(s)$. See Eq. (13).

     | Record $I(n) \leftarrow f_e(s)$.

End

Output: The concatenated 2d-WE feature vector I with N_I features. See Eq. (14).

3.3 ACP Network

    The N_I features are thrown into a feed-forward neural network (FNN)—in which its inner connections do not make a loop. One-hidden-layer FNN (OFNN), represented in Fig. 4, is established due to the universal approximation theory. Assume (x, t) stands for a training case as: $x = [x_1, x_2, \ldots, x_i, \ldots, x_{N_I}]^T$ signifies the input feature vector with N_I-dimension, i denotes the neuron index at the input layer, t is the corresponding target label $t = [t_1, t_2, \ldots, t_k, \ldots, t_{N_O}]^T$, where N_O signifies the number of prediction categories and k the node index at the output layer. Assuming n is the case index and N the number of entire training cases, this study symbolizes the training case (x, t) as $\{x(n), t(n)|n = 1, , \ldots, N\}$. The training of the weights/biases (WBs) of OFNN is considered an optimization problem that minimizes the loss between the target t and the real output y. This study chooses the loss as the sum of the mean-squared error (MSE) E:

$$E = \sum_{n=1}^{N} \sum_{k=1}^{N_O} [y_k(n) - t_k(n)]^2. \tag{15}$$

Figure 4: Diagram of an FNN

    Assume β_2 is the activation function (AF) in the output layer, and (B, S) are the WBs of neurons that connect the hidden layer (HL) to the output layer. $B = \{b(j, k)\}$, $j = 1, \ldots, N_L$, $k = 1, \ldots N_O$, and $S = \{s(k)\}$, $k = 1, \ldots, N_O$. It is easy to reckon the output y_k as

$$y_k(n) = \beta_2 \left[\sum_{j=1}^{N_L} b(j, \ k)z_j(n) + s(k) \right], \tag{16}$$

where $z_j(n)$, $j = 1, \ \ldots, \ N_H$ signifies the output of j-th neuron in the HL. The description of $z_j(n)$ is

$$z_j(n) = \beta_1 \left[\sum_{i=1}^{N_I} a(i, \ j)x_i(n) + r(j) \right]. \tag{17}$$

where $A = \{a(i, \ j)\}$, $i = 1, \ \ldots, \ N_I$, $j = 1, \ \ldots, \ N_L$ and $R = \{r(j)\}$, $j = 1, \ \ldots, \ N_L$ are the WBs of the neurons that connect the input layer with the HL, and β_1 the AF linked to the HL.

The parameter training is an optimization problem that guides us to search for the optimal WB parametric vector $\theta = (A, \ B, \ R, \ S)$. The length of θ is the number of parameters we need to optimize and is calculated as $N_\theta = N_I \times N_L + N_L \times N_O + N_L + N_O$. The training algorithm we choose is adaptive chaotic PSO (ACP) [14].

Recap that two attributes (position x and velocity v) are linked with each particle p in the standard PSO algorithm. Those two attributes are defined as the position of the particle (PoP) and the velocity of the particle (VoP). In each epoch, the fitness function E is re-calculated for the entire particles $\{p\}$ in the swarm. The VoP v is re-evaluated by keeping track of the two best positions (BPs).

The first is the BP a particle p has traversed till now. It is dubbed *pBest* and symbolized as x_{pB}. The second is the BP that any neighbor of p has traversed till now. It is a neighborhood best and is named *nBest* and symbolized as x_{nB}.

If p takes the entire swarm as its neighborhood, the *nBest* turns to the global best and is for that reason named *gBest*. In standard PSO, the VoP v of particle p is updated as:

$$v \leftarrow \omega v + b_1 r_1(x_{PB} - x) + b_2 r_2(x_{nB} - x) \tag{18}$$

where ω signifies the inertia weight (IW) controlling the influence of the preceding velocity of the particle on its present one. b_1 and b_2 stand for two positive constants named acceleration coefficients. r_1 and r_2 mean two random numbers, uniformly distributed in the range of $[0,1]$. r_1 and r_2 are re-calculated whenever they occur. The PoP x of the particle p is updated as:

$$x \leftarrow x + v\Delta t \tag{19}$$

where Δt is the assumed time step and always equals 1 for simplicity.

The ACP algorithm proposed an adaptive IW factor (AIWF) strategy. It uses ω_{AIWF} to replace ω.

$$\omega_{AIWF} = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \times k \tag{20}$$

Here, ω_{max} signifies the maximum IW, ω_{min} the minimum IW, k_{max} the epoch once the IW goes to the final minimum IW, and k the present epoch.

Another improvement in ACP is upon the two random numbers $(r_1, \ r_2)$. In reality, $(r_1, \ r_2)$ are created by pseudo-random number generators (RNG), which cannot guarantee the optimization's ergodicity in solution space since they are pseudo-random. Rossler attractor (RA) is a good choice to calculate the random numbers $(r_1, \ r_2)$. RA equations are defined:

$$\begin{cases} \dfrac{dx}{dt} = -(y+z) \\ \dfrac{dy}{dt} = x + \delta_a y \\ \dfrac{dz}{dt} = \delta_b + xz - \delta_c z \end{cases} , \tag{21}$$

where δ_a, δ_b, and δ_c are inherent parameters of RA. We choose $\delta_a = 0.2$, $\delta_b = 0.4$, $\delta_c = 5.7$ via the trial-and-error method [20]. The corresponding curve is drawn in Fig. 5a. We agree $r_1 = x(t)$ and $r_2 = y(t)$ to implant the chaotic properties of RA into the two parameters (r_1, r_2) in standard PSO. The (x, y) plane of RA is displayed in Fig. 5b.
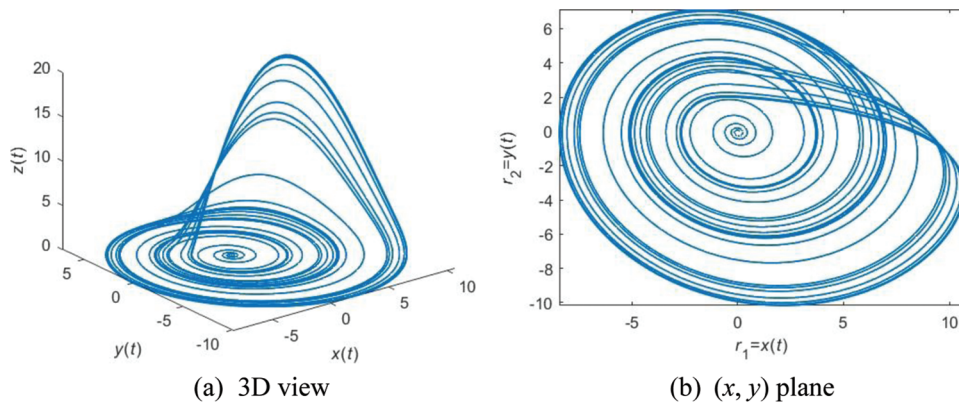

(a) 3D view   (b) (x, y) plane

Figure 5: An example of RA with parameters of (δ_a = 0.2, δ_b = 0.4, δ_c = 5.7)

4 Experiments, Results, and Discussions

Ten runs of 10-fold cross-validation are used to relate a reliable performance of our WACPN model. Besides, we use the following measures—sensitivity (Sen, symbolized as η_1), specificity (Spc, symbolized as η_2), precision (Prc, symbolized as η_3), accuracy (Acc, symbolized as η_4), F1 score (symbolized as η_5), Matthews correlation coefficient (MCC, symbolized as η_6), Fowlkes–Mallows index (FMI, symbolized as η_7), and the area under the curve (AUC)—to appraise the performances of different models.
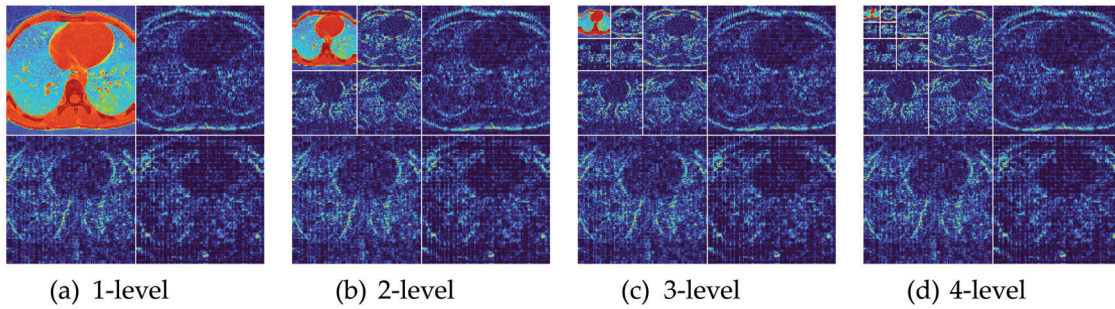
4.1 Parameter Configuration

The parameters of this study are listed in Tab. 3. The sizes of the original images are 1024×1024 if we do not consider the number of color channels. The sizes of MTCed images are 624×624, and the sizes of preprocessed images are 256×256. The DCR is $z_1 = 48$, and the SSR is $z_2 = 97.92\%$. The MDL is $M = 4$. The number of features is $N_I = 13$. The number of neurons in HL is $N_L = 8$. The number of output neurons is $N_O = 2$. The number of parameters to be optimized is $N_\theta = 130$. The parameters in RA are $\delta_a = 0.2$, $\delta_b = 0.4$, $\delta_c = 5.7$.

4.2 Wavelet Decomposition

Fig. 6 shows the wavelet decomposition results with $M = 4$. The raw image is shown in Fig. 2a. The reason why we choose $M = 4$ is the trial-and-error method. We test other values of M and find $M = 4$ can obtain the best result.

Table 3: Parameter Setting

Parameter	Value	Parameter	Value
$(W_0,\ H_0)$	$(1024,\ 1024)$	M	4
$(p_1,\ p_2,\ p_3,\ p_4)$	200	N_I	13
$(W_1,\ H_1)$	$(624,\ 624)$	N_L	8
$(W_2,\ H_2)$	$(256,\ 256)$	N_O	2
z_1	48	N_θ	130
z_2	97.92%	$(\delta_a,\ \delta_b,\ \delta_c)$	$(0.2,\ 0.4,\ 5.7)$

(a) 1-level   (b) 2-level   (c) 3-level   (d) 4-level

Figure 6: Wavelet decomposition results

4.3 Results of Proposed WACPN Model

Tab. 4 shows the ten runs of 10-fold CV via the parameters shown in Tab. 3, where $p_r = 1,\ 2.,\ ..,\ 10$ means the run index. The final row in Tab. 4 presents the mean and standard deviation (MSD) of the results of 10 runs. WACPN attains a sensitivity of $91.87 \pm 1.37\%$, a specificity of $90.70 \pm 1.19\%$, a precision of $91.01 \pm 1.12\%$, an accuracy of $91.29 \pm 1.09\%$, an F1 score of $91.43 \pm 1.09\%$, an MCC of $82.59 \pm 2.19\%$, and an FMI of $91.44 \pm 1.09\%$.

Table 4: Ten-run results of the proposed WACPN model

p_r	η_1	η_2	η_3	η_4	η_5	η_6	η_7
1	93.44	91.28	91.64	92.37	92.53	84.75	92.54
2	92.13	93.29	93.36	92.70	92.74	85.41	92.74
3	91.15	89.93	90.26	90.55	90.70	81.09	90.70
4	93.77	90.27	90.79	92.04	92.26	84.12	92.27
5	90.82	90.94	91.12	90.88	90.97	81.76	90.97
6	89.84	89.93	90.13	89.88	89.98	79.77	89.98
7	91.48	90.27	90.58	90.88	91.03	81.76	91.03
8	91.48	89.26	89.71	90.38	90.58	80.77	90.59
9	93.77	91.95	92.26	92.87	93.01	85.75	93.01
10	90.82	89.93	90.23	90.38	90.52	80.76	90.52
MSD	91.87 ± 1.37	90.70 ± 1.19	91.01 ± 1.12	91.29 ± 1.09	91.43 ± 1.09	82.59 ± 2.19	91.44 ± 1.09

4.4 Effects of AIWF and RA

If we remove the AIWF from our WACPN model, the results using the same configuration are shown in Tab. 5. Similarly, the results of removing RA from our WACPN model are shown in Tab. 6. After comparing the results in Tab. 4 against the results in Tabs. 5 and 6, we can deduce that both strategies—AIWF and RA—are beneficial to our WACPN model.

Table 5: Ten-run results without AIWF

p_r	η_1	η_2	η_3	η_4	η_5	η_6	η_7
1	89.51	90.60	90.70	90.05	90.10	80.11	90.10
2	89.84	86.58	87.26	88.23	88.53	76.47	88.54
3	89.84	90.27	90.43	90.05	90.13	80.10	90.13
4	90.49	87.92	88.46	89.22	89.47	78.45	89.47
5	89.18	90.27	90.37	89.72	89.77	79.44	89.77
6	91.48	87.58	88.29	89.55	89.86	79.15	89.87
7	93.44	93.29	93.44	93.37	93.44	86.73	93.44
8	89.51	89.93	90.10	89.72	89.80	79.44	89.80
9	92.46	92.62	92.76	92.54	92.61	85.07	92.61
10	88.52	88.59	88.82	88.56	88.67	77.11	88.67
MSD	90.43 ± 1.56	89.77 ± 2.15	90.06 ± 1.96	90.10 ± 1.63	90.24 ± 1.58	80.21 ± 3.25	90.24 ± 1.58

Table 6: Ten-run results without RA

p_r	η_1	η_2	η_3	η_4	η_5	η_6	η_7
1	94.43	94.97	95.05	94.69	94.74	89.39	94.74
2	89.84	87.58	88.10	88.72	88.96	77.45	88.97
3	93.44	90.94	91.35	92.21	92.38	84.43	92.39
4	90.16	91.61	91.67	90.88	90.91	81.77	90.91
5	89.51	91.95	91.92	90.71	90.70	81.46	90.71
6	87.54	87.25	87.54	87.40	87.54	74.79	87.54
7	91.15	87.92	88.54	89.55	89.82	79.13	89.83
8	89.84	89.60	89.84	89.72	89.84	79.43	89.84
9	90.16	88.59	89.00	89.39	89.58	78.77	89.58
10	93.44	94.30	94.37	93.86	93.90	87.73	93.91
MSD	90.95 ± 2.16	90.47 ± 2.75	90.74 ± 2.59	90.71 ± 2.29	90.84 ± 2.24	81.44 ± 4.58	90.84 ± 2.24

Fig. 7 represents the ROC curves together with their upper and lower bounds of the proposed WACPN model and its two ablation studies (without AIWF and without RA). The AUC of WACPN model is 0.9577. The AUCs of the models removing AIWF or RA are only 0.9319 and 0.9456, respectively, demonstrating that both AIWF and RA help improve the standard PSO.
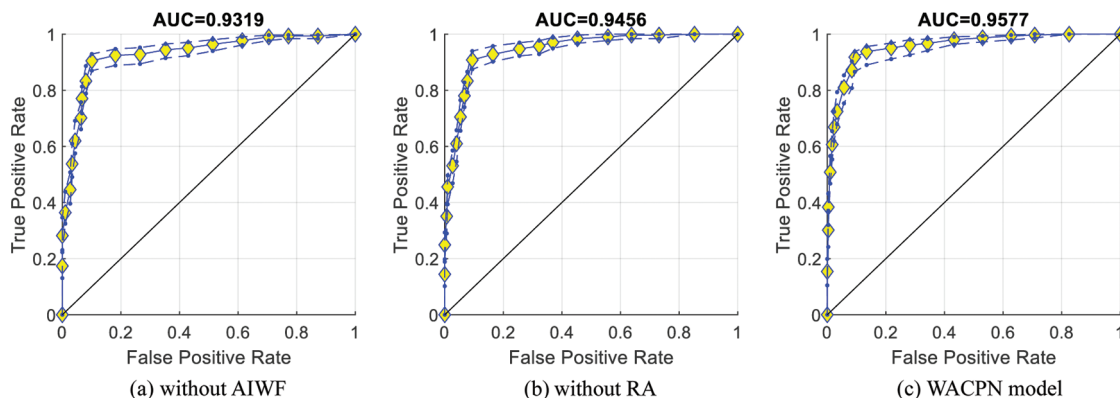
Figure 7: ROC curves

4.5 Comparison with State-of-the-Art Models

The proposed WACPN model is judged with six state-of-the-art models: GAN [4], CADe [5], SVM [6], IQNN [7], DT [8], and CSO [9]. The evaluation results on the same dataset via ten runs of 10-fold CV are listed in Tab. 7.

Table 7: Results of proposed WACPN and SOTA models (Unit: %)

Model	η_1	η_2	η_3	η_4	η_5	η_6	η_7
GAN [4]	85.54 ± 1.57	85.97 ± 1.12	86.20 ± 1.00	85.75 ± 1.02	85.86 ± 1.07	71.52 ± 2.03	85.86 ± 1.07
CADe [5]	86.59 ± 0.96	85.57 ± 1.45	86.02 ± 1.14	86.09 ± 0.62	86.29 ± 0.56	72.18 ± 1.23	86.30 ± 0.56
SVM [6]	87.34 ± 1.29	85.84 ± 1.23	86.33 ± 1.06	86.60 ± 0.95	86.83 ± 0.95	73.21 ± 1.91	86.83 ± 0.95
IQNN [7]	88.36 ± 0.92	86.24 ± 1.55	86.82 ± 1.21	87.31 ± 0.61	87.57 ± 0.54	74.65 ± 1.20	87.58 ± 0.53
DT [8]	82.69 ± 1.79	84.73 ± 1.08	84.73 ± 0.81	83.70 ± 0.80	83.68 ± 0.93	67.44 ± 1.58	83.70 ± 0.92
CSO [9]	91.64 ± 0.99	90.64 ± 2.11	90.96 ± 1.81	91.14 ± 1.12	91.29 ± 1.04	82.31 ± 2.22	91.29 ± 1.03
WACPN	$\mathbf{91.87 \pm 1.37}$	$\mathbf{90.70 \pm 1.19}$	$\mathbf{91.01 \pm 1.12}$	$\mathbf{91.29 \pm 1.09}$	$\mathbf{91.43 \pm 1.09}$	$\mathbf{82.59 \pm 2.19}$	$\mathbf{91.44 \pm 1.09}$

Note: Bold means the best.

Error bar (EB) is an excellent tool for ease of visual evaluation. Fig. 8 presents the EB of model comparison, from which we can observe that the proposed WACPN model is superior to six state-of-the-art models. The causes are triple. First, the 2d-WE layer stands as a proficient way to designate CCT images. Second, ACP is efficient in training FNN. Third, we fine-tune and select the best parameters for the RA. In the future, our model may be applied to other fields [21,22].
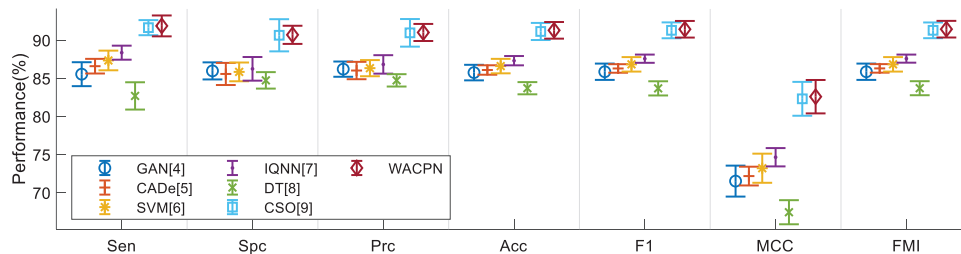

Figure 8: EB of model comparison

5 Conclusions

A novel WACPN method is proposed for diagnosing the CAP in CCT images. In WACPN, the 2d-WE layer works as feature extraction, and the optimization algorithm—ACP—is exercised to optimize the neural network. This proposed WACPN model is verified to have better results than six state-of-the-art models.

Three defects of the proposed WACPN model exist: (i) Deep learning models are not exercised. The reason is the small amount of our image set. (ii) Strict clinical validation is not tested either on-site or in cloud computing (CC) environments. (iii) The model is a black box, which does not go well with patients and doctors.

To work out the three limitations, first, we shall utilize the data augmentation method to enlarge the number of images in the dataset. Second, our team shall circulate the proposed WACPN model to the online CC environment (such as Azure) and summon specialists, clinicians, and physicians to examine its efficiency. Third, trustworthy or explainable Ais, which may provide the heatmaps pointing out the lesions, are two optional models to assist in adding explainability to the proposed WACPN model.

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

References

[1] G. Guarnieri, L. B. De Marchi, A. Marcon, S. Panunzi, V. Batani *et al.,* "Relationship between hair shedding and systemic inflammation in covid-19 pneumonia," *Annals of Medicine*, vol. 54, pp. 869–874, 2022.

[2] J. E. Schneider and J. T. Cooper, "Cost impact analysis of novel host-response diagnostic for patients with community-acquired pneumonia in the emergency department," *Journal of Medical Economics*, vol. 25, pp. 138–151, 2022.

[3] M. T. Olsen, A. M. Dungu, C. K. Klarskov, A. K. Jensen, B. Lindegaard *et al.,* "Glycemic variability assessed by continuous glucose monitoring in hospitalized patients with community-acquired pneumonia," *Bmc Pulmonary Medicine*, vol. 22, Article ID: 83, 2022.

[4] P. S. Heckerling, B. S. Gerber, T. G. Tape and R. S. Wigton, "Use of genetic algorithms for neural networks to predict community-acquired pneumonia," *Artificial Intelligence in Medicine*, vol. 30, pp. 71–84, 2004.

[5] X. L. Liu, F. Hou, H. Qin and A. M. Hao, "A cade system for nodule detection in thoracic ct images based on artificial neural network," *Science China-Information Sciences*, vol. 60, pp. 15, Article ID: 072106, 2017.

[6] A. Strehlitz, O. Goldmann, M. C. Pils, F. Pessler and E. Medina, "An interferon signature discriminates pneumococcal from staphylococcal pneumonia," *Frontiers in Immunology*, vol. 9, Article ID: 1424, 2018.

[7] Y. M. Dong, M. Q. Wu and J. L. Zhang, "Recognition of pneumonia image based on improved quantum neural network," *IEEE Access*, vol. 8, pp. 224500–224512, 2020.

[8] N. Ishimaru, S. Suzuki, T. Shimokawa, Y. Akashi, Y. Takeuchi *et al.,* "Predicting mycoplasma pneumoniae and chlamydophila pneumoniae in community-acquired pneumonia (cap) pneumonia: Epidemiological study of respiratory tract infection using multiplex pcr assays," *Internal and Emergency Medicine*, vol. 16, pp. 2129–2137, 2021.

[9] J. Zhou, "Community-acquired pneumonia recognition by wavelet entropy and cat swarm optimization," *Mobile Networks and Applications*, https://doi org/10.1007/s11036-021-01897-0, 2022 (Online First).

[10] W. Wang, Y. B. Jiang, Y. H. Luo, J. Li, X. Wang *et al.,* "An advanced deep residual dense network (drdn) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, pp. 1592–1601, 2019.

[11] W. Wang, H. Liu, J. Li, H. S. Nie and X. Wang, "Using cfw-net deep learning models for x-ray images to detect covid-19 patients," *International Journal of Computational Intelligence Systems*, vol. 14, pp. 199–207, 2021.

[12] K. Thirugnanasambandam, M. Rajeswari, D. Bhattacharyya and J. Y. Kim, "Directed artificial bee colony algorithm with revamped search strategy to solve global numerical optimization problems," *Automated Software Engineering*, vol. 29, Article ID: 13, 2022.

[13] W. Z. Al-Dyani, F. K. Ahmad and S. S. Kamaruddin, "Binary bat algorithm for text feature selection in news events detection model using markov clustering," *Cogent Engineering*, vol. 9, Article ID: 2010923, 2022.

[14] L. Wu, "Crop classification by forward neural network with adaptive chaotic particle swarm optimization," *Sensors*, vol. 11, pp. 4721–4743, 2011.

[15] M. Sahabuddin, M. F. Hassan, M. I. Tabash, M. A. Al-Omari, M. K. Alam *et al.,* "Co-movement and causality dynamics linkages between conventional and islamic stock indexes in Bangladesh: A wavelet analysis," *Cogent Business & Management*, vol. 9, Article ID: 2034233, 2022.

[16] S. Kavitha, N. S. Bhuvaneswari, R. Senthilkumar and N. R. Shanker, "Magnetoresistance sensor-based rotor fault detection in induction motor using non-decimated wavelet and streaming data," *Automatika*, vol. 63, pp. 525–541, 2022.

[17] A. K. Gupta, C. Chakraborty and B. Gupta, "Secure transmission of eeg data using watermarking algorithm for the detection of epileptical seizures," *Traitement Du Signal*, vol. 38, pp. 473–479, 2021.

[18] A. Meenpal and S. Majumder, "Image content based secure reversible data hiding scheme using block scrambling and integer wavelet transform," *Sadhana-Academy Proceedings in Engineering Sciences*, vol. 47, pp. 1–11, Article ID: 54, 2022.

[19] O. Kammouh, M. W. A. Kok, M. Nogal, R. Binnekamp and A. R. M. Wolfert, "Mitc: Open-source software for construction project control and delay mitigation," *Softwarex*, vol. 18, Article ID: 101023, 2022.

[20] J. M. Malasoma and N. Malasoma, "Bistability and hidden attractors in the paradigmatic rossler'76 system," *Chaos*, vol. 30, Article ID: 123144, 2020.

[21] X. R. Zhang, X. Sun, W. Sun, T. Xu, P. P. Wang *et al.,* "Deformation expression of soft tissue based on bp neural network," *Intelligent Automation and Soft Computing*, vol. 32, pp. 1041–1053, 2022.

[22] X. R. Zhang, X. Sun, X. M. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Cmc-Computers Materials & Continua*, vol. 71, pp. 3035–3050, 2022.