Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 19 No. 8 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijim.v19i08.54043

PAPER

Interactive Mobile English Translation Proficiency Model Based on Particle Swarm Optimisation and Neural Network for Teaching

Yunhui Li[1], Mei Huang[1](✉), Huafeng Dong[2]

[1]Fuzhou Preschool Education College, Fuzhou, China

[2]Chongqing Industry Polytechnic College, Chongqing, China

huangm019@outlook.com

ABSTRACT

This paper studies an interactive mobile English translation ability analysis model based on particle swarm optimisation (PSO) and neural networks (NNs) and explores its application potential in mobile translation teaching. By integrating the global search capability of PSO algorithm and the powerful learning capability of NN, the model aims to optimise the translation quality assessment process and improve the accuracy and efficiency of translation capability analysis. By training NNs to recognise language features, style and accuracy in translated texts, and fine-tuning NN parameters with PSO algorithm, this paper constructs a model that can effectively evaluate and interactively improve mobile English translation ability. The results show that the interactive mobile English translation ability analysis model based on PSO and NN has significant teaching application value, and brings new possibilities to the field of translation education, especially mobile learning scenarios.

KEYWORDS

particle swarm optimisation (PSO), neural network (NN), translate

1 INTRODUCTION

As globalisation speeds up, English's role as the primary language for international communication becomes increasingly prominent, and its translation ability has become increasingly important. In the field of translation, how to improve the ability to translate efficiently and accurately has become a common concern of educators and researchers. In recent years, the field of English translation has ushered in a wave of innovation in artificial intelligence technology and has become the focus of attention in the industry. Hameed and other scholars are committed to exploring new applications of intelligent systems. By extensively collecting and analysing multiple data such as students' learning status, personality characteristics and age, they successfully built an analytical model for assessing learning ability. This model

Li, Y., Huang, M., Dong, H. (2025). Interactive Mobile English Translation Proficiency Model Based on Particle Swarm Optimisation and Neural Network for Teaching. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(8), pp. 109–124. https://doi.org/10.3991/ijim.v19i08.54043

Article submitted 2024-12-23. Revision uploaded 2025-02-01. Final acceptance 2025-02-05.

is then applied to analyse individuals' unique performance in English translation, providing valuable insights for optimising English translation algorithms. With the rapid progress of information technology, this kind of translation system is gradually alleviating the problem of the educational gap caused by the uneven distribution of educational resources, which highlights the importance and urgency of English translation algorithm research. In the actual English application scenario, Behnamian and other innovators use artificial intelligence technology to build an intelligent listening resource corpus, which not only automatically allocates appropriate listening materials, but also realises contextually interactive functions, significantly improving the accuracy and practicality of translation. On the other hand, research on the integration of cloud platforms and artificial intelligence has also become a hot topic. Moradi and other researchers cleverly integrate the powerful big data processing and computing capabilities of cloud platforms into the translation process, so that artificial intelligence can play a greater potential in the field of translation. The cloud platform system can deeply understand the translation characteristics in various language environments, and then realise the accurate and quantitative output of translation results and promote the boundary expansion of English translation technology.

In view of this, this paper introduces a model for analysing English translation proficiency, utilising particle swarm optimisation (PSO) in conjunction with a neural network (NN). By learning and training the extracted translation samples of students, the trained PSO NN model is used to analyse the correct degree of students' English translation ability [1], assisting educators in assessing students' translation skills. To further enhance the quality of English translation instruction. Compared with previous studies, the PSO algorithm is an optimisation technique inspired by the behaviour of particle swarms in nature. By simulating the communication and interaction between particles, it achieves the purpose of swarm intelligence. It has the characteristics of being simple and powerful. The algorithm performs well in solving complex optimisation problems; especially in the realm of machine learning, NNs serve as computational models that emulate the structure and functioning of biological NNs. Through the connection and weight adjustment between large numbers of neurones, it can estimate or approximate complex functions, and has strong learning ability and generalisation ability. By integrating PSO with NNs, we construct an English translation proficiency analysis model. This model aims to enhance the precision and efficiency of translation by optimising the NNs parameters and structure. This model can not only analyse the translator's translation ability but also provide personalised guidance and feedback for translation teaching and promote the rapid improvement of translation ability [2]. In this paper, data collection, model construction, model training and testing, application analysis and other research methods are adopted. Data collection will collect translation sample data of students, including translation tasks of different difficulty and different fields; The model's development employs the PSO algorithm to refine the NNs structure and parameters, ultimately establishing an analysis model for English translation proficiency. The collected data is used to train the model, and the accuracy and generalisation ability of the model are evaluated by cross-validation and other methods. Finally, the trained model is applied to the actual teaching scenario to analyse students' translation ability and provide personalized feedback. The research focuses on the specific application and effect evaluation of the PSO algorithm in NN parameter and structure optimisation and the construction of English translation ability analysis model and its practical application in teaching. Research difficulties include the challenges to model training caused by the diversity of translation sample data (such as different fields and difficulties), how to effectively combine PSO algorithm and NN to achieve accurate optimisation of model parameters, and how to design a scientific and practical personalised teaching feedback mechanism based on model analysis results [3, 4]. The solution can improve data quality and diversity through data cleaning, standardization and enhancement technology, provide strong support for model training, in-depth study of the principles of PSO algorithm and NN, explore the best way to integrate the two, optimize the model parameters through multiple experiments and iterations, combine educational psychology and translation teaching theory, Design a personalized feedback mechanism that meets the needs of students and is constantly verified and adjusted through practical teaching applications. This study will explore the basic principles, construction methods and application of the PSO and NN-based English translation competence analysis model in translation teaching. The findings of this study will contribute to advancing innovative models for analysing translation competence and offer fresh perspectives and techniques for translation instruction. This holds considerable importance for enhancing the quality and efficiency of translation teaching [5].

2 PARTICLE SWARM OPTIMISATION ALGORITHM AND NEURAL NETWORK ALGORITHM ANALYSIS

2.1 Particle swarm optimisation algorithm

Particle swarm optimisation, a renowned evolutionary computational approach developed by J. Kennedy and R. C. Eberhart in 1995, draws inspiration from the collective hunting patterns of birds and fish schools. This algorithm addresses optimisation challenges by mimicking the interactions and collective intelligence exhibited by particle swarms in nature [6]. At its heart lies the principle of achieving the optimal solution through inter-particle communication and collaborative efforts.

The core principle of this algorithm involves representing each potential solution as a "particle" traversing the solution space [7]. These particles dynamically adjust their trajectory and velocity, guided by their own historical search records and the collective behaviour exhibited by neighbouring particles. Each particle possesses two key attributes: position and velocity. The former signifies the current solution, while the latter dictates the next position [8]. The dynamics of particle movement are governed by its instantaneous position and velocity, in conjunction with its personal best-achieved solution and the globally optimal solution discovered, thereby influencing its subsequent trajectory. For each particle, formulas (1) and (2) are used to calculate the updated position and velocity to seek a new solution [9].

$$v_i^{k+1} = \omega^* v_i^k + C_1 * R_{1,i}^k * \left(x_{Pbest,i}^k - x_i^k\right) + C_2 * R_{2,i}^k * \left(x_{Gbest}^k - x_i^k\right) \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2}$$

In the above formula, $x_{i,d}^k$ denotes the position of the i particle in the k generation, and $v_{i,d}^{k+1}$ denotes the velocity of the i particle in the k generation. During the search process, each particle keeps track of the best position it discovers, referred to as its personal best position, and $x_{Pbest,i}^k$ is the individual best position of the i particle in the k generation. At the same time, the whole population will also record the global best position found so far, that is, the overall best position, and x_{Gbest}^k is the overall optimal value position of the population in generation k. When updating speed and position [10], the algorithm uses inertia weight coefficient ω, self-learning factor C_1 and social learning factor C_2, as well as two random numbers R_1 and R_2 uniformly distributed in the interval [0,1]. Together, these parameters dictate how the particle modifies its speed and location based on its current condition, personal best, and overall best values.

Through many iterations, the particle swarm will gradually approach the optimal solution of the problem.

Figure 1 illustrates the flowchart of the PSO algorithm.

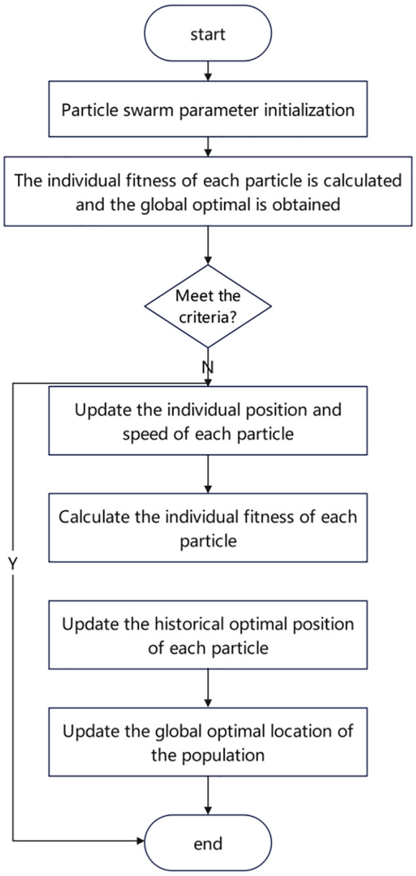

Fig. 1. Illustrative diagram of the particle swarm optimisation methodology

2.2 Neural network algorithm

Neural networks, alternatively known as artificial NNs, constitute a branch of machine learning and serve as the foundation of deep learning algorithms [11]. They represent a mathematical/computational model that mimics the structural arrangement and functional processes of biological NNs to predict or closely approximate functions [12]. It essentially constitutes a network architecture consisting of numerous interconnected neurones. These nodes simulate the functions of biological neurones, process input data through weighted summation [13], activation function and other mechanisms, and output processing results. It is mainly composed of nodes (neurones), levels (input level, concealed level, output level) [14], weights, bias and activation functions and other basic elements. The operational principle of NNs is rooted in mimicking the biological nervous system, processing and transmitting information through the interaction between large numbers of neurones. Each neuron in a NNs receives input signals from the preceding layer [15], applies a weighted summation to these signals, and subsequently processes them through an activation function to generate an output signal [16, 17]. This output then serves as an input for subsequent neurones, contributing to the intricate, interconnected network architecture.

In a NN, each neuron gathers input from the preceding layer of neurones and generates an output after performing a sequence of computations. These calculations usually involve the following steps:

1. Enter weighted summation: A neurone integrates multiple input signals, each scaled by its respective weight, and sums these weighted inputs for further processing [18]. The mathematical expression for this step is formula (3).

$$z = \sum_{i=1}^{n} w_i x_i \tag{3}$$

Where, z is the weighted sum, w_i is the weight of the i th input signal, x_i is the i th input signal, and n is the number of input signals [19].

2. Add bias: In NNs, bias is a learnable parameter used to adjust the activation value of neurones. Bias is usually treated as a constant term that is added to the input of a neurone and thus affects its output [20]. The effect of bias can be understood as adjusting the intercept position of the neuron's activation function when the input is zero. If there is no bias parameter, the activation function of the neurone is always zero when the input is zero, and it cannot exhibit complex nonlinear characteristics. By adjusting the bias parameters, the output value of the activation function of neurones is not zero when the input is zero, so that the complex nonlinear characteristics can be better displayed. Furthermore [21], to tackle the challenge of data imbalance, bias parameters offer a viable solution. In the context of classification, where a dataset may contain a disproportionately small number of samples belonging to a particular class, fine-tuning these bias parameters can effectively elevate the predictive emphasis on that class, ultimately enhancing classification accuracy. On the basis of the weighted sum, a bias term is added to adjust the activation threshold of neurones. The mathematical expression for this step is formula (4).

$$z' = z + b \tag{4}$$

Where, z' is the weighted sum after adding bias, and b is the biased-term.

3. Activation function: Apply the activation function to the weighted sum (optionally adjusted by a bias term) to generate the neuron's output [22]. The activation function serves to convert an unlimited input into a predictable output format and incorporate nonlinear elements. Common activation functions include Sigmoid, ReLU, Tanh and so on. Take the Sigmoid function as an example; its mathematical expression is shown in equation (5) [23].

$$a = \sigma(z') = \frac{1}{1 + e^{-z'}} \tag{5}$$

Where, a is the output of the neuron and $\sigma(\cdot)$ is the Sigmoid function.

3 MODEL ANALYSIS OF ENGLISH TRANSLATION ABILITY BASED ON PARTICLE SWARM OPTIMISATION AND NEURAL NETWORK

By integrating the optimisation principles of the PSO algorithm into the NN, the combined approach significantly enhances its global optimisation capabilities.

In the initial phase, this hybrid algorithm leverages the movement and updating of particles to identify the optimal solution for the NN. This paper presents a model for assessing English translation proficiency through the fusion of the BP neural network algorithm with the PSO algorithm. The following outlines the algorithmic framework and its execution flow [24].

3.1 Construction of neural network model

A BP neural network primarily comprises an arrangement of multiple layers, including input layers, output layers, and intermediary hidden layers, forming the structural foundation of its operation. In a BP neural network, input signals are sequentially transmitted between neurones along a preset path, which is called forward propagation. Each neurone receives weighted inputs from neurones in the previous layer and produces outputs based on activation functions within it [23], which then become inputs to neurones in the next layer. Through this step-by-step approach, the network completes a complete mapping from input to output. To minimise the discrepancy between the networks' output and the desired target, the BP neural network employs an error backpropagation technique that leverages gradient descent for optimisation. This method iteratively computes the error contribution (gradients) of each neurone, starting from the output layer and moving backwards, and updates the weights and biases of each neurone based on these gradients. Through repeated iterations of this learning process, the network gradually adjusts its internal parameters until the error reaches the lowest acceptable level, thus achieving effective training of the network.

Figure 2 displays the algorithm's flowchart.

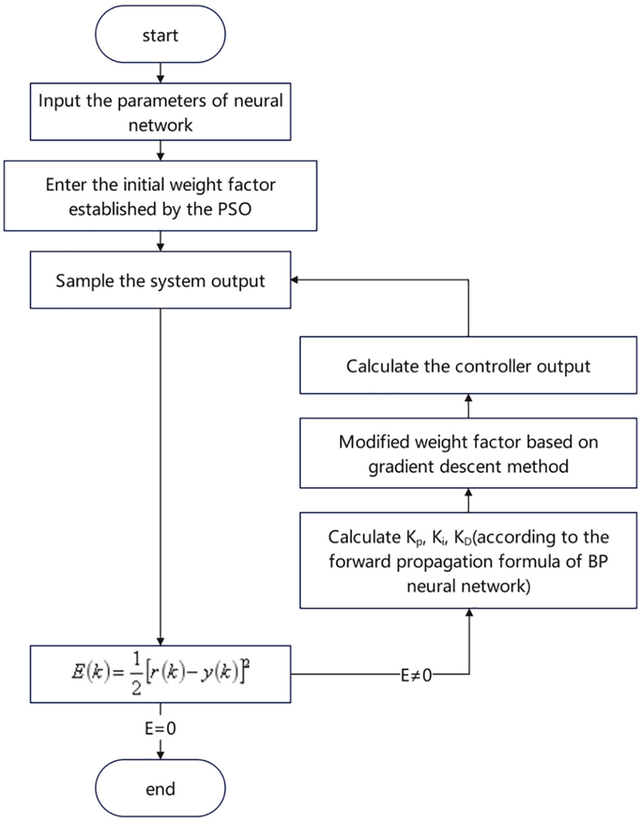

Fig. 2. Flow chart of BP neural network algorithm

The input for the NNs input layer is presented in equation (6).

$$n_i(k) = x_i(k),\ i = 1, 2, 3, 4 \tag{6}$$

Where $x_i(k)$ represents the input value and $n_i(k)$ represents the matrix form of the input value.

The input and output of the hidden layer and the activation function of the hidden layer are shown in formula (7), (8) and (9).

$$I = \sum_{h=1}^{5} w_i \times n_i(k), h = 1, 2, 3, 4, 5 \tag{7}$$

$$O_h(k) = g[I(k)],\ k = 1, 2, 3 \tag{8}$$

$$g(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9}$$

In formula 7, I is the input value of the hidden layer, in formula 8, $O_h(k)$ is the output value of the hidden layer, and in formula 4, the activation function is the sigmoid function.

The input value of the output layer is equation (10), output value is (11), neuronal activation function is equation (12), and the performance index function is equation (13).

$$Q = w_o \times O_h \tag{10}$$

$$K = f(Q) \tag{11}$$

$$f(x) = \frac{e^x}{e^x + e^{-x}} \tag{12}$$

$$E(k) = \frac{1}{2}[r(k) - y(k)]^2 \tag{13}$$

In Formula 10, Q represents the input value of the output layer, in formula 11, K represents the output value of the output layer, and $f(Q)$ represents the activation function.

3.2 Data pre-processing

In order to eliminate the influence of large-value data on model-based prediction or diagnosis, the data must be pre-processed first. In this paper, the normalisation of the data is adopted. Normalisation is a technique of scaling data so that it falls into a specific small range, such as between 0 and 1 [25, 26]. During the normalisation process, the Sigmoid function serves as the activation function for the output layer of the NN. The Sigmoid function possesses a distinctive characteristic: as the input value x tends towards positive or negative infinity, its output value converges to 1 or 0, respectively, thus naturally limiting the output range between (0, 1). Without data normalisation, those features with large absolute values may dominate the training process, causing the model to be insensitive to changes in features with smaller values, thus affecting the overall training effect and the generalisation ability of the model. Through normalization processing [27], it is possible to ensure that all features are fairly considered during training. This, in turn, enhances the learning and prediction capabilities of the network. The formula for normalisation is given in equation (14).

$$x_i = \lambda_1 + (\lambda_2 - \lambda_1)\left(\frac{z_i - z_i^{\min}}{z_i^{\max} - z_i^{\min}}\right) \tag{14}$$

The output value can be reversely normalised, and the formula is shown in equation (15).

$$z_i = \left(\frac{x_i - \lambda_1}{\lambda_2 - \lambda_1}\right)\left(z_i^{\max} - z_i^{\min}\right) + z_i^{\min} \tag{15}$$

In Formula 15, z_i represents the anti-normalised value, and x_i represents the normalised value; λ_1 represents the lower limit and λ_2 represents the upper limit. $z_i^{\max}$ indicates the maximum value in the source data. $z_i^{\min}$ represents the minimum value in the source data.

3.3 Particle initialization

Within the predefined search space, the initial locations and velocities of particles are randomly initialised. Subsequently, the locally optimal position is iteratively refined by comparing and updating each particle's individual best position, fostering a dynamic optimisation process. For particle velocity and position initialisation settings; set the initial position of particle i to be a vector $X_i = (x_{i1}, x_{i2},\ldots, x_{in})$, where n represents the dimensionality of the search space, and each component x_{ij} ($j = 1, 2,\ldots, n$) is randomly generated in the interval (0, 1), denoting the initial coordinates of particle i within the exploration domain, it signifies the origination point of its traversal in the search space. Similarly, the initial velocity of particle i is set to vector $V_i = (v_{i1}, v_{i2},\ldots, v_{id})$. It should be noted that the components of speed are usually not generated directly at random in the interval (0, 1), but according to the specific needs of the problem and the scope of the search space to set a suitable speed range and then randomly select the value of each v_{ij} within this range. This is done to ensure that the particles are able to move through the search space at an appropriate speed. In the PSO algorithm, a local optimal position P_g is also maintained, indicating the optimal position discovered among all current particles, P_g is determined by comparing the current position of all the particles (usually calculated based on some fitness function), $P_g = \min\{P_0, P_1,\ldots, P_s\}$, where P_i ($i = 0, 1, \ldots s$) represents the record of the best position of the particle i (that is, the optimal solution encountered by the particle during the search), and s is the total number of particles in the particle swarm. Therefore, P_g is actually the best of all the individual best positions of all the particles as the local optimal solution for the entire particle population.

3.4 Fitness calculation

For each particle, we calculate its fitness value according to its specific objective function. This step aims to quantify how good the particle's current position is for solving the optimisation problem. We then compare the calculated fitness values with the best fitness values stored in memory (i.e., the function values corresponding to the optimal solutions explored by all the particles so far). If the current particle's fitness value surpasses the best value stored in memory, the best value is updated to ensure that we are tracking the best solution in the entire search space. Then, based on this remembered sweet spot, and the sweet spot in the particle's own history, the particle adjusts the speed and direction of its next phase of search. The sum of squared errors (*SSE*) is a frequently used measure to assess the difference between a model's predictions and actual values, and we utilise it here to determine fitness. A smaller *SSE* value indicates that the model's predictions are closer to the actual values, that is, the higher the fitness of the particle. By minimising *SSE*, we can drive the particles closer to the global optimal solution. Its calculation formula is shown in equation (16).

$$f(i) = SSE = \sum_{i=1}^{n} (T_i - E_i)^2 \tag{16}$$

In Formula 16, T_i represents the data fitted by the algorithm, and E_i represents the original data.

3.5 Particle update and iterative optimisation

The positions of particles are recorded, and their position vectors and velocity vectors are updated continuously. The tracking of each particle's position and velocity vectors is ongoing, ensuring timely updates, and at each iteration, the performance of the current particle (i.e., its best value) is first evaluated and compared to the global best value currently recorded. If a particle's best value surpasses the current global best value, the global best value in memory is updated to that particle's best value. Subsequently, all particles adjust their position and velocity vectors based on the current global best value and their own historical best values, according to the rules defined in the PSO algorithm. This correction process aims to guide the particle to move to a better region in the solution space and prepare for the next global search iteration. The calculation formulas for particle position velocity update are as follows: (17) and (18).

$$V_i^{k+i} = WV_i^k + c_1 rand()\left(s_i^{k^*} - s_i^k\right) + c_2 rand()\left(s_i^{\#} - s_i^k\right) \tag{17}$$

$$s_i^{k+1} = s_i^k + V_i^{k+1} \tag{18}$$

Where c_1 and c_2 are acceleration constants; W stands for inertia factor; *rand*() is a uniform random number of [0,1]; $s_i^{k^*}$ represents the k* dimension of the individual extremum of the i th variable; $s_i^{\#}$ represents the i dimension of the global optimal solution. s_i^k represents the k dimension of the individual extremum of the i th variable.

By repeating the above steps, recording the position, updating the velocity vector, comparing and updating the global best value, and correcting the particle position velocity, the particle swarm can gradually approach the optimal solution of the optimisation problem.

3.6 Training neural network model

Upon completion of the optimisation process by the particle swarm algorithm, the BP neural network is endowed with the optimal weights and thresholds obtained.

These refined parameters then facilitate the network's training, where forward propagation computes the network's output, and error backpropagation adjusts weights and thresholds to minimise the output error. Upon training completion, an independent test set assesses the network's practical performance. During testing, the trained network processes test data, and the output's alignment with actual values is evaluated. Evaluation metrics such as accuracy, recall, and F1 score are obtained by comparing test results with actual values, providing a thorough assessment.

Figure 3 depicts the flowchart of the BP neural network optimised using the PSO technique.

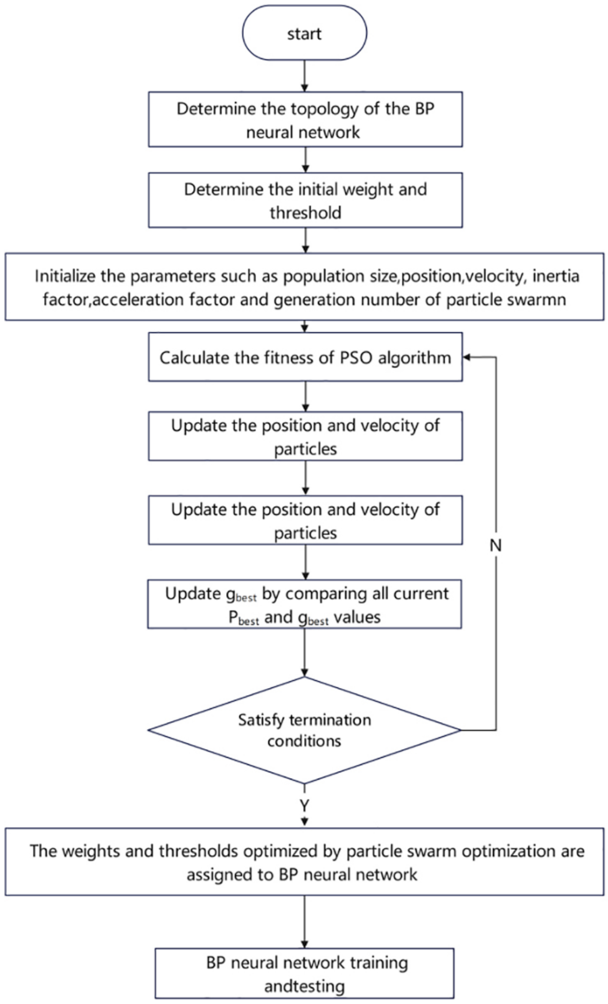

Fig. 3. Flowchart of a BP neural network improved through the application of the particle swarm optimisation algorithm for optimisation

4 EXPERIMENT

4.1 Comparison experiment with Elman neural network

In order to evaluate the accuracy of PSO-BP neural network algorithm in prediction tasks, this study carried out two sets of experiments in parallel: one group used PSO-BP neural network algorithm, the other group used Elman neural network algorithm, and verified the prediction performance by comparing and analysing the experimental data of the two. The Elman neural network, as a leader in the field of regression prediction and time series prediction, with its excellent time series data processing ability and built-in "memory" mechanism, has shown strong strength in many prediction tasks, so it is selected as the benchmark method in this comparison experiment. The Elman neural network is fundamentally a type of recursive NN. Its distinctive design encompasses four primary layers: the input layer, which is tasked with receiving external data; the hidden layer (or intermediate layer), which handles the nonlinear transformation of information; and a key component of the network, the context layer, which serves to store the output state of the hidden layer from the previous time step, so that the network can deal with the temporal dependence of time series data. The output layer outputs the processed result. This design gives Elman NNs a significant advantage when dealing with tasks that feature time series and require memory.

Figure 4 visually illustrates this structural layout of the Elman neural network.

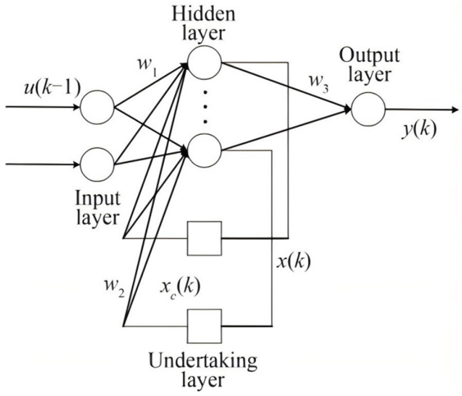

Fig. 4. Architecture of the Elman neural network

$$\begin{aligned} y(k) &= g(w_3 x(k)) \\ x(k) &= f(w_2 x(k) + w_1(u(k-1))) \\ x_c(k) &= x(k-1) \end{aligned} \tag{19}$$

In Formula 14, x represents the node unit of the middle layer, y represents the output, u represents the input, x_c represents the feedback state, $g(*)$ represents the transfer function of the output neuron, $f(*)$ represents the transfer function of the interneuron, k represents the reference points, and w_1 denotes the weight connecting the input layer to the hidden layer in the Elman neural network. w_2 represents the weight between the hidden layer and the successor layer and w_3 represents the weight between the hidden layer and the output layer. $f(*)$ uses logsig as the transfer function, and $g(*)$ uses pure line as the linear transfer function.

In Figure 5, we assessed the predictive accuracy of two NN models: the BP neural network enhanced with PSO and the Elman neural network. The red curve highlights the deviation between the Elman neural network's predictions and the actual values when tested on the dataset, while the blue curve represents the corresponding deviation when making predictions through the PSO-BP neural network. To perform this comparison experiment, the test dataset was reorganized into the same format as the training dataset, i.e., formatted as a matrix with 201 rows and 60 columns. This setup allows the NN to process the input data column by column, each processing corresponding to one training iteration or test instance indicated on the horizontal coordinate in Figure 5, for a total of 60 independent predictions and evaluations.

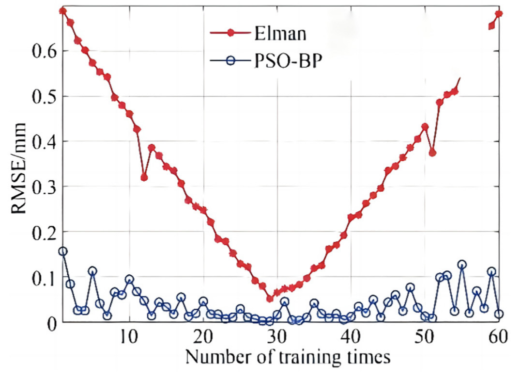

Fig. 5. Comparison of prediction accuracy between PSO-BP and Elman neural network

The analysis presented in Figure 5 reveals that the PSO-BP neural network outperforms the Elman neural network in terms of prediction accuracy. The experimental results clearly showcase this superiority, with the PSO-BP NN algorithm achieving a prediction error of just 0.0177mm, while the error of Elman algorithm is 0.6825mm. This comparison highlights the significant improvement of the prediction accuracy of PSO-BP algorithm, which is specifically reflected in the improvement of its prediction accuracy by about 97.42% compared with Elman algorithm. Further, when dealing with large-scale data sets, the PSO-BP neural network algorithm shows stronger adaptability and efficiency advantages. This is primarily attributed to the effective optimisation provided by the PSO algorithm on the BP neural network's parameters, enabling the network to converge more rapidly to the global optimal solution, so that the prediction speed and accuracy are excellent. To sum up, PSO-BP neural network algorithm is undoubtedly a more advantageous choice for those prediction tasks that require high efficiency and accuracy. It can not only provide more accurate prediction results, but also maintain efficient performance when processing large-scale data, thus meeting the needs of complex prediction scenarios.

4.2 Case application

In order to verify the application of PSO based NN model in English translation teaching, a comprehensive verification process was designed and implemented. First, the data samples of students' English translation learning characteristics are systematically collected, and then the validation procedure encompasses two primary phases. (1) Neural network cultivation and fine-tuning: Here, we leverage the PSO technique to educate NNs. The objective of this phase is to discover the optimal combination of NNs weights and thresholds, ensuring a seamless fit with the training dataset, utilizing PSO's intelligent exploration capabilities. (2) Model evaluation and testing: post-training, an independent test set assesses the optimised NNs model's efficacy and performance. We evaluate its accuracy, recall, and F1 score by gauging how closely its outputs align with actual translation results.

Data quality is the key to the success of NN algorithms. 5000 pieces of students' English translation data were carefully screened from the huge database of many schools. Following rigorous filtering, samples deemed invalid, ambiguous, or incomplete were discarded to uphold the diversity and veracity of the dataset. To enhance the model's generalisation capabilities, data augmentation methods, including the introduction of noise, were employed, culminating in the compilation of a comprehensive sample set comprising 2000 meticulously curated, high-quality test instances.

Then, parameters are set and trained. When using PSO algorithm to train NN, the precocious factor is set to 0.01, and other key parameters are randomly generated. Based on best practices of the PSO algorithm, three different particle swarm sizes (10, 15 and 20) were tried to explore their impact on training efficiency and results. The experimental results show that when the number of iterations reaches 223, 286 and 437 respectively, the model error decreases significantly and tends to be stable, and no better solution appears, indicating that the global optimal solution has been found. At this point, we stop the iteration and record the relevant data, as shown in Table 1.

Table 1. Results of case test of neural network model of particle swarm optimisation

Particle Population	Number of Iterations	Optimal Number of Particles	Training Sample Mean Error	Average Error of Test Sample
10	223	3	0.06	0.13
15	286	9	0.20	0.26
20	437	6	0.29	0.30

The data in the table above clearly indicates that training the NN model using the PSO algorithm can identify the optimal solution for various sizes of particle populations while achieving a low error rate. Utilising the devised learning analysis model framework, the NN model, bolstered by PSO technology, can be employed to precisely assess the proficiency of students in English translation, thereby offering an accurate evaluation of their abilities, providing teachers with an effective estimate of students' translation level, and guiding the subsequent teaching plan and strategy formulation. At the same time, through the quantitative assessment of students' translation ability, students can more intuitively understand their own translation level and see the specific changes in their progress. Such instant feedback can improve students' motivation to learn and stimulate their interest in learning in order to participate more actively in the learning and practice of translation. The application of this model is a concrete embodiment of educational technology innovation. It breaks the traditional way of teaching evaluation and introduces a more scientific and efficient evaluation method. This innovation not only enhances teaching quality but also drives the reform and advancement of the entire educational system and provides strong support to cultivate more innovative talents and practical skills. In summary, the application of the model of analysis of translation capacity to English based on the optimisation of particle swarm and NN in teaching is of great importance, which can improve not only the quality of teaching and the level of personalised teaching but also improve the motivation of students to learn and promote educational innovation. As technology continues to evolve and improve, it is believed that this model will play an even more important role in education in the future.

5 CONCLUSION

This paper delves into the model for analysing English translation ability utilising PSO and NN, and meticulously examines its application in translation education. The key findings of the research are as follows:

1. Successfully constructed an English translation ability analysis model combining PSO algorithm and NN technology. The model makes full use of the advantages of

PSO algorithm in global search and local optimisation, combined with the powerful nonlinear mapping and learning ability of NN, and realises the efficient and accurate evaluation of English translation ability. In the process of model construction, the mathematical model, algorithm flow and convergence characteristics of PSO algorithm are deeply studied. At the same time, the basic principle of artificial neural network, network structure design and parameter optimisation methods are mastered. Through a carefully designed fusion strategy, we find the best balance point for the two to work together, ensuring the accuracy and robustness of the model in the evaluation of translation ability.

2. In order to ensure the effectiveness and generalisation ability of model training, a series of advanced data processing technologies are adopted, including data cleaning, standardisation and enhancement. By removing noise data, filling in missing values, processing outliers and other steps, the data quality is significantly improved. Through standardised processing, the influence of different dimensions on data analysis is eliminated. Through data enhancement technology, the diversity and quantity of training samples are increased, and the overfitting problem is effectively alleviated. In the model training stage, combining educational psychology and translation teaching theory, reasonable objective functions and evaluation indicators were designed, and the model parameters were optimised through multiple experiments and iterations, which gradually improved the prediction accuracy and stability of the model.
3. To assess the practical impact of the NN algorithm model incorporating PSO in English translation education, it is applied in several practical teaching scenarios. Through a comparative analysis of the students' performance under the traditional teaching method and the teaching method based on the model, it is found that the students using the new model have shown significant improvement in translation ability, learning motivation and learning effectiveness. These positive results not only prove the validity and practicability of the model but also provide strong support for the intelligent transformation of translation teaching in the future.

This model offers a novel tool for quantitative assessment of translation quality and an unprecedented personalised guidance and feedback system for translation education. As we look ahead, with the ongoing advancements in artificial intelligence technology and the diversifying needs of translation education, the English translation ability analysis model leveraging PSO and NN holds vast potential for development. It is hoped that the application potential of this model in more teaching scenarios, such as multilingual translation and professional translation, can be further explored in future studies. To cater to various learners' requirements, we also aspire to enhance the model's performance and user experience through ongoing technological innovations and optimizations, and contribute more to the prosperity and development of translation education.

6 REFERENCES

[1] Y. Tian *et al.*, "Post-processing of short-term quantitative precipitation forecast with the multi-stream convolutional neural network," *Atmospheric Research*, vol. 309, p. 107584, 2024. https://doi.org/10.1016/j.atmosres.2024.107584

[2] S. Long *et al.*, "Parameter calibration of the angle of repose of particle materials based on convolutional neural network," *Biosystems Engineering*, vol. 245, pp. 152–163, 2024. https://doi.org/10.1016/j.biosystemseng.2024.07.011

[3] Y. Tang *et al.*, "Multi-subswarm cooperative particle swarm optimization algorithm and its application," *Information Sciences*, vol. 677, p. 120887, 2024. https://doi.org/10.1016/j.ins.2024.120887

[4] Q. Zhou, H. Hashim, and N. A. Sulaiman, "A systemic review (2014–2023) on the mobile-assisted blended learning for English as a foreign language education with a focus on empirical studies," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 24, pp. 153–172, 2023. https://doi.org/10.3991/ijim.v17i24.42019

[5] P. Li *et al.*, "Correction to: Numerical control machining step error calculation based on hybrid particle swarm optimization method," *The International Journal of Advanced Manufacturing Technology*, vol. 133, pp. 3163–3164, 2024. https://doi.org/10.1007/s00170-024-14057-x

[6] K. P. Tyagi and D. Agrawal, "Automatic detection of sleep apnea from a single-lead ECG signal based on spiking neural network model," *Computers in Biology and Medicine*, vol. 179, p. 108877, 2024. https://doi.org/10.1016/j.compbiomed.2024.108877

[7] L. Wang *et al.*, "A particle swarm optimization-based deep clustering algorithm for power load curve analysis," *Swarm and Evolutionary Computation*, vol. 89, p. 101650, 2024. https://doi.org/10.1016/j.swevo.2024.101650

[8] C. Jiao and Q. Zhou, "Path test data generation using adaptive simulated annealing particle swarm optimization," *Soft Computing, (prepublish),* pp. 1–21, 2024. https://doi.org/10.1007/s00500-024-09803-y

[9] Y. Zhang *et al.*, "Particle swarm optimization for a hybrid freight train powered by hydrogen or ammonia solid oxide fuel cells," *International Journal of Hydrogen Energy*, vol. 72, pp. 626–641, 2024. https://doi.org/10.1016/j.ijhydene.2024.05.347

[10] H. Kim and S. J. Yi, "Image generation of hazardous situations in construction sites using text-to-image generative model for training deep neural networks," *Automation in Construction*, vol. 166, p. 105615, 2024. https://doi.org/10.1016/j.autcon.2024.105615

[11] M. M. Draz, O. Emam, and S. M. Azzam, "Software cost estimation prediction using a convolutional neural network and particle swarm optimization algorithm," *Scientific Reports*, vol. 14, 2024. https://doi.org/10.1038/s41598-024-63025-8

[12] Y. Zhang *et al.*, "Non-volatile double-tunable vortex metalens design based on Sb2S3 using deep neural network and particle swarm optimization algorithm," *Optics Communications*, vol. 560, p. 130453, 2024. https://doi.org/10.1016/j.optcom.2024.130453

[13] C. Deng and X. Xue, "Hybrid particle swarm optimization and group method of data handling for the prediction of ultimate strength of concrete-filled steel tube columns," *Advances in Engineering Software*, vol. 195, p. 103708, 2024. https://doi.org/10.1016/j.advengsoft.2024.103708

[14] K. Yan *et al.*, "Deep neural network with adaptive dual-modality fusion for temporal aggressive behavior detection of group-housed pigs," *Computers and Electronics in Agriculture*, vol. 224, p. 109243, 2024. https://doi.org/10.1016/j.compag.2024.109243

[15] L. Yanxinwen *et al.*, "Graph structure estimation neural network-based service classification," *International Journal of Web Information Systems*, vol. 20, no. 4, pp. 436–451, 2024. https://doi.org/10.1108/IJWIS-03-2024-0087

[16] Y. Zhang *et al.*, "Adaptive projection plane and reference point strategy for multi-objective particle swarm optimization," *Alexandria Engineering Journal*, vol. 105, pp. 381–401, 2024. https://doi.org/10.1016/j.aej.2024.07.006

[17] Y. Zhang, "Machine learning-based personalized learning path decision-making method on intelligent education platforms," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 18, no. 16, pp. 68–82, 2024. https://doi.org/10.3991/ijim.v18i16.51009

[18] L. Zhou *et al.*, "Using particle swarm optimization with backpropagation neural networks and analytic hierarchy process to optimize the power generation performance of enhanced geothermal system (EGS)," *Water*, vol. 16, no. 3, p. 509, 2024. https://doi.org/10.3390/w16030509

[19] B. Deng *et al.*, "Improving a fuel cell system's thermal management by optimizing thermal control with the particle swarm optimization algorithm and an artificial neural network," *Applied Sciences*, vol. 13, no. 23, p. 12895, 2023. https://doi.org/10.3390/app132312895

[20] L. Hongxing and L. Yi, "Comprehensive testing technology for new energy vehicle power batteries based on improved particle swarm optimization," *Energy Informatics*, vol. 7, no. 49, 2024. https://doi.org/10.1186/s42162-024-00356-w

[21] W. Meng-qi *et al.*, "Optimization of convolutional neural network for glass-forming ability prediction based on particle swarm optimization," *Materials Today Communications*, vol. 36, p. 106808, 2023. https://doi.org/10.1016/j.mtcomm.2023.106808

[22] K. Bronik and L. Zhang, "Conditional advancement of machine learning algorithm via fuzzy neural network," *Pattern Recognition,* vol. 155, p. 110732, 2024. https://doi.org/10.1016/j.patcog.2024.110732

[23] J. Yang *et al.*, "Rolling bearing residual useful life prediction model based on the particle swarm optimization-optimized fusion of convolutional neural network and bidirectional long–short-term memory–multihead self-attention," *Electronics,* vol. 13, no. 11, p. 2120, 2024. https://doi.org/10.3390/electronics13112120

[24] I. M. A. "Optimization of load balancing and task scheduling in cloud computing environments using artificial neural networks-based binary particle swarm optimization (BPSO)," *Sustainability,* vol. 14, no. 19, p. 11982, 2022. https://doi.org/10.3390/su141911982

[25] L. Zhen and A. Bărbulescu, "Comparative analysis of convolutional neural network-long short-term memory, sparrow search algorithm-backpropagation neural network, and particle swarm optimization-extreme learning machine models for the water discharge of the Buzău river, Romania," *Water,* vol. 16, no. 2, p. 245, 2024. https://doi.org/10.3390/w16020245

[26] G. Coppola, M. Panelli, and F. Battista, "Solution of orifice hollow cathode plasma model equations by means of particle swarm optimization," *Applied Sciences*, vol. 14, no. 13, p. 5831, 2024. https://doi.org/10.3390/app14135831

[27] X. Zhang *et al.*, "Spectral reflectance reconstruction based on wideband multi-illuminant imaging and a modified particle swarm optimization algorithm," *Optics Express*, vol. 32, no. 3, pp. 2942–2958, 2024. https://doi.org/10.1364/OE.506136

7 AUTHORS

Yunhui Li is with the Fuzhou Preschool Education College, Fuzhou 344000, China.

Mei Huang is with the Fuzhou Preschool Education College, Fuzhou 344000, China (E-mail: huangm019@outlook.com).

Huafeng Dong is with the Chongqing Industry Polytechnic College, Chongqing 401120, China.