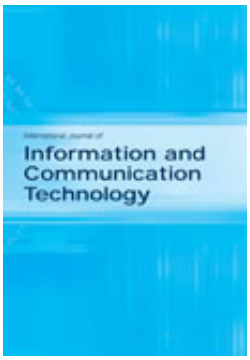
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

A particle swarm optimisation-based deep belief network for traditional Chinese medicine data processing strategies

Wenqiao Ding, Chongli Xu, Ruiqi Zhang

Article History:

Received:	25 September 2025
Last revised:	22 October 2025
Accepted:	26 October 2025
Published online:	12 January 2026

A particle swarm optimisation-based deep belief network for traditional Chinese medicine data processing strategies

Wenqiao Ding*

School of Biological and Food Engineering,
Jilin University of Chemical Technology,
Jilin, 132022, China
Email: ture2025@163.com
*Corresponding author

Chongli Xu

School of Medical Technology,
Chongqing Medical and Pharmaceutical College,
Chongqing, 401331, China
Email: 10953@cqmpc.edu.cn

Ruiqi Zhang

Faculty of Business, Economic and Accountancy,
University Malaysia Sabah,
Jalan UMS, 88400, Malaysia
Email: zhang_ruiqi_db23@iluv.ums.edu.my
and
School of Biological and Food Engineering,
Jilin University of Chemical Technology,
Jilin, 132022, China

Abstract: To address the challenges of high dimensionality, nonlinearity, and small sample size in traditional Chinese medicine data, this study proposes a novel data processing strategy using a particle swarm optimised deep belief network. The method automates deep belief network hyperparameter tuning via particle swarm optimised to enable end-to-end feature learning and classification. On the traditional Chinese medicine systems pharmacology Danshen blood-activation dataset, particle swarm optimised deep belief network achieved an accuracy of 87.8% and an F1-score of 86.8%, surpassing both conventional models, (e.g., XGBoost at 85.2%) and unoptimised deep belief network (83.1%). It also attained 98.8% accuracy on the University of California Irvine Wine dataset, demonstrating strong generalisation. This work offers an automated, high-precision computational tool for traditional Chinese medicine data analysis, significantly enhancing model performance and interpretability.

Keywords: particle swarm optimisation; deep belief networks; DBNs; traditional Chinese medicine data; hyperparameter optimisation; feature learning.

Reference to this paper should be made as follows: Ding, W., Xu, C. and Zhang, R. (2025) 'A particle swarm optimisation-based deep belief network for traditional Chinese medicine data processing strategies', *Int. J. Information and Communication Technology*, Vol. 26, No. 49, pp.36–56.

Biographical notes: Wenqiao Ding is a teacher in the School of Biological and Food Engineering at Jilin University of Chemical Technology, China. She received her PhD from the Harbin Normal University, China, in 2022. She has published over 20 academic papers, including one in SCI top journals. Her research interests include molecular genetics and microbiology.

Chongli Xu is a teacher in the School of Medical Technology at Chongqing Medical and Pharmaceutical College, China. He obtained his Bachelor's degree from the Jilin University (2004) and Master's degree from the Ningxia University (2007), China. He has published more than 70 articles, ten of which have been indexed by SCI. His main research direction is microbial molecular biology.

Ruiqi Zhang is a teacher in the School of Biological and Food Engineering at Jilin University of Chemical Technology, China. She obtained her Bachelor's degree from the Jilin University of Finance and Economics (2015) and Master's degree from the Beihua University (2022), China. At present, she is pursuing a doctoral degree at the University Malaysia Sabah. She has published two articles. Her main research direction is agricultural economics, particle swarm optimisation, and rural development.

1 Introduction

Traditional Chinese medicine (TCM), which is based on the knowledge of the Chinese people for hundreds of years, is slowly getting more attention and acceptance from modern medical systems throughout the world because of its principles about holistic, systematic, and personalised diagnosis and treatment. However, the modernisation and internationalisation of TCM has always faced a core challenge: how to use modern scientific and technological language to explain its profound but abstract theoretical connotations, (e.g., evidence, medicinal properties, and attribution) and complex mechanisms of action. In recent years, with the wide application of high-throughput sequencing, chromatography-mass spectrometry and other technologies in TCM research, a huge amount of biomedical data has been generated, including chemical composition, genomics, proteomics, and clinical diagnostic and therapeutic data of TCM, which signifies that the research of TCM has entered the era of 'big data' (Wang et al., 2012). These data contain valuable information to reveal the scientific principles of TCM, but they also pose unprecedented challenges to traditional data analysis and machine learning (ML) methods due to their high dimensionality, small sample size, high nonlinearity, and

significant noise (Zhang et al., 2023). One of the most important scientific questions for the growth of TCM inheritance and innovation is how to get useful information out of these complicated datasets quickly and accurately.

To address the aforementioned issues, an increasing number of academics have sought to integrate diverse ML algorithms into the study of TCM data. Early research concentrated on employing conventional models, including logistic regression, support vector machine (SVM), and random forest (RF), for tasks such as predicting TCM medicinal qualities, classifying evidence, and mining formula efficacy, yielding notable results (Zhang et al., 2019; Zhou et al., 2010). For example, Zhang et al. (2019) constructed a five-element medicinal property prediction model for Chinese medicine compounds using molecular descriptors and the RF algorithm, while Zhou et al. (2010) applied SVM to classify clinically collected evidence data. Although these shallow models perform well in some scenarios, their representation learning ability is limited, and it is difficult to capture their underlying complex nonlinear feature structures and interaction relationships when dealing with high-dimensional data, and the model performance easily reaches a bottleneck with insufficient generalisation ability (LeCun et al., 2015). In other words, these methods still rely largely on expert experience for manual feature design and screening, failing to realise end-to-end deep feature automatic learning and limiting their ability to mine deep knowledge in complex TCM big data.

Deep belief network (DBN) is a standard deep generative model that can learn very abstract feature representations of input data by stacking its fundamental component, restricted Boltzmann machine (RBM), with unsupervised greedy layer-wise pre-training (greedy layer-wise pre-training). The RBM is the most important aspect of greedy layer-wise pre-training. It can learn very abstract representations of the input data very well. It has worked quite well in several areas, including bioinformatics, speech processing, and picture recognition (Hinton et al., 2006; Fischer and Igel, 2012). It is especially good at processing high-dimensional and complex biomedical data because it can get beyond the problem of gradient vanishing in typical neural networks and learn good initial weights from unlabeled data (Min et al., 2017). Recently, some people have tried to use DBN in Chinese medicine. For example, Liu et al. (2016) utilised DBN to learn features from the chemical composition data of TCM to predict its target proteins; and Fang et al. (2017) explored the use of DBN to integrate multi-omics data for biomarker discovery of TCM evidence. These preliminary explorations demonstrate the great potential of DBN in revealing the laws of complex systems in TCM, and foreshadow that deep learning will become an important engine to promote informatisation and intelligent research in TCM.

However, the successful application of DBN to TCM practice still faces a key technical barrier that cannot be ignored: the optimisation problem of model hyperparameters. The performance of a DBN model is significantly affected by the arrangement of numerous hyperparameters, including its network architecture, (e.g., the count of hidden layers and neurons per layer) and training parameters, (e.g., learning rate and iteration count) (Feurer and Hutter, 2019). Currently, the selection of hyperparameters mostly relies on researchers' a priori knowledge, tedious manual debugging, or extensive grid search, a process that is not only time-consuming and subjective, but also highly susceptible to falling into local optimality, making it difficult to obtain stable and superior model performance (Yang and Shami, 2020). This excessive dependence on expert experience and the challenge of 'manual parameterisation' have significantly obstructed the broad implementation and advocacy of DBN in the domain of

TCM, particularly among TCM researchers who possess limited expertise in ML. Therefore, the development of a method that can automatically and efficiently optimise DBN hyperparameters is of critical importance to unleash the full potential of deep learning in TCM big data analysis.

Meta-heuristic algorithms (MHAs) are great tools for solving these kinds of hard optimisation issues. Many algorithms exist, but particle swarm optimisation (PSO) techniques are preferred because they have simple ideas, few parameters, are easy to use, and can search the whole space well (Wang et al., 2018). PSO mimics how birds behave in groups and find the best solutions in a complicated search space by having people share knowledge and work together finding the optimal solution. It has been effectively utilised to enhance the parameters and architecture of diverse ML models, such as parameter optimisation for SVMs (Lin et al., 2008) and weight training for neural networks (Settles and Soule, 2005). Some studies have started to look into the combination of PSO with deep learning in the last few years. Some studies have started to look into how PSO and deep learning can work together in the last few years. Liang et al. (2006) used PSO to improve the hyperparameters of convolutional neural networks, which made image classification more accurate, Wang et al. (2018) used an improved PSO algorithm to find the best long short-term memory network (LSTM) architecture for a task that needed to predict the future. These works demonstrate the effectiveness and versatility of PSO for automated deep learning model design. However, throughout the existing literature, there is a gap in the research that specifically focuses on the characteristics of TCM data, systematically combines PSO with DBN, and works on solving its hyperparameter automated optimisation problem. Filling this gap can not only provide a powerful and easy-to-use intelligent data analysis tool for TCM researchers, but also promote the deep integration of AI technology and traditional medicine from the methodological level, which has significant theoretical innovation value and practical application prospects.

2 Relevant technologies

2.1 Application of ML in TCM data processing

The complexity and specificity of Chinese medicine data have prompted researchers to continuously explore and apply various advanced ML methods to mine its deep value. Most of the early and current studies are mainly built on traditional ML models. For example, Zhang et al. (2019) worked on constructing an association model between the chemical components of TCMs and their traditional medicinal properties (cold, hot, warm, and cool), and they successfully predicted the medicinal properties of compounds based on a large number of molecular descriptors by using a RF algorithm, which provided a computational basis for understanding the material basis of the action of TCMs. In terms of clinical diagnosis, the work of Zhou et al. (2010) shows that SVM can achieve a certain degree of accuracy in the classification of evidence when dealing with pre-processed clinical evidence questionnaire data, which provides a technical path for the objectification of TCM evidence. In addition, Luo et al. (2020) utilised complex network analysis and Bayesian probabilistic modelling for data mining of classical Chinese medicine formulas, revealing the potential combinatorial mechanism behind the ‘ruler, minister, and enabler’ pattern in the formulas, which demonstrates the potential of

ML in interpreting the holistic concept of Chinese medicine. Despite the promising initial results of these shallow models, they generally rely on a fine-grained feature engineering process that often requires the intervention of expert knowledge. Model performance is largely determined by the quality of the input features rather than the learning ability of the model itself. For the high-dimensionality, nonlinearity and complex interactions among features prevalent in TCM data, the representational ability of these models appears to be insufficient and prone to performance bottlenecks, and their generalisation ability is difficult to ensure (LeCun et al., 2015).

2.2 DBNs and their applications in biomedicine

Breakthroughs in deep learning have brought a new light to address the above limitations. Among many deep learning models, DBN has attracted much attention due to its unique unsupervised pre-training mechanism. DBN is made up of a stack of multi-layer RBMs. There are two sections to its training process: a layer-by-layer pre-training phase that is not overseen and a fine-tuning phase that is supervised. This paradigm, pioneered by Hinton et al. (2006), effectively solves the problem of gradient vanishing in the training of deep neural networks and is capable of learning hierarchical, abstract feature representations of data. The research of Fischer and Igel (2012) elaborated the mathematical foundation and training algorithms of RBM and DBN, laying a solid theoretical foundation for their applications in many fields. In the field of biomedical informatics, the advantages of DBN have been fully realised. Min et al. (2017) systematically reviewed the successful applications of deep learning, including DBN, in biomarker discovery, drug repositioning, and genomics, highlighting its capacity to automatically extract discriminative features from raw high-throughput data. Specifically for TCM informatics research, some cutting-edge explorations have begun to emerge. Liu et al. (2016) developed a bioinformatics analysis platform called biological analysis tool for molecular mechanism of TCM (BATMAN-TCM), in which an attempt was made to utilise DBN to learn features from the chemical composition data of TCM in order to predict its potential protein targets, initially validating the applicability of deep learning in TCM network pharmacology research. Similarly, Fang et al. (2017) in exploring the therapeutic mechanism of TCM for Alzheimer's disease, DBN was applied to perform feature dimensionality reduction and extraction of integrated multi-omics data with a view to discovering more robust biological markers. These studies signify that DBN holds great promise in processing high-dimensional and complex biomedical data, providing compelling evidence for its in-depth application in the field of Chinese medicine.

2.3 Hyperparametric optimisation with metaheuristic algorithms

But there is a huge problem with the amazing performance: DBN and other deep learning models do not work well if their hyperparameters, including the number of layers and neurons per layer in the network and the learning rate and momentum factor during training, are set up wrong. Traditional manual parameter tuning is not only tedious and time-consuming, but also strongly relies on operator's experience, which is difficult to reproduce and prone to fall into local optimal solutions (Yang and Shami, 2020). Automated hyperparameter optimisation (HPO) has thus become a core aspect of ML applications. Standard techniques like grid search and random search work well in

low-dimensional spaces, but as the number of hyperparameters increases, their processing cost rises quickly and inefficiently (Bergstra and Bengio, 2012). For this reason, more efficient sequential model-based optimisation (SMBO) methods, such as Bayesian optimisation, have been proposed, and Feurer and Hutter have systematically combed through them, pointing out that they guide the search by constructing an agent model, thus reducing the time required to achieve superior performance. Direction, thereby reducing the number of verifications required to achieve superior performance (Feurer and Hutter, 2019).

Meanwhile, meta-heuristic (MH) optimisation algorithms provide another powerful way to solve such complex black-box optimisation problems. Kennedy and Eberhart came up with the PSO algorithm, which models how birds act in groups and uses the group's best prior experiences to assist people locate the best answer in a convoluted search area (Wang et al., 2018). PSO is frequently used for optimising ML models because it is straightforward to understand, has few parameters, is easy to implement, and has a high global search ability. Lin et al. (2008) successfully utilised PSO to automatically find the optimal kernel function parameters and penalty coefficients for SVMs, which significantly improved the classification accuracy. Settles and Soule (2005), on the other hand, explored the use of PSO for training neural network connection weights. Researchers have started to use PSO in deeper levels of model automation design in the last few years. The work of Liang et al. (2006) shows that PSO can effectively optimise the structural hyperparameters of convolutional neural networks. And a review by Wang et al. (2018) provides a comprehensive overview of the various variants of PSO and its recent advances in solving real-world engineering optimisation problems, including neural network structure design. These research results strongly demonstrate the significant advantages of PSO as a robust and general-purpose optimiser for automating deep learning model configurations.

In summary, existing research has made significant progress in their respective fields: the application of ML in TCM is on the rise, DBN shows great potential in feature learning of biomedical data, and PSO has proved its effectiveness in automated model optimisation. However, literature combing reveals that there is a gap in research that specifically addresses the intrinsic characteristics of TCM data and systematically and deeply integrates PSO with DBN to thoroughly solve the problem of automatic optimisation of DBN model hyperparameters. This study aims to fill this gap and explore an efficient and automated deep learning model optimisation strategy for TCM data processing.

3 Methodology

3.1 Overall framework

The main idea behind the particle swarm optimised-DBN (PSO-DBN) model for processing TCM data that this paper proposes is to use the PSO algorithm's capacity to scan the whole space to find the best hyperparameter combinations for the DBN. This method tries to solve the problems that come with manually tuning parameters and to improve the DBN's ability to learn features and classify complex TCM data. The model is an iterative closed-loop optimisation process, as indicated in the overall framework diagram in Figure 1. The whole process starts with the initialisation of the PSO

population, where each particle position vector encodes a specific set of DBN hyperparameters. After that, the PSO algorithm tells the particles where to go and how to look for things based on how fit they are (the performance metrics of the DBN model configured with those hyperparameters on the validation set, such as classification accuracy). In each iteration, the combination of hyperparameters represented by each particle is used to instantiate and train a DBN model, whose performance evaluation results are then fed back to the PSO algorithm to update the individual optimal and global optimal solutions. This procedure goes on until the end conditions are met, such as when the maximum number of iterations is reached or the performance levels off. At that point, the hyperparameters for the global optimal particles and the best DBN model trained with those parameters are produced. The framework achieves complete automation of hyperparameter optimisation and model training, providing a powerful end-to-end analysis tool for TCM researchers.

3.2 DBN theory

A DBN is a kind of probabilistic generative model that has a lot of layers of random latent variables. RBMs are the most important parts of a DBN. An RBM has a visible layer **v** and a hidden layer **h**. The neurons in the layers are not connected, but the neurons between the layers are completely coupled. Its energy function is defined as:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{ij} h_j \tag{1}$$

where v_i and h_j stand for the two states of the visible layer cell i and the hidden layer cell j, respectively (in practice, the visible layer can also be a continuous value), a_i and b_j are their bias terms, respectively, and w_{ij} is the weight connecting v_i and h_j. Based on this energy function, the probability that the network is given a (**v**, **h**) joint configuration is:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})} \tag{2}$$

where Z is the component that makes things normal, also known as the collocation function. The conditional probability that a hidden unit is activated given the state of the visible unit is:

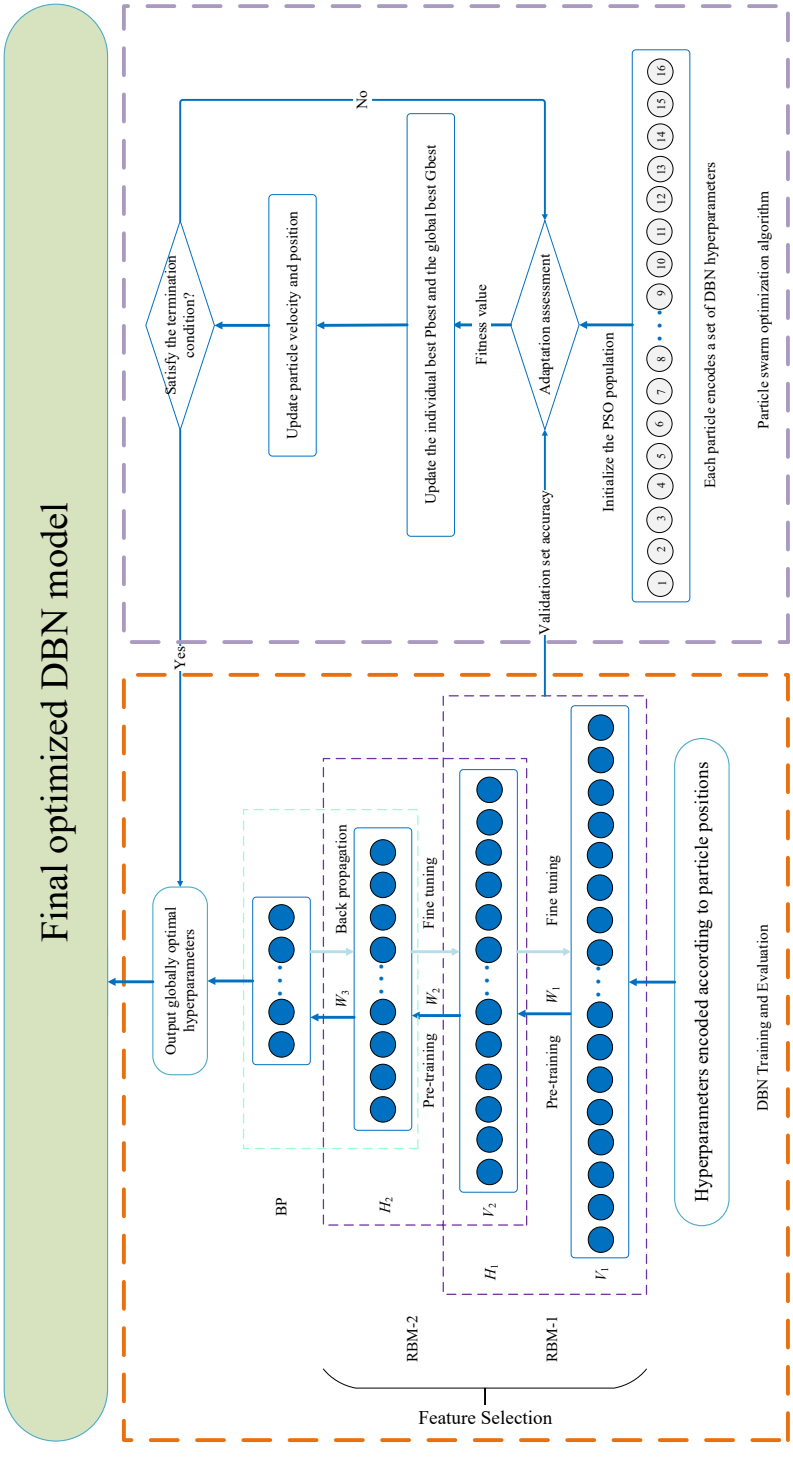
$$P\left(h_j = 1 \mid \mathbf{v}\right) = \sigma\left(b_j + \sum_i v_i w_{ij}\right) \tag{3}$$

Similarly, given the state of the hidden unit, the conditional probability that the visible unit is activated is:

$$P\left(v_i = 1 \mid \mathbf{h}\right) = \sigma\left(a_i + \sum_j h_j w_{ij}\right) \tag{4}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the function that activates the Sigmoid.

Figure 1 PSO-DBN hybrid model schematic framework diagram (see online version for colours)



Hinton et al. (2006) came up with the greedy layer-by-layer unsupervised pre-training approach that is used to train the DBN. The process begins at the bottom layer, when data is sent to the first RBM's visible layer, and learns to obtain the parameters $\mathbf{W}^1$, $\mathbf{a}^1$, $\mathbf{b}^1$ of the first RBM layer by approximating the maximised likelihood function through the contrastive dispersion (contrastive divergence, CD-k) algorithm. The core weight update rule for the CD-k algorithm is:

$$\Delta w_{ij} = \epsilon\left(\left\langle v_i h_j \right\rangle_{data} - \left\langle v_i h_j \right\rangle_{recon}\right) \tag{5}$$

where ϵ is the learning rate, $\langle\cdot\rangle_{data}$ denotes the expectation on the data distribution, and $\langle\cdot\rangle_{recon}$ denotes the expectation on the reconstructed distribution after k steps of Gibbs sampling (usually $k = 1$). The activation probability output of the hidden layer of the first layer of RBM is used as the 'visible' input data for the next layer of RBM, and so on, training all RBMs layer by layer. This pre-training process does not require labels, and can effectively initialise the weights of the network to a better value, capturing the data hierarchical statistical structure. To make a deep neural network, you add a final output layer (such a Softmax classifier) to the top of the DBN. This is because RBMs are already trained. After then, a supervised error backpropagation (backpropagation) technique is used to fine-tune (fine-tuning) all of the parameters to get the best possible final classification result.

3.3 PSO algorithm

PSO is a way to improve population intelligence by copying how birds behave in groups. In PSO, each 'particle' is a probable answer in the search space. In this case, it is a set of DBN hyperparameters. For a D-dimensional search space, the location of the i^{th} particle can be described as:

$$\mathbf{x}_i = \left(x_{i1}, x_{i2}, ..., x_{iD}\right) \tag{6}$$

Its flight speed is expressed as:

$$\mathbf{v}_i = \left(v_{i1}, v_{i2}, ..., v_{iD}\right) \tag{7}$$

Each particle keeps track of the best location $\mathbf{pbest}_i$ that it looks for, and the best position **gbest** that all the particles in the group look for is written down.

At each iteration t, the speed and position of each particle change according to the following formula:

$$\mathbf{v}_i^{t+1} = \omega v_i^t + c_1 r_1 \left(\mathbf{pbest}_i^t - x_i^t\right) + c_2 r_2 \left(\mathbf{gbest}^t - x_i^t\right) \tag{8}$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \tag{9}$$

where ω is the inertia weight, which is used to find a balance between exploring the world and using local resources. A bigger ω is better for searching the whole world, whereas a smaller w is better for searching a narrow area. The acceleration coefficients, c_1 and c_2, control how big the steps are that the particles take toward individual and global optimisation, respectively.

$$c_1 = c_2 = 2 \tag{10}$$

where r_1, r_2 are random numbers that are evenly spread out across the range [0, 1] utilised to introduce randomness in the search.

3.4 PSO-DBN hybrid model

3.4.1 Hyperparametric coding and search spaces

The first step in applying PSO to DBN hyperparameter optimisation is to design a suitable coding scheme to map the hyperparameters to the particle position space. In this study, a particle represents a set of key hyperparameters of DBN. Each dimension x_{id} of a particle corresponds to one hyperparameter, and its search range needs to be predefined. The hyperparameters optimised in this study and their typical search ranges are shown in Table 1.

Table 1 Hyperparameter space for PSO search

Hyperparameter	*Sign*	*Search scope*	*Type*
Number of RBM layers	L	[1, 3]	Integer
Number of hidden neurons in layer 1	U_1	[50, 500]	Integer
Number of hidden neurons in layer 2	U_2	[50, 300]	Integer
Number of hidden neurons in layer 3	U_3	[10, 100]	Integer
Pre-training learning rate	ϵ_{pre}	[0.001, 0.1]	Continuous
Fine-tuning learning rate	ϵ_{ft}	[0.0001, 0.01]	Continuous
Momentum factor	α	[0.5, 0.9]	Continuous

For a particle $\mathbf{x}_i$, its position may be encoded as $[L, U_1, U_2, U_3, \epsilon_{pre}, \epsilon_{ft}, \alpha]$. For integer-type parameters, it is necessary to perform a rounding operation on them when decoding.

3.4.2 Design of the fitness function

The PSO algorithm uses the fitness function to help it find the right path. The main purpose of this study is to make a DBN model that is very good at classifying things. To do this, the validation accuracy is used as the fitness function:

$$Fitness(\mathbf{x}_i) = Accuracy_{val} \tag{11}$$

where $Accuracy_{val}$ is the accuracy of the classification calculated on a separate validation set of the DBN model that was built and trained using the hyperparameters stored by particle $\mathbf{x}_i$. The PSO algorithm's goal is to maximise the fitness function.

3.4.3 Algorithmic processes

The full algorithmic flow of the PSO-optimised DBN is as follows: first, initialisation is done: set the parameters that the PSO algorithm needs, like the population size M, the inertia weight ω, the individual learning factor c_1, and the social learning factor c_2, as well as the maximum number of iterations $T_{\max}$; randomly initialise all particles with their positions $\mathbf{x}_i^0$ and velocities $\mathbf{v}_i^0$ in a defined hyperparameter search space; and set each

particle's individual optimum $\mathbf{pbest}i^0$ as its initial position. $\mathbf{v}_i^0$ and set the individual optimum $\mathbf{pbest}i^0$ of each particle as its initial position, which in turn determines the initial global optimum $\mathbf{gbest}^0$. Subsequently, we enter the main loop with iterations from $t = 1$ to $T_{\max}$: for each particle, its current position $\mathbf{x}_i^t$ is decoded as a set of specific DBN hyperparameters; the DBN structure is constructed using these hyperparameters, and unsupervised layer-by-layer pre-training and supervised fine-tuning are performed sequentially on a training set; followed by prediction of the validation set using the trained DBN to predict the validation set, calculate the fitness $Fitness(\mathbf{x}_i^t)$, and update the individual optimum accordingly: if the current fitness is better, let $\mathbf{pbest}_i^t = \mathbf{x}_i^t$. After completing the evaluation of all particles, update the global optimum $\mathbf{gbest}^t$, the position of the most adapted particle in the current population. Finally, update the velocity $\mathbf{v}_i^{t+1}$ and position $\mathbf{x}_i^{t+1}$ of all particles according to the velocity and position update formula, while ensuring that the particle positions do not go beyond the search boundary. At the end of the loop, the global optimal solution **gbest**, the optimal hyperparameter combination, and its corresponding DBN model are output.

4 Experimental validation

4.1 Dataset and experimental setup

This work employs two datasets to thoroughly assess the performance of the proposed PSO-DBN model: a real dataset tailored for a specific TCM application situation and a commonly utilised standard dataset to validate the model's generalisability.

The methodology for constructing the dataset on danshen's role in activating blood circulation and alleviating blood stasis for the Traditional Chinese Medicine Systems Pharmacology Database and Analysis Platform (TCMSP) was previously detailed (Ru et al., 2014). This study involved downloading all the chemical parts of danshen (Salvia miltiorrhiza) and their absorption, distribution, metabolism, excretion (ADME) properties from the TCMSP database. After data cleaning (removing duplicates, processing missing values), 202 samples of valid compounds were finally retained. Each sample was described by 12 numerical type features, including key pharmacological attributes such as oral bioavailability (OB), drug-likeness (DL), Apparent Log P (AlogP), and topological polar surface area (TPSA). The samples were manually labelled based on the Encyclopaedia of TCM (ETCM) database and authoritative literature. 89 compounds that had been tested and shown to 'activate blood circulation and remove blood stasis' were labelled as positive samples (1), while the other 113 compounds were labelled as negative samples (0). The dataset is very imbalanced; the ratio of positive to negative samples is roughly 1:1.27, which increases the difficulty of the classification task and is more in line with real-world scenarios.

The University of California, Irvine (UCI) Wine dataset is a classic ML standard dataset that comes from the chemical study of three different types of wine grown in the same area of Italy (Bender et al., 2023). The dataset contains 178 samples, each characterised by 13 physicochemical metrics, (e.g., alcohol content, malic acid, ash, magnesium content, flavonoid content, etc.), and the task is to classify the wines into one of the three varieties. A lot of people use this dataset to test how well categorisation

algorithms work. The dataset's statistical information is shown in Table 2. This dataset is widely used to evaluate the performance of classification algorithms. Table 2 shows the statistical information of the dataset. The UCI Wine dataset was chosen to test the generalisation ability of the PSO-DBN model on different domains and data structures. This dataset is a classic benchmark dataset in the field of ML, which has the following advantages: first, its clear structure, moderate feature dimensions (13 physicochemical metrics), and completely balanced sample categories (59 samples for each of the three categories) can effectively exclude the interference of data imbalance on the performance of the model; second, all of its features are continuous-type values, which are similar to the attributes of the TCMSP dataset, and ensure that the comparative experiments are fairness; third, as a public benchmark, its research results are easy to be compared with other algorithms at home and abroad horizontally, thus more strongly proving the generality and superiority of the model proposed in this study.

Table 2 Statistical information on datasets

Dataset	*Number of samples*	*Number of features*	*Number of classes*	*Number of positive samples*	*Number of negative samples*
TCMSP	202	12	2	89	113
UCI Wine	178	13	3	-	-

A ratio of 7:1.5:1.5 is used to divide all datasets into training, validation, and test sets for the experiment. The PSO algorithm's fitness and hyperparameter optimisation are verified using the validation set, while the final test set is employed to report the final performance and ensure that the results are not distorted. The training set is employed to learn the model parameters, while the final test set is only used once. The selection of the dataset division ratio mainly considers the total size of the dataset. The total sample sizes of the TCMSP and Wine datasets are 202 and 178, respectively, which both belong to the small sample category. If a smaller validation/test set ratio, (e.g., 10%) is used, its absolute sample size will be less than 20, and the variance of the evaluation results will be larger and less stable. Increasing both the validation and test set ratios to about 15% is intended to ensure that these two key evaluation sets have sufficient sample size (about 30 samples) to obtain more stable and credible performance estimates and to reduce the evaluation bias due to the randomness of a single division.

The Z-score was normalised for all continuous features. The following parameters were established for the PSO algorithm: a population size of $M = 20$, a maximum number of iterations $T_{\max} = 50$, an inertia weight ω that decreases linearly from 0.9 to 0.4 and an acceleration constant $c1 = c2 = 2.0$. The inertia weight ω controls the extent to which a particle's previous velocity affects its current velocity. In this study, we adopt the linearly decreasing inertia weight (LDIW) strategy, which is a standard optimisation technique in the PSO field. The principle lies in the following: at the early stage of the search, setting a larger value of ω (0.9) helps the particles to maintain a higher flight speed, enhances the global exploration ability, and avoids falling into the local optimum prematurely; at the late stage of the search, a smaller value of ω (0.4) can weaken the global exploration and enhance the local exploitation ability, so that the particles can carry out a fine search in the area of the suspected optimal solution, and thus improve the convergence accuracy. This range of values (0.9 -> 0.4) is a widely adopted empirical validity interval and has been shown to be well-balanced in pre-experiments for this particular problem (Natekin and Knoll, 2013). The population size is set as a trade-off between search capability and

computational efficiency. The dimension of the hyperparametric search space in this study is seven dimensions, which is of medium complexity. A population that is too small, (e.g., < 10) may be insufficiently diverse and prone to premature convergence, while a population that is too large, (e.g., > 50) will significantly increase the computational burden as each particle is required to fully train a DBN model. Setting the population size to 20 was suggested by the classical PSO literature (the population size is usually set between 20–50), and it was observed in the pre-experiments that this size provides enough diversity in a reasonable computational time to explore the search space efficiently.

All comparative experiments were repeated ten times on the same training/validation/testing divisions and reported their average performance metrics ± standard deviation to eliminate the effect of randomness. Repeating the experiment ten times is a common and accepted practice in ML research to counteract the effects of random factors. These random factors include random partitioning of the dataset, random initialisation of the PSO algorithm, and random weight initialisation of the DBN network itself. Repeating the experiment 10 times strikes a good balance between computational cost and statistical reliability: on the one hand, it is sufficient to smooth out the fluctuations caused by one extreme random result by averaging out the results, providing a more representative central tendency of the performance; on the other hand, the results of 10 experiments are sufficient to perform a simple statistical test, (e.g., paired t-tests) to assess the significance of the differences in the performances, which is much more reliable than repeating the experiment only 3–5 times. Five metrics: accuracy, precision, recall, F1-score, and macro-average area the receiver operating characteristic curve (AUC) was utilised to evaluate performance in a thorough way.

4.2 Contrasting models and ablation experiments

Various representative state-of-the-art ML algorithms were employed in this study as benchmark comparison models to illustrate the superiority of the PSO-DBN model. The Scikit-learn library and the Keras library, which were developed by Pedregosa et al. (2011), were employed to implement all of them, and each was hyper-parametrically tuned to attain optimal performance.

1 SVM: SVM with radial basis function (RBF) kernel with penalty coefficients C and kernel coefficients gamma optimised using lattice search. SVMs excel at handling high-dimensional data and are powerful benchmark models (Zhou et al., 2010).

2 RF: a representative algorithm for integrated learning that sets the number of trees to 100 and uses default values for the rest of the parameters. RF is effective in capturing nonlinear relationships between features and is insensitive to overfitting (Zhang et al., 2019).

3 Extreme gradient boosting (XGBoost): an efficient, state-of-the-art gradient boosting decision tree algorithm proposed by Natekin and Knoll (2013), with extensive parameter tuning in this study, including the learning rate, the maximum depth, and the proportion of subsamples.

4 Standard DBN: for the purpose of ablation study, a standard DBN model without PSO optimisation is set up for this experiment. Its hyperparameters (number of layers, number of nodes, learning rate) are set manually based on common

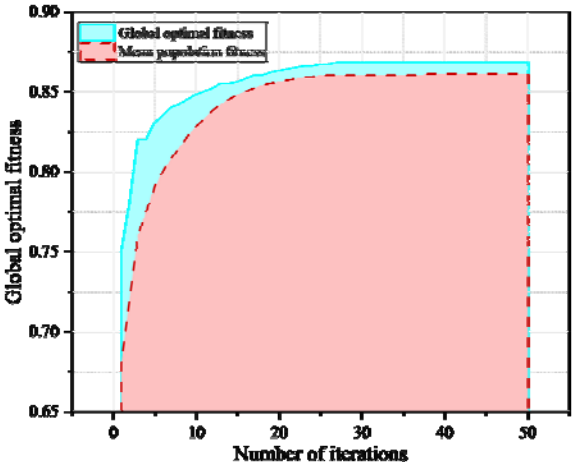
experience and preliminary experiments, (e.g., structure of [12-50-30], pre-learning rate of 0.01, and fine-tuned learning rate of 0.001) with the aim of verifying the necessity and effectiveness of the PSO optimisation session.

5 Bayesian optimisation-DBN (BO-DBN): this experiment uses Bayesian optimisation (BOP) based on Gaussian processes to find the best hyperparameters for the DBN. This is done so that it can be compared to another popular method for hyperparameter optimisation, with a core library based on Gaussian process optimisation (GPyOpt), in order to fairly compare the efficiency of PSO with that of another black-box optimisation approach (Snoek et al., 2012).

4.3 Analysis of the hyperparameter optimisation process

The process of PSO optimisation of DBN hyperparameters is shown in Figure 2, whose vertical axes are the average fitness (validation set accuracy) and the global best fitness of the population. From the figure, it can be clearly observed that in the early stage of optimisation (the first 15 generations), both the average fitness of the population and the global best fitness increase rapidly, which indicates that the PSO algorithm efficiently explores and quickly discovers the hyperparameter regions with excellent performance in the vast search space. After about 25 generations, the growth of the curves flattens out and finally converges, indicating that the algorithm finds a relatively stable optimal solution region around which most of the particles in the population are clustered. The entire optimisation process shows how the PSO method can explore the full world and take advantage of local resources. It can automate and quickly finish the hyperparameter optimisation operation, which would normally take a lot of work.

Figure 2 PSO-DBN hybrid model schematic framework diagram (see online version for colours)



4.4 Model performance comparison and analysis

A comparison of the comprehensive performance of all models on the two dataset test sets is shown in Table 3, and a comparison of their F1-scores and accuracies is shown in Figure 3.

Figure 3 Performance comparison of different models (see online version for colours)

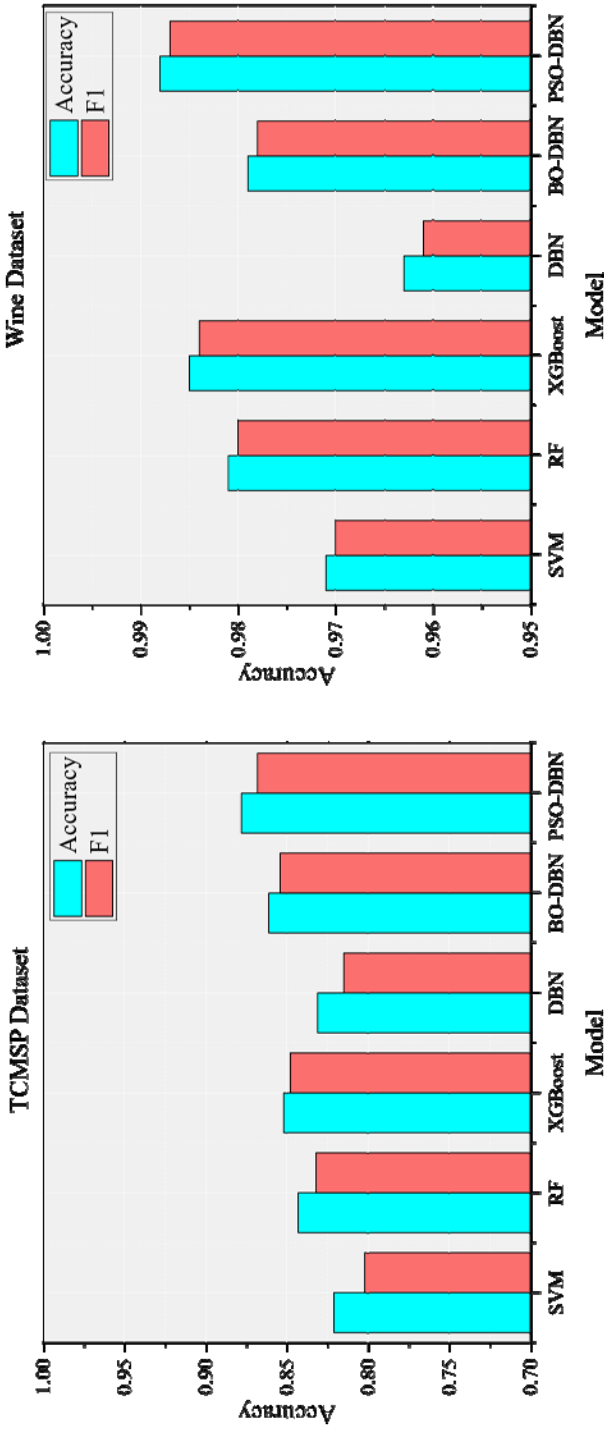


Table 3 Detailed performance metrics comparison for all models

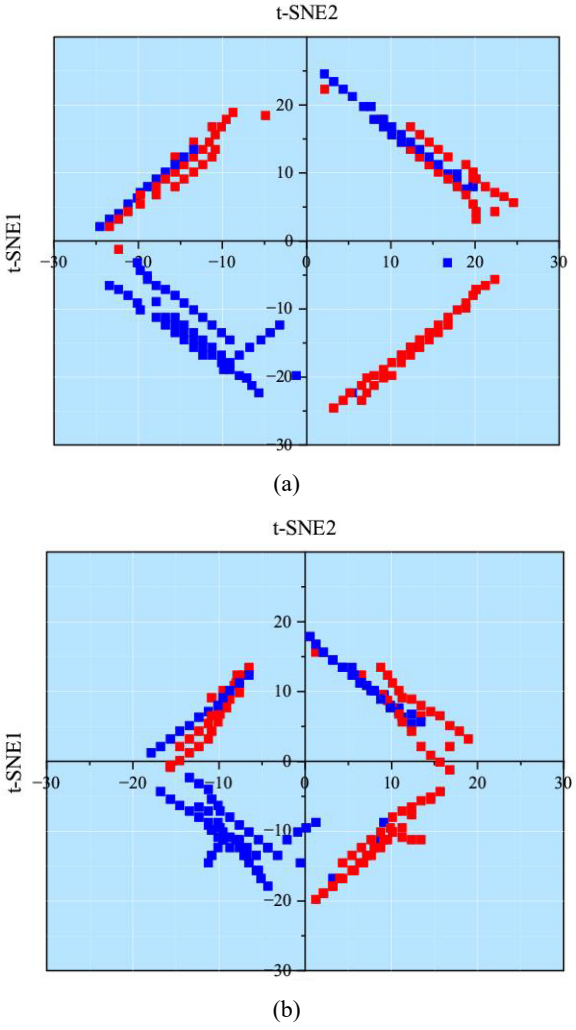
Model	*TCMSP dataset*			*Wine dataset*		
	Accuracy	*F1-score*	*AUC*	*Accuracy*	*F1-score*	*AUC*
SVM	0.821 ± 0.024	0.802 ± 0.031	0.888 ± 0.019	0.971 ± 0.018	0.970 ± 0.020	0.997 ± 0.005
RF	0.843 ± 0.019	0.832 ± 0.022	0.925 ± 0.015	0.981 ± 0.015	0.980 ± 0.016	0.999 ± 0.002
XGBoost	0.852 ± 0.021	0.848 ± 0.025	0.932 ± 0.017	0.985 ± 0.012	0.984 ± 0.013	0.999 ± 0.001
DBN	0.831 ± 0.032	0.815 ± 0.038	0.901 ± 0.028	0.963 ± 0.022	0.961 ± 0.024	0.995 ± 0.007
BO-DBN	0.861 ± 0.017	0.854 ± 0.020	0.941 ± 0.014	0.979 ± 0.016	0.978 ± 0.017	0.998 ± 0.003
PSO-DBN	0.878 ± 0.014	0.868 ± 0.016	0.958 ± 0.011	0.988 ± 0.010	0.987 ± 0.011	0.999 ± 0.001

Analysing Table 3 and Figure 3 shows that our proposed PSO-DBN model achieves the most superior performance (accuracy: 87.8%, F1-score: 86.8%, AUC: 0.958) in all evaluation metrics on the TCMSP danshen dataset. Its performance was significantly better than that of the manually tuned DBN model (accuracy: 83.1%), which fully demonstrates the key role of automatic hyperparameter optimisation (AHO) in unlocking the potential of the DBN model. Compared with another automated optimisation method, BO-DBN, PSO-DBN also shows some advantages, which may be attributed to the powerful exploration capability of PSO's population parallel search mechanism in complex parameter spaces. Compared with traditional ML models, PSO-DBN improves about 2.6 percentage points in accuracy compared with the best-performing XGBoost model, which reflects the advantage of DBNs in automatically learning deep abstract features of high-dimensional data. On the UCI Wine dataset, PSO-DBN also achieves a leading level (accuracy: 98.8%), demonstrating the model's good generalisation.

4.5 Feature visualisation and analysis

This study employed the t-SNE dimensionality reduction technique to graphically depict the original feature space and the feature space obtained from the last hidden layer of the PSO-DBN model, therefore clarifying the model's advantages in a more intuitive manner. The findings are depicted in Figure 4. On the left side of the initial feature visualisation map, the positive and negative sample points are heavily mixed together, and the decision boundary is very vague. While in the deep feature visualisation graph on the right side, which has been learned by PSO-DBN, it can be clearly seen that the sample points of the two categories form two tighter and more separated clusters. This shows that the PSO-DBN model can learn the most important and unique high-level features from the original high-dimensional ADME attributes. This powerful feature learning capability is the fundamental reason for its excellent classification performance.

Figure 4 Feature space t-SNE visualisation, (a) original features, (b) PSO-DBN learned features (see online version for colours)



4.6 Experimental results and analysis

In this study, a PSO algorithm was successfully combined with a DBN to construct an automated deep learning framework (PSO-DBN) for high-dimensional, nonlinear TCM data. The model demonstrates markedly superior performance compared to conventional ML models and non-optimised DBNs in the binary classification test of TCMSP Danshen, which activates blood circulation and alleviates blood stasis. This success is mainly attributed to two core factors: first, the powerful unsupervised feature learning capability possessed by the DBN itself, which is able to automatically extract deep, abstract feature representations related to the efficacy of TCMs from the raw ADME attributes, as intuitively evidenced by the t-SNE visualisation. As LeCun et al. (2015) have pointed out the main power of deep learning is that it can automatically find

distributed feature representations of data using multi-layer nonlinear transformations, thus dispensing with the tedious and expert knowledge-dependent manual feature engineering. Second, the PSO algorithm, as an efficient global optimiser, successfully automates the most labour-intensive and critical hyperparameter debugging process in the DBN model. This study confirms that the problem of deep learning models being highly sensitive to hyperparameters, as pointed out by Yang and Shami (2020), can be efficiently solved by MHAs. PSO's population-intelligent searching mechanism enables it to efficiently find a combination of configurations that are close to the global optimum in the complex space of hyperparameters, which fully unleashes the potential performance of the DBN model. This is consistent with the conclusion of the robustness of PSO in complex optimisation problems mentioned in Wang et al. (2018).

The importance of this work is not only that it suggests a better way to classify things, but also that it makes a methodological contribution. It offers an innovative, comprehensive solution for the analysis of 'small-sample, high-dimensional' data within the realm of Chinese medicine. End-to-end' means that the model receives the raw data input and directly outputs the final task result without human intervention in the middle process. Compared with traditional methods: manual feature engineering relies heavily on the knowledge of domain experts, and the researcher needs to carefully design, filter, and transform features based on experience, (e.g., constructing new ratios or cross terms from ADME attributes), which is a time-consuming, subjective, and easy to lose information process. The PSO-DBN framework, on the other hand, achieves full automation: the DBN module automatically learns hierarchical abstract feature representations from raw data, and the PSO module automatically optimises the network structure and training parameters. This frees researchers from tedious 'parameter tuning' and 'feature engineering', and allows them to obtain a high-performance model by simply providing regularised raw data, which greatly reduces the technical barriers and allows domain experts to focus more on the problem itself rather than model implementation details. For a long time, Chinese medicine researchers have been limited by their deep domain expertise and limited computer science skills when applying complex ML models, and the PSO-DBN model greatly lowers this technical threshold, as researchers only need to prepare the data, and the model can automatically complete the whole process from feature learning to optimal model construction. This automated and intelligent data processing strategy is highly compatible with the trend of modernisation of TCM empowered by AI as envisioned by Zhang et al. (2023), it will speed up the change of TCM from 'empirical medicine' to 'evidence-based medicine' and 'precision medicine.' It is a strong computer program that speeds up the change of TCM from 'empirical medicine' to 'evidence-based medicine' and 'precision medicine'.

However, this study also has some limitations. First, the operation of the PSO algorithm requires multiple trainings of the DBN model to evaluate the particle fitness, and its computational cost is still high compared to the training of a single model. To save computational cost, parallel computing can be used to evaluate multiple particles at the same time to shorten the time rigidly, and combined with agent modelling or early-stopping strategy to intelligently skip ineffective training, which significantly reduces the overall computational elapsed time. Second, although the database size in this study makes this cost within acceptable range, the computational efficiency still needs to be further optimised for ultra-large datasets. To improve the computational efficiency, parallel computing or distributed processing strategies can be introduced to assign the particle evaluation task to multiple nodes for simultaneous execution; early stopping

mechanisms, proxy models, or adaptive particle updating strategies can also be considered to reduce the number of unnecessary DBN trainings and accelerate the convergence process. Finally, like most deep learning models, PSO-DBN is still a 'black box' model to some extent. Although we indirectly explain its internal mechanism through feature visualisation, the final decision logic, (e.g., which combination of features determines the 'blood circulation' effect) is still not transparent. This may affect its application in clinical decision-making scenarios that require high interpretability.

Future work will be carried out in the following directions: first, we will explore more computationally efficient optimisation strategies, such as using PSO variants with adaptive parameters or combining with early stopping strategy to reduce the time cost of model evaluation. Secondly, we will talk about how to make the models easier to understand, and plan to introduce the explainable artificial intelligence (XAI) technique to interpret the trained DBN models and identify the original features and their interactions that contribute the most to the classification, so as to provide testable scientific hypotheses for the study of TCM pharmacological mechanisms. In addition, we will extend the application of the PSO-DBN model to more complex TCM research tasks, such as the mining of monarch-executive-ordination patterns of TCM compound prescriptions, the multi-marker classification of TCM evidence, and the prognosis prediction of TCM therapeutic effects, with a view to validating its broader applicability. Ultimately, we expect to develop an integrated software platform that encapsulates the algorithms in this study into a user-friendly graphical interface, which can be used 'out-of-the-box' by the majority of Chinese medicine researchers, and promote the popularisation and application of AI technology in the field of Chinese medicine.

5 Conclusions

In summary, this study proposes and validates a new strategy for Chinese medicine data processing based on PSO-DBN. By integrating the global optimisation capability of PSO and the deep feature learning advantage of DBN, the model effectively solves the core bottleneck of DBN model in TCM applications, which is difficult to manually debug hyperparameters. Experiments on both the publicly available TCMSP Danshen dataset and the UCI Wine dataset demonstrate that the PSO-DBN model significantly outperforms traditional ML models such as SVMs, RFs, and XGBoost, as well as the DBN benchmark model with manually tuned parameters, in terms of classification accuracy, F1-score, and AUC value. The primary result of this work is that AHO is essential for realising the potential of deep learning techniques in TCM, and PSO-DBN offers an effective and viable implementation pathway for this objective. This work not only advances the integration of intelligent optimisation algorithms and deep learning in theory, but also provides a powerful and easy-to-use data analysis tool for TCM researchers in practice, which is of positive significance in promoting the modernisation and international development of TCM.

Acknowledgements

This work is supported by the Jilin Province Science and Technology Development Plan Project Task Book (No. 20240602099RC), the Scientific research project of Jilin Provincial Department of Education (No. JJKH20230291KJ), the Major category of natural science and technology research projects in Chongqing Medical College (No. ygzzd2024105), and the Key Subject Project of Chongqing Medical College (No. ygz2025702).

Declarations

All authors declare that they have no conflicts of interest.

References

Bender, T., Sørensen, S., Kashani, A., Eldjarn Hjorleifsson, K., Hyldig, G., Hauberg, S., Belongie, S. and Warburg, F. (2023) 'Learning to taste: a multimodal wine dataset', *Advances in Neural Information Processing Systems*, Vol. 36, pp.7351–7360.

Bergstra, J. and Bengio, Y. (2012) 'Random search for hyper-parameter optimization', *The Journal of Machine Learning Research*, Vol. 13, No. 1, pp.281–305.

Fang, J., Wang, L., Wu, T., Yang, C., Gao, L., Cai, H., Liu, J., Fang, S., Chen, Y. and Tan, W. (2017) 'Network pharmacology-based study on the mechanism of action for herbal medicines in Alzheimer treatment', *Journal of Ethnopharmacology*, Vol. 196, pp.281–292.

Feurer, M. and Hutter, F. (2019) 'Hyperparameter optimization', *Automated Machine Learning: Methods, Systems, Challenges*, Vol. 1, pp.3–33.

Fischer, A. and Igel, C. (2012) 'An introduction to restricted Boltzmann machines', *Iberoamerican Congress on Pattern Recognition*, Vol. 18, pp.14–36.

Hinton, G.E., Osindero, S. and Teh, Y-W. (2006) 'A fast learning algorithm for deep belief nets', *Neural Computation*, Vol. 18, No. 7, pp.1527–1554.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) 'Deep learning', *Nature*, Vol. 521, No. 7553, pp.436–444.

Liang, J.J., Qin, A.K., Suganthan, P.N. and Baskar, S. (2006) 'Comprehensive learning particle swarm optimizer for global optimization of multimodal functions', *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp.281–295.

Lin, S-W., Ying, K-C., Chen, S-C. and Lee, Z-J. (2008) 'Particle swarm optimization for parameter determination and feature selection of support vector machines', *Expert Systems with Applications*, Vol. 35, No. 4, pp.1817–1824.

Liu, Z., Guo, F., Wang, Y., Li, C., Zhang, X., Li, H., Diao, L., Gu, J., Wang, W. and Li, D. (2016) 'BATMAN-TCM: a bioinformatics analysis tool for molecular mechanism of traditional Chinese medicine', *Scientific Reports*, Vol. 6, No. 1, p.21146.

Luo, L., Jiang, J., Wang, C., Fitzgerald, M., Hu, W., Zhou, Y., Zhang, H. and Chen, S. (2020) 'Analysis on herbal medicines utilized for treatment of COVID-19', *Acta Pharmaceutica Sinica B*, Vol. 10, No. 7, pp.1192–1204.

Min, S., Lee, B. and Yoon, S. (2017) 'Deep learning in bioinformatics', *Briefings in Bioinformatics*, Vol. 18, No. 5, pp.851–869.

Natekin, A. and Knoll, A. (2013) 'Gradient boosting machines, a tutorial', *Frontiers in Neurorobotics*, Vol. 7, p.21.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. (2011) 'Scikit-learn: machine learning in Python', *The Journal of Machine Learning Research*, Vol. 12, pp.2825–2830.

Ru, J., Li, P., Wang, J., Zhou, W., Li, B., Huang, C., Li, P., Guo, Z., Tao, W. and Yang, Y. (2014) 'TCMSP: a database of systems pharmacology for drug discovery from herbal medicines', *Journal of Cheminformatics*, Vol. 6, No. 1, p.13.

Settles, M. and Soule, T. (2005) 'Breeding swarms: a GA/PSO hybrid', *Genetic and Evolutionary Computation*, Vol. 5, pp.161–168.

Snoek, J., Larochelle, H. and Adams, R.P. (2012) 'Practical bayesian optimization of machine learning algorithms', *Advances in Neural Information Processing Systems*, Vol. 25, pp.2951–2959.

Wang, D., Tan, D. and Liu, L. (2018) 'Particle swarm optimization algorithm: an overview', *Soft Computing*, Vol. 22, No. 2, pp.387–408.

Wang, X., Zhang, A. and Sun, H. (2012) 'Future perspectives of Chinese medical formulae: chinmedomics as an effector', *Omics: a Journal of Integrative Biology*, Vol. 16, Nos. 7–8, pp.414–421.

Yang, L. and Shami, A. (2020) 'On hyperparameter optimization of machine learning algorithms: theory and practice', *Neurocomputing*, Vol. 415, pp.295–316.

Zhang, R., Zhu, X., Bai, H. and Ning, K. (2019) 'Network pharmacology databases for traditional Chinese medicine: review and assessment', *Frontiers in Pharmacology*, Vol. 10, p.123.

Zhang, S., Wang, W., Pi, X., He, Z. and Liu, H. (2023) 'Advances in the application of traditional Chinese medicine using artificial intelligence: a review', *The American Journal of Chinese Medicine*, Vol. 51, No. 5, pp.1067–1083.

Zhou, X., Chen, S., Liu, B., Zhang, R., Wang, Y., Li, P., Guo, Y., Zhang, H., Gao, Z. and Yan, X. (2010) 'Development of traditional Chinese medicine clinical data warehouse for medical knowledge discovery and decision support', *Artificial Intelligence in Medicine*, Vol. 48, Nos. 2–3, pp.139–152.