International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-021-01285-w

ORIGINAL ARTICLE

A PSO-based deep learning approach to classifying patients from emergency departments

Weibo Liu[1] · Zidong Wang[1] · Nianyin Zeng[2] · Fuad E. Alsaadi[3] · Xiaohui Liu[1]

Received: 24 November 2020 / Accepted: 8 February 2021

Abstract
In this paper, a deep belief network (DBN) is employed to deal with the problem of the patient attendance disposal in accident & emergency (A&E) departments. The selection of the hyperparameters of the employed DBN is automated by using the particle swarm optimization (PSO) algorithm that is known for its simplicity, easy implementation and relatively fast convergence rate to a satisfactory solution. Specifically, a recently developed randomly occurring distributedly delayed PSO (RODDPSO) algorithm, which is capable of seeking the optimal solution and alleviating the premature convergence, is exploited with aim to optimize the hyperparameters of the DBN. The developed RODDPSO-based DBN is successfully applied to analyze the A&E data for classifying the patient attendance disposal in the A&E department of a hospital in west London. Experimental results show that the proposed RODDPSO-based DBN outperforms the standard DBN and the modified DBN in terms of the classification accuracy.

Keywords Accident & emergency department · Classification · Deep belief network · Deep learning · Particle swarm optimization

1 Introduction

In the UK, the accident & emergency (A&E) departments are operational for 24 hours a day and 365 days a year. The quality of patient treatment in A&E departments, as part of the National Health Service (NHS), has attracted an ever-increasing public concern. In response to the necessity of improving the NHS performance, A&E departments are required to treat 98% of patients within 4 hours from their arrival to admission, transfer or discharge [9]. Unfortunately, due to the unpredictable patient attendances at the A&E departments and the growing number of emergency cases, A&E departments suffer from the overcrowding issue, which poses significant pressure on the limited resources (e.g. staffing and finance).

Clearly, the overcrowding problem in the A&E departments would cause unnecessarily long patient waiting-time and undesirable low treatment quality. Under these circumstances, patients with severe illness might not be able to be treated on time, thereby resulting in negative patient outcomes [4]. As such, it is of vital importance to reduce the length of stay of non-urgent patients in A&E departments so as to decrease the financial costs while delivering high-quality service to the patients with severe illness [17]. In this case, it becomes a prerequisite to have an accurate yet efficient identification of the patient attendance disposal for the purposing of improving inpatient services in A&E departments, which can be seen as a data classification problem.

Recently, a large number of machine learning (ML) algorithms have been successfully employed in the healthcare informatics [17, 22, 33, 38]. For example, the random forests and the gradient boosted decision tree algorithms have been utilized in [33] to predict clinical outcomes (critical care and hospitalization) in A&E departments. In [22], an ensemble learning-based scoring system has been developed to predict the acute cardiac complications for patients with chest pain in A&E departments. Due to their promising performance in

✉ Zidong Wang
Zidong.Wang@brunel.ac.uk

[1] Department of Computer Science, Brunel University London, Uxbridge, Middlesex UB8 3PH, UK

[2] Department of Instrumental and Electrical Engineering, Xiamen University, Fujian 361005, China

[3] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Published online: 06 March 2021

dealing with high-dimensional data, the popular deep learning techniques have been recognized as a powerful family of ML approaches with successful applications to a wide range of research fields such as signal processing, telecommunication, healthcare informatics, natural language processing and computer vision [6, 12, 20, 48].

It is worth mentioning that the deep belief network (DBN) proposed in [14], a breakthrough in the deep learning area, has proven to be effective for various ML problems (e.g., object recognition, image classification, and hand-written character recognition) [2, 30]. In this case, it seems natural to apply the DBN to the patient classification problem in A&E departments. Importantly, with a proper prediction of the patient attendance disposal, patients with serious illness could be allocated with hospital beds and patients who require specific treatments could be transferred to other clinical departments on time.

It should be pointed out that the performance of the DBN is highly related to the selection of hyperparameters, which can be regarded as an optimization problem [3, 31, 32]. As the evolutionary computation (EC) algorithms have shown competitive performance in solving global optimization problems, it makes practical sense to solve the optimal selection problem for the hyperparameters by using the EC algorithms. In fact, various EC algorithms (e.g., the harmony search, the genetic algorithm and the particle swarm optimization (PSO) algorithms) have recently been employed to effectively choose suitable hyperparameters of a DBN [3, 31, 32].

As a particularly attractive EC algorithm, the PSO algorithm has exhibited outstanding performance in discovering the optimal solution with relatively fast convergence rate, and has thus been successfully applied to a wide range of practical applications such as power systems, healthcare informatics, and signal processing [35, 41]. Unfortunately, like other EC algorithms, the PSO algorithm has the issue of premature convergence and may easily be trapped into the local optima. Therefore, it is of practical importance to develop advanced PSO algorithms in order to improve the search capability of the algorithm and thus alleviate the premature convergence problem [26–28, 37, 45–47]. Very recently, a randomly occurring distributedly delayed PSO (RODDPSO) algorithm has been proposed in [25] with the purpose of improving the search performance of the algorithm to enhance the ability of escaping from the local optima. According to its competitive performance in thoroughly exploring the search space, the RODDPSO algorithm is adopted in this paper to optimize the hyperparameters of the DBN. In this context, the established predictions of the patient disposal contribute to a better management of the medical and human resources in A&E departments, which could help deliver high-quality treatment to the high-risk patients.

In this paper, we aim to propose a novel RODDPSO-based DBN in order to launch a study on the patient classification problem with an A&E department. The main contributions can be outlined in the following two aspects:

1. A recently developed RODDPSO algorithm is employed to optimize the hyperparameters of the DBN including the momentum, the penalty parameter and the learning rate, which significantly helps with the training procedure of the DBN. With the optimized hyperparameters, the DBN is effectively and efficiently trained with satisfactory classification accuracy.
2. The introduced RODDPSO-based DBN is successfully applied to analyze the A&E data with aim to solve the patient classification problem based on the attendance records. With an accurate prediction of the patient conditions (i.e., discharge, transfer or admission), the patient routing as well as the medical resource management in A&E departments could be better regulated, which may also reduce the financial pressure on A&E departments.

The rest of this paper is organized in the following manner. The background information of the DBN is presented in Sect. 2. The RODDPSO algorithm is discussed in Sect. 3. Section 4 introduces the RODDPSO-based DBN with detailed information. The parameter setting and experiment results are presented in Sect. 5. Finally, conclusions are drawn and some potential research directions are highlighted in Sect. 6.

2 Deep belief network

It is well known that the DBN has been recognized as the breakthrough of the deep learning community [1, 11, 12, 14]. A DBN is constructed by a series of simple learning modules which are the restricted Boltzmann machines (RBMs).

2.1 Restricted Boltzmann machine

RBMs are undirected probabilistic graphical models which are stacked to build DBNs [30]. The schematic diagram of an RBM is depicted in Fig. 1. An RBM is composed of one hidden layer and one visible layer, where the hidden layer represents the features, and the visible layer represents the input data. A hidden layer consists of a number of binary stochastic hidden units represented by a hidden vector h, and a visible layer consists of several binary stochastic visible units represented by a visible vector v. In one visible layer, all the visible units are fully connected to all the hidden units in the hidden layer with certain weights. It should be

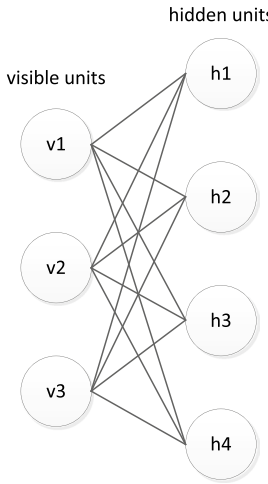

Fig. 1 The schematic diagram of an RBM

mentioned that there are no inner connections among the units in the same layer.

As an energy-based model, the energy function of an RBM is defined by the following function:

$$E(v,h|\theta) = -\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}v_i h_j - \sum_{i=1}^{m} b_i v_i - \sum_{j=1}^{n} c_j h_j \tag{1}$$

where $\theta = (w, b, c)$ represents the parameters of an RBM. In this RBM, the weight (also known as the symmetric interaction term) between the visible unit v_i and the hidden unit h_j is defined by w_{ij}. The number of hidden and visible units are represented by n and m, respectively. The bias of the visible unit v_i is denoted by b_i, and the bias of the hidden unit h_j is represented by c_j. The joint probability distribution $p(v, h|\theta)$ over the visible layer and the hidden layer is computed by the following function:

$$p(v,h|\theta) = \frac{e^{-E(v,h|\theta)}}{Z(\theta)} \tag{2}$$

where $Z(\theta)$ is the partition function (also known as a normalizing constant) that is given by summing over all possible configurations of the visible and hidden vectors. The partition function $Z(\theta)$ is defined by:

$$Z(\theta) = \sum_{v,h} e^{-E(v,h|\theta)} \tag{3}$$

The marginal probability of a visible vector v is defined as follows:

$$p(v|\theta) = \frac{\sum_{h} e^{-E(v,h|\theta)}}{Z(\theta)} \tag{4}$$

It is worth mentioning that there are no intra-layer connections in an RBM, and all the hidden and visible units are conditionally independent. Taking above discussions into consideration, the conditional probabilities $p(v|h, \theta)$ and $p(h|v, \theta)$ can be derived from the joint distribution by:

$$\begin{aligned} p(v_i = 1|h,\theta) &= \sigma\left(\sum_{j=1}^{n} w_{ij}h_j + b_i\right) \\ p(h_j = 1|v,\theta) &= \sigma\left(\sum_{i=1}^{m} w_{ij}v_i + c_j\right) \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ is a sigmoid function shown as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

A fast learning algorithm named by the contrastive divergence (CD) algorithm has been proposed in [14] to train the RBMs. According to the CD algorithm, the parameters of an RBM are updated by the following functions:

$$\begin{aligned} \Delta w_{ij} &= \epsilon_1(\langle v_i h_j\rangle_{\text{data}} - \langle v_i h_j\rangle_{\text{rec}}) \\ \Delta b_i &= \epsilon_1(\langle v_i\rangle_{\text{data}} - \langle v_i\rangle_{\text{rec}}) \\ \Delta c_j &= \epsilon_1(\langle h_j\rangle_{\text{data}} - \langle h_j\rangle_{\text{rec}}) \end{aligned} \tag{7}$$

where ϵ_1 denotes the learning rate, $\langle\cdot\rangle_{\text{data}}$ represents the expectation with respect to the distribution of the input data, and $\langle\cdot\rangle_{\text{rec}}$ is the expectation with respect to the distribution defined by the reconstruction model.

2.2 Deep belief network

DBNs have drawn tremendous attention in the past few years. As mentioned previously, the DBNs are formed by stacking a series of RBMs. The upper layers of a DBN are expected to extract high-level features which explain the input data, and the lower layers extract low-level features from the input data. The learning algorithm of the DBNs is a greedy layer-wise unsupervised learning algorithm [14].

The learning algorithm of the DBNs can be summarized into two stages: (1) the unsupervised pre-training stage; and (2) the supervised fine-tuning stage. In general, the purpose of the pre-training process is to work as a network pre-conditioner and produce suitable parameters for further supervised training which demonstrates better performance than randomly initialize the parameters of the network. The RBMs are trained in the bottom-up manner layer by layer, where the input of the upper RBM is provided by the output of the lower RBM. At the fine-tuning stage, an additional output layer is added to the network in order to predict the desired labels, and the parameters of network are further tuned by employing the back-propagation algorithm [1, 12, 30].

It should be noticed that the weight decay (used as a penalty term) is adopted in the DBN with hope to prevent overfitting [18]. In this case, the capacity of the DBN is

controlled by adding the weight decay term, and therefore improving the performance of the model on the testing dataset. It is also well known that the momentum is an important factor in training a deep neural network which is used to control the learning speed and smooth out the update of the weights during the training process. In our work, the momentum and the penalty terms are used in both of the pre-training and the fine-tuning stages in order to control possible oscillations during the training process.

The model parameters $\theta = (w, b, c)$ in the pre-training process at tth epoch are thus updated by the following equations:

$$\begin{aligned}\Delta w_{ij}^{t} &= m_1 \times \Delta w_{ij}^{t-1} - \lambda_1 \times w_{ij}^{t} \\ &\quad + \epsilon_1 \times (\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{rec}}) \\ \Delta b_i^{t} &= m_1 \times \Delta b_i^{t-1} + \epsilon_1 \times (\langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{rec}}) \\ \Delta c_j^{t} &= m_1 \times \Delta c_j^{t-1} + \epsilon_1 \times (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{rec}})\end{aligned} \tag{8}$$

where m_1 represents the momentum term in the pre-training process, ϵ_1 is the learning rate in the pre-training process, λ_1 denotes the penalty parameter (which is used to penalize weights with large magnitude) in the pre-training process.

In the fine-tuning process, the pre-trained DBN is further tuned to improve the performance. Additionally, the momentum is also used to make the learning process stable, and the weight decay is added on the weights to prevent overfitting at the fine-tuning stage.

3 The RODDPSO algorithm

In our work, a recently proposed RODDPSO algorithm is utilized to improve the performance of the DBN by optimizing the hyperparameters [25]. Note that the major advantages of the RODDPSO algorithm over the conventional PSO algorithms can be summarized as follows: (1) the distributed time-delays (DTDs) have been added in the velocity updating process such that historical information of the particles can be adequately taken into consideration during the evolution process; (2) the utilization of the randomly-occurring strategy contributes to a balance between the global search and local search; and (3) the capability of escaping from the local optimal solutions is enhanced. As discussed above, the RODDPSO algorithm demonstrates its strong ability and high effectiveness in discovering the optimal solution.

3.1 Framework

The kinematics equations of the RODDPSO algorithm at kth iteration are as follows:

$$\begin{aligned}v_i(k+1) &= \omega v_i(k) + c_1 r_1 (p_i(k) - x_i(k)) \\ &\quad + c_2 r_2 (p_g(k) - x_i(k)) \\ &\quad + m_i(k) c_3 r_3 \sum_{\tau=1}^{N} \alpha_{(\tau)} (p_i(k-\tau) - x_i(k)) \\ &\quad + m_g(k) c_4 r_4 \sum_{\tau=1}^{N} \alpha_{(\tau)} (p_g(k-\tau) - x_i(k)) \\ x_i(k+1) &= x_i(k) + v_i(k+1)\end{aligned} \tag{9}$$

where ω is the inertia weight; c_1 and c_2 are the cognitive acceleration coefficient and the social acceleration coefficient, respectively; c_3 and c_4 denote the acceleration coefficients for DTDs with $c_1 = c_3$ and $c_2 = c_4$; N is the upper bound of the DTDs; p_i and p_g denote the personal best position and global best position of the particle, respectively; $m_i(k)$ and $m_g(k)$ are the intensity factors of the DTDs depending on the evolutionary state; $\alpha_{(\tau)}$ is a N-dimensional vector whose elements are random scalars from 0 or 1; and random numbers $r_i (i = 1, 2, 3, 4)$ obey the uniform distribution in [0, 1]. Notice that ω is updated according to the linearly decreasing strategy in [36], and c_1, c_2 are controlled based on the time-varying strategy introduced in [34].

Figure 2 shows the flowchart of the RODDPSO algorithm.

3.2 Evolutionary state

In the RODDPSO algorithm, four evolutionary states (exploitation, exploration, convergence and jumping-out) are defined based on the searching characteristics of the particles. The velocity and position of the particles are updated depending on the evolutionary state accordingly [25, 47].

It is worth mentioning that the evolutionary factor is introduced to determine the evolutionary state, which is computed according to the distance between the particles. The average distance between the mth particle and others is defined by:

$$d_m = \frac{1}{P-1} \sum_{n=1, n\neq m}^{P} \sqrt{\sum_{k=1}^{D} (x_{mk} - x_{nk})^2} \tag{10}$$

where D is the dimension of the particle, and P is the population size of the entire swarm.

The evolutionary factor (F) is computed by the following equation.

$$F = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \tag{11}$$

where d_g represents the average distance between the global best particle and other particles; $d_{\min}$ and $d_{\max}$ represent the minimum and maximum distances of d_m, respectively.

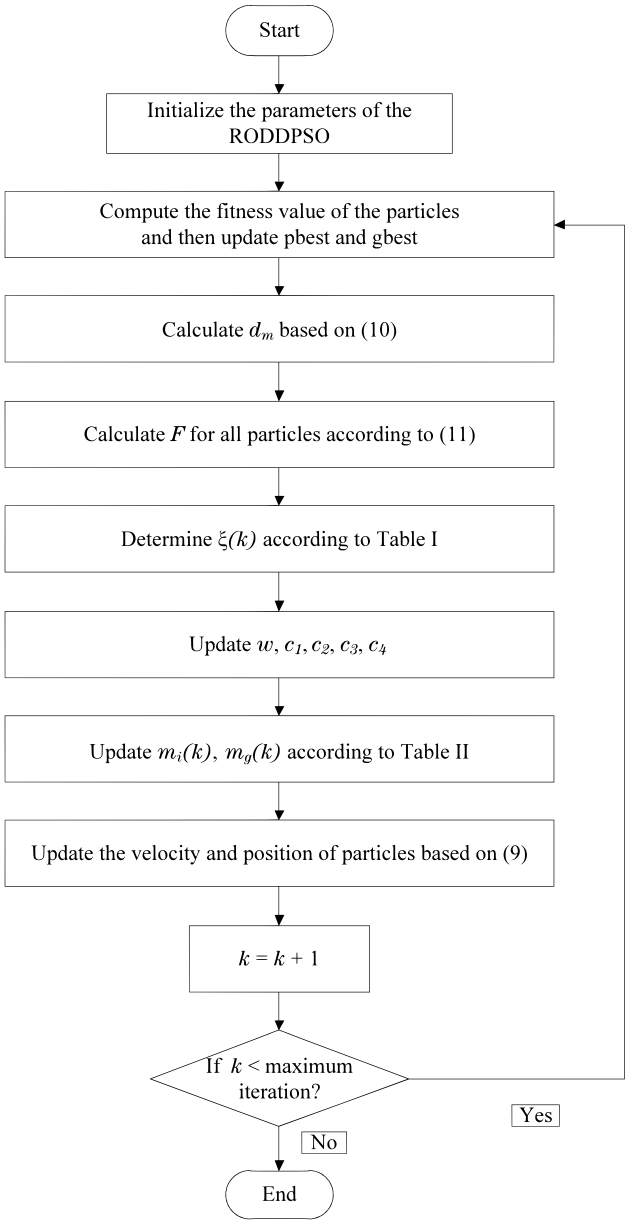

Fig. 2 Flowchart of the RODDPSO algorithm

The evolutionary states $\xi(k) = 1, 2, 3, 4$ are displayed in Table 1, and the selection of the intensity factors $m_i(k)$ and $m_g(k)$ are given in Table 2. Broadly speaking, large values of m_i and m_g could highly increase the influence of the distributed time-delays on the updating of the velocity which may reduce the convergence speed. If the values of m_i and m_g are very small (e.g., 0.00001), the changes of the distributed time-delays in the velocity updating model may not reduce the possibility of escaping from the local optima.

Table 1 Evolutionary states

Evolutionary state	Mode	Evolutionary factor
Convergence	$\xi(k) = 1$	$0.00 \le F < 0.25$
Exploitation	$\xi(k) = 2$	$0.25 \le F < 0.50$
Exploration	$\xi(k) = 3$	$0.50 \le F < 0.75$
Jumping-out	$\xi(k) = 4$	$0.75 \le F \le 1.00$

Table 2 Velocity updating strategy for the RODDPSO algorithm

State	$m_i(k)$	$m_g(k)$
Convergence	0	0
Exploitation	0.01	0
Exploration	0	0.01
Jumping-out	0.01	0.01

4 The RODDPSO-based DBN

The RODDPSO algorithm is employed to select suitable hyperparameters of a DBN. As mentioned previously, the momentum and weight decay terms are utilized in both of the pre-training and fine-tuning stages. In this case, there are six hyperparameters we need to optimize including m_1 (the momentum parameter for pre-training), m_2 (the momentum parameter for fine-tuning), λ_1 (the penalty parameter for pre-training), λ_2 (the penalty parameter for fine-tuning), ϵ_1 (the learning rate for pre-training), and ϵ_2 (the learning rate for fine-tuning). To sum up, the dimension of the problem space is $D = 6$ and a candidate solution of our optimization problem is a vector $[m_1\ m_2\ \lambda_1\ \lambda_2\ \epsilon_1\ \epsilon_2]$.

4.1 Objective function

The objective function of the RODDPSO algorithm is given as follows:

$$J = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \frac{\lambda_2}{2}\sum_{l=1}^{M-1}\sum_{j=1}^{m_{l+1}}\sum_{i=1}^{m_l} w_{ijl}^2 + \frac{\lambda_2}{2}\sum_{l=2}^{M}\sum_{i=1}^{m_l} d_{il}^2 \tag{12}$$

where y_i and $\hat{y}_i$ indicate the real class and the predicted class of the ith data point, respectively. N is the number of total data points. λ_2 represents the penalty parameter for the fine-tuning process. M is the total number of layers in the DBN. m_l represents the number of neurons in the lth layer. w_{ijl} denotes the weight between the ith neuron in the lth layer and the jth neuron in the $(l+1)$-th layer. d_{il} represents the threshold of the ith neurons in the lth layer. It should be mentioned that the objective function of the RODDPSO algorithm is also used as the loss function of the DBN at the fine-tuning stage.

In (12), the term $\frac{\lambda_2}{2}\sum_{l=1}^{M-1}\sum_{j=1}^{m_{l+1}}\sum_{i=1}^{m_l} w_{ijl}^2$ is the weight decay which is introduced to penalize large weights of the neural network [12, 19]. The term $\frac{\lambda_2}{2}\sum_{l=2}^{M}\sum_{i=1}^{m_l} d_{il}^2$ is the

penalty term to constrain the threshold of the neural network. By adding the two penalty terms, the weight and threshold will not be too large. In this context, the neural network will not be sensitive to the magnitude of the input data, and the generalization ability of the neural network is enhanced which could prevent the overfitting problem.

4.2 RODDPSO-based DBN framework

It should be pointed out that the RODDPSO algorithm is utilized to optimize the hyperparameters of a DBN in order to improve the classification performance. In this paper, each particle is represented by a vector, which contains six hyperparameters. The training procedure of the introduced RODDPSO-based DBN is shown in Algorithm 1.

Algorithm 1 The Training Algorithm of the RODDPSO-based DBN

1. Parameter initialization (consisting of the swarm size P, the velocity and position of ith particle v_i, x_i, inertia weight ω, acceleration coefficients c_1, c_2, the maximum value of velocity $V_{\max}$ and intensity factors m_i, m_g.)
2. Randomly initialize every particle containing the penalty parameters λ_1, λ_2, momentum terms m_1, m_2, and the learning rate ϵ_1, ϵ_2.
3. Train the DBN and adjust the weights based on (8).
4. Evaluate the fitness value of particles according to (12).
5. Update p_i and p_g.
6. Determine the evolutionary state $\xi(k)$ according to the evolutionary factor F.
7. Adjust the velocity and position of each particle by (9).
8. Repeat Steps 3-7 till the algorithm reaches the maximum iteration number.
9. Calculate the accuracy of the RODDPSO-based DBN.

5 Results and discussions

5.1 Data description

The A&E data is collected from a hospital in London including three emergency units. The detailed information of the three emergency units is displayed in Table 3.

In the raw data, a total number of 126,986 incidents are recorded where each incident denotes a record of a patient attendance at the A&E departments. Once a patient enters the A&E department, a nurse will make records on the patient from arrival to discharge (transfer to another clinic or admit to a hospital bed). There are 25 attributes in the raw data, some of which are redundant attributes (the description of existing attributes). The incidents are recorded in real-time, hence the patient treatment time and the patient waiting time are computed as new attributes. Note that the data is normalized, and incidents with missing data or null values are abandoned [25].

Table 3 Patients attendance at the emergency departments

Department	Number of incident
Accident and emergency department	51,713
Minor injury unit	15,151
Urgent care centre	60,122

In our work, the mode of patient arrival, the age of patient, the first diagnosis, the treatment time and the healthcare resource group are selected as the inputs of the classification problem, and the attendance disposal is used as the output class. Notice that there are 12 categories of the patient attendance disposal in A&E departments, and the detailed information of the patient attendance disposal is presented in Table 4.

Due to the suggestions of experts in the A&E departments, we reduce the number of class to 5 by merging related categories into 1 class and removed 6 classes. Category 3 is abandoned because patients belonging to this category do not require further following-up treatment, and there may exist misdiagnosed cases. In recent years, most of the hospitals stop providing A&E clinic because they are dealing with acute emergency conditions since GP and other outpatients specialists are capable of dealing with the follow up cases, and hence category 4 is removed. Category 9 is also abandoned because this category is part of categories 5–7. Furthermore, categories 10, 11 and 12 are canceled based on the domain knowledge from the experts. Additionally, categories 1 and 8 are merged into the same class due to patients within these two categories are highly possibility have life-threaten illness.

In summary, the modified output class of the neural network is displayed in Table 5. Specifically, class 1 represents patients who are admitted to a hospital bed, become a logged patient of the same health care provider or die in emergency department. Class 2 includes patients who are discharged with follow up treatment to be provided by the general practitioner. Patients in class 3 and class 4 are referred to fracture clinic and other outpatient clinic, respectively. Class 5 includes patients who are transferred to other healthcare provider.

5.2 Parameter setting

In the simulation, 64,800 incidents are selected from the data where 48,600 incidents are used for training and the rest 16,200 incidents are utilized for testing. The number of particle is 10 and the maximum iteration is set to be 10. The number of the hidden layers in the standard DBN,

Table 4 Patient attendance disposal

Category	Description	Number of incidents
1	Admitted to a hospital bed or became a logged patient	20,468
2	Discharged—follow up treatment to be provided by GP	31,540
3	Discharged—did not require any follow up treatment	51,795
4	Referred to A&E clinic	115
5	Referred to fracture clinic	4010
6	Referred to other outpatient clinic	7130
7	Transferred to other healthcare provider	1598
8	Died in department	62
9	Referred to other healthcare professional	5627
10	Left department before being seen for treatment	546
11	Left department having refused treatment	235
12	Other	80

Table 5 Modified class for patient attendance disposal

Output class	Original category	Number of incidents
1	1, 8	20,530
2	2	31,540
3	5	4010
4	6	7130
5	7	1598

Table 6 Configuration of the standard DBN at the pre-training stage

Parameter	Standard DBN	Penalized DBN	RODDPSO-based DBN
Learning rate	0.01	0.01	[0, 0.01]
Momentum	0	0.5	[0.5, 1]
Penalty parameter	0	1e-5	[0, 1e-5]
Mini-batch size	50	50	50

penalized DBN and the RODDPSO-based DBN are all set to be 3, and the number of hidden units in three hidden layers is 100, 64 and 50, respectively. The pre-training epochs of the three DBNs are all 100. The numbers of epoch of the fine-tuning process of the standard DBN, the penalized DBN, and the RODDPSO-based DBN are set to be 300. It should be mentioned that the activation function of the three variant DBNs is the sigmoid function. The parameter setting of three DBNs at the pre-training stage are given in Table 6. Note that we give the range of the learning rate, the momentum, and the penalty parameters for the RODDPSO-based algorithm in Table 6. At the fine-tuning stage, the learning rate of the RODDPSO-based DBN is in the range of [0, 1], the momentum is in the range of [0.5, 1], and the penalty parameter is in the range of $[0, 5e-6]$.

5.3 Performance indicator

In this paper, the classification accuracy is utilized as the performance indictor to evaluate the classification performance of the RODDPSO-based DBN. The standard DBN and the penalized DBN (with momentum and weight decay) are employed in comparison with the RODDPSO-based DBN. The classification accuracy is computed as follows:

$$A_c = \frac{N_c}{N_c + N_f} \times 100\% \tag{13}$$

where A_c is the classification accuracy; N_c is the number of correct prediction, and N_f represents the number of incorrect prediction. As such, a larger value of classification accuracy indicates a better classification performance.

5.4 Experiment results

To comprehensively evaluate the performance of the RODDPSO-based DBN, the standard DBN and the penalized DBN are used for comparison. The full-batch training mean squared error (MSE) of the standard DBN is displayed in Fig. 3 where the vertical coordinate indicates the full-batch MSE and the horizontal coordinate is the epoch number. In Fig. 3, we can see that the MSE of the DBN decreases very fast, which indicates that the pre-trained DBN performs well. In Fig. 4, it is apparent that the learning curve of the penalized DBN is

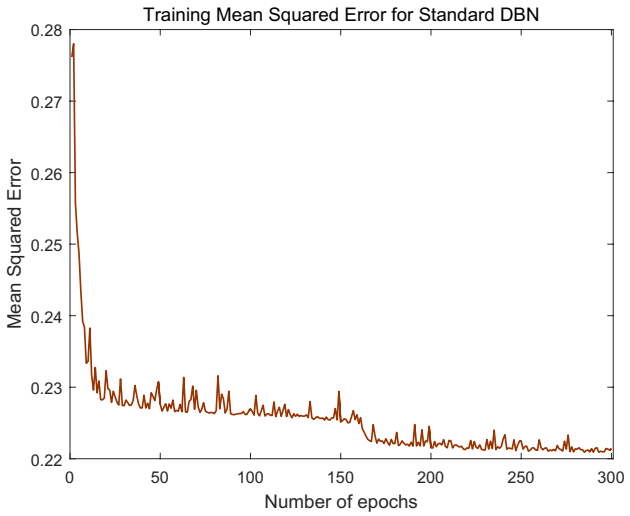

Fig. 3 Full-batch training mean squared error results of the standard DBN

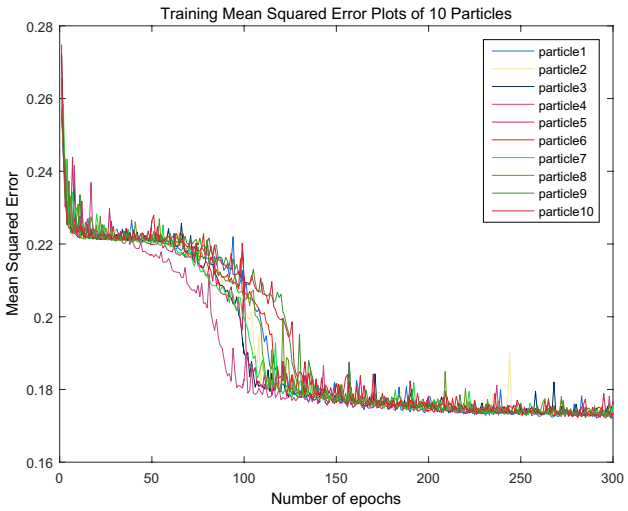

Fig. 5 Full-batch training mean squared error results of the ROD-DPSO-based DBN

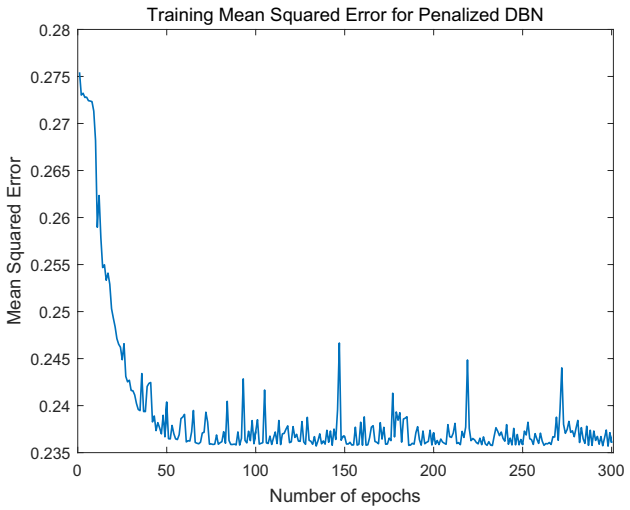

Fig. 4 Full-batch training mean squared error results of the Penalized DBN

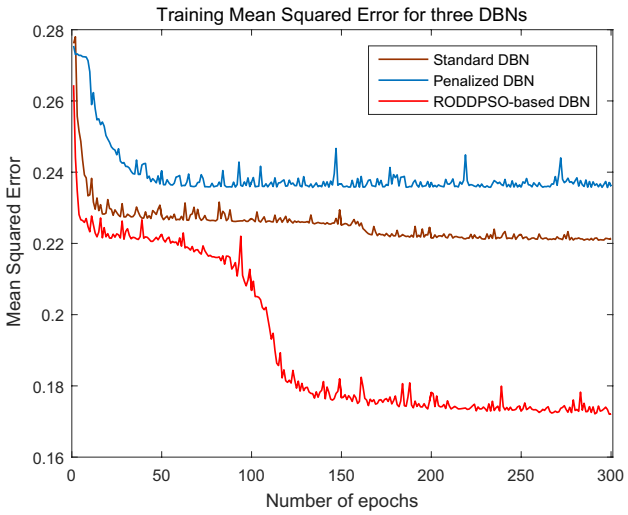

Fig. 6 Comparison of the full-batch training mean squared error results of three DBNS (RODDPSO-based DBN, Penalized DBN, and standard DBN

relatively smooth. The full-batch training MSE of the DBNs which use different sets of hyperparameters (in each particle) in the final iteration are demonstrated in Fig. 5. The full-batch training MSE of the DBN with hyperparameters in particle 1 performs better than other DBNs with hyperparameters in other particles.

According to [39], the time complexity of the ROD-DPSO-based DBN algorithm is:

$$\text{Time} = O\left(\sum_{i=1}^{D_L}(Lx_i Hn_i(Lx_i + T(f_a)))PM\right) \tag{14}$$

where D_L is the total number of layer in the DBN; Lx_i denotes the input size of layer i, Hn_i represents the number of hidden nodes in the ith layer; $T(f_a)$ is the time complexity of the activation function f_a for one single data point; P denotes the number of particle; and M is the maximum iteration number.

The running time of the standard DBN, the penalized DBN, and the RODDPSO-based DBN for one epoch is 114.112s, 140.917s, and 277.383s, respectively. By employing the RODDPSO algorithm to automatically choose suitable hyperparameters, the running time of the

RODDPSO-based DBN is more than that of the standard DBN and the penalized DBN.

The results of classification accuracy of the three DBNs are depicted in Fig. 6. It is clear that the full-batch training MSE of the RODDPSO-based DBN is less than that of the standard DBN and the penalized DBN, which indicates that the RODDPSO-based DBN outperforms the other two DBNs. In addition, the MSE curve of the RODDPSO-based DBN decreases faster than that of the standard DBN and the penalized DBN. The classification accuracy of the RODDPSO-based DBN, the penalized DBN, and the standard DBN is 76.06%, 68.10% and 69.83%. To sum up, the RODDPSO-based DBN demonstrates superior classification performance over the penalized DBN and the standard DBN. As such, we can draw the conclusion that the RODDPSO-based DBN performs well on the patient attendance data in A&E departments and could efficiently classify the patient attendance disposal. With the output class (patient discharge) obtained by the RODDPSO-based DBN, it becomes easier to verify the patient attendance disposal category, which may improve the patient care and discharge the non-urgent patients to release the overcrowding problem and save the NHS costs in terms of both the medical and human resources.

6 Conclusion

In this paper, an advanced deep learning approach has been developed and successfully applied to patient classification problem in an A&E department. Due to its promising performance in discovering the optimal solution, the recently developed RODDPSO algorithm has been adopted to optimize hyperparameters of a DBN. Experiment results demonstrate that the proposed RODDPSO-based DBN effectively classifies the patient attendance disposal and outperforms the traditional DBN and the DBN with penalty in terms of the training curve of the model as well as the classification accuracy. In the future, we aim to: (1) improve the performance of the RODDPSO-based DBN by introducing sparse terms and optimizing the activation of the sparse terms by employing the PSO algorithms; (2) employ other EC algorithms to choose appropriate hyperparameters of the deep neural networks [7, 8]; (3) further optimize the topology of the DBN in terms of the number of hidden layers and the number of hidden units in each hidden layer; and (4) extend our results to other research fields, such as signal processing [5, 15, 16, 49–51], healthcare [13], telecommunication [10, 21, 23, 24, 40], and recommender systems [29, 42–44].

Funding This work was supported in part by the National Natural Science Foundation of China under Grants 61873148 and 61933007, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany.

References

1. Bengio Y, Lamblin P, Popovici D, Larochelle H (2006) Greedy layerwise training of deep networks. In: Proceedings of advances in neural information processing systems 19, Vancouver, BC, Canada, pp 153–160
2. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127
3. Bergstra JS, Bardenet R, Bengio Y, Kegl B (2011) Algorithms for hyper-parameter optimization. In: Proceedings of advances in neural information processing systems 24, Vancouver, BC, Canada, pp 2546–2554
4. Bhattacharjee P, Ray PK (2014) Patient flow modelling and performance analysis of healthcare delivery processes in hospitals: a review and reflections. Comput Ind Eng 78:299–312
5. Chen D, Chen W, Hu J, Liu H (2019) Variance-constrained filtering for discrete-time genetic regulatory networks with state delay and random measurement delay. Int J Syst Sci 50(2):231–243
6. Cheng H, Wang Z, Wei Z, Ma L, Liu X (2020) On adaptive learning framework for deep weighted sparse autoencoder: a multiobjective evolutionary algorithm. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.3009582
7. Cui L, Li G, Lin Q, Du Z, Gao W, Chen J, Lu N (2016) A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. Inf Sci 367:1012–1044
8. Cui L, Li G, Lin Q, Chen J, Lu N (2016) Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations. Comput Oper Res 67:155–173
9. Eatock J, Clarke M, Picton C, Young T (2011) Meeting the four-hour deadline in an A&E department. J Health Org Manag 25(6):606–624
10. Geng H, Wang Z, Liang Y, Cheng Y, Alsaadi FE (2017) Tobit Kalman filter with time-correlated multiplicative sensor noises under redundant channel transmission. IEEE Sens J 17(24):8367–8377
11. Gong M, Liu J, Li H, Cai Q, Su L (2015) A multiobjective sparse feature learning model for deep neural networks. IEEE Trans Neural Netw Learn Syst 26(12):3263–3277
12. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
13. Gul M, Guneri AF (2015) A comprehensive review of emergency department simulation applications for normal and disaster conditions. Comput Ind Eng 83:327–344
14. Hinton G, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
15. Hu J, Wang Z, Liu G-P, Jia C, Williams J (2020) Event-triggered recursive state estimation for dynamical networks under randomly switching topologies and multiple missing measurements. Automatica 115:108908
16. Hu J, Wang Z, Liu G-P, Jia C, Zhang H (2020) Variance-constrained recursive state estimation for time-varying complex networks with quantized measurements and uncertain inner coupling. IEEE Trans Neural Netw Learn Syst 31(6):1955–1967
17. Jiang S, Chin K-S, Wang L, Qu G, Tsui KL (2017) Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. Expert Syst Appl 82:216–230

18. Jiang S, Chin K-S, Tsui KL (2018) A universal deep learning approach for modeling the flow of patients under different severities. Comput Methods Programs Biomed 154:191–203
19. Kienzle W, Chellapilla K (2006) Personalized handwriting recognition via biased regularization. In: Proceedings of the 23rd international conference on machine learning, Pennsylvania, USA, pp 457–464
20. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
21. Liu H, Wang Z, Fei W, Li J, Alsaadi FE (2021) On inite-horizon H-infinity state estimation for discrete-time delayed memristive neural networks under stochastic communication protocol. Inf Sci 555:280–292
22. Liu N, Koh ZX, Chua ECP, Tan LML, Lin Z, Mirza B, Ong MEH (2014) Risk scoring for prediction of acute cardiac complications from imbalanced clinical data. IEEE J Biomed Health Informatics 18(6):1894–1902
23. Liu S, Wang Z, Wang L, Wei G (2018) On quantized H_∞ filtering for multi-rate systems under stochastic communication protocols: the finite-horizon case. Inf Sci 459:211–223
24. Liu S, Wang Z, Wei G, Li M (2020) Distributed set-membership filtering for multirate systems under the Round-Robin scheduling over sensor networks. IEEE Trans Cybern 50(5):1910–1920
25. Liu W, Wang Z, Liu X, Zeng N, Bell D (2019) A novel particle swarm optimization approach for patient clustering from emergency departments. IEEE Trans Evol Comput 23(4):632–644
26. Liu W, Wang Z, Yuan Y, Zeng N, Hone K, Liu X (2021) A novel sigmoid-function-based adaptive weighted particle swarm optimizer. IEEE Trans Cybern 51(2):1085–1093
27. Liu W, Wang Z, Zeng N, Yuan Y, Alsaadi FE, Liu X (2021) A novel randomised particle swarm optimizer. Int J Mach Learn Cybern 12(2):529–540
28. Luo X, Yuan Y, Chen S, Zeng N, Wang Z (2020) Position-transitional particle swarm optimization-incorporated latent factor analysis. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3033324
29. Luo X, Yuan Y, Zhou M, Liu Z, Shang M (2019) Non-negative latent factor model based on β-divergence for recommender systems. IEEE Trans System Man Cybern Syst. https://doi.org/10.1109/TSMC.2019.2931468
30. Mohamed AR, Dahl G, Hinton G (2009) Deep belief networks for phone recognition. In: Proceedings of NIPS workshop on deep learning for speech recognition and related applications
31. Papa JP, Scheirer W, Cox DD (2016) Fine-tuning deep belief networks using harmony search. Appl Soft Comput 46:875–885
32. Passos LA, Rodrigues DR, Papa JP (2018) Fine tuning deep boltzmann machines through meta-heuristic approaches. In: Proceedings of IEEE 12th international symposium on applied computational intelligence and informatics, Timisoara, Romania, pp 419–424
33. Raita Y, Goto T, Faridi MK, Brown DF, Camargo CA, Hasegawa K (2019) Emergency department triage prediction of clinical outcomes using machine learning models. Critical Care 23:64
34. Ratnaweera A, Halgamuge SK, Watson HC (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans Evol Comput 8(3):240–255
35. Sheng W, Shan P, Chen S, Liu Y, Alsaadi FE (2017) A niching evolutionary algorithm with adaptive negative correlation learning for neural network ensemble. Neurocomputing 247:173–182
36. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 IEEE congress on evolutionary computation, Washington, DC, USA, pp 1945–1950
37. Song B, Wang Z, Zou L (2017) On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm. Cogn Comput 9(1):5–17
38. Tan PN, Steinbach M, Kumar V (2005) Introduction to data mining. Addison Wesley, Boston
39. Tang F, Mao B, Fadlullah ZM, Liu J, Kato N (2020) ST-DeLTA: an novel spatial-temporal value network aided deep learning based intelligent network traffic control system. IEEE Trans Sustain Comput 5(4):568–580
40. Wang L, Wang Z, Wei G, Alsaadi FE (2019) Observer-based consensus control for discrete-time multi-agent systems with coding-decoding communication protocol. IEEE Trans Cybern 49(12):4335–4345
41. Xiao X, Dow ER, Eberhart R, Miled ZB, Oppelt RJ (2003) Gene clustering using self-organizing maps and particle swarm optimization. In: Proceedings of the international parallel and distributed processing symposium, Nice, France
42. Yue W, Wang Z, Tian B, Payne A, Liu X (2020) A collaborative-filtering-based data collection strategy for Friedreich's ataxia. Cogn Comput 12:249–260
43. Yue W, Wang Z, Liu W, Tian B, Lauria S, Liu X (2020) An optimally weighted user-and item-based collaborative filtering approach to predicting baseline data for Friedreich's ataxia patients. Neurocomputing 419:287–294
44. Yue W, Wang Z, Tian B, Pook M, Liu X (2021) A hybrid model-and memory-based collaborative filtering algorithm for baseline data prediction of Friedreich's ataxia patients. IEEE Trans Industr Inf 17(2):1428–1437
45. Zeng N, Wang Z, Zhang H, Alsaadi FE (2016) A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. Cogn Comput 8(2):143–152
46. Zeng N, Wang Z, Liu W, Zhang H, Hone K, Liu X (2020) A dynamic neighborhood-based switching particle swarm optimization algorithm. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.3029748
47. Zhan Z-H, Zhang J, Li Y, Chung HS-H (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B Cybern 39(6):1362–1381
48. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning and its applications to machine health monitoring. Mech Syst Signal Process 115:213–237
49. Zhao Z, Wang Z, Zou L, Guo J (2020) Set-Membership filtering for time-varying complex networks with uniform quantisations over randomly delayed redundant channels. Int J Syst Sci 51(16):3364–3377
50. Zou L, Wang Z, Hu J, Zhou D (2020) Moving horizon estimation with unknown inputs under dynamic quantization effects. IEEE Trans Autom Control 65(12):5368–5375
51. Zou L, Wang Z, Zhou D (2020) Moving horizon estimation with non-uniform sampling under component-based dynamic event-triggered transmission. Automatica 120:13