Ramalingaswamy Cheruku* and Damodar Reddy Edla

Selector: PSO as Model Selector for Dual-Stage Diabetes Network

https://doi.org/10.1515/jisys-2017-0394
Received August 3, 2017; previously published online April 7, 2018.

Abstract: Diabetes is a chronic disease caused by insulin deficiency, and it should be detected in the early stages for effective treatment. In this paper, the Diabetes-Network (Dia-Net) is proposed to increase diabetes predictive accuracy. The proposed Dia-Net is a dual-stage network. It combines both optimized probabilistic neural network (OPNN) and optimized radial basis function neural network (ORBFNN) in the first stage. Hence, Dia-Net possesses the advantages of both the models. In the second stage, the linear support vector machine is used. As the dataset size increases, both RBFNN and PNN perform better, but both suffers from complexity and computational problems. To address these problems, in this paper, particle swarm optimization-based clustering is employed for discovering centers in high-dense regions. This reduces the size of the hidden layer of both RBFNN and PNNs. Experiments are carried out on the Pima Indians Diabetes dataset. The Experimental results showed that the proposed Dia-Net model outperformed individual as well as state-of-the-art models.

Keywords: Diabetes classification, RBFNN, PNN, optimal number of clusters, highly dense regions.

2010 Mathematics Subject Classification: 62M45, 82C32, 92B20, 68U35, 60G25.,

1 Introduction

Diabetes is a major health problem in both developed and developing countries. Its prevalence is rising every year. Diabetes occurs when the body fails to produce insulin or produce insufficient insulin hormone. Insulin is a hormone produced by the pancreas that helps to regulate glucose levels in the blood. The most common form of diabetes is type-2 diabetes; in this, pancreas loses the ability to appropriately produce and release insulin. Uncontrolled diabetes causes rise in blood sugar levels, this increases the risks of developing diseases like kidney failure, heart attack, blindness, nerve damage, and blood vessel damage. About half of the patients with type-2 diabetes are undiagnosed. The detection of diabetes disease in earlier stages improves the patient's life span. Thus, classification algorithms play a vital role in the prediction of diabetes [2, 25].

The multi-layer feed forward neural networks (MLFFNNs) and the multi-layer perceptron neural networks (MLPNNs) are the most popular techniques for classification and use iterative process for training. Contrary to the MLFFNNs and MLPNNs, the radial basis function neural networks (RBFNNs) and the probabilistic neural networks (PNNs) are trained in single iteration and learn applications quickly. Thus, the RBFNNs and PNNs draw the researcher's attention for classification tasks. Moreover, the performances of these neural networks are on par with the MLFFNNs and MLPNNs [1, 26].

Although the existing rule-based and non rule-based classification algorithms are popular, they show moderate performance. Hence, an ensemble technique gained attention, which performs better than the

***Corresponding author: Ramalingaswamy Cheruku,** Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda 403401, Goa, India, e-mail: rmlswamygoud@nitgoa.ac.in, rmlswamygoud@gmail.com; and Department of Computer Science and Engineering, Mahindra École Centrale College of Engineering, Bahadurpally, Hyderabad-500034, Telangana, India. https://orcid.org/0000-0003-1677-5321
Damodar Reddy Edla: Department of Computer Science and Engineering, National Institute of Technology Goa, Ponda 403401, Goa, India

individual classifiers. There exist multiple ensemble techniques in the literature, but most commonly bagging [19], boosting [16], and stacking [13] are used.

Already in the literature, Kaynak and Alpaydin [14] proposed the multistage cascading of multiple classifiers. They focused not only on accuracy but also on computational and space complexity. They have used single, multi-layer perceptrons and kNN in implementations. The proposed cascading model obtained more accuracy than the individual classifier accuracy. The proposed model obtained nearly 77% accuracy on the Pima Indians Diabetes (PID) dataset.

Next, Polat et al. [20] proposed a new cascade learning system based on the generalized discriminant analysis (GDA) and least square support vector machine (LS-SVM). The proposed system consists of two stages. In the first stage, they used the GDA in the discriminant feature variables between healthy and patient (diabetes) data as the pre-processing process. In the second stage, they used the LS-SVM in order to classify the diabetes dataset. The proposed system GDA-LS-SVM obtained an 82.05% classification accuracy on the PID dataset.

Moreover, Bashir et al. [3] proposed multiple ensemble classification techniques for improving the performance of diabetes classification. They used three types of decision trees ID3, C4.5, and CART (Classification and Regression Tree) as the base classifiers. They used majority voting, AdaBoost, Bayesian boosting, stacking, and bagging ensemble techniques for experimental evaluation. The experimental results showed that the bagging ensemble technique shows better performance compared to individual as well as other ensemble techniques.

Kandhasamy and Balamurali [12] applied the random forest (RF) classifier on the PID dataset. Bashir et al. [4] proposed the HMV (hierarchical majority voting) ensemble model for disease classification and prediction with a three-layered approach and obtained an accuracy of 77.08% on the PID dataset. Again, Bashir et al. [5] proposed a medical decision support system called HM-BagMoov using a novel weighted multi-layer classifier ensemble framework. The proposed HM-BagMoov obtained an accuracy of 78.21% on the PID dataset.

Especially, in medical diagnostic systems, a small increment in the classifier-predictive accuracy also matters as it saves many people lives. In order to increase the diabetes predictive accuracy while balancing the model complexity, in this paper we proposed:

- Cascaded dual-stage Dia-Net that combines both the optimized probabilistic neural network (OPNN) and optimized radial basis function neural network (ORBFNN) in the first stage and the linear SVM in the second stage.
- Particle swarm optimization (PSO)-based clustering to reduce the Dia-Net complexity.

2 Preliminaries

2.1 Probabilistic Neural Network

Specht [24] first proposed the PNNs in 1990. The learning speed of the PNN model is very fast, making it suitable in real-time disease diagnosis. A few advantages for the PNN over the conventional MLFFNN and MLPNN are [18]:

- PNNs are computationally faster than the MLFFNN and MLPNNs.
- PNNs provide robust performance on noisy data and easily incorporate additional samples.

The architecture of the PNN is displayed as four layers and is shown in Figure 1. The figure displays a PNN that recognizes two classes and extended to multi-class problems [6].

- **Input layer:** The input neurons supply the same input values to the hidden layer neurons. The size of this layer is determined by the dataset dimensionality (D).
- **Hidden layer:** There is one neuron per training pattern. The response of each hidden layer neuron is computed using the equation below.

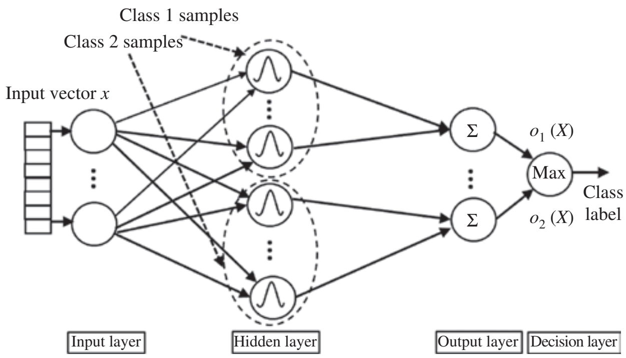

Figure 1: PNN model for classification task.

$$\varphi_i(X)=\frac{1}{\sqrt[D]{2\Pi}(\sigma_i)^D}e^{-\frac{((X-\mu_i)\cdot(X-\mu_i)^T)}{2(\sigma_i)^2}}, \tag{1}$$

- **Output layer:** This layer has one neuron for each class. Each output neuron receives the output from the hidden layer neurons associated with a given class, and the summation is carried out as follows:

$$0_j(X)=\frac{1}{N_j}\sum_{i=1}^{N_j}\phi_i(X),\quad i=1,\ 2,\ \ldots,\ N_j \tag{2}$$

  where, N_j denotes number of patterns in the j^{th} class.
- **Decision layer:** The size of this layer is one. This layer determines the class label of the given input vector (X) present at the input layer using Eq. (3).

$$\text{class}(X)=\underset{j}{argmax}O_j(X),\quad j=1,\ 2,\ \ldots,\ C. \tag{3}$$

2.2 Optimal PNN

The traditional PNN estimates each class probability density function (PDF) using a training set. These estimated PDFs approach the true PDFs as the training set size increases. Consequently, the PNN asymptotically converges to the Bayes optimal classifier. On the other hand, the PNNs have two limitations [21]:
1. The entire training set must be stored and used during testing (memory limitation), and
2. The amount of computation necessary to classify an unknown pattern is proportional to the size of the training set (computation limitation).

These limitations hinder the PNN performance. In order to increase the PNN performance, under the memory and computation limitations, it is a good practice grouping closer patterns by employing any clustering algorithm (k-means, k-medoids, etc.). Once we employ the clustering algorithm, we have to carefully choose cluster centers and assign one neuron for every cluster center.

In the OPNN, the sizes of input and output layers are determined by the number of features and the number of distinct classes in the training dataset, respectively. The hidden layer is constituted by assigning one node for each cluster center.

2.3 Radial Basis Function Neural Network

The RBFNN is an alternative model to the MLPNNs and MLFFNNs for the classification. It is explained in Figure 2 for a two-class problem. It can be extended to any number of classes.

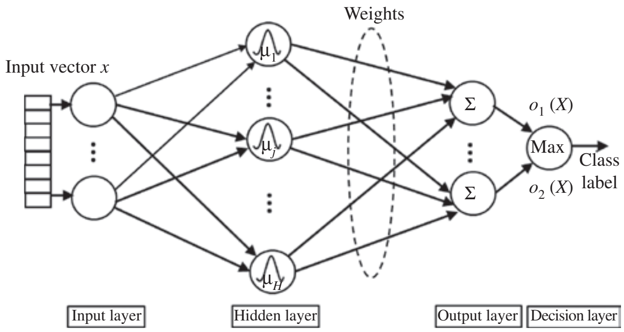

Figure 2: RBFNN model for classification task.

- **Input layer:** It functions similar to the PNN.
- **Hidden layer:** It also functions similar to the PNN. The output value of each hidden layer neuron is computed using Eq. (3).
- **Output layer:** The output layer is made up of two neurons, where 2 is the number of distinct classes. The response of the output layer neuron is a weighted sum of the hidden layer outputs, which is computed using Eq. (4).

$$0_j(X)=\sum_{i=1}^{H} w_{ji}\phi_i(X),\quad i=1,\ 2,\ \ldots,\ H,\ j=1,\ 2. \tag{4}$$

- **Decision layer:** It works similar to the PNN decision layer.

2.4 Optimal RBFNN

In the traditional RBFNN, the sizes of the input layer and output layer are determined by the number of features and the number of distinct classes in the training dataset, respectively. The problem lies on the size of the hidden layer. Usually, it is equal to the size of the training dataset. Although it is simple, it is not practical as most of the applications have numerous training patterns with high dimensionality. It is a good practice to cluster training patterns by employing clustering techniques. It is desirable to select proper cluster locations for better performance as this problem requires exponential time (NP-hard problem). This problem can be solved using meta-heuristic optimization techniques.

3 Proposed Methodology

3.1 Proposed Objective Function

The proposed multi-objective function has taken into account three metrics such as the spread (compactness) of the intra-clusters, separability between the inter-clusters, and loss function. The verbal notation of the fitness function is defined in Eq. (7).

$$\text{Fitness function}=\min\left\{\frac{\text{Compactness}}{\text{Separability}}+\text{Loss function}\right\} \tag{5}$$

$$\text{Fitness function}=\min\left\{\frac{\frac{1}{N}\sum_{i=1}^{N} S_i}{d(i,\ j)}+H(p,\ q)\right\} \tag{6}$$

S_i is a measure of the scatter within the cluster, which is defined as

$$S_i = \frac{1}{T_i}\sum_{i=1}^{T_i} \text{Euclidian distance}\,(X_j - A_i) \tag{7}$$

Here, A_i is the centroid of C_i, and T_i is the size of the cluster i. $d(i, j)$ is a measure of the separation between cluster C_i and cluster C_j [23].

In mathematical optimization, loss function (cross entropy) for classification problems represents the price paid for inaccurate predictions. It is defined for a two-class problem as follows:

$$H(p,q) = -\sum_{i=0}^{1} p_i \log q_i, \quad p \in \{y,\ 1-y\},\ q \in \{y,\ 1-y\} \tag{8}$$

p and q are true and predicted distributions, respectively.

For a better set of cluster positions, the fitness function needs to be minimized.

3.2 Proposed Diabetes-Network (Dia-Net)

The Dia-Net consists of two stages. In the first stage, it combines the OPNN and the ORBFNN and keeps the linear SVM [9, 11] in the second stage. The outputs of the ORBFNN and OPNNs are the inputs to the linear SVM classifier. The Dia-Net architecture is shown in Figure 3.

3.3 PSO-Based Clustering

The PSO [15, 27] is a population-based meta-heuristic optimization algorithm. In the PSO-based clustering, each particle is encoded to represent a set of cluster centers. Each particle is evaluated using fitness function.

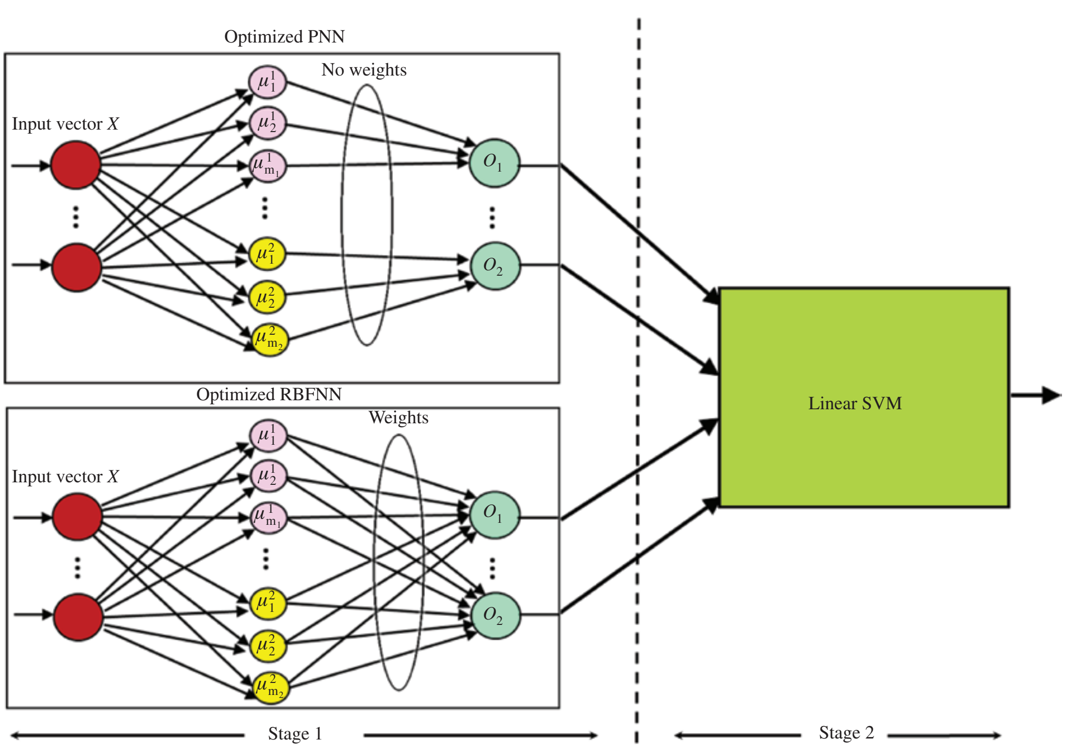

Figure 3: Proposed Dia-Net model.

Algorithm 1: PSO-based clustering.

Input: K initial clusters center
Output: Best K clusters center positions
$K \leftarrow$ number of clusters; $gBest \leftarrow [\,]$
for $i \leftarrow 1$ **to** *Population* **do**
  Initialize each particle
  $P_{velocity} = \text{rand}(\,)$; $P_{position} = \text{rand}(K)$
  $pBest \leftarrow P_{position}$
  gBest = gBest $\cup$ pBest
Compute the each particle's best position
pBest = min{*gBest*}
while *maximum iterations* **do**
  for $i \leftarrow 1$ **to** *Population* **do**
    Update particle velocity using below equation
    $v_i^{t+1} = v_i^t + c_1 * rand() * (pBest_i^t - p_i^t) + c_2 * rand() * (gBest_i^t - p_i^t)$
    Where, c_1 and c_2 are learning factors
    Update particle position using below equation
    $p_i^{t+1} = p_i^t + v_i^{t+1}$
    if fitness($P_{position}$) < fitness(pBest) **then**
      $pBest \leftarrow P_{\text{position}}$
    if fitness(pBest) < fitness(gBest) **then**
      $gBest \leftarrow pBest$
return *gBest*

In order to obtain high-density regions in a given dataset, the PSO-based clustering algorithm is applied on each class. The pseudo code for this algorithm is shown in Algorithm 1. It takes a number of clusters (K) as input and output best K cluster positions using the training dataset. It is used for the determination of the hidden layer size in the ORBFNN and OPNNs.

4 Experimental Results and Discussion

4.1 Experimental Setup

We used the PID dataset obtained from the University of California, Irvine repository [17] whose detail specifications are shown in Table 1. The PID dataset consists of a total of 768 diabetes patient data in which 500 records are related to diabetes negative (class label 0) and 268 records are related to diabetes positive (class label 1). For experimental purposes, the PID dataset is partitioned into the training and testing datasets. The training dataset constitutes 538 patterns (350 class 0 patterns and 188 class 1 patterns), and the testing dataset constitutes the remaining patterns.

Table 1: PID dataset attribute description.

Feature	Description	Feature	Description
1	Number of times pregnant	5	Serum insulin
2	Plasma glucose concentration	6	Body mass index
3	Diastolic blood pressure	7	Diabetes pedigree function
4	Triceps skin fold thickness	8	Age

Class 0 or 1: 0, diabetes negative; 1, diabetes positive.

Table 2: Fine-tuned parameters of OPNN and ORBFNN.

Parameter	Value		Explanation
	For OPNN	For ORBFNN	
Population	50	100	Population of particles
c_1	0.5	0.5	Importance of personal best value
c_2	1.5	1.5	Importance of neighborhood best value
Dimension of particles	1 to 180	1 to 180	Each particle dimension
Max-clusters-count	180	180	Maximum number of clusters
σ	1.2	1	Spread of radial basis functions

4.2 Parameter Tuning

In order to obtain the optimal cluster positions, it is necessary to fine tune the PSO parameters for the ORBFNN and OPNN using the training dataset. These fine-tuned parameter values for the OPNN and ORBFNN are listed in Table 2.

Once the PSO parameters are fixed, the PSO-based clustering is applied to fix the hidden layer neurons in the ORBFNN and OPNN. To figure out the number of hidden layer neurons, the PSO-based clustering is applied on each class training dataset. A performance plot is drawn for the number of hidden layer neurons versus the training accuracy. This plot is shown in Figure 4. From the figure, it is clear that the ORBFNN and OPNN obtained the highest accuracies at 155 and 119 centers per class, respectively.

Once the hidden layer size is determined, the ORBFNN and OPNN classifiers are constructed. Next, the dual stage Dia-Net is constructed by keeping the OPNN and ORBFNN classifiers in the first stage and the

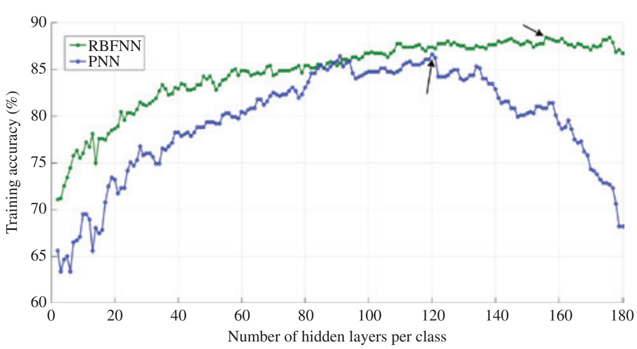

Figure 4: Performance plot.

Table 3: Simulation results on the PID training dataset for various *C* values.

C	Training accuracy (%)	Training sensitivity (%)	Training specificity (%)
0.1	90.00	95.68	81.32
0.2	89.13	95.62	79.57
0.4	**90.43**	**95.71**	**83.22**
0.5	90.43	95.71	82.22
0.6	89.13	95.62	79.57
0.7	89.57	95.65	80.43
0.8	89.57	95.65	80.43
0.9	89.57	95.65	80.43
1	90.00	95.68	81.32
10	90.00	95.68	81.32
100	90.00	95.68	81.32

linear SVM in the second stage, respectively. This Dia-Net is trained serially, i.e. the outputs of previous classifiers are used for the training of the next-level classifiers. During the training phase, the training dataset is provided to the OPNN and ORBFN, and the outputs of these classifiers are supplied as the inputs to the linear SVM. The final outputs are given by the linear SVM classifier. This Dia-Net has been trained with the trained dataset in order to fix the linear SVM regularization parameter (C) value. The performance of the linear SVM for various C values is given in Table 3. It is clear from the table results that at $C=0.4$, the linear-SVM has achieved a better performance on the training dataset.

4.3 Performance Analysis

4.3.1 Effect of PSO-Based Clustering

The performances of the RBFNN, ORBFNN, PNN, OPNN, and Dia-Net are compared in terms of the hidden layer size, network complexity, and percentage reduction in network complexity. These results are shown in Table 4. It is observed from the results that the proposed PSO-based clustering approach generated a few proper cluster center locations for the ORBFNN (i.e. 310), OPNN (i.e. 238) hidden layer neurons. This helps in reducing the network complexity of the RBFNN, PNN, and Dia-Net.

4.3.2 Effect of Cascaded Ensemble Framework

Once the RBFNN, PNN hidden layer sizes, and SVM regularization parameter values are fixed, the Dia-Net is experimented on testing dataset to evaluate its performance. The performance results of the RBFNN, ORBFN, PNN, OPNN, and Dia-Net on the testing dataset are shown in Table 5. It is clear from the table results that the Dia-Net outperformed all the classifiers in terms of accuracy, sensitivity, and specificity.

4.4 Comparative Analysis

The proposed Dia-Net is compared with the RBFNN variants in the literature. These results are shown in Table 6. It is clear from the results that the proposed network outperformed the other methods in terms of accuracy, sensitivity, and specificity.

Table 4: Comparison of the proposed method with other RBFNN variants of the same domain.

	RBFNN	ORBFNN	PNN	OPNN	Dia-Net
# Hidden layer neurons	768	310	768	238	548
# Links (complexity of network)	7680	3100	6912	2142	5242
% Reduction in network complexity	0	59.63	0	69.01	64.08

Table 5: Performance comparison of the proposed Dia-Net.

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)
RBFNN	65.22	100	0
ORBFNN	74.78	77.33	70.00
PNN	68.26	74.00	57.50
OPNN	63.04	59.33	70.00
Dia-Net	90.87	95.74	83.15

Table 6: Comparison of the proposed method with other RBFNN variants of the same domain.

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Reference
MEPGANf1f3	68.35	20.37	94.00	Qasem et al. [22]
MEPGANf1f2	72.78	45.20	87.11	Qasem et al. [22]
PSO-RBFN	72.60	77.34	63.75	Cheruku et al. [8]
Bee-RBF	71.13 ± 1.06	–	–	Cruz et al. [10]
RBFNN + SCVI	70.00	77.34	56.25	Cheruku et al. [7]
Proposed Dia-Net	90.87%	95.74%	83.15%	This paper

Table 7: Comparison of proposed method with other ensemble approaches.

Classifiers	PID dataset			
	Accuracy (%)	Sensitivity (%)	Specificity (%)	Reference
Casc	76.92 ± 0.6	–	–	Kaynak and Alpaydin [14]
GDA-LS-SVM	82.50	90.00	67.85	Polat et al. [20]
Bayesian boosting	73.18	82.60	55.60	Bashir et al. [3]
Stacking	68.23	76.00	53.73	Bashir et al. [3]
RF	71.74	53.81	80.40	Kandhasamy and Balamurali [12]
AdaBoost	76.43	52.99	89.00	Bashir et al. [5]
Bagging	77.99	75.96	85.00	Bashir et al. [5]
Majority voting	76.30	50.00	**90.40**	Bashir et al. [5]
Accuracy weighting	77.00	65.54	85.55	Bashir et al. [5]
HMV	77.08	78.93	88.40	Bashir et al. [4]
HM-BagMoov	78.21	78.65	92.60	Bashir et al. [5]
Proposed Dia-Net	**90.87**	**95.74**	83.15	**This study**

Finally, the proposed Dia-Net is compared with various ensemble techniques in the literature. These results are shown in Table 7. It is clear from the results that the proposed network outperformed in terms of accuracy, sensitivity and specificity compared to the other ensemble methods in the literature.

Overall, the proposed dual-stage cascade ensemble network called Dia-Net achieved the highest diabetes classification accuracy.

5 Conclusion

In this paper, to improve the diabetes prediction accuracy, a dual-stage Dia-Net is designed. The Dia-Net is constituted by combining the ORBFNN and OPNN in the first stage and keeping the SVM in the second stage. A supervised PSO-based clustering is proposed to obtain high density regions in the dataset. Moreover, a novel multi-objective fitness function is proposed for the PSO. The proposed Dia-Net is experimented on PID dataset. The experimental results proved that the proposed Dia-Net achieved more accuracy than the individual accuracies of the RBFNN, ORBFNN, PNN, and OPNN, and state-of-the-art models. It also reduced the network complexity and size of the hidden layer a lot.

Bibliography

[1] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl and J. Havel, Artificial neural networks in medical diagnosis, *J. Appl. Biomed.* **11** (2013), 47–58.

[2] J. Assal and L. Groop, Definition, diagnosis and classification of diabetes mellitus and its complications, *World Health Organ.* (1999), 1–65.
[3] S. Bashir, U. Qamar, F. H. Khan and M. Y. Javed, An efficient rule-based classification of diabetes using ID3, C4. 5, and CART ensembles, in: *Frontiers of Information Technology (FIT), 2014 12th International Conference on*, pp. 226–231, IEEE, 2014.
[4] S. Bashir, U. Qamar, F. H. Khan and L. Naseem, HMV: a medical decision support framework using multi-layer classifiers for disease prediction, *J. Comput. Sci.* **13** (2016), 10–25.
[5] S. Bashir, U. Qamar and F. H. Khan, IntelliHealth: a medical decision support application using a novel weighted multi-layer classifier ensemble framework, *J. Biomed. Inform.* **59** (2016), 185–200.
[6] B. Chandra and K. V. N. Babu, An improved architecture for probabilistic neural networks, in: *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pp. 919–924, IEEE, 2011.
[7] R. Cheruku, D. R. Edla and V. Kuppili, Diabetes classification using radial basis function network by combining cluster validity index and BAT optimization with novel fitness function, *Int. J. Comput. Intell. Syst.* **10** (2017), 247–265.
[8] R. Cheruku, D. R. Edla, V. Kuppili and R. Dharavath, PSO-RBFNN: a PSO-based clustering approach for RBFNN design to classify disease data, in: *International Conference on Artificial Neural Networks*, pp. 411–419, Springer, Cham, Switzerland, 2017.
[9] C. Cortes and V. Vapnik, Support-vector networks, *Mach. Learn* **20** (1995), 273–297.
[10] D. P. F. Cruz, R. D. Maia, L. A. da Silva and L. N. de Castro, BeeRBF: a bee-inspired data clustering approach to design RBF neural network classifiers, *Neurocomputing* **172** (2016), 427–437.
[11] T.-M. Huang and V. Kecman, *Linear Support Vector Machine*, http://www.linearsvm.com, Accessed: 30 September, 2016.
[12] J. P. Kandhasamy and S. Balamurali, Performance analysis of classifier models to predict diabetes mellitus, *Procedia Comput. Sci.* **47** (2015), 45–51.
[13] S. Kang, S. Cho and P. Kang, Multi-class classification via heterogeneous ensemble of one-class classifiers, *Eng. Appl. Artif. Intell.* **43** (2015), 35–43.
[14] C. Kaynak and E. Alpaydin, Multistage cascading of multiple classifiers: one man's noise is another man's data, in: *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*, pp. 455–462, CiteSeerX, The Pennsylvania State University, 2000.
[15] J. Kennedy, R. C. Eberhart and Y. Shi, *Swarm intelligence*, 1st Ed., Elsevier, Morgan Kaufmann, Amsterdam, Netherlands, 2001.
[16] M.-J. Kim, D.-K. Kang and H. B. Kim, Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction, *Exp. Syst. Appl.* **42** (2015), 1074–1082.
[17] M. Lichman, *UCI Machine Learning Repository*, School of Information and Computer Sciences, University of California, Irvine, 2013.
[18] M. Mirzaei, M. Z. A. Ab. Kadir, H. Hizam and E. Moazami, Comparative analysis of probabilistic neural network, radial basis function, and feed-forward neural network for fault classification in power distribution systems, *Electr. Power Compon. Syst.* **39** (2011), 1858–1871.
[19] F. Moretti, S. Pizzuti, S. Panzieri and M. Annunziato, Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling, *Neurocomputing* **167** (2015), 3–7.
[20] K. Polat, S. Güneş and A. Arslan, A cascade learning system for classification of diabetes disease: generalized discriminant analysis and least square support vector machine, *Expert Syst. Appl.* **34** (2008), 482–487.
[21] R. Priya and P. Aruna, A new eyenet model for diagnosis of diabetic retinopathy, *Appl. Artif. Intell.* **27** (2013), 924–940.
[22] S. N. Qasem, S. M. Shamsuddin, S. Z. M. Hashim, M. Darus and E. Al-Shammari, Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems, *Inf. Sci.* **239** (2013), 165–190.
[23] S. Ray and R. H. Turi, Determination of number of clusters in k-means clustering and application in colour image segmentation, in: *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pp. 137–143, The Pennsylvania State University, CiteSeerX, 1999.
[24] D. F. Specht, Probabilistic neural networks, *Neural Netw.* **3** (1990), 109–118.
[25] WHO, *World Health Organization*, http://www.who.int/diabetes/action_online/basics/en/, Accessed: 30 September, 2016.
[26] B. Yegnanarayana, *Artificial neural networks*, PHI Learning Pvt. Ltd., Delhi, 2009.
[27] Y. Zhang, S. Wang and G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, *Math. Probl. Eng.* **2015** (2015), 1–38.