Cluster Computing (2019) 22:S1329–S1344
https://doi.org/10.1007/s10586-017-1624-7

Cluster-based distributed architecture for prediction of student's performance in higher education

L. Ramanathan[1] · G. Parthasarathy[2] · K. Vijayakumar[1] · L. Lakshmanan[3] · S. Ramani[1]

Received: 14 November 2017 / Revised: 18 December 2017 / Accepted: 23 December 2017 / Published online: 9 January 2018

Abstract
Educational data mining (EDM) has emerged as a research area in recent years for researchers all over the world from different and related research areas. The EDM obtained knowledge can be used to offer suggestions to the academic planners in higher education institutes to enhance their decision-making process. Literature has suggested various prediction models for predicting the student's performance. This work proposes the cluster based distributed architecture for predicting the student's performance. The proposed cluster-based distributed architecture performs the prediction with clustering through Bayesian fuzzy clustering, feature extraction through Kernel-based principal component analysis, and prediction through the proposed Lion–Wolf based deep belief network (LW-DBN). The proposed architecture uses the LW algorithm to find the optimal weights for the DBN. The experimentation of the proposed work is done by collecting a real-time database and measuring the prediction performance through the mean square error (MSE) and root MSE (RMSE). The proposed LW-DBN model has achieved lower error performance than other models with MSE and RMSE values of 0.222606 and 0.050435, for the database.

Keywords EDM · Prediction · Distributed architecture · DBN · Real-time database

1 Introduction

Educational data mining (EDM) can be explained as processes designed for the analysis of data from educational settings for a better understanding of students and the environment, in which they learn in [1]. EDM [2] module tries to understand the learning behavior of the students and tries to predict the status of the student's performance shortly [3]. The EDM learns the study pattern of the students by analyzing the performance of the students in the past years. The data available in the college records is larger, and hence, the data mining schemes are to be adapted to the big data for building the efficient EDM [4]. The data mining applied to the big data in the EDM recommends various suggestions to the user regarding their academics, future planning, policy framing, etc. [5].

The prediction schemes are related to the information available in the educational institutions, which has the collection of the student's details, such as personal details, educational details, and other curricular activities. The information from the database can be subjected to the prediction based on the standard data mining schemes, such as classification, clustering, statistics, visualization, etc. Also, the clustering techniques suggested in the EDM only try to gather the information from the data. Predicting the performance of the students from the data available in the universities has faced considerable challenges in the recent years. Organizations use the predicted information for pre-planning the academic activities of the students hence; prediction scheme helps the student to achieve a better result at the end of the course. Some works considered the prediction, as predicting the student's performance for the upcoming semesters based on the past performance of the student may act as an early warning system for both the learners and the educators [6]. It also provides a learner to design a corrective system or extra coaching to the underperforming students [7,8].

✉ L. Ramanathan
lramanathan@vit.ac.in

[1] SCOPE, VIT, Vellore, Tamil Nadu 632014, India

[2] Jeppiaar Maamallan Engineering College, Jeppiaar Nagar, Sriperumpudur, Kancheepuram 602 108, India

[3] Department of CSE, School of Computing, Sathyabama University, Chennai, Tamil Nadu 600119, India

Literature has suggested the clustering based schemes for the prediction since the application of the clustering schemes in the fields, such as bioinformatics, image analysis, etc., have produced significant results. The clustering scheme divides the database into various groups by calculating the similarity measures between the data points in the database. One of the major challenges involved in the clustering is dividing the database with the vagueness [9]. The literature [10] has used the fuzzy rule-based classifiers (FRBCs) for constructing the prediction system for the students, and [11] has used the particle swarm optimization (PSO) for selecting the suitable centroids for clustering. Moreover, the existing works have suggested the prediction schemes based on the neural networks (NN) [12], decision tree, support vector machine (SVM) [13] and Naive Bayes [14] suitable for the EDM [15]. Several works [16] suggested the use of the academic performance of the students in the high school to predict the performance of the students in the future. This can be studied by the prediction scheme to identify the study pattern of the students.

This research intends to propose the prediction scheme for identifying the student's performance in the future. The database required for the prediction is collected in the real-time, which comprises of the information, such as individual, environmental, family, etc. The research introduces the cluster based distributed architecture for the predicting the performance of the students. The database may comprise of the missing information hence, the missing data is imputed with the use of the clustering based neural network defined in [17]. The architecture is provided with the set of computers for the processing, and it processes the data clustered through the Bayesian fuzzy clustering (BFC). For every cluster computer, Kernel-based principal component analysis (KPCA) performs the feature dimension reduction and provides it to the training algorithm. Once the best features are identified from the data, the prediction is performed using the proposed Lion–Wolf-deep belief network. Here, Lion–Wolf developed in the previous paper [18] is employed in the deep belief network (DBN) for the training process. Finally, the prediction results are merged in the final cluster nodes to aggregate the prediction results.

This paper contributes towards the EDM and the major beneficiary of this work is listed as follows,

- Firstly, this work intends to develop the cluster based distributed architecture for predicting the student's performance.
- Secondly, the LW-DBN algorithm is developed combining Lion–Wolf algorithm with the DBN, for predicting the student's performance from the available academic performance.

Further, the paper is organized as follows: Sect. 1 introduces the EDM model and the various techniques used for the prediction of the student's performance. Section 2 reviews the literary work, which deals with the EDM and the big data. Section 3 presents the proposed cluster-based distributed architecture and the proposed LW-DBN algorithm. The simulation results achieved by the proposed LW-DBN are analyzed in Sect. 4 and conclusion is presented in Sect. 5.

2 Motivation

2.1 Literature survey

This section presents the various literary works dealing with the clustering of the big data involved with the student's performance.

Kotsiantis [5] presented the prediction model based on the machine learning techniques, and they used the demographic characteristics of the students along with their marks for the training purpose. This prediction model had not considered the missing attributes present in the database hence, the model had failed to leverage the effect of the bias. Another prediction model was proposed by Romero et al. [18] based on the classification algorithms. The database for the training purpose was collected from the quantitative, qualitative and social network environments. The model had improved the prediction but the selected features by the model had not influenced the accuracy of the classifier. Wolff et al. [19] proposed the prediction model with the combined data sources, and the model had included various regression models for designing the prediction model. The model had defined the prediction capacity for the prediction of the student's performance. Guarín et al. [20] presented the prediction model based on the Naive Bayes and the decision tree classifier suitable for the EDM. The proposed model performed the prediction by the database containing both the academic and the non-academic information.

Chen et al. [21] presented the big data platform for clustering the data from the health records, and the model used the Hadoop framework for the implementation of the clustering model. The model had improved the scalability and the flexibility. However, the cost of implementation of the model is high. Hussain et al. [4] proposed a prediction framework for analyzing the big data involved during the EDM, and the framework tried to predict the student's performance through the verification strategy. Even though the model provided improved prediction results, it lacked in the integration of various data mining strategies. Bharara et al. [1] had presented a model for learning the context in the database, and they presented the learning model based on the interrelationships between the attributes of the database. K-means clustering model used in the model clustered the database based on the feature set obtained from the learning context. Asif et al. [16] presented the data mining methods to study the performance of undergraduate students. Two aspects of students' performance have been focused for building the model. The first model found the prediction based on the performance of the students end of a four-year study course. The other model concentrated in the typical progressions of the students or the study pattern, and then, these patterns were combined with the results of the prediction model. The work lacked in the generalization of the results.

Vojt [25] had focused on three different kinds of deep neural networks: (i) the multilayer perceptron, (ii) the convolutional neural network, and (iii) the deep belief network. The comparison results represent the superior performance of multilayer perceptrons and convolutional neural networks. The deep belief network implementation performs slightly better for restricted Boltzmann machine (RBM) layers with up to 1000 hidden neurons. Yazdani and Jolai [26] have proposed a population-based algorithm named as the Lion optimization algorithm (LOA). The results of LOA were compared with some well-known meta-heuristics for benchmark problems. The results showed that the LOA had high performance than the other algorithms.

Mirjalili et al. [27] have proposed a meta-heuristic named grey wolf optimizer (GWO). The GWO algorithm copied the hunting mechanism and leadership hierarchy of grey wolves. It includes the three main steps, namely (i) searching for prey, (ii) encircling prey, and (iii) attacking prey. The results depicted that the GWO algorithm provides competitive results than the well-known meta-heuristics. Montana and Davis [28] have described a set of experiments performed on data from a sonar image classification problem. These experiments demonstrated the improvements obtained by a genetic algorithm rather than backpropagation and the evolution of the performance of the genetic algorithm.

2.2 Challenges

Various challenges involved in the process during the clustering of the big data is described as follows,

- Digitization of the academic records allowed the universities to store the data on the websites or the electronic form. Also, the universities store the data of a large number of students, and hence, the concept of the big data needs to be incorporated into the data mining schemes while designing EDM [16].
- The quality of the information available in the academic records mainly influences the prediction schemes. Also, the database may contain missing attributes that severely reduce the accuracy of the prediction [9].
- In work [20], both the academic and the non-academic information have been used for predicting the student's performance, but the prediction model requires only the academic records for the prediction. Hence, for this database feature selection requires high attention.
- The use of the machine learning schemes to develop the prediction model has gained popularity nowadays. But, the features for the training need to be carefully selected [10].

3 Proposed method

This section presents the proposed cluster-based distributed architecture for predicting the student's performance in the future. Figure 1 presents the diagram for the proposed cluster-based distributed architecture. The proposed prediction scheme contains two merger layers, namely merger layer 1 and merger layer 2, and two distributed layers namely, distributed layer 1 and distributed layer 2. The data for the prediction scheme is collected from the web pages of institutions; the database D is provided to the proposed architecture as different data sources. Each data source provided for the distributed layer 1 contains the information of the students belonging to the individual colleges. The distributed layer 1 imputes the missing data into the data sources, and clusters the data sources as smaller groups. Each merger present in the merger layer 1 combines the clusters and provides it to the distributed layer 2. In the distributed layer 2, the features of the data group from the merger are extracted and provided for the training in the proposed LW-DBN network. The LW-DBN predicts the student's performance for the forthcoming semesters based on the features. Then, the predicted information is passed over the merger layer 2, and it provides the information about the marks secured by the students in the upcoming semesters.

One of the major challenges in constructing the prediction model is the data collection, and the collected data must be genuine and free-from errors. The proposed cluster-based architecture used the academic records of the students from the college websites and the records. The database collected from each college corresponds to student information from the individual college. Let us consider there are |S| students available in each college, and the collected data sources are represented as,

$$\mathrm{D} = \{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_\mathrm{i}, \ldots, \mathrm{D}_\mathrm{D}\} \quad (1)$$

where D_i refers to the data source collected from the ith college or the institution, and the size of the individual data source varies as $|\mathrm{S}|^* \mathrm{N}$. The term |S| indicates the student count in each data source and N refers to the total attributes available in the data source. The data source contains the information about the personal information, academic records, extracurricular activities, and the schooling records. The proposed cluster-based distributed architecture

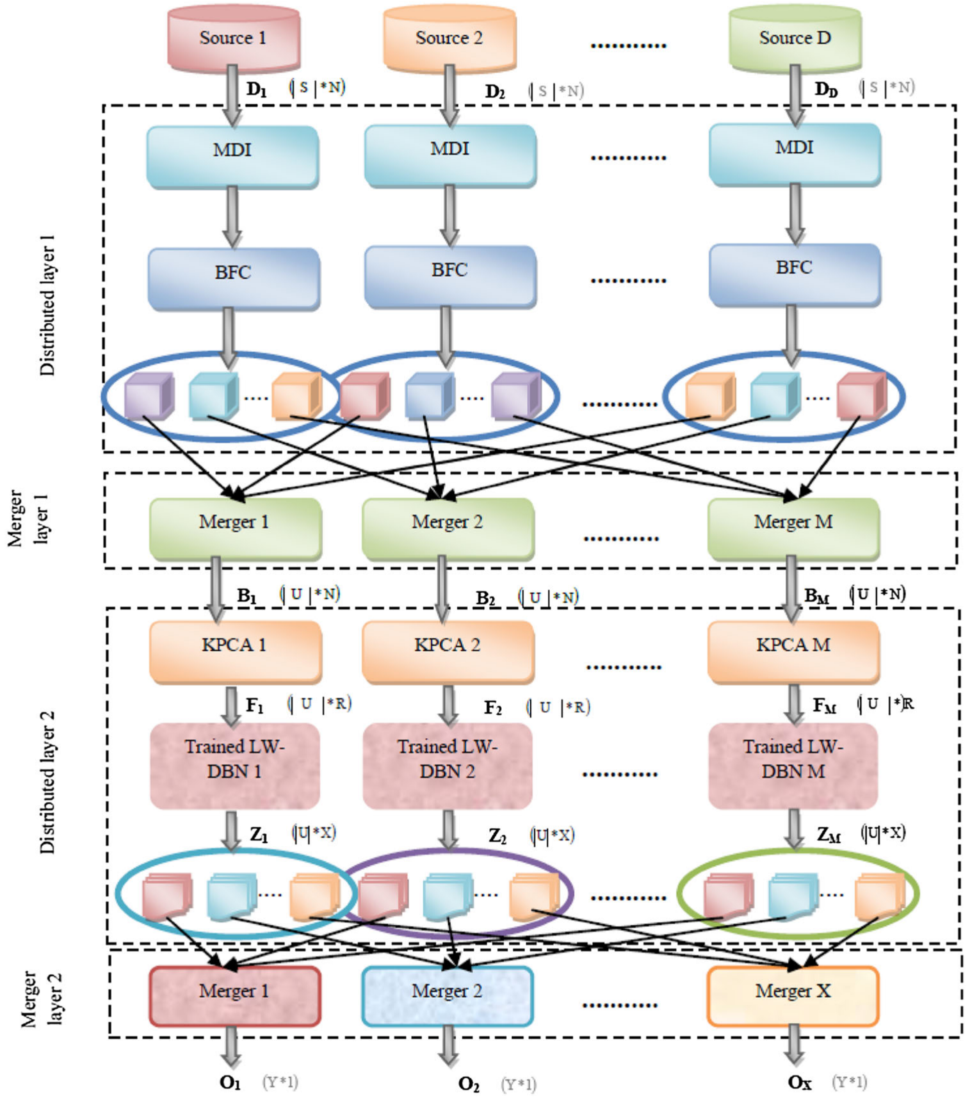


Fig. 1 Block diagram of the proposed cluster-based distributed architecture for predicting the student's performance

has D number of data sources, M number of nodes in the merger layer 1, M number of nodes for the feature selection, and X number of nodes in the merger layer 2, and hence, the total number of cluster nodes in the proposed cluster-based distributed architecture is expressed as,

$$A = D + M + M + X \tag{2}$$

where, A refers to the total number of cluster nodes in the proposed architecture.

3.1 Construction of the distributed layer 1

The distributed layer 1 forms the primary layer of the proposed prediction scheme, and it has the missing data imputation (MDI) block and the clustering block. Since the D data sources contain a large number of data, the proposed architecture tries for the simultaneous processing of the data. The distributed layer 1 has D number of processors for the clustering process.

3.1.1 Missing data imputation

The data sources collected from information sources contains missing data, and hence, this factor needs to be analyzed clearly for predicting the performance of the students. The MDI block present in the proposed cluster-based distributive framework depends on the clustering based neural network defined in [17]. Here, the data present in the data sources are clustered with the WLI fuzzy clustering model, which finds the centroid by averaging the data points. The clustered information is provided to the hybrid neural network, where the GWO is involved in training the weights of the NN. The hybrid NN provides the information about the missing data available in the data source. Finally, the MDI block fills the missing attribute information along with the data source and sends to the next layer.

3.1.2 Bayesian fuzzy clustering

The data source contains large information, which is clustered into smaller groups for the processing. The proposed architecture utilized the BFC [22] scheme for dividing the data source into smaller groups. The BFC model eliminates the manual interpretation of choosing a number of clusters for the data clustering, and hence, the model is more advantageous while clustering large data. The distributed layer 1 has D number of systems for clustering the data from each MDI block. Consider the data source D_i from the ith MDI block is provided to the ith BFC block for data clustering. The BFC utilizes the membership function Q to find the suitable cluster prototypes. The BFC uses the uniform symmetric Dirichlet proposal for the clustering and is defined as,

$$g_n^+ \sim \text{Dirichlet}\,(\eta = 1_M) \tag{3}$$

where M and g_n^+ refer to the number of clusters and the uniform symmetric Dirichlet proposal for the data clustering. The cluster prototypes are defined through the distribution, and it is expressed as,

$$\tilde{P}(d_n, g_n|\tilde{G}) = P(d_n|g_n, G)\tilde{P}\left(g_n|G\right) \tag{4}$$

$$\tilde{P}(d_n, \tilde{g}_n|G) = \sum_{m=1}^{M} N\left(d_n\middle|h_n, g_{nm}^{-v}\right)g_{nm}^{-v\beta/2}\text{Dirichlet}(g_n\,|\eta\,) \tag{5}$$

The cluster prototypes are subjected to the variation due to the Gaussian distribution, and hence, the Markov chain state rule is applied. The modified value of the cluster prototype is given as,

$$P(D_i, h_m|Q) = P(D_i|Q, h_m)P\,(h_m) \tag{6}$$

where Q refers to the membership function used for the data clustering. Finally, the clustered information of the data source D_i from the ith BFC is represented as,

$$M_i = \left\{M_1^i, M_2^i, \ldots, M_m^i, \ldots, M_M^i\right\} \tag{7}$$

where M_i represents the output of the ith BFC in the distributed layer 1, and M is the total number of clusters obtained from the ith BFC. Each clustered output from the BFC has the size of $|J|^*\,N$. Hence, the final output of the distributed layer 1 is the data clusters from each BFC, and it is represented as,

$$M = \{M_1, M_2, \ldots, M_i, \ldots, M_D\} \tag{8}$$

where D is the total number of nodes required for BFC, which is equivalent to the number of data sources.

3.2 Construction of the merger layer 1

The merger layer 1 combines the results of the distributed layer 1. The merger layer 1 has M number of mergers, represented as $\{B_1, B_2, \ldots, B_m, \ldots, B_M\}$, for collecting the M clusters from each BFC. The function of the mth merger in the merger layer 1 is to combine the mth cluster group produced by each BFC. The output of the mth merger is defined as,

$$B_m = \left\{M_m^1, M_m^2, \ldots, M_m^i, \ldots, M_m^D\right\} \tag{9}$$

where M_m^i indicates the mth cluster group from the ith BFC. The data size of each merger is defined as $|U|^*\,N$.

3.3 Construction of the distributed layer 2

The distributed layer 2 gets the merged data from the merger layer 1 for the feature selection, followed by the prediction. The distributed layer is provided with M number of feature selector blocks and the predictor blocks. The proposed architecture employs the KPCA model for feature selection and the proposed LW-DBN algorithm for the prediction.

3.3.1 Feature selection: KPCA

It is necessary to select the suitable features from the clustered information. The proposed architecture makes use of the existing KPCA [23] model to select the features from each merger. The distributed layer 2 has M KPCA feature selector blocks, defined as $\{F_1, F_2, \ldots, F_m, \ldots, F_M\}$. Consider the data group available in the mth merger as B_m, which is of the size $|U|^*$ N, used as the training data for the feature selection by the KPCA model. Consider the data present in the merger as $B_m = \left\{B_{uq}\right\}$, where the value of the p and q vary as the size of the data $|U|^*$ N. The training sample for the feature selection is defined as $\frac{1}{|U|}\sum_{u=1}^{|U|}\gamma\,(b_u)$. The KPCA uses the kernel function for selecting the features from the training data, and it is expressed as,

$$T\,(b{,}b_u) = \gamma^T(b_u)\gamma(b) \tag{10}$$

where the term T refers to the kernel function. The application of the kernel function to the data yields the covariance matrix defined as,

$$B = \frac{1}{|U|}\sum_{u=1}^{|U|}\gamma^T(b_u)\gamma(b) \tag{11}$$

The selection of the features from the data is considered as the Eigenvalue problem, and it is stated as follows,

$$B \cdot V = \lambda \cdot V \tag{12}$$

where V indicates the associate Eigenvectors. λ represents the eigenvalue of B. Rearranging the above equation, the required Eigenvectors for the feature selection are obtained, and it is expressed as,

$$T \cdot \phi = u \cdot \lambda \cdot \phi \tag{13}$$

where ϕ indicates the associative Eigenvector. u indicates the total eigenvalue coefficients. The features extracted from the training data $\mathrm{B_m}$ is represented as,

$$\mathrm{f(B_{uq})} = \sum_{\mathrm{u=1}}^{|\mathrm{U}|} \phi_{\mathrm{uq}} \gamma^{\mathrm{T}}(\mathrm{b_q}) \gamma(\mathrm{b_u}) \tag{14}$$

The features form the mth KPCA is represented by the following equation,

$$\mathrm{F_m} = \left\{ \mathrm{f_m^1, f_m^2, \ldots, f_m^r, \ldots, f_m^R} \right\} \tag{15}$$

where f_m^r represents the rth feature from the mth KPCA and the feature set from each KPCA has the size of |U|* R, such that the value of $R$is less than the total number of attributes N.

3.3.2 Prediction of student's performance using the proposed LW-DBN

The features selected from the KPCA are provided to the LW-DBN for the training purpose. The distributed layer has M number of proposed LW-DBN for predicting the students' performance. The features selected from each KPCA are fed to the proposed LW-DBN network for the training purpose. Figure 2 presents the architecture of the LW-DBN model containing the two RBM layers and one (multilayer perception) MLP layers.

The proposed LW-DBN network has two RBM layers and one MLP layer, and the features from the KPCA are provided to the first RBM layer for training. The neurons corresponding to the visible layer of the first RBM layer corresponds to the input feature size, and the hidden layers of the RBM 1 have the corresponding weights. The input provides to the LW-DBN is represented as,

$$\mathrm{L^1} = \left\{ \mathrm{L_1^1, L_2^1, \ldots, L_r^1, \ldots, L_R^1} \right\}; \quad 1 \leq \mathrm{r} \leq \mathrm{R} \tag{16}$$

where, $\mathrm{L_r^1}$ refers to the feature input to the rth neuron layer of the RBM layer 1 and R is the total number of neurons in the input layer. The RBM hidden layers in the first RBM layer of the LW-DBN is represented as,

$$\mathrm{H^1} = \left\{ \mathrm{H_1^1, H_2^1, \ldots, H_s^1, \ldots, H_a^1} \right\}; \quad 1 \leq \mathrm{s} \leq \mathrm{a} \tag{17}$$

where, $\mathrm{H_s^1}$ refers to the hidden neuron in the first RBM layer. The weights present in the RBM layer of the LW-DBN is represented as,

$$\mathrm{W^1} = \left\{ \mathrm{W_{rs}^1} \right\} \tag{18}$$

where, $\mathrm{W_{rs}^1}$ corresponds to the weights between the rth visible and the sth hidden layer of the RBM 1. The expression of the output of the RBM layer 1 in the LW-DBN corresponds to the weight and the feature input, and it is expressed as,

$$\mathrm{K_c^1} = \tau \left[\mathrm{y_s^1} + \sum_{\mathrm{s}} \mathrm{L_r^1 H_{rs}^1} \right] \tag{19}$$

where, τ indicates the activation function and the expression for each output is combined and expressed as,

$$\mathrm{K^1} = \left\{ \mathrm{K_s^1} \right\}; \quad 1 \leq \mathrm{s} \leq \mathrm{a} \tag{20}$$

Similarly, the output of the RBM layer 1 is provided as the input to the RBM layer 2, and the output of the RBM layer two is expressed as $\mathrm{K^2}$. The output of the RBM layer two is directly fed to the MLP layer, and it is given as,

$$\mathrm{I} = \{\mathrm{I_1, I_2, \ldots, I_s, \ldots, I_a}\} = \left\{ \mathrm{K_s^2} \right\}; \quad 1 \leq \mathrm{s} \leq \mathrm{a} \tag{21}$$

where, $\mathrm{u_j}$ represents the jth input neuron of the MLP layer, and similarly RBM layer has X number of hidden layers. The weights corresponding to the input and the hidden layer of the MLP layer is expressed below,

$$\mathrm{w^{I\_MLP}} = \left\{ \mathrm{w_{sx}^{I\_MLP}} \right\}; \quad 1 \leq \mathrm{s} \leq \mathrm{a}; \quad 1 \leq \mathrm{x} \leq \mathrm{X} \tag{22}$$

$$\mathrm{w^{H\_MLP}} = \left\{ \mathrm{w_{xc}^{H\_MLP}} \right\}; \quad 1 \leq \mathrm{x} \leq \mathrm{X}; \quad 1 \leq \mathrm{c} \leq \psi \tag{23}$$

where, $\mathrm{w^{I\_MLP}}$ indicates the weight in the input of the MLP, $\mathrm{w^{H\_MLP}}$ indicates the weights in the hidden layer of the MLP, and ψ is the number of output neurons. The output of the hidden layer present in the MLP depends on the following equation,

$$\mathrm{e_x} = \left[\sum_{\mathrm{s=1}}^{\mathrm{a}} \mathrm{w_{sx}^{I\_MLP}{*}I_s} \right] \mathrm{E_X} \forall \mathrm{I_s} = \mathrm{K_s^2} \tag{24}$$

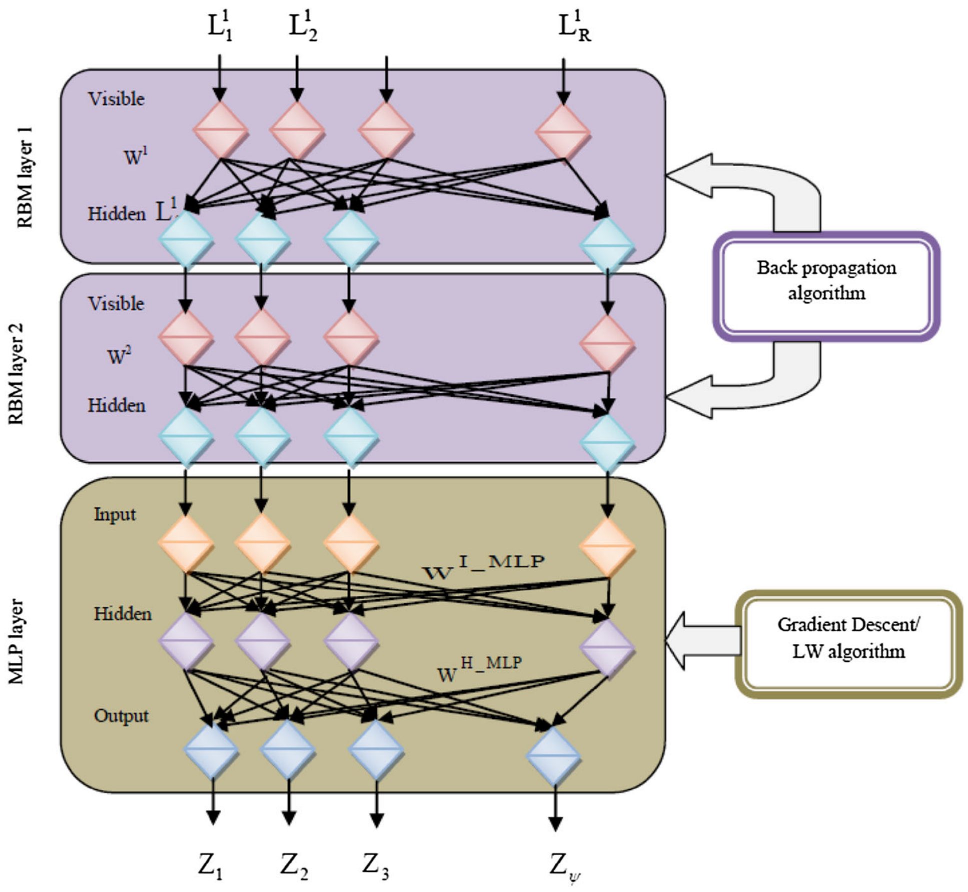


Fig. 2 Architecture of the proposed LW-DBN model

The final output of the proposed LW-DBN is expressed as,

$$Z = \sum_{x=1}^{X} w_{xc}^{H-MLP*} e_x \tag{25}$$

where, e_x is the output of the hidden layer.

A. Training phase of the DBN with the existing LW algorithm

The optimal weights for the RBM and the MLP layers are found with the use of the existing LW algorithm defined in [24]. The LW algorithm is the integration of the LOA and GWO algorithms. In the existing work [24], the LW algorithm is used for the training of the NN, for the better learning of the deep features in the database, this work replaced the NN with the DBN. In the training phase of the DBN [25], the optimal weights of the MLP layers are found with the use of the LW algorithm along with the gradient descent algorithm. The weights involved in the RBM layers are trained with the use of the backpropagation. The weights in the input and the hidden layer of the MLP are trained with the use of the LW and the gradient descent algorithm. The training procedure for finding the optimal weights in the MLP layer is defined as follows,

(1) The weights present in the input and the hidden layer of the MLP l are initialized based on Eqs. (22) and (23), respectively.

(2) Provide the training input to the MLP layer obtained from the second RBM layer.

(3) Find the MLP layer output Z and the bias e_x based on Eqs. (25) and (24), respectively.

(4) The computed output of the MLP layer has some deviation from the ground value. Thus, the error value is expressed as follows,

$$C_{avg} = \frac{1}{x} \sum_{c=1}^{X} (Z_c - \psi_c)^2 \tag{26}$$

where, C_{avg}indicates the deviation of the response of the classifier form the actual response. Z_c indicates the output of cth layer of the LW-DBN and the term ψ_c indicates the desired response.

(5) The error computed from the previous step contributes to the weight adjustment in the input and the hidden layer of the LW-DBN, and it is expressed in the following equations,

$$\Delta w_{sx}^{\text{I-MLP}} = -\chi \frac{\partial C_{avg}}{\partial w_{sx}^{\text{I-MLP}}} \tag{27}$$

$$\Delta w_{x}^{\text{H-MLP}} = -\chi \frac{\partial C_{avg}}{\partial w_{x}^{\text{H-MLP}}} \tag{28}$$

(6) Update the weight of the input and the hidden layer of MLP based on the gradient descent algorithm, and it is defined as,

$$w_{sx(Gr)}^{\text{I-MLP}}(t+1) = w_{sx}^{\text{I-MLP}}(t) + \Delta w_{sx}^{\text{I-MLP}} \tag{29}$$

$$w_{x(Gr)}^{\text{H-MLP}}(t+1) = w_{x}^{\text{H-MLP}}(t) + \Delta w_{x}^{\text{H-MLP}} \tag{30}$$

(7) The weight update based on the LW algorithm for both the input and the hidden layer of the MLP is defined as follows,

$$w_{sx(LM)}^{\text{I-MLP}}(t+1) = \frac{\vec{w}_{sx(LM)}^{\text{I-MLP}}(1) + \vec{w}_{sx(LM)}^{\text{I-MLP}}(2) + \vec{w}_{sx(LM)}^{\text{I-MLP}}(3) + \vec{w}_{sx(LM)}^{\text{I-MLP}}(4)}{4} \tag{31}$$

where, the terms $\vec{w}_{sx(LM)}^{\text{I-MLP}}(1)$ refer to the position of the first best search agent, $\vec{w}_{sx(LM)}^{\text{I-MLP}}(2)$, $\vec{w}_{sx(LM)}^{\text{I-MLP}}(4)$ indicate the position of second best search agent, and $\vec{w}_{sx(LM)}^{\text{I-MLP}}(3)$ position based on the third search agent for the input layer. The values of the $\vec{w}_{sx(LM)}^{\text{I-MLP}}(1)$, $\vec{w}_{sx(LM)}^{\text{I-MLP}}(2)$, and $\vec{w}_{sx(LM)}^{\text{I-MLP}}(3)$ are obtained from the position update done based on the GWO algorithm, and their expressions are defined as follows,

$$\vec{w}_{sx(LM)}^{\text{I}-\text{MLP}}(1) = \vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_1) - \vec{z}_1 \cdot \vec{l}(\omega_1) \tag{32}$$

$$\vec{w}_{sx(LM)}^{\text{I-MLP}}(2) = \vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_2) - \vec{z}_2 \cdot \vec{l}(\omega_2) \tag{33}$$

$$\vec{w}_{sx(LM)}^{\text{I-MLP}}(3) = \vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_3) - \vec{z}_3 \cdot \vec{l}(\omega_3) \tag{34}$$

where, the value of the $\vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_1)$, $\vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_2)$ and $\vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_3)$ indicates the search position at the direction of ω_1, ω_2, and ω_3 in the 2D space, respectively. The value of $\vec{w}_{sx(LM)}^{\text{I-MLP}}(\omega_1)$ indicates the solution update from the fertility evaluation of the female Lion used in the LOA and the value of the $\vec{w}_{sx(LM)}^{\text{I-MLP}}(4)$ is indicated as follows,

$$\vec{w}_{sx(LM)}^{\text{I-MLP}}(4) = \min\left[\vec{w}_{sx(LM)}^{\text{I-MLP}}(\max), \max\left[\vec{w}_{sx(LM)}^{\text{I-MLP}}(\min), \nabla c\right]\right] \tag{35}$$

where, the value of the ∇c indicates the female update function used in the LOA. Similarly, the weight update for the hidden layer of the MLP is expressed as follows,

$$w_{sx(LM)}^{\text{H-MLP}}(t+1) = \frac{\vec{w}_{sx(LM)}^{\text{H-MLP}}(1) + \vec{w}_{sx(LM)}^{\text{H-MLP}}(2) + \vec{w}_{sx(LM)}^{\text{H-MLP}}(3) + \vec{w}_{sx(LM)}^{\text{H-MLP}}(4)}{4} \tag{36}$$

(8) Calculate the output of the algorithm based on the weight update done through the gradient descent algorithm and the calculated error is expressed as $C_{avg(Gr)}$

(9) Similarly, compute the output of the algorithm with the weights updated based on the LW algorithm and the defined error is expressed as $C_{avg(LW)}$.

(10) Finally, the final weight of the input and the hidden layers are calculated based on Eqs. (37) and (38). The weight of the algorithm with the minimal error value replaces the actual weight of the MLP layers.

$$w_{jx}\text{I-MLP}\,(t+1) = \begin{cases} w_{sx(LW)}\text{I-MLP}\,(t+1)\,; & \text{if } C_{avg(LW)} < C_{avg(Gr)} \\ w_{sx(Gr)}\text{I-MLP}\,(t+1)\,; & \text{otherwise} \end{cases} \tag{37}$$

$$w_{jx}^{\text{H-MLP}}(t+1) = \begin{cases} w_{sx(LW)}^{\text{H-MLP}}(t+1)\,; & \text{if } C_{avg(LW)} < C_{avg(Gr} \\ w_{sx(Gr)}^{\text{H-MLP}}(t+1)\,; & \text{otherwise} \end{cases} \tag{38}$$

(11) At the end of the iteration, the optimal weights of the MLP layer are returned by the proposed LW-DBN algorithm.

B. Testing of the proposed LW-DBN classifier

The optimal weights from the LW-DBN training are used in the testing phase. For the features of the test input D^{test}, the LW-DBN classifier finds the marks of students in the next semester.

3.4 Construction of the merger layer 2

The merger layer 2 in the proposed architecture has X number of mergers, and the merger provides the performance of the students in various semesters with the predicted results. The output O from the merger present in merger layer 2 has the size of Y*1. The term *Y* represents a group of the students present in the database, and thus, each merger in the merger layer 2 provides the marks of Y students in the semester 1.

Thus, the X merger present in the merger layer two present in the marks of Y students for the X semesters. The following equation represents the output of the X mergers present in the merger layer 2.

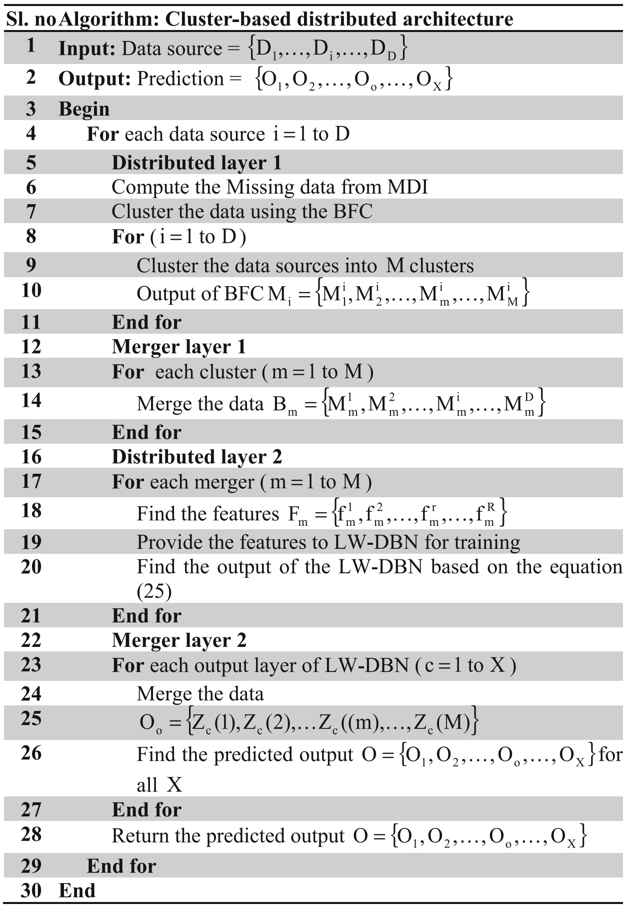
$$\mathrm{O} = \{\mathrm{O}_1, \mathrm{O}_2, \ldots, \mathrm{O}_o, \ldots, \mathrm{O}_X\} \tag{39}$$

where, O_o refers to the output of the oth merger present in the merger layer 2. The oth layer of the merger layer 2 combines the results of each mth output layer of the LW-DBN present in the distributed layer 2. Thus, the first layer of the merger layer 1 contains the semester one marks of Y students present in the database D. The expression of the output of the oth layer of the merger layer two is defined as follows,

$$O_o = \{Z_c(1), Z_c(2), \ldots Z_c(m), \ldots, Z_c(M)\} \tag{40}$$

where, the term $Z_c(m)$ refers to the cth output layer in the mth LW-DBN network, and its size is Y*1. Finally, each merger provides the individual semester marks of the Y students in the (t + 1)th semester.

Algorithm 1 presents the pseudocode of the proposed cluster based distributed architecture for predicting the student's performance.



```
Sl. no Algorithm: Cluster-based distributed architecture
Input: Data source = $\{D_1,\ldots,D_i,\ldots,D_D\}$
Output: Prediction = $\{O_1,O_2,\ldots,O_o,\ldots,O_X\}$
Begin
    For each data source $i = 1$ to D
        Distributed layer 1
        Compute the Missing data from MDI
        Cluster the data using the BFC
        For ( $i = 1$ to D )
            Cluster the data sources into M clusters
            Output of BFC $M_i = \{M^i_1, M^i_2,\ldots,M^i_m,\ldots,M^i_M\}$
        End for
        Merger layer 1
        For each cluster ( $m = 1$ to M )
            Merge the data $B_m = \{M^1_m, M^2_m,\ldots,M^i_m,\ldots,M^D_m\}$
        End for
        Distributed layer 2
        For each merger ( $m = 1$ to M )
            Find the features $F_m = \{f^1_m, f^2_m,\ldots,f^r_m,\ldots,f^R_m\}$
            Provide the features to LW-DBN for training
            Find the output of the LW-DBN based on the equation (25)
        End for
        Merger layer 2
        For each output layer of LW-DBN ( $c = 1$ to X )
            Merge the data
            $O_o = \{Z_c(1), Z_c(2),\ldots Z_c((m),\ldots,Z_c(M)\}$
            Find the predicted output $O = \{O_1,O_2,\ldots,O_o,\ldots,O_X\}$ for all X
        End for
        Return the predicted output $O = \{O_1,O_2,\ldots,O_o,\ldots,O_X\}$
    End for
End
```

4 Results and discussion

This section presents simulation results achieved by the proposed LW-DBN classifier for predicting the student's performance.

4.1 Experimental setup

The experimentation of the proposed work is implemented in the MATLAB tool, and the setup requires the configurations of PC with the Windows 10 OS, Intel I3 processor, and the 4 GB RAM.

4.1.1 Database description

The data for the experimentation is collected in the real time, and it contains information about the student's details, environment, and performance of the student in school, and the academic record in each semester during the college.

4.1.2 Evaluation metrics

The evaluation of the proposed system is measured based on the metrics, such as MSE, and RMSE, which define the error performance. The mathematical expression for the MSE and the RMSE is expressed as follows,

MSE The MSE metric defines the quality of the prediction system by calculating the deviation of the predicted value of the model from the actual value provided by the database, and it is expressed as follows,

$$\mathrm{MSE} = \frac{1}{\mathrm{X}} \sum_{o=1}^{\mathrm{X}} \left(\mathrm{O}_o - \hat{\mathrm{O}}_o \right)^2 \tag{41}$$

where, O_o indicates the predicted value and $\hat{\mathrm{O}}_o$ refers to the ground truth information.

RMSE The RMSE also defines the error performance of the models and is expressed as the root value of the MSE metric.

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{42}$$

Since both the evaluation metrics RMSE and the MSE defines the error performance of the model, lower value of the RMSE and the MSE indicates improved performance.

4.1.3 Comparative models

The experimentation of the proposed cluster-based distributed architecture along with the LW-DBN is compared with the various existing techniques, such as DBN, Lion based DBN (L-DBN), Grey wolf based DBN (W-DBN), and

Genetic Algorithm based DBN (GA-DBN). The description of the existing models used for the analysis of the proposed LW-DBN is explained as follows,

DBN [25] The prediction model with the DBN network uses the backpropagation algorithm for training the weights and bias present in the RBM layer and the MLP layer.

L-DBN In the L-DBN, the weights present in the MLP layer of the DBN is trained with the Lion optimization algorithm [26].

W-DBN Similar, to the L-DBN algorithm, the DBN is trained with the grey wolf optimization [27] algorithm for predicting the student's performance.

GA-DBN In the existing DA-DBN algorithm, the weights involved in the MLP layer of the DBN is trained with the genetic algorithm (GA) [28].

4.2 Comparative analysis of the proposed LW-DBN

The performance of the proposed LW-DBN is compared with the DBN, L-DBN, W-DBN, and the GA by varying the training percentage of the database (k), crossfold validation (cf), iteration (t), step ratio (sr), and the dropout ratio (dr). The results of the proposed LW-DBNare analyzed based on the MSE, and the RMSE metrics.

4.2.1 Comparative analysis of models without the crossfold validation

A. Analysis based on varying the step ratio (sr)

Here, the comparative analysis is done by varying the step ratio of the DBN network and training the database without the crossfold validation. Figure 3a depicts the comparative analysis of the models based on MSE metric for varying sr values. For sr = 0.3, the existing models DBN, L-DBN, W-DBN, and the GA-DBN achieved the MSE values of 0.337239, 0.238128, 0.559461, 0.536805, respectively, while the proposed LW-DBN algorithm achieved the lowest MSE value of 0.234044. Similarly, for the analysis based on RMSE as shown in Fig. 3b, the existing DBN, L-DBN, W-DBN, and the GA-DBN has the RMSE values of 0.116509, 0.056862, 0.314537, and 0.28957, respectively for sr = 0.3. But, the proposed LW-DBN model has achieved the lower RMSE value of 0.054907 for the sr value of 0.3.

B. Analysis based on varying the dropout ratio (dr)

Figure 3c, d presents the comparative analysis of the models based on the varying values of dr for the database trained without crossfold validation. Analysis based on the MSE, as shown in Fig. 3c, shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN metrics achieved the MSE values of 0.374352, 0.228539, 0.251737, and 0.539084 respectively, while the proposed LW-DBN model has the lower MSE value of 0.222606 for dr the value of 0.3. Likewise, analysis based on the RMSE depicted in Fig. 3d shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.132749, 0.053137, 0.064499, and 0.292894 respectively for the dr value of 0.3. The proposed LW-DBN algorithm achieved the RMSE value of 0.050436 for the dr value of 0.3 thus outperforms the other models.

C. Analysis based on varying the iteration (T)

Figure 4a, b presents the comparative analysis of the models based on the varying values of iteration (T) without crossfold validation. Analysis based on the MSE, shown in Figure 4a, shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN metrics achieved the MSE values of 0.53122, 0.237958, 0.563879 and 0.563879, respectively, while the proposed LW-DBN model has the lower MSE value of 0.233653 for T = 1100. Similarly, analysis based on the RMSE depicted in Fig. 4b shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.282533, 0.057077, 0.318471, and 0.318471 respectively for the T = 1100. The proposed LW-DBN algorithm achieved the RMSE value of 0.054969 for the T = 1100 and thus, has better performance over the other models.

D. Analysis based on varying the training % (k)

Analysis based on the MSE, as presented in Fig. 4c, shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN metrics achieved the MSE values of 0.54612, 0.289413, 0.547157 and 0.547157 respectively, while the proposed LW-DBN model has the lower MSE value of 0.240637 for k = 75. Similarly, analysis based on the RMSE depicted in Fig. 4d shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.274548, 0.055957, 0.058315, and 0.062019 respectively for the k = 75. The proposed LW-DBN algorithm achieved the RMSE value of 0.052342 for the k = 75 thus have improved performance over the other models.

4.2.2 Comparative analysis of models with the crossfold validation

A. Analysis based on varying the step ratio (sr)

Here, the comparative analysis is done by varying the step ratio of the DBN network and training the database with the crossfold validation. Figure 5a depicts the comparative analysis of the models based on MSE metric for varying sr values. For sr = 0.3 and cf = 6, the existing models DBN, L-DBN, W-DBN, and the GA-DBN achieved the MSE values of 0.337239, 0.238128, 0.559461, 0.536805, respectively, while the proposed LW-DBN algorithm achieved the lowest MSE value of 0.234044. Similarly, for the analysis based on RMSE as shown in Fig. 5b, the proposed LW-DBN model has achieved the lower RMSE value of 0.054907 for the sr value of 0.3 than the existing methods.

B. Analysis based on varying the dropout ratio (dr)

The analysis based on the MSE, as shown in Fig. 5(c), depicts that the existing DBN, L-DBN, W-DBN, and the

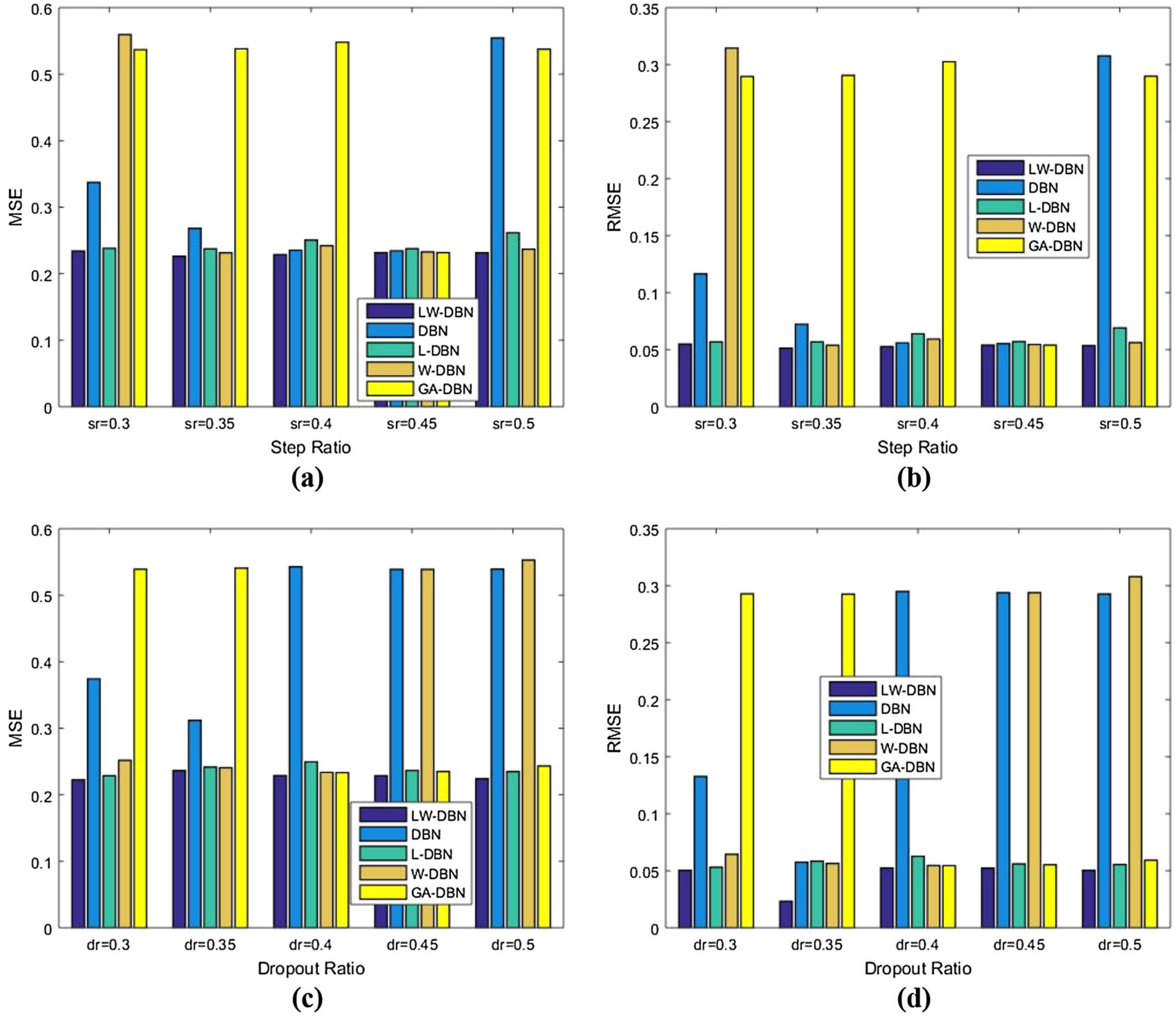


Fig. 3 Comparative analysis of models without the crossfold validation based on **a** MSE by varying sr, **b** RMSE by varying sr, **c** MSE by varying dr, **d** RMSE by varying dr

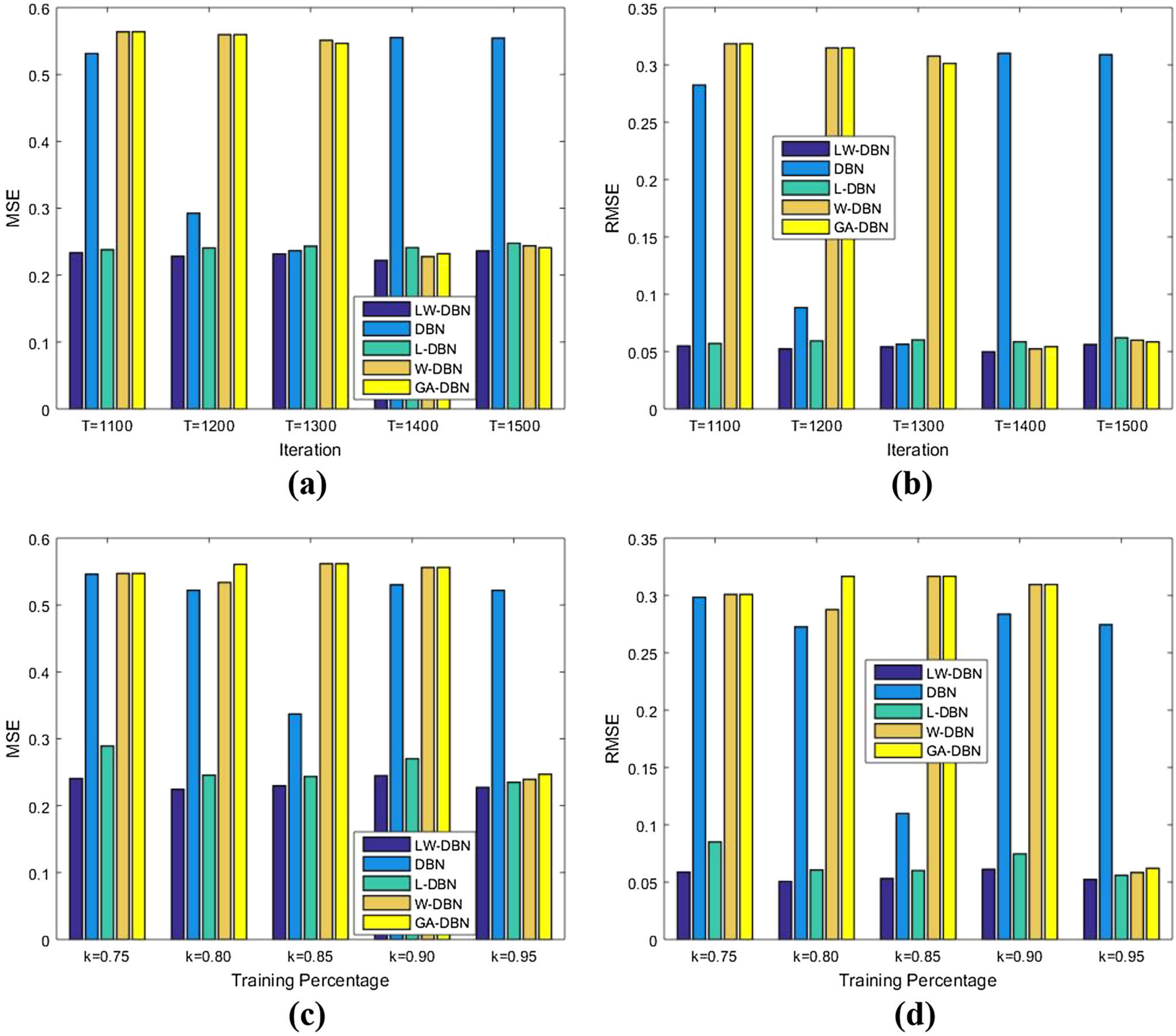
GA-DBN metrics achieved the MSE values of 0.502074, 0.240744, 0.401396, and 0.237185, respectively, while the proposed LW-DBN model has the lower MSE value of 0.236999 for dr = 0.3 and cf = 6. Likewise, analysis based on the RMSE depicted in Fig. 5d shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.254469, 0.058816, 0.185429, and 0.056942, respectively, for the dr = 0.3 and cf = 6. The proposed LW-DBN algorithm achieved the RMSE value of 0.056731 for the dr = 0.3 and cf = 6 thus outperforms the other models.

C. Analysis based on varying the iteration (T)

Figure 6a, b presents the comparative analysis of the models based on the varying values of iteration (T) with crossfold validation. Analysis based on the MSE, as shown in Fig. 6a, depicts that the existing DBN, L-DBN, W-DBN, and the GA-DBN metrics achieved the MSE values of 0.404957, 0.233036, 0.561111, and 0.382309, respectively, while the proposed LW-DBN model has the lower MSE value of 0.229603 for T = 1100 and cf = 6. Similarly, analysis based on the RMSE, depicted in Fig. 6b, shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.172144, 0.054514, 0.316843, and 0.171357 respectively for the T = 1100. The proposed LW-DBN algorithm achieved the RMSE value of 0.053109 for the T = 1100 and cf = 6.

D. Analysis based on varying the training % (k)

The comparative analysis based on the MSE, as given in Fig. 6c, shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN metrics achieved the MSE values of 0.542483,



(a) **(b)** **(c)** **(d)**

Fig. 4 Comparative analysis of models without the crossfold validation based on **a** MSE by varying iteration (T), **b** RMSE by varying iteration (T), **c** MSE by varying training % (k), **d** RMSE by varying training % (k)

0.235306, 0.385438, and 0.400784, respectively, while the proposed LW-DBN model has the better MSE value of 0.234182 for k = 75 and cf = 6. Similarly, analysis based on the RMSE depicted in Fig. 6d shows that the existing DBN, L-DBN, W-DBN, and the GA-DBN model has the RMSE value of 0.298485, 0.085124, 0.300974, and 0.300974 respectively for the k = 75 and cf = 6. The proposed LW-DBN algorithm achieved the RMSE value of 0.058772 for the k = 75 thus have improved performance over the other models.

4.3 Discussion

This section presents the comparative discussion of the models based on the crossfold validation, and without the crossfold validation and it is shown in Table 1. The bold values in Table 1 indicate the best performance. As depicted in Table 1, while not training the database without crossfold validation, the existing DBN, L-DBN, W-DBN, and the GA-DBN achieved the MSE values of 0.374351, 0.22853, 0.25173, and 0.53908, respectively. Training the database with various values of crossfold validation makes the existing works DBN, L-DBN, W-DBN, and the GA-DBN to achieve the MSE value of 0.404957, 0.23303, 0.56111, and 0.38230, respectively. Similarly, the DBN, L-DBN, W-DBN, and the GA-DBN models have attained the RMSE values of 0.13274, 0.05313, 0.06449, and 0.29289, respectively, for the database without crossfold validation, and the values of 0.175203, 0.05652, 0.18272, and 0.05442, respectively, for the database

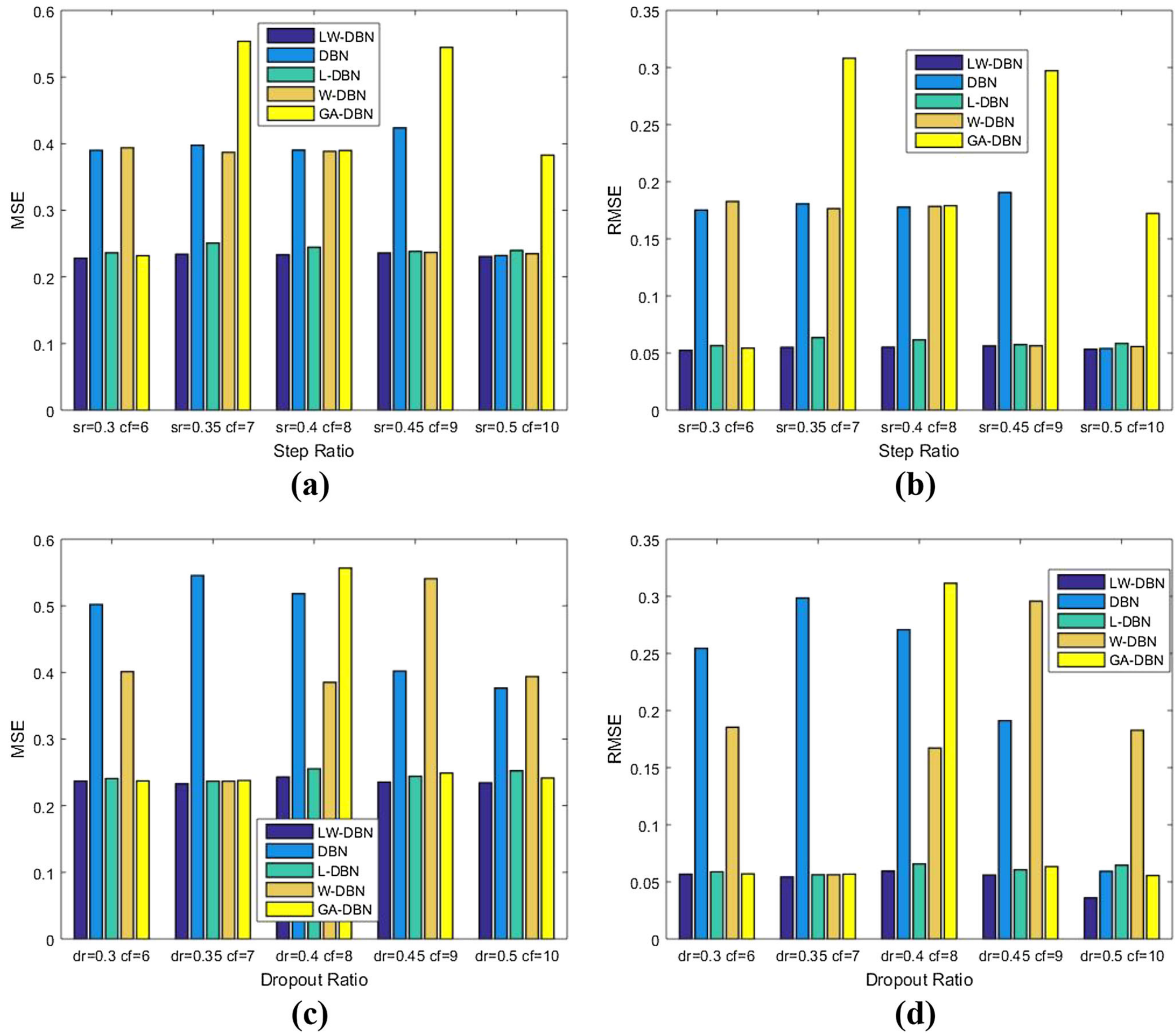


Fig. 5 Comparative analysis of models with the crossfold validation based on **a** MSE by varying sr, **b** RMSE by varying sr, **c** MSE by varying dr, **d** RMSE by varying dr

with crossfold validation. The proposed LW-DBN model has achieved lower values of MSE and RMSE with 0.222606, and 0.050435, for the database analyzed without the crossfold validation. Also, training the database with the crossfold validation makes the proposed LW-DBN more suitable for the prediction as it achieves minimum MSE and RMSE value of 0.229602 and 0.052406, respectively.

5 Conclusion

This paper primarily contributes towards the EDM by introducing the cluster based distributed architecture for predicting the student's performance for the forthcoming academics. The proposed cluster-based distributed architecture predicts the student's performance by collecting the academic performance of the students in previous records. The proposed architecture has two merger layer, and two distributive layers for performing various operations, such as clustering, feature extraction, and the prediction. The architecture uses the BFC for clustering the database, and the KPCA model for the feature selection from the selected clusters. The features are provided to the proposed LW-DBN algorithm for the training purpose, and the LW algorithm chooses the optimal weights for the prediction. The experimentation of the proposed work is done by varying the iteration, training percentage, dropout ratio, and the step out the ratio of the DBN network. The simulation results of the proposed LW-DBN is compared with

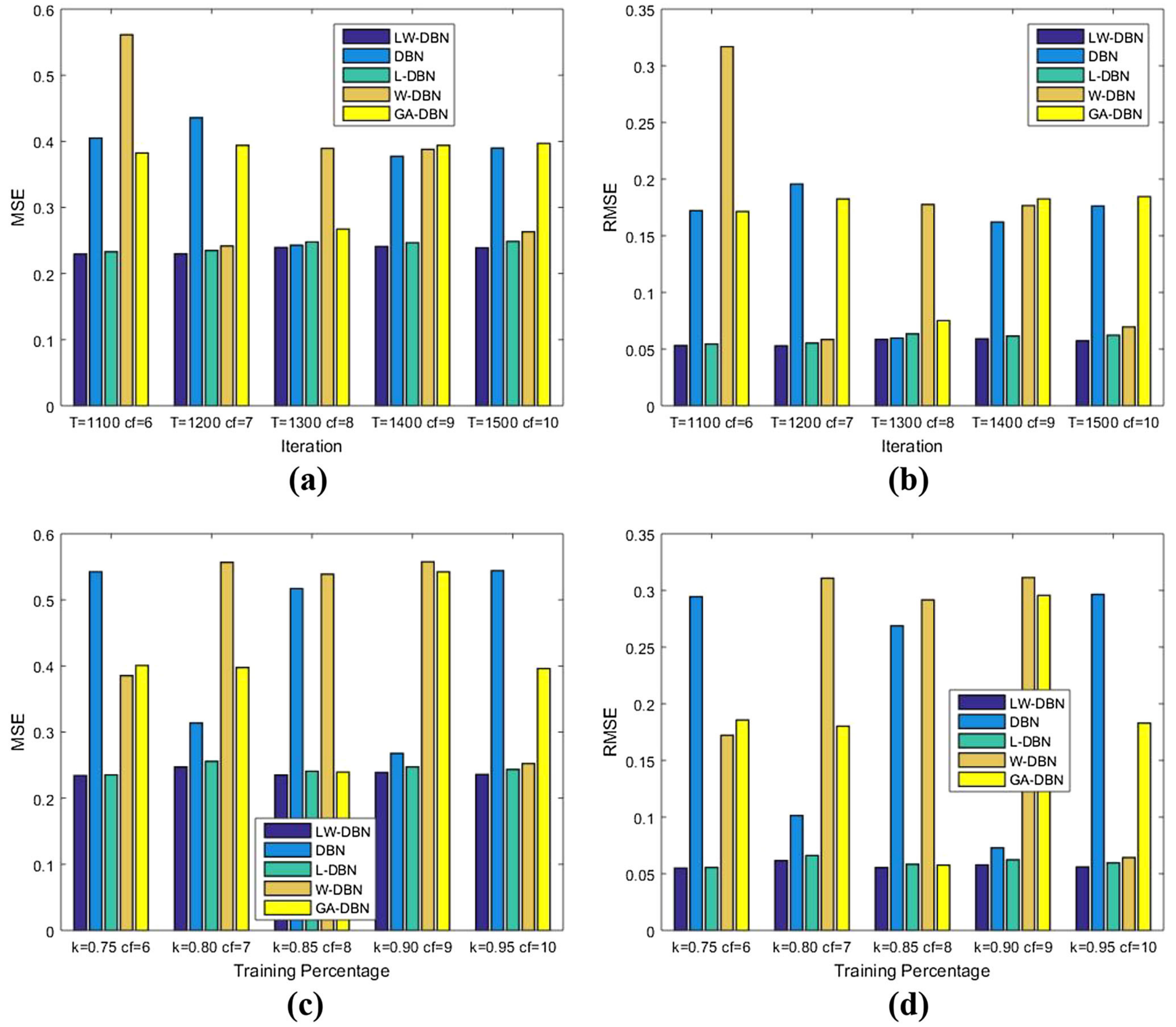


Fig. 6 Comparative analysis of models with the crossfold validation based on **a** MSE by varying T, **b** RMSE by varying T, **c** MSE by varying k, **d** RMSE by varying k

Table 1 Comparative discussion for LW-DBN

Comparative methods	Crossfold validation	Evaluation metrics	
		MSE	RMSE
DBN	Nil	0.374351	0.13274
	cf = 6	0.404957	0.175203
L-DBN	Nil	0.22853	0.05313
	cf = 6	0.23303	0.05652
W-DBN	Nil	0.25173	0.06449
	cf = 6	0.56111	0.18272
GA-DBN	Nil	0.53908	0.29289
	cf = 6	0.38230	0.05442
Proposed LW-DBN	Nil	**0.222606**	**0.050435**
	cf = 6	**0.229602**	**0.052406**

the various existing works and analyzed based on the MSE and the RMSE metrics. The proposed LW-DBN model has achieved lower error performance than other models with MSE and RMSE values of 0.222606 and 0.050435 for the database. The future enhancement of this work is to utilize the advanced techniques for clustering and optimization to further improve the performance.

References

1. Bharara, S., Sabitha, S., Bansal, A.: Application of learning analytics using clustering data mining for students' disposition analysis. Educ. Inf. Technol. **2017**, 1–28 (2017)
2. Wook, M., Yusof, Z.M., Nazri, M.Z.A.: Educational data mining acceptance among undergraduate students. Educ. Inf. Technol. **22**(3), 1195–1216 (2017)
3. Baker, R.S.: Educational data mining: an advance for intelligent systems in education. IEEE Intell. Syst. **29**(3), 78–82 (2014)
4. Hussain, M., Al-Mourad, M., Mathew, S., Hussein, A.: Mining educational data for academic accreditation: aligning assessment with outcomes. Glob. J. Flex. Syst. Manag. **18**(1), 51–60 (2017)
5. Kotsiantis, S.B.: Use of machine learning techniques for educational proposes: a decision support system for forecasting students grades. Artif. Intell. Rev. **37**(4), 331–344 (2012)
6. Barber, R., Sharkey, M.: Course correction: using analytics to predict course success. In: Proceedings of the 2nd International Conference On Learning Analytics and Knowledge, pp. 259–262 (2012)
7. Polyzou, A., Karypis, G.: Grade prediction with course and student specific models. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 89–101 (2016)
8. Almutairi, F.M., Sidiropoulos, N.D., Karypis, G.: Context-Aware recommendation-based learning analytics using tensor and coupled matrix factorization. IEEE J. Sel. Top. Signal Process. **11**(5), 729–741 (2017)
9. Liu, L., Sun, L., Chen, S., Liu, M., Zhong, J.: K-PRSCAN: a clustering method based on PageRank. Neurocomputing **175**, 65–80 (2016)
10. Ferranti, A., Marcelloni, F., Segatori, A., Antonelli, M., Ducange, P.: A distributed approach to multi-objective evolutionary generation of fuzzy rule-based classifiers from big data. Inf. Sci. **415**, 319–340 (2017)
11. Yahya, A.A.: Swarm intelligence-based approach for educational data classification. J. King Saud Univ. Comput. Inf. Sci. (2017). https://doi.org/10.1016/j.jksuci.2017.08.002
12. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
13. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Mach. Learn. **29**(2), 103–130 (1997)
14. Nürnberger, A., Pedrycz, W., Kruse, R.: Data mining tasks and methods: Classification: neural network approaches. In: Handbook of Data Mining and Knowledge Discovery, New York, pp. 304–317 (2002)
15. Costa, E.B., Fonseca, B., Santana, M.A., de Araújo, F.F., Rego, J.: Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. Comput. Hum. Behav. **73**, 247–256 (2017)
16. Asif, R., Merceron, A., Ali, S.A., Haider, N.G.: Analyzing undergraduate students' performance using educational data mining. Comput. Educ. **113**, 177–194 (2017)
17. Kuppusamy, V., Paramasivam, I.: Integrating WLI fuzzy clustering with grey neural network for missing data imputation. International. J. Intell. Enterp. **4**(1–2), 103–127 (2017)
18. Romero, C., López, M.I., Luna, J.M., Ventura, S.: Predicting student's final performance from participation in online discussion forums. Comput. Educ. **68**, 458–472 (2013)
19. Wolff, A., Zdrahal, Z., Herrmannova, D., Knoth, P.: Predicting student performance from combined data sources. In: Alejandro, P.-A. (ed.) Educational data mining, pp. 175–202. Springer, Cham (2014)
20. Guarín, C.E.L., Guzman, E.L., González, F.A.: A model to predict low academic performance at a specific enrolment using DATA mining. IEEE J. Learn. Technol. **10**(3), 119–125 (2015)
21. Chen, D., Chen, Y., Brownlow, B.N., Kanjamala, P.P., Arredondo, C.A.G., Radspinner, B.L., Raveling, M.A.: Real-time or near real-time persisting daily healthcare data into HDFS and ElasticSearch Index inside a big data platform. IEEE Trans. Ind. Inf. **13**(2), 595–606 (2017)
22. Glenn, T.C., Zare, A., Gader, P.D.: Bayesian fuzzy clustering. IEEE Trans. Fuzzy Syst. **23**(5), 1545–1561 (2015)
23. Ebied, H.M.: Feature extraction using PCA and Kernel-PCA for face recognition. In: Proceedings of 8th International Conference on Informatics and Systems (INFOS), Cairo, Egypt, pp. 72–77 (2012)
24. Ramanathan, A.G., Khalid, M., Swarnalatha, P.: Student performance prediction model based on Lion-Wolf neural network. Int. J. Intell. Eng. Syst. **10**(1), 114–123 (2017)
25. Vojt, B.: Deep neural networks and their implementation. Thesis, Charles University in Prague (2016)
26. Yazdani, M., Jolai, F.: Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J. Comput. Des. Eng. **3**(1), 24–36 (2016)
27. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
28. Montana, D.J., Davis, L.: Training feedforward neural networks using genetic algorithms. Proc. IJCAI **89**, 762–767 (1989)

L. Ramanathan has received his B.E. in Computer Science & Engineering from Bharathidasan University, Trichirappali, India and M.E. in Computer Science from Sathyabama, Chennai, India. He is currently an Assistant Professor (Selection Grade) in VIT, Vellore, India. His area of interest is in Data Mining and Database systems, Software Egg, Cloud Computing and Virtualization. He is currently pursuing his Ph.D. in the field of Educational Data mining. His main focus is on Prediction, Classification and Clustering for Educational systems.

G. Parthasarathy was born in 1979. He received the B.Sc. Degree in Computer Science from Madras University in 2000. He Received the M.Sc. Degree in Computer Science from Madras University in 2002. He received the M.Phil. Degree in Computer Science from Alagappa University in 2005. He received the M.E. degree from the Department of Computer Science and Engineering, Sathyabama University, Chennai, India in 2006 and Ph.D. degree in Computer Science and Engineering from Sathyabama University, Chennai, India in 2017. Currently, he is an Associate Professor and HOD (CSE) at Jeppiaar Maamallan Engineering College. He has published ten papers in International level Journals and Conferences. He has published three papers in National Journal. His research interests are in Data Mining, Web Mining, Big Data Analytics, and Information Retrieval.

K. Vijayakumar received his M.Tech. Degree in 2006 from VIT, Vellore, India. He is pursuing Ph.D. in Data Mining & Warehousing. Working as Assistant Professor Selection Grade, in Department of Software and Systems Engineering, at School of Computer Science and Engineering (SCOPE), in Vellore Institute of Technology, Vellore-632014.

L. Lakshmanan having more than 17 years of experience in Teaching (in the field of Engineering and Technology). Currently Working as a Associate Professor in Computer Science and Engineering Department of Satyabhama. Previously Worked as an Asst. Professor in Computer Science and Engineering Department of Oxford Engineering College, Trichy, India. From June-2000 to May-2006 and worked as a System Associate in AARJAY Exports Chennai from June-1998 to March-2000. He has published 35 papers in International level Journals and Conferences. He is a member of CSI and IEEE. He is an Editorial board member/reviewer of International/National Journals and Conferences. Her current research interest includes Image Processing, Remote Sensing, Artificial Intelligence and Software Engineering. His research interests are in Wireless Sensor Networks, Big Data Analytics, IOT & Mobile Application development, Web Mining.

S. Ramani is an Assistant professor (Senior) in the School of Computing Science and Engineering at VIT, Vellore, India. He received his B.E. (Computer Science and Engineering) from MNM Jain Engineering College, Madras University and M.Tech. (Computer Science and Engineering) from Bharathidasan University and currently pursuing his Ph.D. (Computer Science and Engineering) from VIT, Vellore. He has 9 years of experience in teaching and 2 years of experience in Consultancy and Software Industry. He has published about 20 papers in International Journals on Data Mining, Nature Inspired Algorithms and Health Care. His research interest includes Data Mining, Database Systems, Optimization Techniques, Internet and Web Technology. He is a life member of Computer Society of India.