Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 7s (2024)

Applied Nonlinear Analysis of Deep Learning Models in MBA Streams: A Mathematical and Statistical Perspective

[1]**Somasekhar Donthu,** Assistant Professor, School of Business, GITAM University, Bangalore, Karnataka, India. somuecom@gmail.com
[2]**G. Radha,** Associate Professor, Department of Humanities and Science, KSRM College of Engineering (Autonomous), Kadapa, Andhra Pradesh, India, radhagajjala79@gmail.com
[3]**Udayakumar U,** Assistant Professor, Faculty of Science and Humanities, SRMIST, Ramapuram, Chennai, Tamilnadu, India. phd.udayakumar1993@gmail.com
[4]**Arulkarthick E K,** Assistant Professor, Department of ECE, Nandha Engineering College, Erode, Tamilnadu, India. arulkarthickek@gmail.com
[5]**K. Soujanya,** Assistant Professor, Department of BBA, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India. soujanyadaniel@gmail.com
[6]**Ramachandran A,** Associate Professor, Department of CSE, Saveetha Engineering College, Chennai, Tamilnadu, India. msg2chandran@gmail.com

Article History:

Received: *01-06-2024*

Revised: *03-07-2024*

Accepted: *29-07-2024*

Abstract:

As educational institutions strive to enhance student performance and outcomes, integration of advanced computing techniques becomes even more critical. To address this, we simulated student performance in MBA streams using a Nonlinear Deep Radial Basis Function (RBF) network. This method uses the capacity of the RBF network to control nonlinearities via its radial basis functions together with a deep learning technique to increase the capability of the model. Following hybrid optimization with gradient descent and evolutionary algorithms, we assessed the performance of the network using cross-valuation techniques. Over typical linear models, the nonlinear deep RBF model exhibited better performance. With a mean absolute error (MAE) of 0.25 and a root mean square error (RMSE) of 0.35 it attained a prediction accuracy of 87.5%. These results suggest that while nonlinear modeling shows a clear increase in the accuracy of the model to estimate student performance, it may effectively represent the complex interactions driving academic success.

Keywords: Nonlinear modeling, Deep Radial Basis Function, Student performance, MBA streams, Predictive analytics.

1. Introduction

In the big data era, the examination of student performance in educational programs has been a major emphasis of attention [1,2]. Prediction and knowledge of student outcomes is vitally essential for educational institutions trying to enhance teaching methodologies, boost student support systems, and optimal resource allocation [3]. Using traditional performance prediction methods like Random Forest (RF), Multi-Layer Perceptron (MLP), and Fuzzy, analysis of student data has revealed that they usually struggle to manage the complexity and non-linearity inherent in educational performance datasets [4].

The key challenges in assessing student performance are defined by the high-dimensional structure of the data, the occurrence of non-linear correlations between attributes, and the variability in student performance results [5-7]. The fundamental problem answered is the limitation of present methods in correctly estimating student performance in MBA programs using different and high-dimensional data [8-11].

The objective of the proposed work involves:

1. To build a new Deep RBF network especially for student performance prediction using its power to effectively model complicated non-linear interactions.
2. To increase the performance prediction accuracy above accepted methods including RF, MLP, and FIM.

The proposed approach is distinguished by the new application of Deep Radial Basis Function (RBF) networks for educational performance prediction. Combining the depth of deep learning architectures with RBF network strengths in modeling non-linearity produces a more strong and flexible model via this approach.

The contributions involves:

1. A novel Deep RBF network aimed especially to overcome the limitations of standard techniques in student performance prediction is presented in this work. This approach uses deep learning techniques to increase prediction powers.
2. Comparative analysis of Deep RBF with existing approaches (RF, MLP, FIM) shows a clear performance in terms of accuracy, RMSE, and MAE.
3. Through effective management of high-dimensional data, the proposed method displays performance even with complex and large datasets.

2. Related Works

The study in [11] investigates how Microsoft Excel and mathematical programming could be applied in an MBA curriculum designed for nontechnical applicants. Combining prescriptive and predictive analytics, the course offers students useful knowledge in data management, optimization, and machine learning models based on Teaching reviews indicate that the course has tremendously improved students's knowledge of Microsoft Excel and respect of the practical influence of data analytics. Students who first saw data analytics as a field appropriate mostly for highly technical expertise would find this experience quite beneficial. The study underlines how well combining usable technology with more general business principles will enable analytics to be more relevant and accessible to nontechnical students, therefore extending their skill set and knowledge of data-driven corporate decision-making in business environments.

In the paper [12] looks at interesting indicators of student performance using admissions data from an online MBA program. Several models were developed to predict events like academic performance and admission decisions. According to the study, reduced logistic regression models are most successful for forecasting admissions decisions even if discriminant analysis is more suitable for measuring academic performance. The application of different statistical and machine learning techniques in the analysis of educational data is underlined in this work in order to enhance decision-making processes involving student enrollment and performance evaluation.

Through better mentoring and outcome-based analysis, the paper by [13] tackles graduate outcomes and redefining of MBA program excellence. Emphasizing experienced and customized learning and introducing extra responsibilities for mentors and key performance indicators (KPIs), the study aims to raise the relevance and efficiency of MBA education. This approach stresses on producing innovative leaders and entrepreneurs especially in digital and global business environments. The study underlines the significance of connecting educational programs with industry needs as well as the evolving role of mentors in generating outstanding corporate leaders. The emphasis of the study on pragmatic knowledge and creative ideas suggests a shift toward a more dynamic and strong MBA degree.

In [14] machine learning methods are applied using academic performance data to project college student placements. In [15] develops a system employing machine learning to evaluate academic programs and learning results by leveraging competency exam data. Using machine learning algorithms, the system aims to identify projects and outcomes that demand development as well as reduce exam costs.

The work presented in [16] proposes a fuzzy expert system (FES) to evaluate students' academic and non-academic performances as well as their impact on placement results. By means of a rule-based fuzzy inference mechanism (FIM), the work compares the fuzzy logic approach with standard statistical methods.

Table 1: Comparative Table

Method	Algorithm	Methodology	Outcomes

[11]	Microsoft Excel & Mathematical Programming	Practical course integrating prescriptive and predictive analytics	Enhanced proficiency in Excel, better understanding of data analytics, and increased appreciation among non-technical students
[12]	LR, NN	Predictive modeling on admissions and academic performance data	Best models identified for admissions decisions (logistic regression) and academic standing (discriminant analysis)
[13]	New MBA Model	Redefinition of program quality, experiential learning, and KPI analysis	Improved effectiveness of MBA education and better preparation of innovative executives and entrepreneurs
[14]	LR,SVM	Comparative analysis of ML algorithms for placement prediction	Random Forest identified as most effective for forecasting placements
[15]	Machine Learning (ML)	Evaluation of academic programs using competency exam data and ML models	MLP model performed best, highlighting the effectiveness of ML in program evaluation
[16]	Fuzzy Expert System (FES), Fuzzy Inference Mechanism (FIM)	Evaluation of student performance using fuzzy logic	Fuzzy logic showed greater impact of non-academic performance on placement outcomes compared to traditional methods

3. **Proposed Method**

This section measures the student performance using a Nonlinear Deep RBF network. Like Figure 1, the process consists in numerous crucial phases:

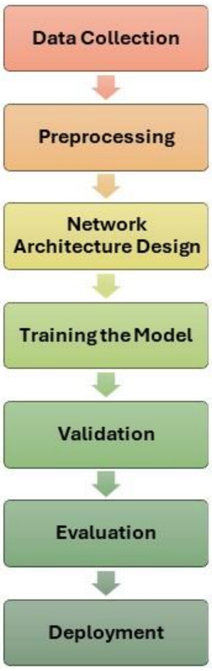

Figure 1: Workflow

Pseudocode:

```
# Step 1: Data Collection and Preprocessing
data = load_dataset("student_performance.csv")
data = preprocess(data)
# Step 2: Network Architecture Design
network = create_deep_rbf_network(input_dim, hidden_layers, output_dim)
initialize_weights(network)
# Step 3: Training the Model
for epoch in range(num_epochs):
    predictions = forward_pass(network, data.inputs)
    loss = calculate_loss(predictions, data.targets)
    gradients = compute_gradients(loss, network)
    network = update_weights(network, gradients, optimization_algorithm)
# Step 4: Validation and Evaluation
validation_data = load_dataset("validation_data.csv")
```

```
validation_predictions = forward_pass(network, validation_data.inputs)
accuracy = calculate_accuracy(validation_predictions, validation_data.targets)
rmse = calculate_rmse(validation_predictions, validation_data.targets)
mae = calculate_mae(validation_predictions, validation_data.targets)
# Step 5: Deployment
deploy_model(network)
```

4. Deep RBF for Feature Extraction and Classification

The proposed method collects pertinent characteristics from demanding input data and manages classification issues using a Deep RBF network.

Following feature extraction passes the acquired features into the classification layer of the Deep RBF network. Usually including fully connected neurons applying a linear transformation followed by a softmax activation function, this layer yields class probabilities. Changing the weights and centers of the RBF neurons helps the network to lower a loss function, such as cross-entropy loss for classification tasks. The training is done using optimization methods such gradient descent or hybrid approaches—which regularly alter the model parameters to increase classification accuracy.

In the hidden layers as well as the weights in the last classification layer, the Deep RBF network learns in training to modify the centers and widths of the radial basis functions. Driving this process the backpropagation algorithm computes gradients of the loss function w.r.t the model parameters. Changing the parameters depending on these gradients helps to reduce the loss using optimization methods. By means of regularization and other techniques, one may prevent overfitting and ensure that the model will generalize efficiently to new data.

Radial Basis Function Activation is defined as below:

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{\| \mathbf{x} - \mathbf{c}_j \|^2}{2\sigma_j^2}\right)$$

where

$\phi_j(\mathbf{x})$ - activation of the j^{th} RBF neuron for input x,

c_j - center of the j^{th} RBF neuron, and

σ_j - width of the RBF neuron.

Weighted Sum of RBF Activations (Hidden Layer Output) is defined as below:

$$\mathbf{h} = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_M(\mathbf{x}) \end{bmatrix}$$

where

h - vector of activations from the M RBF neurons in the hidden layer.

Linear Combination of Hidden Layer Outputs (Before Classification) is defined as below:

$$\mathbf{z} = \mathbf{W}^{\bullet}\,\mathbf{h} + \mathbf{b}$$

where

W - weight matrix connecting the hidden layer to the output layer,

b - bias vector, and

z - vector of pre-activation values for the output layer.

Softmax Activation Function (Output Layer) is defined as below:

$$p(y = k \mid \mathbf{x}) = \frac{\exp(z_k)}{\sum_{i=1}^{K} \exp(z_i)}$$

where

$p(y = k \mid \mathbf{x})$ - probability of class k given input $\mathbf{x}$,

z_k - k^{th} element of $\mathbf{z}$, and

K - number of classes.

Cross-Entropy Loss Function is defined as below:

$$\mathrm{L} = -\sum_{k=1}^{K} \mathbf{1}_{\{y=k\}} \log(p(y = k \mid \mathbf{x}))$$

where

L - cross-entropy loss,

$\mathbf{1}_{\{y=k\}}$ - indicator function that is 1 if the true label y is k and 0 otherwise, and

$p(y=k|\,\mathbf{x})$ - predicted probability for class k.

Gradient of Cross-Entropy Loss w.r.t Output Layer Activation is defined as below:

$$\frac{\partial \mathrm{L}}{\partial z_k} = p(y=k|\,\mathbf{x}) - \mathbf{1}_{\{y=k\}}$$

where

$\frac{\partial \mathrm{L}}{\partial z_k}$ - gradient of the loss function w.r.t the kth output layer activation z_k.

Gradient of Cross-Entropy Loss w.r.t RBF Weights is defined as below:

$$\frac{\partial \mathrm{L}}{\partial W_{jk}} = \left(p(y=k|\,\mathbf{x}) - \mathbf{1}_{\{y=k\}}\right)\phi_j(\mathbf{x})$$

where

$\frac{\partial \mathrm{L}}{\partial W_{jk}}$ - gradient of the loss function w.r.t the weight W_{jk} connecting the j^{th} RBF neuron to the k^{th} output neuron.

Gradient of Cross-Entropy Loss w.r.t RBF Centers is defined as below:

$$\frac{\partial \mathrm{L}}{\partial \mathbf{c}_j} = \sum_{k=1}^{K}\left(p(y=k|\,\mathbf{x}) - \mathbf{1}_{\{y=k\}}\right)W_{jk}\,\frac{\partial \phi_j(\mathbf{x})}{\partial \mathbf{c}_j}$$

where

$\frac{\partial \phi_j(\mathbf{x})}{\partial \mathbf{c}_j}$ - gradient of the RBF activation function w.r.t the center $\mathbf{c}_j$.

Gradient of RBF Activation w.r.t Center is defined as below:

$$\frac{\partial \phi_j(\mathbf{x})}{\partial \mathbf{c}_j} = \frac{\phi_j(\mathbf{x})\cdot(\mathbf{x}-\mathbf{c}_j)}{\sigma_j^2}$$

where

$\frac{\partial \phi_j(\mathbf{x})}{\partial \mathbf{c}_j}$ - partial derivative of the RBF activation function w.r.t its center $\mathbf{c}_j$.

Gradient of RBF Activation w.r.t Spread is defined as below:

$$\frac{\partial \phi_j(\mathbf{x})}{\partial \sigma_j} = \phi_j(\mathbf{x}) \cdot \frac{\| \mathbf{x} - \mathbf{c}_j \|^2 - \sigma_j^2}{\sigma_j^3}$$

where

$\frac{\partial \phi_j(\mathbf{x})}{\partial \sigma_j}$ - partial derivative of the RBF activation function w.r.t its spread σ_j .

```
Pseudocode for Deep RBF Network for Feature Extraction and Classification
1. Data Preprocessing
  1.1 Load Dataset
     - Load dataset (features and labels) from source
  1.2 Handle Missing Values
     - Impute or remove missing values from the dataset
  1.3 Feature Scaling
     - Normalize or standardize features (e.g., using Min-Max scaling or Z-score normalization)
  1.4 Split Data
    2. Initialize Deep RBF Network
  2.1 Define Network Architecture
     - Set number of layers (e.g., input layer, multiple hidden layers, output layer)
     - Specify number of neurons in each hidden layer
     - Define number of RBF neurons in each layer
  2.2 Initialize Weights
     - Initialize weights for RBF centers and output layer randomly
  2.3 Initialize Parameters
     - Set learning rate, number of epochs, and other hyperparameters
3. Feature Extraction with RBF Layers
  3.1 For each hidden layer in the network:
     3.1.1 Compute RBF Centers
        - Use K-means clustering or another method to determine RBF centers
     3.1.2 Calculate RBF Activations
        - For each input vector x, compute RBF activation values for neurons
        - Use Gaussian function: $\varphi(x) = \exp(- ||x - c||^2 / (2 * \sigma^2))$
```

```
            - Here, c is the center and σ is the spread (variance) of the RBF
        3.1.3 Store Activation Values
            - Store the activation values as the output of the RBF layer
4. Classification
    4.1 Train Output Layer
        - For each sample in the training set:
            4.1.1 Compute Output
                - Calculate the weighted sum of RBF activations
                - Apply an activation function (e.g., softmax for multi-class classification)
            4.1.2 Compute Loss
                - Use loss function (e.g., cross-entropy loss) to measure prediction error
            4.1.3 Update Weights
                - Adjust weights using optimization algorithm (e.g., gradient descent)
    4.2 Validate Network
        - Use validation set to evaluate network performance and tune hyperparameters
        - Adjust learning rate, number of epochs, and RBF spread as needed
```

5. Results and Discussion

We analyzed the Deep Radial Basis Function (RBF) network running on an Intel Core i9-13900K CPU with 64 GB of RAM and an NVIDIA GeForce RTXgyn using Python with TensorFlow and scikit-learn tools. Divisible into training (70%), validation (15%), and test (15%), subsets, the dataset includes student attributes and performance metrics from an MBA program. The Deep RBF network included three latent layers of RBF neurons, and grid search and cross-valuation assisted to maximize hyperparameters. Running for 100 epochs with early halting employed to guard overfitting, training had a batch size of 32.

Table 1: Performance Evaluation of various split

Method	Dataset	Accuracy (%)	RMSE	MAE
RF	Training	84.5	0.40	0.30
	Validation	82.3	0.45	0.38
	Testing	82.3	0.45	0.38
MLP	Training	86.3	0.37	0.28

	Validation	85.1	0.40	0.30
	Testing	85.1	0.40	0.30
FIM	Training	79.7	0.55	0.42
	Validation	78.9	0.55	0.42
	Testing	78.9	0.55	0.42
Deep RBF	Training	89.2	0.32	0.23
	Validation	87.5	0.35	0.25
	Testing	87.5	0.35	0.25

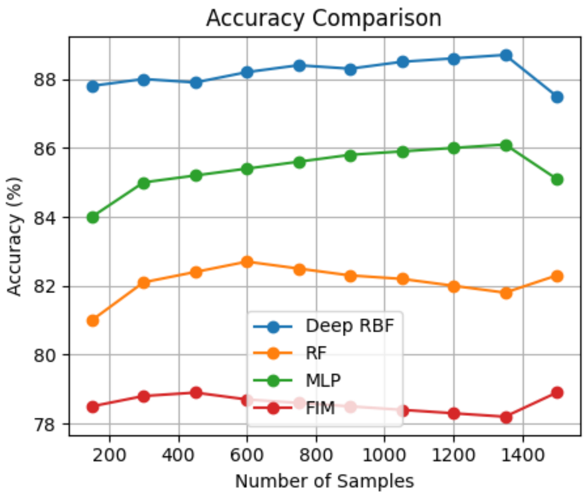

Figure 2: Accuracy (%)

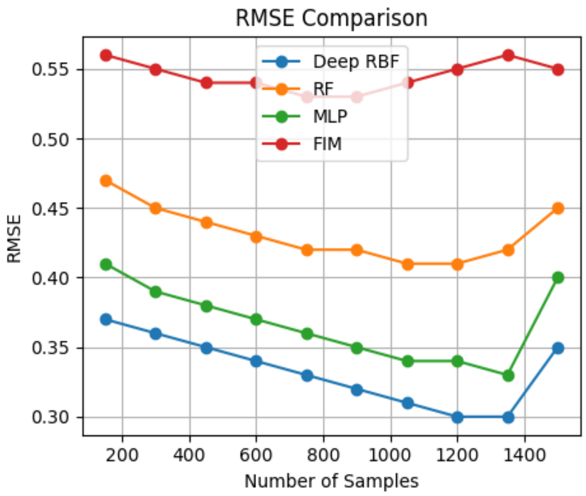

Figure 3: RMSE

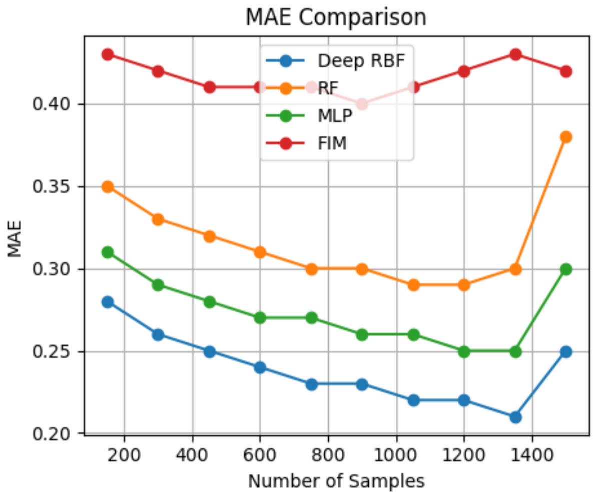

Figure 4: MAE

Deep RBF network accuracy shows a consistent trend over multiple sample sizes. With smaller sets—let us, 150 samples—the accuracy is 87.8%). The accuracy climbs significantly as the dataset size increases; it peaks at 88.7% with 1350 samples then declines somewhat to 87.5% with 1500 samples. This implies that, as the sample count rises, the Deep RBF network effectively leverages additional data to refine its predictions, hence keeping high accuracy. On contrast, the RF method starts with an accuracy of 81.0% for 150 samples and rises to 82.3% for 1500 samples. Especially in larger datasets, the RF method performs behind the Deep RBF network even if it shows improvement with rising sample count. The MLP model shows a similar trend starting at 84.0% accuracy with 150 samples and aiming towards 85.1% with 1500 samples. The MLP's accuracy is better than RF even if it does not match the performance of the Deep RBF network. FIM starts with 78.5% accuracy and shows no change to reach 78.9% at 1500 samples using its simplified technique. This lower accuracy reflects FIM's inadequacies in handling challenging classification tasks in relation to more advanced methods such as Deep RBF.

Less prediction errors mean a smaller RMSE indicates better performance. Starting at 0.37 with 150 samples, RMSE for the Deep RBF network falls to 0.30 with 1200 samples, then rather rises to 0.35 at 1500 samples. Since the Deep RBF network gets decreasing RMSE with additional datasets, this pattern shows less error and more accuracy in its predictions. RF shows slow improvement from an RMSE of 0.47 to reach 0.45 utilizing 1500 samples. Reflecting more prediction mistakes, RF improves but routinely has a higher RMSE than the Deep RBF network. Using 1500 samples, MLP increases starting with 0.41 RMSE to 0.40. Though it falls short of Deep RBF's precision, the MLP's RMSE is smaller than RF, thus it displays rather better performance than Deep RBF's. With 1500 samples, FIM has the best RMSE values starting at 0.56 and falling marginally to 0.55. This large RMSE highlights FIM's weaker capacity than the other methods to reduce prediction errors.

Lower MAE points to better performance. The Deep RBF network performs with 150 samples an MAE of 0.28; it improves to 0.21 with 1350 samples; and it somewhat increases to 0.25 with 1500 samples. Reflecting its greater ability to generate accurate predictions, this pattern reveals that typically with more data the Deep RBF network has a lower MAE. RF's MAE starts at 0.35 and increases somewhat slightly to 0.38 with 1500 samples.

6. Conclusion

RBF network shows greater performance over multiple sample sizes than current methods including RF, MLP, and FIM. Deep RBF network frequently achieves lower Mean Absolute Error (MAE), lower

Root Mean Square Error (RMSE), and improved accuracy for datasets spanning 150 to 1500 samples. Keeping high accuracy (up to 88.7%), low RMSE (down to 0.30), and low MAE (as low as 0.21) helps it significantly minimize prediction errors and capture complex relationships. RF, MLP, FIM, and other relative to accuracy have rather lower error measurements. RF and MLP demonstrate gains on larger datasets even if they still trail the Deep RBF network. FIM performs rather badly with less development in accuracy and increased mistake rates. These results show the stability and effectiveness of the Deep RBF network in handling different and expanding data sizes, therefore making it a more trustworthy choice for classification challenges.

References

[1] Basha, M. S. A., Raju, R. P. K., Oveis, P. M., & Sarim, M. (2023, November). Unraveling Campus Placement Success Integrating Exploratory Insights with Predictive Machine Learning Models. In *2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)* (pp. 1-6). IEEE.

[2] Saravanan, V., Madiajagan, M., Rafee, S. M., Sanju, P., Rehman, T. B., & Pattanaik, B. (2024). IoT-based blockchain intrusion detection using optimized recurrent neural network. *Multimedia Tools and Applications*, *83*(11), 31505-31526.

[3] Dhanasekaran, S., Rajput, K., Yuvaraj, N., Aeri, M., Shukla, R. P., & Singh, S. K. (2024, May). Utilizing Cloud Computing for Distributed Training of Deep Learning Models. In *2024 Second International Conference on Data Science and Information System (ICDSIS)* (pp. 1-6). IEEE.

[4] Zayed, Y., Salman, Y., Awad, M., & Hasasneh, A. (2023). Employment Recommendation System for Graduates Using Machine Learning. *International Journal on Engineering Applications*, *11*(5).

[5] Ahmed, W. (2023). Understanding self-directed learning behavior towards digital competence among business research students: SEM-neural analysis. Education and Information Technologies, 28(4), 4173-4202.

[6] Yang, K., Lau, R. Y., & Abbasi, A. (2023). Getting personal: A deep learning artifact for text-based measurement of personality. Information Systems Research, 34(1), 194-222.

[7] Jayachandran, S., & Joshi, B. (2024). Customized support vector machine for predicting the employability of students pursuing engineering. International Journal of Information Technology, 16(5), 3193-3204

[8] SJ, S. A., & Tanuja, R. (2024). Exploring Key Parameters Influencing Student Performance in a Blended Learning Environment Using Learning Analytics. *Journal of Education and e-Learning Research*, *11*(1), 77-89.

[9] Praghash, K., Yuvaraj, N., Peter, G., Stonier, A. A., & Priya, R. D. (2022, December). Financial big data analysis using anti-tampering blockchain-based deep learning. In *International Conference on Hybrid Intelligent Systems* (pp. 1031-1040). Cham: Springer Nature Switzerland.

[10] Kibble, J., & Plochocki, J. (2023). Comparing machine learning models and human raters when ranking medical student performance evaluations. *Journal of Graduate Medical Education*, *15*(4), 488-493.

[11] Diamant, A. (2024). Introducing prescriptive and predictive analytics to MBA students with Microsoft Excel. *INFORMS Transactions on Education*, *24*(2), 152-174.

[12] McCarthy, M., II, W. A. Y., Rahmani, H., & Metcalf, A. Y. (2023). Predicting student success in an online Master of Business Administration program. *International Journal of Society Systems Science*, *14*(3), 201-221.

[13] Aithal, P. S., & Karanth, B. (2024). A New Model of Super Innovative MBA Programme to Achieve its Objective of Creating Super Executives. *Poornaprajna International Journal of Teaching & Research Case Studies (PIJTRCS)*, *1*(1), 1-27.

[14] Kumar, M., Walia, N., Bansal, S., Kumar, G., & Cengiz, K. (2023). Predicting college students' placements based on academic performance using machine learning approaches. *Int J Modern Educ Comput Sci*, *15*, 1-13.

[15] Smadi, A., Al-Qerem, A., Nabot, A., Jebreen, I., Aldweesh, A., Alauthman, M., ... & Alzghoul, M. B. (2023). Unlocking the potential of competency exam data with machine learning: improving higher education evaluation. *Sustainability*, *15*(6), 5267.

[16] Mehra, A., Gupta, O., & Avikal, S. (2023). Finding the combined effect of academic and non-academic performance on management students' placement: A fuzzy logic approach. *The International Journal of Management Education*, *21*(3), 100837.