International Transactions on Electrical Engineering and Computer Science
Kavi Priya et.al., Vol: 4, No: 2, pp: 91-102, June 2025

Enhanced Butterfly Optimization and Deep Learning Algorithm for Student Placement Prediction

T. Kavi Priya[1], N. Kumar[1]

Abstract: Campus Placement (CP) is regarded in India as a crucial factor in determining universities or college's ranking and recognition. A university's standing and reputation are greatly influenced by the number of students it places in jobs and the average compensation offered to those students. It would be extremely beneficial to develop Deep Learning (DL)-based algorithms that can assist individuals in getting placement guidance, analyses labor market trends, and help educational institutions evaluate opportunities and expanding fields. Numerous realistic and potential placement criteria, such as the kinds of organizations a junior year student can be put in or the companies that are likely to seek out a student's particular skill sets, can be estimated with the use of a DL model based on Predictive Analysis (PA). The Objectives can be predicted using a variety of characteristics, including projects, technical proficiency, training experiences, and academic performance. Initially, pre-processing is applied on the student placement dataset using K-Means Clustering (KMC) algorithm which handles the Missing Values (MV) and error values efficiently. Then, Enhanced Butterfly optimization algorithm (EBOA) is used to select the best students for placement based on their qualities. It is done by generating the optimal Fitness Values (FV). At last, the DL algorithm Improved Long Short-Term Memory (ILSTM) is used for predicting student placement and the results are superior. It is used to assist students and educational institutions in navigating the multifaceted landscape of placement prediction. Finding pupils with academic potential is assisted by this study. Their prospects of getting a placement are increased because this course allows students to concentrate on and develop their social and technical abilities. From the result, it proved that the suggested ILSTM based DL algorithm gives superior performance by means of greater Accuracy (Acc), Precision (P), Recall (R) and f-measure rather than the existing algorithms.

Keywords: Campus placement (CP), K-Means Clustering (KMC) algorithm, Enhanced Butterfly optimization algorithm (EBOA), Deep learning (DL) algorithm, Improved Long short-term memory (ILSTM).

History
Received: 14-02-2025; Revised: 19-04-2025;
Accepted: 30-05-2025

T. Kavi Priya
kkavipriya09@gmail.com

[1]Department of Computer Science with Cyber Security, Sri Rama Krishna College of Arts and Science, Coimbatore – 641006, India
[2]Department of Computer Science, Dr. N. G. P. Arts and Science College, Coimbatore – 641048, India

https://doi.org/10.62760/iteecs.4.2.2025.139

1. Introduction

As a standard for assessing the quality of educational institutions, CP is essential in determining the career paths of students. The placement procedure helps students make a seamless transition into the working world by addressing the gap among academia and industry. In India, placement outcomes significantly influence the reputation and ranking of universities, with metrics like the number of students placed and the average salary packages being key indicators of institutional success [1]. Traditional placement prediction methods rely on linear models or basic statistical analyses, which often fall short in capturing the complex interplay of factors affecting placement outcomes [2]. These factors include academic performance, technical skills, internships, and extracurricular activities, among others. The dynamic nature of the Job Market (JM) further complicates the prediction process, making it imperative to adopt more sophisticated and adaptive approaches.

The advent of Machine Learning (ML) and Artificial Intelligence (AI) has opened new avenues for PA in education. These technologies enable the analysis of large datasets to uncover hidden patterns and trends that traditional methods cannot identify. By leveraging these tools, educational institutions can better understand the strengths and weaknesses of their students and provide targeted interventions to improve their employability [3 - 4]. Additionally, this approach helps streamline institutional efforts to meet evolving industry demands and enhance collaboration with recruiters [5].

Incorporating optimization techniques such as the Enhanced Butterfly Optimization Algorithm (EBOA) further enhances the predictive power of these systems. By selecting the most promising candidates based on a variety of attributes, these techniques ensure a more focused and effective placement process. This not only benefits students but also helps institutions maintain their competitive edge in an increasingly globalized education sector. The ability to optimize candidate selection translates into a more efficient placement pipeline and a higher success rate in securing quality job opportunities for students. Moreover, the integration of deep learning models allows for a more nuanced understanding of placement dynamics. These models can analyze non-linear relationships between variables and adapt to changes in the JM, offering a level of precision and adaptability that is unmatched by traditional approaches. DL frameworks are particularly effective in handling multi-dimensional data, enabling the identification of intricate patterns that influence placement outcomes. By addressing the limitations of existing methods, this research aims to set a new benchmark in placement prediction systems.

This paper introduces a comprehensive framework that leverages Enhanced (BOA) Butterfly Optimization Algorithm and deep learning Improved LSTM (ILSTM) to address these challenges. By integrating advanced preprocessing techniques with State-Of-The-Art predictive models, the proposed system aims to deliver accurate and actionable insights into placement opportunities. The Enhanced Butterfly Optimization algorithm ensures the selection of optimal candidates, while the deep learning model ILSTM captures intricate patterns in the data to predict placement outcomes effectively. Beyond individual placements, this framework supports broader institutional goals. By providing data-driven insights, it enables educational institutions to evaluate and refine their programs continuously, ensuring alignment with industry requirements. For students, it creates a roadmap for career preparation, highlighting key areas for improvement and development. The framework also serves as a strategic tool for recruiters, helping them identify talent that aligns with organizational needs more efficiently. Through this research, we aim to empower students with personalized placement advice, assist educational institutions in aligning their programs with market demands, and provide recruiters with a reliable tool for identifying top talent. The results demonstrate how DL and optimisation techniques can be combined to revolutionise placement prediction.

The remainder of this study is structured in the following way: Several approaches for the CP approach in the most current relevant work are summarised in Section 2. Details of the suggested approach are presented in Section 3. Section 4 describes the result and discusses the obtained experimental results. Section 5, concludes the contributions of this suggested research work.

2. Literature Survey

The literature review explores various advanced techniques and methodologies employed in predictive analytics and optimization for placement prediction. It highlights significant contributions in clustering, optimization algorithms, classification and DL frameworks, show casing their effectiveness in improving accuracy and Decision Making (DM) processes. By combining clustering and classification methods, a single prediction model (PM) was suggested by Thakar and Mehta [2017] [6]. The pre-processing step uses two level clustering (k-means kernel) with chi-square analysis to automatically select pertinent attributes. The employability of students is then predicted using an ensemble vote classification technique that combines four classifiers: k-star, Random Tree (RT), simple cart, and Random Forest (RF). A generalised solution for predicting students' employability is offered by the suggested framework. Model performance across different classification strategies is readily illustrated by comparative outcomes. Additionally, the classification accuracy is 96.78% and the kappa value is 0.937 when the suggested model is used at the state level.

To solve the teacher placement problem, the original placement dataset will be clustered and then processed through a Genetic Algorithm (GA), and it has been suggested by Sriwindono et al [2022] [7]. The Ordered Crossover (OX) operator and the Partial Shuffle Mutation (PSM) Mutation Operator (MO) are employed in the GA, while the K-Means Clustering (KMC) approach is employed. This study demonstrated that prior to the optimisation process with the GA, KMC produced better outcomes than clustering alone.

Premalatha and Sujatha [2022] introduced a hybrid strategy that combines the advantages of Fuzzy Clustering Means (FCM) and Particle Swarm Optimisation (PSO) [8]. The outcomes demonstrate that compared to previous clustering strategies, the suggested method aids in achieving higher Acc. PSO-FCM, the suggested clustering algorithm, outperforms the current approach in accuracy by 34.4%, 36.45%, and 28.45%.

A SOTA DL-based system that combines Long Short-Term Memory (LSTM), Recurrent (NN) Neural Networks (RNN), and Feedforward NN (FNN) has been suggested by Maragatham et al. [2024] [9]. The objective is to develop an advanced predictive model capable of efficiently analyzing diverse student data sources. By integrating FNNs, RNNs, and LSTMs, the proposed system can interpret complex student information and generate valuable insights for educational institutions. Its accuracy in predicting placement probabilities and mapping out career trajectories is tailored to students' abilities and career goals.

In order to handle global optimisation challenges, Arora and Singh [2019] [10] developed a new Nature-Inspired Algorithm (NIA) called the Butterfly Optimisation Algorithm (BOA), which imitates the food search and mating behaviour of butterflies. The architecture is primarily based on how butterflies forage, using their sense of fragrance to locate nectar or a potential mate. This study compares the performance of the suggested method with that of other Metaheuristic (MH) algorithms after testing and validating it on a set of thirty benchmark test functions.

3. Proposed Methodology

The proposed methodology begins with pre-processing the student placement dataset using the K-Means Clustering (KMC) algorithm, which efficiently handles missing and erroneous values. Next, the Enhanced Butterfly (EB) optimization algorithm is employed to select the best students for placement by generating optimal fitness values based on their qualities. The final step involves using a deep learning algorithm to predict student placement outcomes. This approach aims to provide accurate placement predictions, aiding students and educational institutions in navigating the complex landscape of placement decisions. The general block diagram of the suggested approach is shown in Fig. 1.

3.1 Description

The college placement dataset contains information about students who have graduated from college and have been placed in jobs. The dataset available at https://www.kaggle.com/datasets/durgeshrao9993/college-placement-data-set. The dataset includes information such as the student's name, college, major, GPA, and salary. The dataset can be used to study factors that affect college placement, such as the student's academic performance, college choice, and extracurricular activities. The dataset can also be used to develop models that can predict a student's chances of being placed in a job after graduation. The dataset is a valuable resource for students, college counselors, and employers. Students can use the dataset to learn about the factors that affect college placement and to make informed decisions about their education and career. College counselors can use the dataset to help students develop strategies for college placement. Employers can use the dataset to identify qualified candidates for jobs.

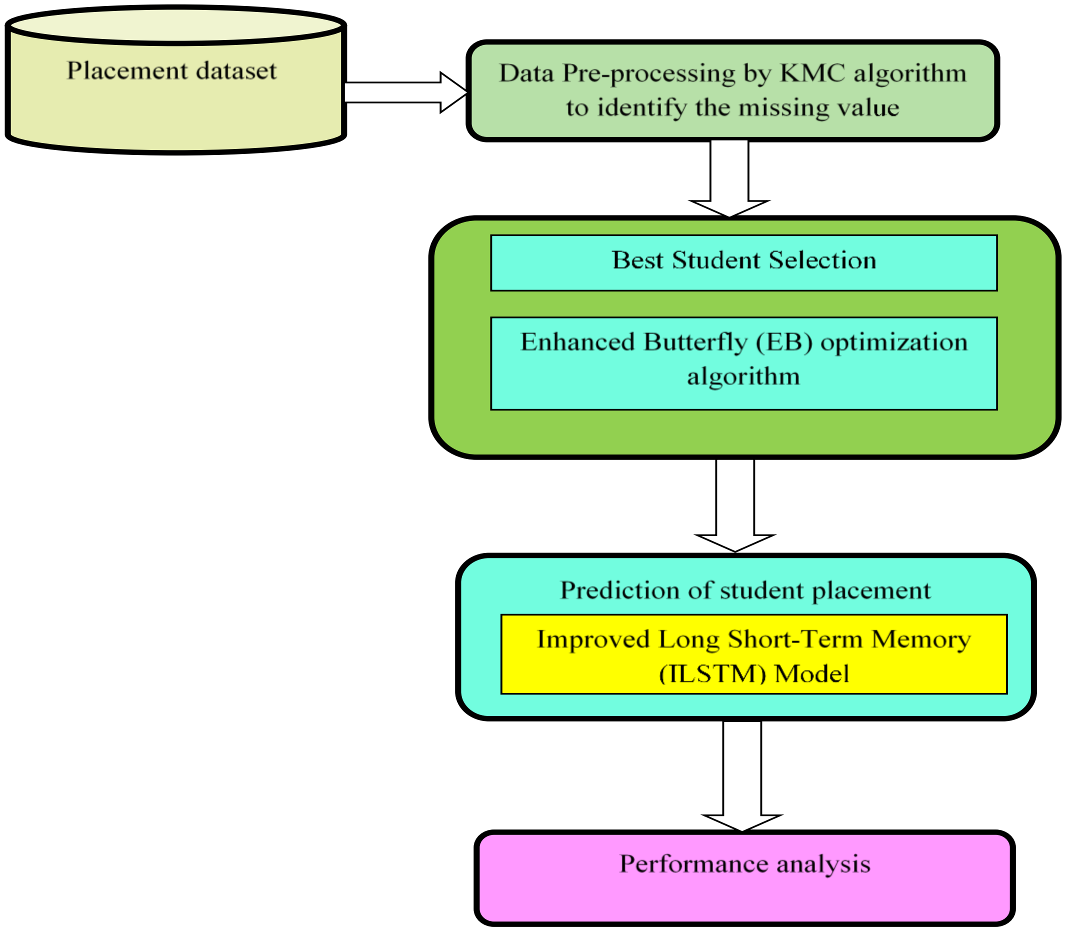

Fig. 1: Overall block diagram of the suggested method

3.2 Preprocessing via KMC procedure

Pre-processing is performed using the KMC algorithm, which is employed to improve the prediction of student placement outcomes in the given dataset. K-means is an effective clustering algorithm for clustering comparable data by using the initial centroids of clusters [11 - 12].

Euclidean Distances (ED) identified cluster centroids. Starting with random partitions, the method iteratively computed (i) current cluster centers (average vectors in data spaces for clusters), and (ii) sent information to clusters closest to them. The iterations stopped when there were no more assignments. The technique reduced local Intra-Cluster (IC) variances, or the sum of the squares of the differences among the cluster centres and the data features. Fig. 2 displays a KMC algorithm. 1, instance [13 - 14].

Operating with a linear runtime and demonstrates remarkable efficiency were the two benefits of K-means implementation. A maximum of one cluster per class is maintained. To get the cluster centroids, use the following formula to calculate the ED (1).

$$d(i,j) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

In this case, two points in Euclidean n-space are x_i and y_i.

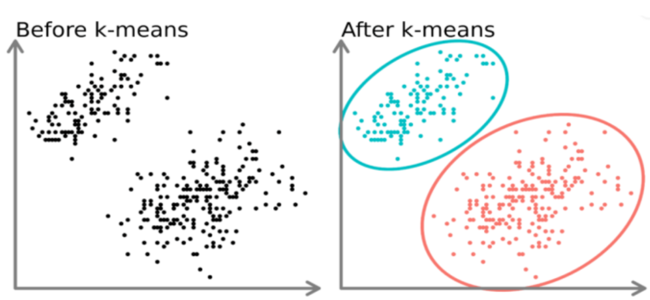

Fig. 2: Example of KMC algorithm

Algorithm 1: KMC algorithm
Input: Placement dataset
Output: Pre-processed data

i. Select **k** clusters from the dataset **D**.
ii. Initialize **k** cluster centers.
iii. Choose **k** initial Data Points (DP) and allocate the corresponding cluster centers to them.
iv. Calculate the mean for each cluster and randomly assign DP to the clusters.
v. Determine the nearest cluster centre for every DP, then use the distance to compute the noise values.
vi. Assign the DP to its corresponding cluster.
vii. Update the cluster centers by recalculating the mean of the DP within each cluster.
viii. Identify and remove any errors values or MV from the dataset.
ix. Terminate the process when no new assignments occur.

The instances from the original dataset with missing attributes were excluded and dataset split into two groups, one with examples that were complete and had no missing values while the other contained instances that were partially complete and had missing values. KMC is used to collect entire instances in order to form clusters of full instances. Each occurrence is therefore looked at independently. Potential values are used to fill up any lacking attributes. Once KMC has been applied to the dataset from the generated clusters, the newly added instance is evaluated to see if it has been clustered in the correct class. Until the value that places the instance in the correct cluster is found, the corresponding value that places the instance in the incorrect cluster will be examined. The allocated value becomes permanent in the appropriate cluster, and the process begins for the next instance. Hence, the usage of KMC as a preprocessing strategy increases accuracy of prediction [15].

3.3 Feature selection by enhanced butterfly optimization algorithm

Once the missing values are computed, it is vital to find the salient features with a powerful and positive association with significant features to diagnose the disease. Extraction of the vector features removes the unimportant features for prediction and those with no relevance is a hurdle in developing a reliable diagnostic model. In this work, Enhanced Butterfly Optimization Algorithm (EBOA) is presented for extracting the most important features of a classification. BOA is one of the innovative MH optimization algorithms that imitates the food foraging and discovery of (choosing the optimum features of Placement dataset) mating partner behaviours of butterflies (*bf*). The *bf* make use of these chemoreceptors to get the optimum mating partner (features from the dataset). Butterflies produce a scent (fitness higher accuracy) having a degree of intensity when it is migrating to different locations. This scent directs the traversal of search-agents (*bf*) in the BOA algorithm. Depending on the intensity of the scent, in case a certain *bf* cannot perceive the scent of any *bf* located within the search space (SS), this *bf* will carry out exploitation ((LS) Local Search) through its movement to a new randomly chosen position (feature position). If the *bf* perceives the scent of the best *bf*, then it will go towards it, and this is referred as exploration ((GS)-Global Search). As per the BOA algorithm, the fragrance is given by a function of the stimulus intensity and can be decided with the help of the equation (2).

As a function of stimulus intensity, the fragrance f is described as follows (2):

$$f = c \cdot Ia \tag{2}$$

Here, I *is* the stimulus intensity, an is the power exponent regulating the degree of fragrance emission, and c is the sensory modality (a constant).

The equation (3) is accountable for global search and expressed using equation, whereas the local search is defined using equation changed as below,

$$x_i^{t+1} = x_i^t + (r^2 * g^* - x_i^t) * f_i * T_w \tag{3}$$

$$x_i^{t+1} = x_i^t + \left(r^2 * x_j^t - x_i^t\right) * f_i * T_w \tag{4}$$

In equation (3, 4), T_w is defined to be the triangular fuzzy weight number that the fuzzy membership function generates. It is treated as adaptive weight value to improve the feature selection from the Placement

A triangular fuzzy number is a fuzzy subset A⊂ R defined using a membership function $\varphi_A: R \rightarrow [0,1]$ expressed using equation (5), as follows

$$\varphi_A(w) = \begin{cases} \frac{w-l}{m-l} if l \leq w \leq m \\ \frac{n-w}{n-m} if m \leq w \leq n \\ 0 \; if else \end{cases} \quad (5)$$

Where l, m, and n indicate the real numbers which is generated via the least, middle and highest values of weight from the random weight values. g^*valuestands for the optimal solution (selected features) at the present iteration, $f_i$indicates the magnitude of fragrance emitted by the i^th^ butterfly, r stands for a random value ranging between [0, 1], x_j^tindicates the j^th^ butterfly, and x_k^tsignifies the k^th^ butterfly of the existing solutions space. In order to improve the outcomes of the BOA method, a novel particle swarm optimisation algorithm based on a MO (PSOMO) is presented in this work. Algorithm. 2, presents this algorithm's pseudocode.

PSO [16 - 17] is worked based on two factors, which is either through the improvement of the current best solution (Selected features) or through the improvement in the quality of the remaining solutions. PSOMO initially gets the current best solution g^* from BOA for every iteration. Next, PSOMO will use the mutation operator ong^*.In case the newly mutated (Features) solution is superior compared to g^*, then g^* will be substituted with it.Else, PSOMO will choose a random solution (features) from rest of the solutions and the same earlier process is repeated. In case the FV (Accuracy) of the mutated (selected) solution is improved compared to the FV of the chosen (features) solution, then the new mutated solution will be substituted for the chosen solution. PSOMO will iterate through these steps Num_iterations times. At last, when it gets an optimal solution (selected features), it will allocate it to g^*. The proposed PSOMO algorithm [18 - 20], as seen in algorithm1, will run through specific number of iterations with the aim of improving the superior solution. During every iteration of PSOMO, it will perform the mutation of the current best solution. Next, PSOMO will verify the FV (Accuracy) of the newly found solution, and in case it is found to be improved compared to the current best solution (features), then the current best solution will be replaced with the new- mutated solution position; else, PSOMO will choose a single solution(new features) arbitrarily from the other existing solutions (features), in case the fitness (Accuracy) pertaining to the newly mutated solution is superior compared to the chosen solution. Then PSOMO will have the randomly chosen solution (features) replaced with the newly mutated solution position. PSOMO will go on with improving the process in accordance with the Num_iterations. This way, PSOMO can be helpful in improving the best solution (selected features) to prevent local optima and make improvement in other solutions (selected features) versatility through the fitter one replacing the worst solution.

Algorithm 2: PSO Based on A Mutation Operator (PSOMO) Algorithm

Input: g^* (current best solution), fitness (), Mutation_Rate, Max_iterations,

Swarm parameters: n (population size), w (inertia weight),

c1 (cognitive parameter), c2 (social parameter)

Output: g^* (optimized best solution)

…………………………………………………………

1. Initialize swarm with n particles (random positions and velocities)

2. For each particle:- Assign initial personal best positions and fitness values

3**. Set Old_fitness = fitness(g^*)**

4. Repeat until Max_iterations:

a. For each particle in the swarm:

i. Compute fitness (Position[i])

ii. If fitness (Position[i]) < fitness (Personal_Best[i]):

Update Personal_Best[i] = Position[i]

iii. If fitness (Position[i]) < Old_fitness:

Update g^* = Position[i]

Old_fitness = fitness (Position[i])

b. Update velocity for each particle:

Velocity[i] = w * Velocity [i] + c1 * rand () * (Personal_Best [i] - Position[i]) + c2 * rand () * (g^* - Position[i])

c. Update position for each particle:

Position[i] = Position[i] + Velocity[i]

Ensure Position [i] is within valid bounds

d. Apply Mutation:

i. Generate Mutated_Position by applying Mutation_Rate to g^*

ii. Compute New_fitness = fitness (Mutated_Position)

iii. If New_fitness<Old_fitness: Update g^* = Mutated_Position

Old_fitness = New_fitness

5. Return g^*

As per the earlier sections, algorithm. 3, demonstrates the full pseudocode of the proposed AWDBOA algorithm. The primary enhancement to DBOA [19-20] involves developing PSOMO algorithm, and weight generation of features that can aid in improving the exploitation of BOA and solutions versatility. In order to improve the current best solution or augment the other population entities, AWDBOA will use the PSOMO algorithm at the end of each DBOA iteration. This is done by randomly selecting the optimal features from among the other feature solutions that have been chosen [21].

Algorithm 3: Adaptive Weight Dynamic BOA (AWDBOA)

1. Initialize the positions of n butterflies population $x_i (i = 1, ... n)$
2. Fix the initial value of the parameters including switch probability p, sensor modality c, the power exponent a, and the maximum number of iterations (max_iter)
3. While(max_iter) do
   - 3.1. For every bf in the population (number of features in the Palacement dataset) do
     Compute the fragrance value f of bf by equation (1)
   - 3.2. End for
   - 3.3. Find the best bf (feature)
   - 3.4. set the best feature (butterfly) to g^*
   - 3.5. For each bf in the population (number of samples and features in the dataset) do
     - 3.5.1. Create a random number r with the interval [0,1]
     - 3.5.2. If (r<p)
       Update the location of bf (features) in the LS
     - 3.5.3. Else
       Update the location of bf (features) in the GS
     - 3.5.4. End if
     - 3.5.5. Evaluate new bf (features)
       If the new butterfly is better, update it in the population
   - 3.6. End for
   - 3.7. Update the value of power exponent C
   - 3.8. Use PSOMO on the current best solution applying MO by algorithm 2
   - 3.9. Update global best solution
4. End while
5. Return the best features got using AWDBOA

The proposed Enhanced Butterfly Optimization Algorithm (EBOA) with the PSO-Based Mutation Operator (PSOMO) effectively identifies the most significant features from the campus placement dataset. By leveraging the GS and LS capabilities of EBOA and enhancing it with PSOMO's mutation-based exploration, the algorithm improves classification accuracy while reducing feature redundancy.

3.4 Classification by ILSTM

For the classification step, the ILSTM model is applied to the preprocessed and feature-selected data to predict placement outcomes. ILSTM ensures high classification accuracy and dependability by efficiently capturing complex patterns and temporal correlations in the dataset. Standard RNNs, the LSTM [22] is an RNN architecture designed to more precisely imitate temporal sequences and their long-term interactions. LSTM cells having input gate, forget gate, and output gates (o_t) as well as an activating cell component are shown in Fig. 3.

Cell activations are controlled by the use of specifically designed multipliers.

The input gates of LSTM are defined as (6)

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (6)$$

The forget gate is defined as (7)

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (7)$$

The cell gate is defined as (8)

$$c_t = f_t c_{t-1} + i_c \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (8)$$

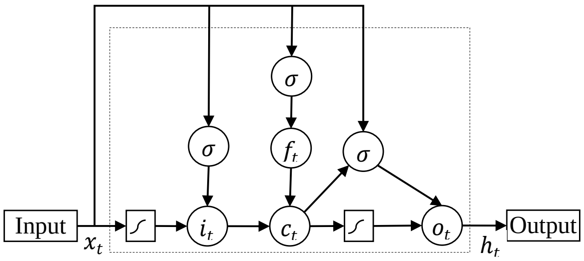

Fig. 3: LSTM cell

The o_t is expressed as (9)

$$o_t = \sigma\left(W_{x0} x_t + W_{h0} h_{t-1} + W_{c0} c_t + b_0\right) \quad (9)$$

Ultimately hidden states are computed using (10)

$$h_t = o_t \tanh(c_t) \quad (10)$$

tanh represents hyperbolic tangent activations, x_t- implies inputs at times t while W and b stand for network's Weights and Biases respectively.

LSTM gates avoid data changes from the rest of networks and store it in in the memory cells with different time steps. LSTM networks propagate errors and hold onto signals far longer than ordinary RNNs [23]. In order to circumvent these problems, the work that is being proposed uses modified long-term memory, which includes a weighted scheme that is based on weighted mean data. According to this method, the relevance of each feature to the estimation point determines how much weight it is given to in the overall scheme. Based on 3D distance metrics between observation and estimated estimate points, the weight corresponding to each characteristic may be established.

The weight function was discovered to have the following form (11)

$$\omega = \begin{cases} (1 - \left(\frac{d}{h}\right)^3)^3 \rightarrow \text{if}|d| \leq h \\ 0 \rightarrow \text{if}|d| > h \end{cases} \quad (11)$$

Where,

ω - indicates weights,

d - denotes separations between measurement and observation points.

h - half window widths.

The σ I, f, o, and c stand for input gate, forget gate, output gate, and cell state, respectively. Logistic (SF) sigmoid function. Weight matrices represent Wci, Wcf, and Wco for peephole connections.

Three gates in the LSTM help with Vanishing Gradient Problems (VGP) and exploding gradient problems. Recurrent Hidden Layers (HL) are swapped with LSTM cells in LSTM-RNN architectures. Placement data categorization is provided via the output gate. The application of the ILSTM model to the campus placement dataset has demonstrated superior performance in predicting placement outcomes with high accuracy and reliability. By leveraging advanced gating mechanisms and a weighted scheme, ILSTM effectively captures temporal dependencies and complex patterns in the dataset, addressing challenges like vanishing gradients. This approach ensures that students' placement predictions are precise, enabling better decision-making for both students and institutions.

4. Experimental results

In MATLAB, the experimental analysis is conducted. Here conducted student placement process on placement dataset. In terms of Acc, P, R, F-measure, and error, the performance of the suggested ILSTM is contrasted with the current LSTM and RF approach scheme. Table. 1, shows the performance comparison.

Table. 1: Performance comparison

Performance metrics	Methods		
	RF	LSTM	ILSTM
Accuracy	84.12	93.75	97.3
Precision	84.16	93.75	97.5
Recall	84.12	93.75	96
F-measure	84.14	93.75	97.8
Error	15.87	6.25	2.7

Performance metrics

4.1 Accuracy

Basically, Acc is the ratio of accurately predicted observations to total observations, making it the most intuitive performance metric (12).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (12)$$

Here, TP stands for "True Positive." False Positive (FP) and False Negative (FN), TN- True Negative.

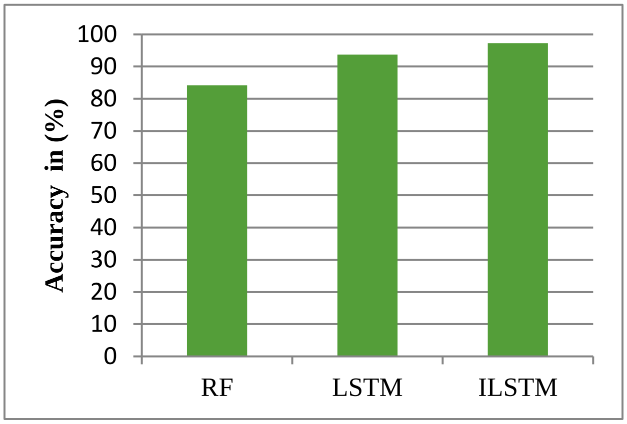

Fig. 4: Accuracy comparison

Fig. 4, compares the accuracy performance of the suggested ILSTM with LSTM with the current RF and LSTM approaches. Accuracy is regarded as the y-axis and methods as the x-axis. Enhanced BOA is used to choose placement data in the suggested research, and ILSTM is used for classification. It raises the rate of accuracy. According to the outcomes, the suggested approach obtains 97.3% accuracy, but alternative approaches, like RF and LSTM, achieve 84.12% and 93.75%, respectively.

4.2 Precision

The ratio of accurately predicted positive observations to all predicted positive observations is known as precision (13)

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

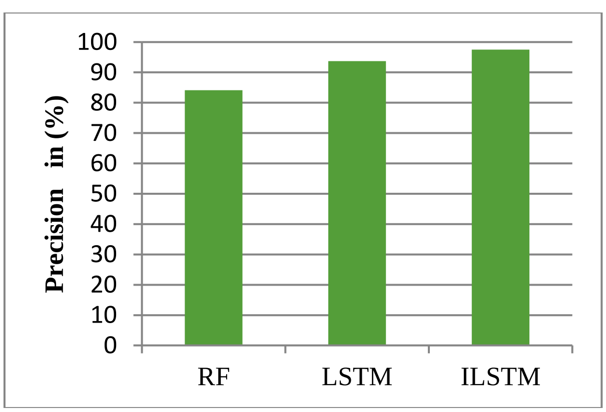

Fig. 5: Precision comparison

The precision of the suggested ILSTM with LSTM scheme is contrasted with that of the current RF and LSTM techniques. Fig. 5 shows the comparison of precision. The experimental results show that the suggested ILSTM with LSTM attains 97.56% of precision but current RF and LSTM method provides 84.16% and 93.75 % respectively.

4.3 Recall

The proportion of accurately predicted positive observations to all observations in the actual class is known as recall (14)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

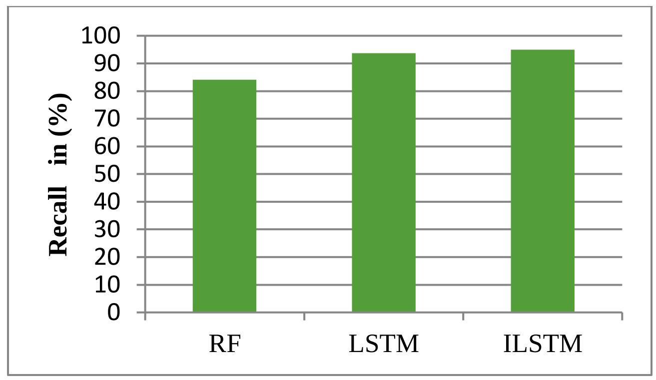

Fig. 6: Recall comparison

Fig. 6 compares the recall performance of the ILSTM with LSTM, LSTM, and RF techniques. In this suggested study, the ILSTM is used for classification. It raises the rate of recall. According to the graph, the suggested method obtains 97% recall, while RF and LSTM achieve 84.12% and 93.75%, respectively.

4.4 F-measure

The weighted average of P and R is known as the F1 Score. Consequently, both FP and FN are considered in this score (15).

$$\text{F-measure} = 2*\frac{(Recall*Precision)}{(Recall+Precision)} \quad (15)$$

The f-measure performance of the ILSTM with LSTM, LSTM, and RF techniques is shown in Fig. 7. Methods are taken in the x-axis, and the f-measure is obtained in the y-axis. The experimental outcomes show that the f-measure of the suggested method is 94.8 % when RF and LSTM method attains 84.14% and 93.75% respectively.

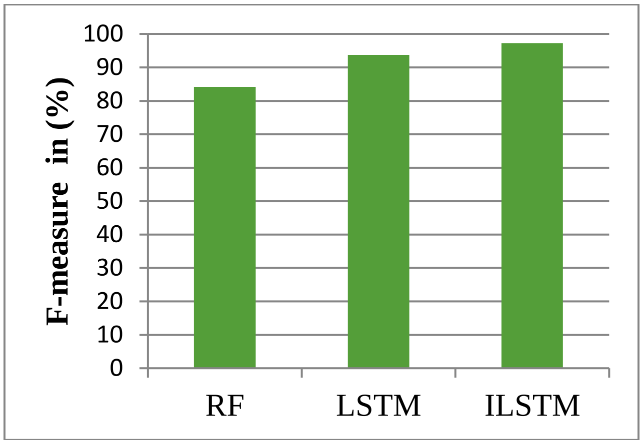

Fig. 7: F - measure comparison

4.5 Error

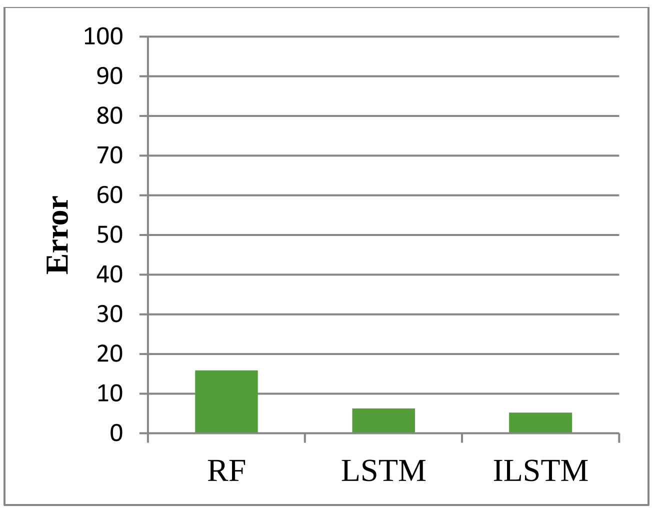

Fig. 8: Error comparison

The performance of the ILSTM with LSTM, LSTM and RF methods are compared in terms of error in Fig. 8. ILSTM is used to classify the dataset for the suggested work placement in order to lower the classifier's error rate. According to the graph, the suggested ILSTM system obtains an error rate of 2.7%, whereas the RF and LSTM methods achieve 15.87% and 6.25%, respectively.

5. Conclusion

The proposed method utilizing an ILSTM network for student placement prediction has demonstrated exceptional performance, achieving an accuracy of 97.3%. This superior accuracy highlights the robustness of the improved LSTM model in capturing complex patterns and dependencies within the student placement dataset. The integration of K-Means Clustering (KMC) for efficient pre-processing, coupled with the Enhanced Butterfly (EB) Optimization algorithm for selecting the best candidates, ensures that the model processes clean and high-quality data. With the use of this foundation, the enhanced LSTM can remarkably accurately predicting placement outcomes by analysing key factors such as academic performance, technical proficiency, training experiences, and projects. In terms of Acc, P, R, F-measure, and error rate, the results show that the improved LSTM model performs better than conventional PM and optimisation techniques. In conclusion, the proposed Improved LSTM based placement prediction framework not only enhances prediction accuracy but also contributes significantly to optimizing the placement process. Better placement prospects and a more effective talent pipeline are made possible by its ability to address the gap among academic preparation and industrial requirements.

Conflict of interest

The authors declared 'No conflict of interest'.

References

[1] Z. Wang, G. Liang, and H. Chen "Tool for predicting college student career decisions: an enhanced support vector machine framework", *Applied Sciences*, Vol. 12, No. 9, art. no. 4776, 2022. https://doi.org/10.3390/app12094776

[2] A. K. Srivastava, V. Tripathi, and B. Pant "Computer Vision based Online Job Interview Proctoring for Campus Placement", 2024 *11th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 663-668, 2024. https://doi.org/10.23919/INDIACom61295.2024.10499075

[3] D. S. Kumar, Z. B. Siri, D. S. Rao, and S. Anusha "Predicting student's campus placement probability using binary logistic regression", *International Journal of Innovative Technology and Exploring Engineering*, Vol. 8, No.9, pp. 2633-2635, 2019. http://dx.doi.org/10.35940/ijitee.I8984.078919

[4] M. S. Byagar, R. Patil, and J. Pawar "Maximizing Campus Placement Through Machine Learning",

Journal of Advanced Zoology, Vol. 45, No. S4, pp. 06-12, 2024. https://doi.org/10.53555/jaz.v45iS4.4141

[5] A. Gupta, S. Gupta, P. K. Mall, S. Srivastava, A. S. Saluja, N. Yadav, V. Narayan, M. K. Verma, and S. Sriramulu "ML-CPC: A Pathway for Machine Learning Based Campus Placement Classification", *Journal of Electrical Systems*, Vol. 20, No. 3s, pp. 1453-1464, 2024. https://doi.org/10.52783/jes.1521

[6] P. Thakar, and A. Mehta, Manisha "A unified model of clustering and classification to improve students' employability prediction", *International Journal of Intelligent Systems and Applications*, Vol. 9, No. 9, pp. 10-18, 2017. http://dx.doi.org/10.5815/ijisa.2017.09.02

[7] H. Sriwindono, P. P. Rosa, and K. Pinaryanto "The Teacher Placement using K-Means Clustering and Genetic Algorithm. *Conference Series*, Vol. 4, pp. 43-51, 2022. http://dx.doi.org/10.34306/conferenceseries.v4i1.669

[8] N. Premalatha, and S. Sujatha "Prediction of students' employability using clustering algorithm: A hybrid approach", *International Journal of Modeling, Simulation, and Scientific Computing*, Vol. 13, No. 06, art. no. 2250049, 2022. http://dx.doi.org/10.1142/S1793962322500490

[9] T. Maragatham, P. Yuvarani, S. Harishri, and R. Swetha. "Student Placement Prediction using Deep Learning Techniques", *2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 620-626, 2024. https://doi.org/10.1109/ICECA63461.2024.10800998

[10] S. Arora, and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization", *Soft computing*, Vol. 23, pp. 715-734, 2019. https://doi.org/10.1007/s00500-018-3102-4

[11] S. S. Yu, S. W. Chu, C. M. Wang, Y. K. Chan, and T. C. Chang "Two improved k-means algorithms", *Applied Soft Computing*, Vol. 68, pp. 747-755, 2018. https://doi.org/10.1016/j.asoc.2017.08.032

[12] K. P. Sinaga, and M. S. Yang "Unsupervised K-means clustering algorithm", *IEEE Access*, Vol. 8, pp. 80716-80727, 2020. https://doi.org/10.1109/ACCESS.2020.2988796

[13] M. Cui, "Introduction to the k-means clustering algorithm based on the elbow method", *Accounting, Auditing and Finance*, Vol. 1, No. 1, pp. 5-8, 2020.

[14] H. Amin "Clustering Analysis of Admission of New Students Using K-Means Clustering and K-Medoids Algorithms to Increase Campus Marketing Potential", *Tech-E*, Vol. 7, No. 1, pp. 43-56, 2023. http://dx.doi.org/10.31253/te.v7i1.2264

[15] L. Kesheng, N. Yikun, L. Zihan, and D. Bin "Data mining and feature analysis of college students' campus network behaviour", *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pp. 231-237, 2020. https://doi.org/10.1109/ICBDA49040.2020.9101257

[16] Z. Liang, X. Di, Z. Liu, X. Liu, X. Zhang, and Z. Yu "Predicting students' academic performance based on improved PSO-Xgboost: a campus behavior perspective", *International Conference on Algorithms and Architectures for Parallel Processing*, pp. 402-421, 2021. https://doi.org/10.1007/978-3-030-95384-3_26

[17] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh "An overview of variants and advancements of PSO algorithm", *Applied Sciences*, Vol. 12, No. 17, art. no. 8392, 2022. http://dx.doi.org/10.3390/app12178392

[18] F. A. Şenel, F. Gökçe, A. S. Yüksel, and T. Yiğit "A novel hybrid PSO–GWO algorithm for optimization problems", *Engineering with Computers*, Vol. 35, pp. 1359-1373, 2019. https://link.springer.com/article/10.1007/s00366-018-0668-5

[19] X. S. Yang "Metaheuristic optimization: Nature-inspired algorithms and applications", *Artificial intelligence, Evolutionary computing and Metaheuristics*, Vol. 4, No. 27, pp. 405-420, 2013. https://doi.org/10.1007/978-3-642-29694-9_16

[20] S. N. Makhadmeh, M. A. Al-Betar, A. K. Abasi, M.A. Awadallah, I. A. Doush, Z. A. A. Alyasseri, and O. A. Alomari "Recent advances in butterfly optimization algorithm, its versions and applications", *Archives of Computational Methods in Engineering*, Vol. 30, No. 2, pp. 1399-1420, 2023. https://doi.org/10.1007/s11831-022-09843-3

[21] M. Alweshah, S. A. Khalaileh, B. B. Gupta, A. Almomani, A. I. Hammouri, and M. A. A. Betar "The monarch butterfly optimization algorithm

for solving feature selection problems", *Neural Computing and Applications*, Vol. 34, pp. 11267-11281, 2022. https://doi.org/10.1007/s00521-020-05210-0

[22] A. B. Nadaf, S. S. Reddy, R. Sanwal, T. Thulasimani, K. K. V. Satyanarayana, and M. Krishnaraj "An Efficient Method for Empowering engineering students through employability skills Based on New A-ResNet-LSTM Model", *2024 3rd International Conference for Advancement in Technology (ICONAT)*, pp. 1-6, 2024. https://doi.org/10.1109/ICONAT61936.2024.10775238

[23] O. Ahmad, and B. Farooq "Auxiliary-LSTM based floor-level occupancy prediction using Wi-Fi access point logs", *Journal of Smart Cities and Society*, Vol. 1, No. 3, pp. 195-211, 2022. https://doi.org/10.3233/SCS-220012

***