Predicting Student Placement Outcomes with Bi-GRU: A Deep Learning Approach

Dr. T Kavipriya[1] and Mr. N. Kumar[2]

[1]Assistant Professor in Computer Science with Cyber Security
Sri Ramakrishna College of Arts and Science, Coimbatore-641 006, Tamil Nadu, India
kkavipriya09@gmail.com
&
[2] Assistant Professor (SG) in Computer Science
Dr.N.G.P. Arts and Science College, Coimbatore-641048, Tamil Nadu, India
pkn.kumaar@gmail.com

Abstract

For students, the shift from academia to the workforce is a pivotal time, and it is becoming more and more important to be able predict their placement success. Placements are the main determinant of admission and establishment names. The primary goal of this effort is to predict current students' placement prospects by analysing past student data from prior years, which helps the institutions' placement rates rise. Based on the data of students who have been put in the past, this system offers a Recommendation System (RS) that predicts whether the current student will be placed or not. If the student is placed, the company is also predicted. To predict the outcomes, Deep Learning (DL) based classification algorithm, namely Bi-directional Gated Recurrent Unit (BiGRU) algorithm is presented in this study. Afterwards, the dataset-based algorithms' efficiency was assessed. Initially, the dataset is pre-processed using Min-Max Normalization (MMN) algorithm which is used to remove the noise rate effectively. Then, Feature Selection (FS) is applied using Improved Cuckoo Search Optimization (ICSO) algorithm for the given Student Placement (SP) dataset. It is used to generate best Fitness Values (FV) via Objective Function (OF). At last, the Placement cell (PC) periodically assists companies in identifying suitable students and focussing on enhancing their technical and social abilities using a DL model based on BiGRU. By examining the dataset of students from the previous year, a predictor model is presented with the rise of Data Mining (DM) and DL. The experimental results show that the suggested ICSO-BiGRU technique performs better than the existing methods in terms of Accuracy (Acc), Precision (P), Recall (R), and F-measure.

Keywords: Student Placement Data (SPD), prediction, Bi-directional Gated Recurrent Unit (BiGRU) algorithm and Improved Cuckoo Search Optimization (ICSO) algorithm.

1. Introduction

In order to get an effective profession, the majority of higher education students enrol in courses. Therefore, after finishing a particular course, it is essential for students to make a well-informed career decision regarding their placement. There are several student records in an educational institution [1]. Therefore, it is not difficult to detect patterns and qualities in this vast volume of data. Professional Education and non-professional education are the two divisions of higher education. Students who complete professional education gain professional knowledge that will help them succeed in the business world. It can be technology-focused or solely focused on enhancing a candidate's managerial abilities [2].Professional computer technology education is offered to students enrolled in Masters in Computer Applications (MCA) courses. Students graduate from this course with state-of-the-art (SOTA) information technologies theoretical and practical skills. They will be able to take part in the quickly changing information sector as a result. Students' efforts to make appropriate progress will be aided by the prediction of where MCA students can be placed after completing the course [3]. During the course, it will assist teachers in paying appropriate attention to students' progress. It will help establish the institute's reputation among other IT education institutions in the same category. A key component of successfully completing any graduate or postgraduate (PG) coursework is placement [4]. In order to accomplish student's aims and objectives, every student wants to get placed in a prestigious MultiNational Corporations (MNC). Universities and other institutions have stepped up their efforts to place many students as possible by providing training and PC to equip and upgrade their students.

Extracting or "mining" knowledge from vast amounts of data is referred to as DM, or Knowledge Discovery (KD) in databases [5]. In order to find hidden patterns and relationships that aid in Decision-Making (DM), DM techniques are applied to vast amounts of data. Although KD and DM are sometimes used interchangeably in databases, DM is simply a step in the KD procedure. Finding strategies and trends in large databases to inform decisions regarding upcoming projects is known as DM. Therefore, DM techniques will be able to identify the model with minimal user input [6].

In order to make strategic decisions and get Business Intelligence (BI), the model that is being provided can be helpful in understanding the unexpected and in providing Data Analysis (DA). This is followed by the application of additional tools to analyse DM. "Exploring the knowledge of database" is a more suitable term to describe the process of extracting and exploring knowledge from large amounts of data. A database contains knowledge regarding the process of discovery. Preparation and result interpretation are part of this procedure [7]. The most often used DM approach is classification. In order to create a model that can categorise the entire population of records, this classification uses a collection of pre-classified attributes.

Classification algorithms based on Decision Trees (DT) or Neural Networks (NN) are commonly used in this method. Learning and classification are steps in the Data Classification (DC) process. A classification algorithm analyses the training data during learning [8]. Test data is used to estimate the classification rules' accuracy. If the accuracy is acceptable, the rules can be applied to the new data sets.

The classifier-training method determines the set of parameters needed for appropriate discrimination based on these pre-classified attributes. These parameters are subsequently encoded by the algorithm into a classifier model.

Machine Learning (ML) is the technology that enables computers to learn without explicit programming [9]. Because your computer has learnt to differentiate between spam and non-spam emails, a spam filter saves you from having to sift through a tonne of spam emails every time you need your email. Because different classifiers yield contradictory results, it can be difficult to model and predict student actions from a student dataset using ML approaches. Because of this, researchers use student datasets to test several types of models based on ML techniques [10].

Ten of the most popular conventional classification methods, such as ZeroR, Naive Bayes (NB), DT, Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Adaptive Boosting (AdaBoost), and Logistic Regression (LR) classifiers, are used to assess the effectiveness of an ML-based prediction model. They chose to employ these techniques since they are frequently used to predict a student's future in the workforce. In the background and associated work section, the ML classifiers are briefly explained. Following the development of the ML classifier-based framework, tests are conducted on real student datasets that include pertinent background information as well as information about each student's activities to assess each model's performance [11].

These datasets were created by graduates from a wide range of academic disciplines. In this study, researchers assess various classifier-based models using metrics such as Acc, P, R, F-Score, Receiver Operating Characteristic (ROC) value, and Error Rate (ER). The outcomes are predicted using a DL-based classification technique called the BiGRU algorithm. Afterwards, the dataset-based algorithms' efficiency was assessed.

The remaining study is organised in the following way: The most modern approaches to student placement, such as DL and ML techniques, are examined in Section 2. The process of the suggested methodology is shown in section 3. The results and discussion are presented in section 4. The conclusion and Future work are covered in section 5.

2. Literature review

Research in the subject of education has advanced significantly due to DM tools. This section examines some of the most recent methods for utilising ML techniques for predicting student placement.

Recently, effective ML and Kaggle competitions have been dominated by the eXtreme Gradient Boost (XGBoost) Technique, a structured or tabular data-focused method developed by Manike et al. [12]. A gradient boosted DT implementation that is quick and efficient is called XGBoost. To ascertain if the student would be placed and what kind of organisation he would be placed in, researchers developed a model and conducted multiple EDAs.

Information from students who have completed their courses at various colleges was acquired by Rao et al. [13]. Communication is the way information is obtained. Enquiring about their performance, families, abilities, habits, personal information, academic background, and other factors that kept them from seizing the chance. Next, in order to build a model using synthetic data, researchers created a dataset that included every component that influenced a student's career. Colleges and universities can also benefit from student placement prediction since it offers insightful data on the career outcomes of students. Colleges can provide services and initiatives to help their students become more prepared for their professions by knowing the factors that affect student job placement. In comparison to conventional classification methods, the performance of the XGBoost ML model was assessed using Acc and P. The results show that the suggested method outperforms other methods by a substantial amount.

To predict student performance, Harihar et al. [14] employ logistic classifiers, Logistic Model Trees (LMT), logistic MultiLayer Perceptron (MLP), sequential minimal optimisation (SMO), and simple logistic. Based on the data set, these classifiers independently predict the outcomes before comparing the algorithms' accuracy. They are able to determine which ML algorithm is performing better than the others on the student data set by conducting analysis on various matrices (time taken to build classifier, correctly classified instances, Root Mean Squared Error (RMSE), incorrectly classified instances, P, R, F-measure, ROC area) by various algorithms. This enables us to set standards for future enhancements in the educational placement performance of students.

A RS that predicts whether or not the current student will be placed, it was presented by Archana et al. [15]. If the student is placed, the company is also predicted using data from students who have already been placed. Here, employing 2 distinct ML classification methods, the KNN algorithm and the NB Classifier, to predict the outcomes. Based on the dataset, the techniques' efficiency was analysed.This strategy helps a company's PC in identifying possible students on a regular basis and concentrate on improving their technical and social abilities.

According to student's academic performance in classes ten, twelve, graduation, and backlog to date in graduation, Maurya et al. [16] offered a few Supervised ML (SML) classifiers that might be used to forecast a student's placement in the IT sector. Additionally, we contrast the outcomes of various suggested classifiers. Accuracy score, percentage accuracy score, Confusion Matrix (CM), heatmap, and classification report are some of the criteria used to compare and examine the output of various classifiers that have been constructed. P, R, f1-score, and support are the characteristics that make up the classification report produced by developed classifiers. To create the classifiers, the following classification methods are used: SVM, Gaussian NB, KNN, RF, DT, Stochastic Gradient Descent (SGD), LR, and NeuralNET. New data that is not included in the experiment's dataset. This new data is utilised to test each of the generated classifiers.

To predict the students who will be placed for the current year by examining the data gathered from students in prior years, an automatic model was presented by Srinivas et al. [17] . An algorithm to predict the same is suggested for this model. In order to make predictions and use appropriate data pre-processing procedures, the institution has gathered the data. The LR technique is used to prepare this model. Using the dataset as a basis, researchers compare the algorithm's efficiency after it independently predicts the outcomes. The PC will be better able to concentrate on the prospective students and assist them in developing their social and technical abilities via this model.

Over 185 students' personal information is included in a dataset that was created by Muthusenthil et al. [18]. The dataset, which includes information on 2018 and 2019 graduating students, was gathered from SRM Valliammai Engineering College. The dataset includes information about the students' other characteristics in addition to their academic performance. The accuracy score would be improved by the approximately 20 attributes in the dataset. The objective is to create a training model that uses this information to predict a student's final CGPA as well as their placement result. This model was able to accurately predict the student's placement outcome and final CGPA.
Algorithms such as Lasso regression (LR3), LR (LR2), KNN, DT, and LR (LR1) have been used by researchers to predict the outcomes. And with an accuracy of over 94%, they accomplished outstanding outcomes. The accuracy scores of the aforementioned algorithms are compared to determine the predicted results. The final result was the one with highest accuracy.

A prediction model was presented by Kumar et al. [19]. The study examines a variety of ML techniques, such as KNN, RF, Gaussian NB, SVM, and LR. Performance metrics including R, Acc, P, and F1-score are used to assess the prediction models. A comparison analysis determines the best ML technique for predicting students' placements according to academic performance. Based on the results, it is possible to predict college student placements using RF techniques. The results demonstrate the significant influence of academic factors on prediction

accuracy, including grades and test scores. Career counsellors, students, and educational institutions may find significance in the study's outcomes.

A classification method that would be as accurate as feasible while working for DC was presented by Divya et al. [20]. The accuracy of an algorithm depends on the kind of problem that has to address and the data it must gather. This study finally selected SVM, RF, and DT as Current Techniques and LR, KNN, and Gradient Boosting Classifier as Suggested Techniques in order to assess the accuracy levels of each method with respect to the problem and data set. In order to choose which approach to utilise when incorporating the information into the placement analysis system, this study may utilise the test outcomes.

A placement analysis system was created by Nalayani et al. [21].It uses factors such as completed internships, papers published, aptitude scores, etc. to forecast a person's placement, whether it be in a dream business, core firm, normal company, or somewhere else. 3 fundamental algorithms are utilised in ML for predicting the outcomes. To identify the best algorithm, the algorithms' accuracy is assessed using RF, DT, and K-Means Clustering (KMC). A portion of the training dataset, which included historical placement results data, was used to assess the model's accuracy. The probability that a student will be placed can thus be predicted if the accuracy is high.The model can be used to identify the areas in which the student has to develop in order to get his intended outcome if the student is not satisfied with the outcome. This helps to satisfy each student's goal by offering solutions if it is executed properly and the students work consistently.

A system for automatically predicting a student's placement selection based on their past and current academic records was given by Nagamani et al. [22]. The organisation that uses the algorithms provides the necessary data. Applying classification techniques like RF and SVM comes after the first step of pre-processing the collected data. Each algorithm may produce different results, thus Acc, P, and R values are compared in order to determine which method performs best.
The aforementioned analysis indicates that the accuracy of the student placement based on ML needs to be improved. Therefore, the focus of this study is on DL-based student placement prediction.

3. Proposed Methodology

The BiGRU method, a DL-based classification system, is used in this study for predicting the outcomes. The algorithms' respective efficiency is then compared using the dataset. Primarily, the dataset is pre-processed using min-max normalization algorithm which is used to remove the noise rate effectively. Then, feature selection is applied using Improved Cuckoo Search Optimization (ICSO) algorithm for the given student placement dataset. It is used to generate best fitness values via objective function. At last, the PC periodically assists a firm in identifying suitable students and focussing on enhancing their technical and social abilities using a DL model based on BiGRU. By examining the dataset of students from the previous year, a predictor model is presented with the rise of DM and DL.

3.1. Data Normalization (DN)

All occurrences of Missing Values (MV) in the form of noise or nulls were eliminated or substituted with their mean values in order to improve the predictive models' performance efficiency [23]. For instance, the date values that indicate the date the assessments were taken / submitted were missing from the assessments table. The date mean value was used to replace all date instances with N/A, null, or MV because the date is a crucial variable in the early prediction of at-risk students. MMN is used to carry out these operations.

- **MMN**

Using Linear Transformation (LT), the MMN technique converts the input data into a new fixed range and normalises the dataset [24]. The relationships between the scaled value and the original input value are maintained by the min-max approach. Additionally, when the normalised values diverge from the original data range, an out of bound error occurs. Extreme input values are assured to be contained within a given range due to this technique. Equation (1) states that MMN converts a value X_0 to X_n that falls within the specified range.

$$X_n = \frac{X_0 - X_{min}}{X_{max} - X_{min}} \quad (1)$$

For variable X, X_n is a new value . The current value of variable X is X_0. The dataset's minimum and maximum Data Points (DP) are denoted by X_{min} and X_{max}, respectively.

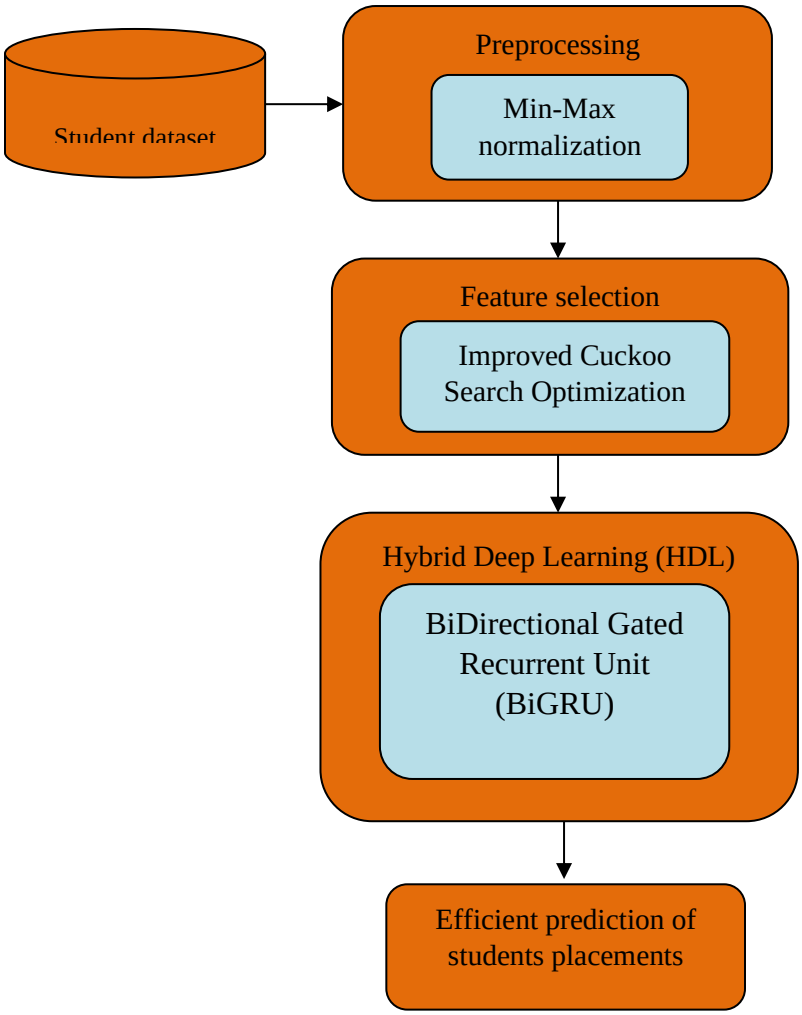

Figure 1. the procedure of the suggested approach

3.2. FS using ICSA

In order to improve the fundamental cuckoo search (CS) algorithm's optimisation capability, an ICSA is suggested in this section. To enhance its exploitation search capabilities, the CS algorithm uses orthogonal design and Opposition Learning (OL). In order to initialise the population and generate new candidate solutions in evolutionary generations, CS incorporates OL, which can disperse the population as widely as possible across the Searching Space (SS) and direct it towards the more promising locations.

Basic CS Algorithm.

Some cuckoo species are brood parasites, laying their eggs in the nests of other host birds. This is the basis for the fundamental CS algorithm. The three ideal norms listed below are used to simplify the description of basic CS: The

number of available host nests is fixed, and the host bird has a probability of $p_a \in [0, 1]$of finding the cuckoo's egg. (1) A randomly chosen clutch of eggs is dumped by each cuckoo after it has laid one egg at a time. (2) The next generations will inherit the best nests with superior eggs [25].
The host bird has two options in this situation: either remove the egg or just leave the nest and construct a sophisticated new one.

The effect of substituting new eggs for cuckoo eggs: (eggs found by the host bird) at each generation is reflected in this probability. Consider that a solution is represented by an egg. As a selection procedure for the optimisation algorithm, these presumptions guarantee that the best solutions will endure from one generation to the next. Therefore, the objective of the CS algorithm is to substitute new, higher-quality solutions for the ones found in the nests. The following provides a novel solution X_i^{t+1} for cuckoo *i*:

$$X_i^{t+1} = X_i^t + \alpha \otimes L'evy(\lambda) \quad (2)$$

$$\alpha = \alpha_0 \otimes (X_j^t - X_i^t) \quad (3)$$

Here, the dimension of the problem is equal to α, which is the step size (α > 0). Entry-wise multiplications (EWM) are represented by the product ⊗. The randomly selected solution is denoted as X_j^t. Lévy Flights (LF) random walks are the *L*évy(λ). To improve the search abilities, the parameter α0 is set to 0.01 as suggested. One type of random walk whose step duration is determined by the Lévy distribution is called a LF. A Probability Density Function (PDF) with a power law tail that is produced by a sequence of instantaneous jumps defines this distribution:

$$L'evy(\lambda) \approx S = t^{-\lambda}, (1 < \lambda \leq 3) \quad (4)$$

A uniform distribution that complies with the Lévy distribution is used to determine the step length S of LF. Additionally, using a switching parameter pa, the approach used an equitable balance of the global explorative (RW) Random Walk and a local RW (LRW). One way to express the LRW is as

$$X_i^{t+1} = X_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes (X_j^t - X_k^t) \quad (5)$$

Here, random permutation is used to find two distinct solutions, $X_j^t$and X_k^t. Heaviside functions is denoted as H. ε represents a random number, and it is selected from a uniform distribution. Then, s denotes the step size.

Alternatively, LF are used to execute the global random walk:

$$X_i^{t+1} = X_i^t + \alpha \oplus L'evy(s, \lambda) \quad (6)$$

The step size Scaling Factor (SF) is represented as $\alpha > 0$. The s that are distributed in accordance with the probability distribution depicted in (5), which has an infinite variance and an infinite mean, are denoted by L´evy(s, λ).

$$L'evy(s,\lambda) = \frac{\lambda\Gamma(\lambda)\sin\left(\frac{\pi\lambda}{2}\right)}{\pi}\frac{1}{s^{1+\lambda}} \quad (7)$$

Modified Orthogonal design (OD) and OL based CSA

The OL operation and OD are used into the CS algorithm to enhance its searching capabilities [26]. Utilising the fractional experiment's qualities to effectively identify the optimal level combination is the fundamental concept behind the OD. $L_M\ (Q^K)$ is an orthogonal array of K factors with Q levels and M combinations. Here, $M = Q^J$and Q is the prime number. $K = (Q^J - 1)/(Q - 1)$ is satisfied by the positive integer J. The following approach explains the short process of creating the orthogonal array $L_M\ (Q^K) = [a_{i,j}]_{M.K}$. Algorithm 1 describes the OD method.

Procedure 1: Stages for OD

Stage 1: Create the basic columns
For k = 1 to J
j = $\frac{Q^{k-1}-1}{Q-1} + 1$
For i =1to Q^J
$a_{i,j} = \left\lfloor \frac{i-1}{Q^J-k} \right\rfloor \bmod Q$
End for
End for
Step 2: Create the non-basic columns
For k = 2 to J
j = $\frac{Q^{k-1}-1}{Q-1} + 1$
For s =1 to j-1
For t = 1 to Q-1
$a_{j+(s-1)(Q-1)+1} = (a_s \times t + a_j) mod\ Q$
End for
End for
Step 3: Augmented $a_{i,j}$ by one for $1 \le i \le M, 1 \le j \le N$

Algorithm 1: OD

Begin
(1) Use the steps listed above to create the orthogonal array.

(2) Arbitrarily choose 2 solutions from the population
(3) Quantize the domain formed through the 2 solutions
(4) Randomly create $k - 1$ integers $p_1 \dots p_{k-1}$
(5) Apply $L_M\ (Q^K)$ to create Mpotential offspring
(6) From the population, a solution is selected at random
(7) To obtain the superior solution, compare it with the best solution from the orthogonal *M*offspring.
End

In Computational Intelligence (CI), OL is a novel concept. The fundamental principle of OL is to provide a better approximation of the existing candidate solutions by taking into account both a solution and its corresponding opposing solution. It has been shown to be a useful technique for improving different optimisation strategies. To further boost diversity and improve convergence, the suggested algorithm incorporates the OL concept.

Let $x_i \epsilon [Lx_i, Ux_i], (i = 1,2, \dots, n)$ be a solution in an n-dimensional space, such that $X = (x_1, x_2, \dots ., x_n)$. $X' = (x'_1, x_{\prime 2}, \dots ., x'_n)$ is the opposite solution.

$$x'_i = Lx_i + Ux_i - x_i \quad (8)$$

To evaluate the FV, consider f (.) be a fitness function. X is replaced by X' if $f(X') \leq f(X)$, and X is retained otherwise, according to the definitions of X and X'provided above. Thus, to determine the _tter solution, the solution and its opposing solution are examined concurrently. Algorithm 2 describes how OBL is used to initialise the population and generate new solutions during the evolution process.

Algorithm 2: Modified Orthogonal OL based CSA (MOOLCSA)

Input: student dataset
Output: Optimized Features
(1) Set the iteration number t = 1, initialise the method's parameter values, and generate random initial vector values.
(2)Determine the individual's current best individual with the highest objective value by calculating their FV. Verify if the stopping criterion is satisfied. If the stopping criterion is satisfied, output the optimal solution; if not, update the iteration number $t = t + 1$ and keep going until the stopping criterion is satisfied.
(3) Maintain the best solution of the final iteration, and obtain a series of new solutions $Xnew = \left[x_1^{(t+1)}, \dots ., x_i^{(t+1)}, \dots x_K^{(t+1)}\right] by$ LF,
(4) Calculate the FV $F_i^{(t+1)}$ of the new solution $x_i^{(t+1)}$, and compare $F_i^{(t+1)}$ with $F_i^{(t)}$ which denotes the solution of the t^{th} iteration.
5) New nests are constructed and a proportion (pa) of the poorer ones are abandoned.
(6) Put the OD strategy methods into practice.
(7) Retain the best solution.
(8) Use the OL method with Eqs to find a new solution.
(9) Use high-quality solutions to keep the best nest.
(10) Sort the nests and choose the best one at the moment. (11) Transfer the best nest to the following generation.
(12) Go to step (2)
End

3.3. Bi- GRU based DL

One type of Recurrent NN (RNN) is GRU. It is suggested to solve the issues of gradient in Back-Propagation (BP)and long-term memory, just like LSTM (long short-term memory). Because GRU is computationally less expensive than LSTM while still performing comparably, researchers have chosen it for this work [27]. Figure 2 depicts the structure of GRU. The new hidden state h_t, the update gate z_t, the reset gate r_t, the current input x_t, and the prior hidden state ht−1 make up GRU. The following procedure uses x_t and h_{t-1} as the inputs and h_t as the output of GRU.

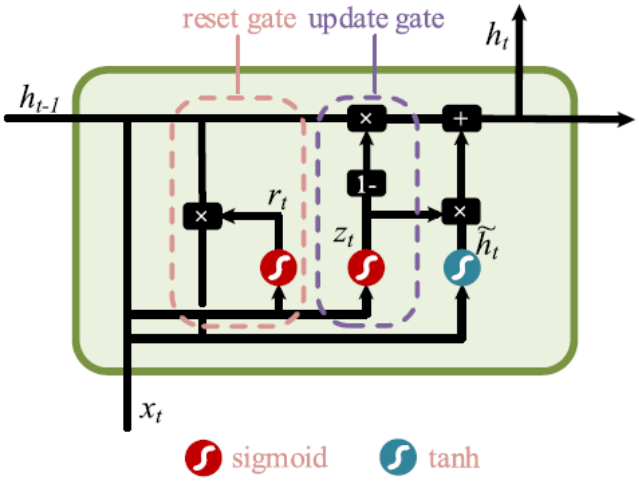

Figure 2. The GRU structure

• z_t

The degree to which the information from the preceding moment is incorporated into the current state is controlled by the z_t. More information is brought in from the preceding moment the higher the z_t. One can compute z_t by:

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \qquad (9)$$

Here, the weight matrix and bias of the update gate are denoted by W_z and b_z, accordingly. The Sigmoid function is denoted as σ(x) = 1/[1+exp(−x)]. As the gate-control signal, the data is transformed into values between 0 ∼ 1.

• r_t

The amount of the Hidden Layer (HL) data from the previous instant that must be forgotten is controlled by the r_t. It can be computed using:

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \qquad (10)$$

Here, the weight matrix of the r_t is denoted by W_r and the bias of r_t by b_r. In order to forget the data about the hidden state of the preceding moment, the sigmoid will set the output of the r_t to 0 if there is no need to remember it. On the other hand, all of the concealed states from the previous instant will be able to pass through if a sigmoid sets the output of the r_t to 1. That is, less information from the prior state is preserved when smaller the reset gate.

• **Candidate output state ($\bar{h}_t$)**

The $\bar{h}_t$ can be computed as follows:

$$\bar{h}_t = tanh(W_h.[r_1 \odot h_{t-1}, x_t] + b_h) \qquad (11)$$

Here, the weight matrix and bias of the $\bar{h}_t$ are denoted by W_h and b_h, correspondingly. The Hadamard Product, or the multiplication of the matching matrix elements, is represented as $\odot$. Tanh activates the function to scale the data to a range between −1 and 1. The GRU unit's $\bar{h}_t$ is directly impacted by the r_t output. The amount of the neuron's output from the previous moment is kept depends on the product of r_t and the HL state of that moment. The more the neuron's output from the previous moment is preserved, the higher the value of r_t, and vice versa.

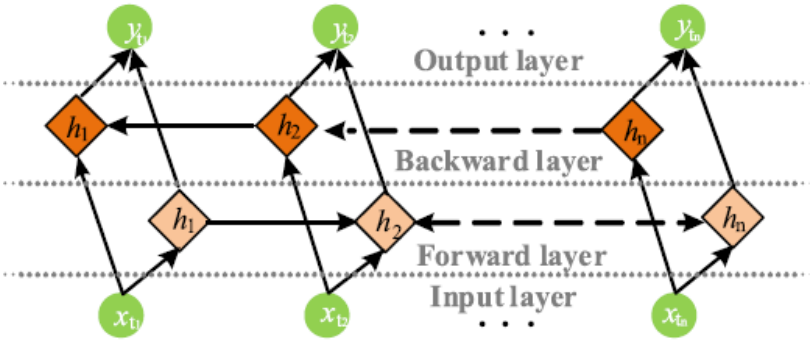

Figure 3. The Bi-GRU structure

• h_t

The new HL state h_t, which is defined by z_t, ht−1 and h˜ t, is the output of GRU. It is mathematically expressed as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \bar{h}_t \qquad (12)$$

Here, h_t depends more on $\bar{h}_t$ and h_{t-1} has a smaller determining effect on the output as z_t increases. The selective "forgetting" to the prior hidden state h_{t-1} is shown by $(1 - z_t) \odot h_{t-1}$.Selective "memory" to $\bar{h}_t$, which contains the current node information, is indicated by $z_t \odot \bar{h}_t$. Bi-GRU integrates the input sequence information in both forward and backward directions, in contrast to regular GRU, which can only predict the output of the subsequent instant based on the temporal sequence information of the preceding moment. Fig. 3 presents the bidirectional structure. 2 HL with opposing Data Transfer (DT) and GRU unit directions can be considered as the bidirectional structure.

The inputx_t simultaneously generates the HL in two opposing directions at time t. These two one-way HL work together to determine the output y_t at time t. Moment t and the preceding moment in the input sequence are known to the forward GRU layer, and moment t and the succeeding moment in the input sequence are known to the backward GRU layer. The Bi-GRU HL propagation technique is described as follows:

$$\vec{h}_t = f(\vec{W} x_t + \vec{V} \vec{h}_{t-1} + \vec{b} \qquad (13)$$

$$\overleftarrow{h}_t = f(\overleftarrow{W} x_t + \overleftarrow{V} \overleftarrow{h}_{t+1} + \overleftarrow{b} \qquad (14)$$

$$y_t = \sigma(\vec{h}_t, \overleftarrow{h}_t) \qquad (15)$$

Here, the HL states of forward and backward computation are denoted by $\vec{h}$ and $\overleftarrow{h}_t$. In forward and backward computation, $\vec{W}$, $\vec{V}$, $\overleftarrow{W}$, $\overleftarrow{V}$ denote the input weight and the HL state of the preceding moment. The forward and backward computation biases are denoted by $\vec{b}$ and $\overleftarrow{b}$. A bidirectional cyclic HL structure can substantially enhance the model's ability to process for Nonlinear (NL) data by capturing the forward and backward dependence in the period of student placement. Consequently, it makes sense to predict student placement using a Bi-GRU structure.

4. Results and Discussion

The following tests are conducted and the results are presented in order to assess how well the suggested DL classifier performs when predicting various students using student placement data and benchmark data. The student placement dataset was considered for the prediction of various students in this research project. To generate updated classification rules for the prediction of various students, the Bi-GRU technique was used, and the suggested classification model has been applied to the student placement dataset

Precision is defined as the ratio of correctly found positive observations to all of the expected positive observations.

$$Precision = TP/(TP + FP) \qquad (16)$$

The ratio of accurately identified positive observations to all observations is known as sensitivity or R.

$$Recall = TP/(TP + FN) \qquad (17)$$

The weighted average of P and R is known as the F-measure. It thus accepts FN and FP.

$$F - measure = 2 * (Recall * Precision)/(Recall + Precision) \qquad (18)$$

The following describes how Acc is determined in terms of positives and negatives:

$$Accuracy = (TP + FP)/(TP + TN + FP + FN) \qquad (19)$$

Here, FN stands for False Negative, TP for True Positive, FP for False Positive, and TN for True Negative.

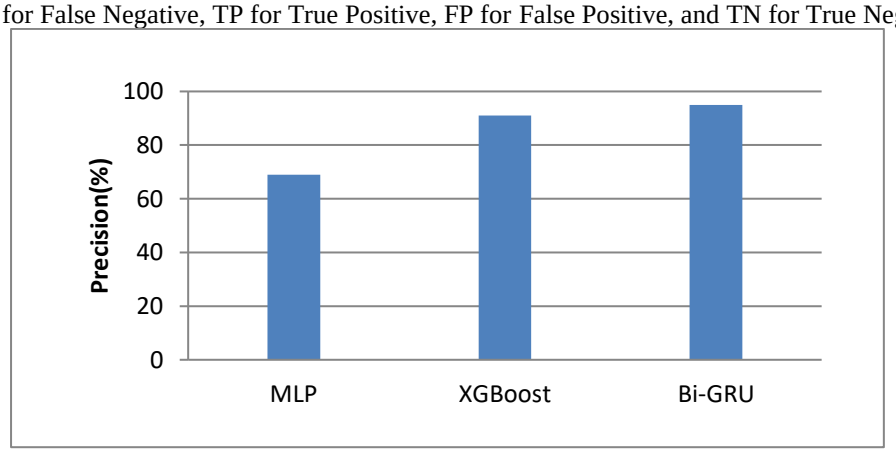

Fig.4. Precision comparison outcomes among the current and suggested student placement model

The precision comparison findings among the suggested Bi-GRU and the current approach for classifying student placement levels are shown in fig. 4. The results showed that the suggested predictive model (PM) was effective in predicting at-risk students performance as early as possible.

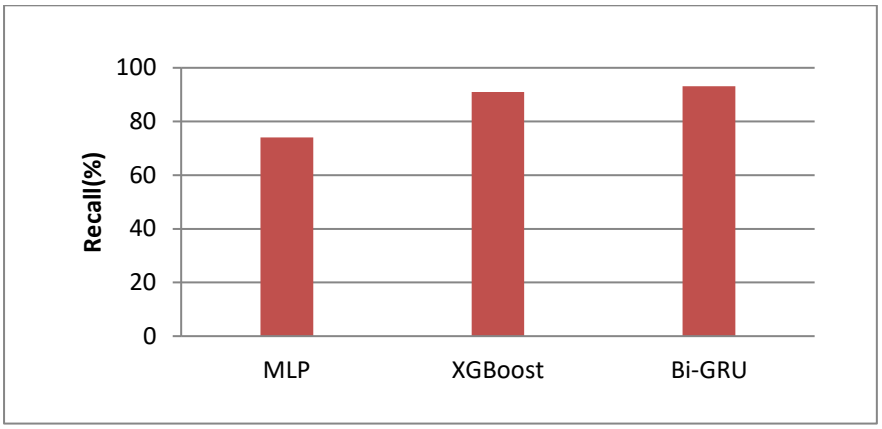

Fig.5. R- comparison outcomes among the current and suggested student placement model

The R- comparison outcomes among the current and suggested student placement models are shown in fig. 5. Based on the length of the course, the prediction model is therefore seen as a classification problem that arises from either a high-risk or low-risk student. In order to accomplish the objective, the suggested Bi-GRU models were used, and the outcomes were examined and assessed.

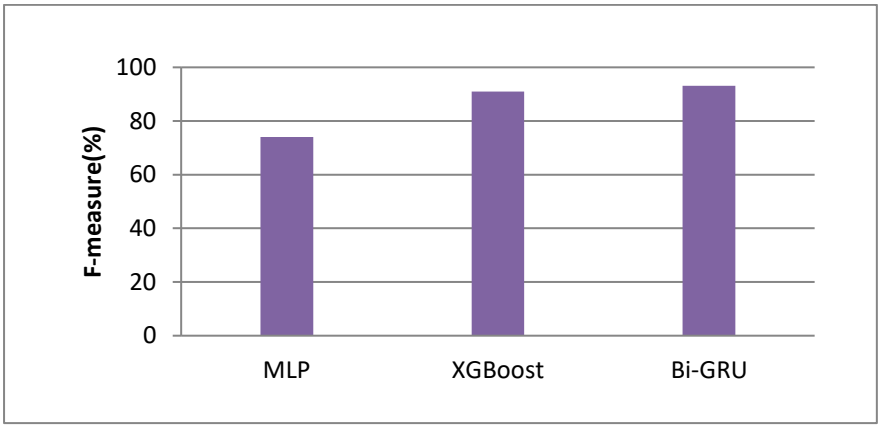

Fig.6. F-measure comparison outcomes among the current and suggested student placement PM

The F-measure comparison findings between the current and suggested student PM are shown in fig. 6. In general, the outcomes showed that the Bi-GRU model performed better on the given datasets than the ML algorithms that were taken into consideration. These results are consistent with the error rate that was previously produced and can be ascribed to the rule sets that the proposed Bi-GRU classification model generates. According to the outcomes, the suggested Bi-GRU strategy outperforms the current classification methods in terms of f-measure.

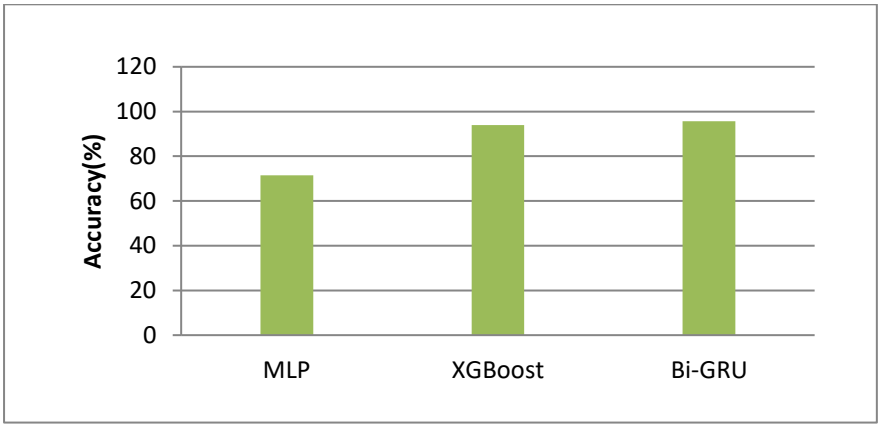

Fig.7. Acc comparison outcomes among the current and suggested student PM

A effective DL model is one that accurately predict the target and generalise predictions to new instances. Accuracy, which has two subtypes: sensitivity and specificity, is typically used for evaluating model validation. The accuracy comparison between the suggested and current student PM is shown in fig. 7. The simulation results show that the suggested Bi-GRU model has a high accuracy of 95.67%, compared to 93.9% for the current DFFNN model and 71.41% for the MLP model. In terms of accuracy, the results show that the suggested FDL approach performs better than the conventional classification techniques.

5. Conclusion

Improving student placement performance is one of the main issues that have been faced by higher education institutions today. As educational organisations get more complex, the placement prediction becomes more intricate. Educational institutions search for more effective technology that may help them develop new strategies, improve DM processes, and help with better management. Based on the data of students who have been put in the past, this system offers a RS that predicts whether the current student will be placed or not. If the student is placed, the company is also predicted. The BiGRU method, a DL-based classification system, is used in this work to predict the outcomes. The algorithms' respective efficiency is then compared using the dataset. To effectively reduce the noise rate, the dataset is initially pre-processed using the MMN algorithm. Then, FS is applied using ICSO algorithm for the given student placement dataset. It is used to generate best FV via OF. At last, the PC periodically assists a firm in identifying suitable students and focussing on enhancing their technical and social abilities using a DL model based on BiGRU. By examining the dataset of students from the previous year, a predictor model is presented with the rise of DM and DL. Based on the testing results, it was determined that the suggested ICSO-BiGRU algorithm outperforms the current methods in terms of Acc, P, R, and f-measure. The effectiveness of the various classifiers assessed for this study was examined through a comparative analysis. In terms of P, R, and F-score, this model has performed well. The use of temporal features for predicting students' evaluation grades is a potential avenue for future research.

REFERENCES

1. Joy, L. C., & Raj, A. (2019, March). A review on student placement chance prediction. In *2019 5th International conference on advanced computing & communication systems (ICACCS)* (pp. 542-545). IEEE.
2. Abed, T., Ajoodha, R., & Jadhav, A. (2020, January). A prediction model to improve student placement at a south african higher education institution. In *2020 International SAUPEC/RobMech/PRASA Conference* (pp. 1-6). IEEE.
3. Maragatham, T., Yuvarani, P., Harishri, S., & Swetha, R. (2024, November). Student Placement Prediction using Deep Learning Techniques. In *2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 620-626). IEEE.
4. Rao, K. E., Pydi, B. M., Vital, T. P., Naidu, P. A., Prasann, U. D., & Ravikumar, T. (2023). An Advanced Machine Learning Approach for Student Placement Prediction and Analysis. *International Journal of Performability Engineering*, *19*(8), 536.

5. Maurya, L. S., Hussain, M. S., & Singh, S. (2021). Developing classifiers through machine learning algorithms for student placement prediction based on academic performance. *Applied Artificial Intelligence*, *35*(6), 403-420.
6. Pal, A. K., & Pal, S. (2013). Classification model of prediction for placement of students. *International Journal of Modern Education and Computer Science*, *5*(11), 49.
7. Thangavel, S. K., Bkaratki, P. D., & Sankar, A. (2017, January). Student placement analyzer: A recommendation system using machine learning. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 1-5). IEEE.
8. Surya, M. S., Kumar, M. S., & Gandhimathi, D. (2022, April). Student placement prediction using supervised machine learning. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 1352-1355). IEEE.
9. Çakıt, E., & Dağdeviren, M. (2022). Predicting the percentage of student placement: A comparative study of machine learning algorithms. *Education and Information Technologies*, *27*(1), 997-1022.
10. Agarwal, K., Maheshwari, E., Roy, C., Pandey, M., & Rautray, S. S. (2019). Analyzing student performance in engineering placement using data mining. In *Proceedings of International Conference on Computational Intelligence and Data Engineering: Proceedings of ICCIDE 2018* (pp. 171-181). Springer Singapore.
11. Khandelwal, J., Pareek, G., Dey, R., & Pareek, S. (2023, February). The Study of Machine Learning Classification Algorithm for Student Placement Prediction. In *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)* (pp. 1-7). IEEE.
12. Manike, M., Singh, P., Madala, P. S., Varghese, S. A., & Sumalatha, S. (2021, December). Student Placement Chance Prediction Model using Machine Learning Techniques. In *2021 5th Conference on Information and Communication Technology (CICT)* (pp. 1-5). IEEE.
13. Rao, K. E., Pydi, B. M., Vital, T. P., Naidu, P. A., Prasann, U. D., & Ravikumar, T. (2023). An Advanced Machine Learning Approach for Student Placement Prediction and Analysis. *International Journal of Performability Engineering*, *19*(8), 536.
14. Harihar, V. K., & Bhalke, D. G. (2020). Student placement prediction system using machine learning. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, *12*(SUP 2), 85-91.
15. Archana, P., Pravallika, D., Priya, P. S., Sushmitha, S., & Amitha, S. (2023). Student Placement Prediction Using Machine Learning. *Journal of Survey in Fisheries Sciences*, 2734-2741.
16. Maurya, L. S., Hussain, M. S., & Singh, S. (2021). Developing classifiers through machine learning algorithms for student placement prediction based on academic performance. *Applied Artificial Intelligence*, *35*(6), 403-420.
17. Srinivas, C., Yadav, N. S., Pushkar, A. S., Somashekar, R., & Sundeep, K. R. (2020). Students placement prediction using machine learning. *International Journal for Research in Applied Science and Engineering Technology*, *8*(5), 2771-2774.
18. Muthusenthil, D., Mugesh, V. S., Thansh, D., & Subaash, R. (2020). Predictive analysis tool for predicting student performance and placement performance using ml algorithms. *Int. J. Adv. Res. Innovative Ideas Educ*, *6*.
19. 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.
20. Divya, N., Namburu, S., & Raja, R. (2023, June). Student Placement Analysis using Machine Learning. In *2023 8th International Conference on Communication and Electronics Systems (ICCES)* (pp. 1027-1031). IEEE.
21. Nalayani, C. M. (2023). Placement Analysis for Students using Machine Learning. *Journal of Information Technology and Digital World*, *5*(3), 223-237.
22. Nagamani, S., Reddy, K. M., Bhargavi, U., & Kumar, S. R. (2020). Student placement analysis and prediction for improving the education standards by using supervised machine learning algorithms. *J. Crit. Rev*, *7*(14), 854-864.
23. Ahmed, S., Zade, A., Gore, S., Gaikwad, P., & Kolhal, M. (2018). Performance Based Placement Prediction System. *IJARIIE-ISSN (O)*, *4*(3), 2395-4396.
24. Patro, S. (2015). Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.
25. Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, *29*, 17-35.

26. Yang, X. S., & Deb, S. (2014). Cuckoo search: recent advances and applications. *Neural Computing and applications*, *24*, 169-174.
27. Zhang, J., Jiang, Y., Wu, S., Li, X., Luo, H., & Yin, S. (2022). Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliability Engineering & System Safety*, *221*, 108297.