Adaptive Weight Deep Convolutional Neural Network (AWDCNN) Classifier for Predicting Student's Performance in Job Placement Process

T.Kavipriya, Assistant Professor, Department of Computer Technology, Hindusthan College of Arts and

Science, Coimbatore.

Dr. M. Sengaliappan, Dean of Computer Science, Kovai Kalaimagal College of Arts and Science, Coimbatore.

ABSTRACT:

Most of the educational institutes and entities have an urgent desire to predict and measure student performance. This prediction helps them in assuring student retention, facilitating learning experience with necessary resources and increasing the university reputation and ranking. These requirements can be considered and Educational Data Mining (EDM) can be used to discover student learning environments with college context to estimate the student performance. The educational institutions are curious to predict the student failure every time and this can be addressed by prediction method referred as Decision Support System (DSS) in a given arrangements. This DSS Prediction method used doesn't provide an accurate measure of the student failures since it lack the details of the parameters influencing the achievements of the students in a specific source in college context. Many researchers algorithm addresses only the classification and didn't provide any solution for the data mining issues such as data pre-processing and classification error, etc. The proposed prediction algorithm addresses the student performance extracting the specific keys and uses deep learning techniques. A classifier based on Deep Convolutional Neural Network (DCNN) predicts the student performance better compared to other classifiers. Three types of regularization methods are considered to compromise the problem of DCNN overfitting and thereby attaining early convergence such as weight regularization via the Genetic Algorithm (GA), Batch normalization and dropout. In this work, a new classifier is proposed namely Adaptive Weight Deep Convolutional Neural Network (AWDCNN) optimized by the use of GA algorithm to predict the performance of the students. The DCNN classifier weights are estimated using GA algorithm and these optimized weights are used to better classify the student results and the corresponding metrics used for performance analysis. The proposed classifier AWDCNN exhibited better student result prediction than other existing prediction methods and it is proved using training datasets in simulation results.

KEYWORDS: Prediction of students' performance, educational data mining, Classification algorithms, Machine learning, Deep Learning analytics.

1. INTRODUCTION

Many academic institutions and its associated domains are curious to predict and measure the student performance which leads to many research activities in areas such as personalized teaching, academic early warnings in recent years. To provide effective learning resources at critical time, a trustful prediction algorithm to predict the student performance is very essential during the course. The Data Mining [1-2] is used rarely in education environment and such type of mining is referred to as Educational Data Mining (EDM) [3-4]. The main aim of this EDM is to provide solutions to educational research issues where it is

used as a tool in educational settings. [5-6].

The main objective is to provide solutions to educational research issues [7], thereby improving the overall teaching learning process of the educational sectors [8]. The result of this mining will enhance teaching performance, organize institutional resources, student learning performance, personalized recommendations to students, and evaluate learning effectiveness, educational offer and much more [9]. Thus the EDM application includes modeling the accurate student performances and characteristics to enhance the learning experience of the students [10].

To analyze processes such as clustering, classification and association rule, the methods used in EDM are very essential. The prediction type classification rule, based on the training set classifies data which construct a pattern and using this pattern classifies a new data referred as testing set. The process of record grouping in classes as similar and dissimilar is referred to as Clustering. To discover the relationship between parameters [11-12], the relationship mining is used. Predicting definite outputs from various datasets is a predictive data mining technique used in Classification. Considering the known values of other variables, these models have the potential to estimate the unknown values of variable in a dataset. Mapping a set of vector input measurements to a scalar output is a learning process for predictive modeling. The predefined class groups are mapped using classification and since the classes are predetermined before data examination, it is often referred to as Supervised Learning.

The EDM community and learning analytics is analyzed to predict student performance through student dropout's prediction issues which is a prominent problem in performance prediction. The term examination performance and student's background data at various educational environmental levels are used to predict the student performance and these parameters in applying the learning methods in the proposed model. Previous works of the learning methods are categorized into two models as follows:

- The linear models such as Linear SVM, Survival Analysis and logistic regression [13] are the generalized linear models used traditionally first. Different models have specific type of predictive and behavioural features to be extracted from raw activity records for ex: grades, forum and clickstream).
- The second approach employs Neural networks (NN). Existing models explored models such as Recurrent Neural Network (RNN) model [14], Recurrent Neural Network (RNN) model [14], Convolutional Neural Networks (CNN) followed by RNN [15] and Deep Neural Network (DNN) model [13]. The performance of these models are primitive since they consider feature engineering in order to lower the dimensions(input) which helps in developing better NN models.

There are a lot of attractive challenges in designing a prediction models [16-17]. The datasets belongs only to a single class and it doesn't obey the traditional class distribution and this leads to design complexity [18]. There are two major reasons for this challenge, first the Decision Support System (DSS) is not accurate in predicting the student performance and secondly, insufficient data on factors influencing the student performance in a particular course leads to the design challenge. The selected prediction model is applied on the data collected about the student performance. The prime focus is to identify the key factors deciding the student performance in designing the deep learning algorithm using prediction model.

The performance of the prediction model can be improved using methods which are based on Deep Convolutional Neural Network (DCNN).The proposed algorithm predicts the performance of the students using a deep learning model which provide better optimized values in deciding the weights and this new classifier is termed as Adaptive Weight Deep Convolutional Neural Network (AWDCNN).

This classifier is able to predict whether the student performance meet up the placement criteria and their skills based on the course specific data analysis. The accuracy rate attained through this AWDCNN model is reasonable and is proven to achieve high accuracy compared to many existing machine learning algorithms given the database of the students.

2. LITERATURE REVIEW

To measure the student performance various methods are adapted focusing on different aims such as, student retention assurance, allocation of resources and respective courses, student risk level detection and many others. This section deals with performance prediction models of student data.

To predict the academic performance of the engineering students, Pandey and Taruna [19] presented a Decision Tree based Multilevel Classification Model (MLCM). This model consists of two levels and the first level consists of four different classification models and their construction, evaluation and comparison and they are listed as, Multi Layer Perception (MLP), Naïve Bayes Tree (NBT), Lazy Learner (IBK), and Decision Tree (J48). The classifier model selected in this first step is decision tree classifier. Then the level 2 deals with enhancing the individual classes as well the performance of overall classifiers by ignoring the low priority data sets is the student database by using MLCM-filtered dataset model. The student yearly CGPA is predicted as presented by Sikder at al [20] using NN and the values were predicted with the real CGPA values. The authors used actual datasets for better prediction efficiency. The Bangabandhu Sheikh Mujibur Rahman Science and Technology University (BSMRSTU) provided their student datasets for the prediction analysis.

Based on neuro fuzzy concept, a classification model to predict student performance proposed by Hidayah at al [21] a combination of neural network and fuzzy's IF-THEN rules. This model has the ability to generate best classification model which learn from the rules generated. Using the Adaptive Neuro-Fuzzy Inference System (ANFIS) Editor-Matlab Fuzzy Logic the student datasets were processed. The combination of three parameters values such as, motivation, interest and talent is the widely used best model for student performance classification with RMS error value of 0.123and average testing RMS error value of 0.25611. The psychometric indicators which can be analyzed during the enrolment, proposed by Gray et al [22] consider motivation, learning strategies and personality. Model accuracy was assessed using six models and cross validation done and the results were compared in subsequent years. It is event from the results that, modeling is complex older students than the younger ones. Also, the prediction accuracy is more close in the case of 10 fold cross validation in younger ones than the older students (over-estimated model).

A pre-warning system, based on a deep learning multiclass model which classifies the students based upon their risk of failure is presented by Guo et al [23]. The proposed deep neural system is six layered, feed-forward NN fully connected with Rectified Linear Units(ReLU). The softmax classifier is positioned with five neurons in the outer layer. Each layer maps the students final grade such as O, A, B, C and D. This proposed algorithm produced results with high accuracy compared to the traditional classification algorithm such as MLP, NB and SVM. To predict the graduating student performance in a degree program Asif et al [24] presented a model and this model considered the student pre-admission marks data and the score they attained during their four years of study. The marks they obtain are divided into five categories, A, B, C, D, and E. Several classification algorithms such as, k-Nearest Neighbor (kNN), NN, RF, NB and Decision Tree were studied and a result accuracy of 83.65% is obtained in NB.

A hybrid algorithm with feature selection considering ML Classifiers, Decision Trees and NB

is presented by Turabieh [25] to evaluate student performance. The valuable features from the student database are extracted and the Binary Genetic Algorithm is presented. This is used for wrapper feature selection datasets. A standard dataset from ML repository of University of California Irvine (UCI) and the proposed algorithm showed excellent performance results. Xu et al [26] proposed a system which predicts the CGPA based on the known performance states and course prediction of the student. Two–layer architecture is proposed for progressive predictions. The proposed algorithm has several layers and the first layer construct the base predictors based upon the course whose performance state is mapped to the course of target. A course clustering technique is used to discover relevant courses. The ensemble predictors, in the second layer improve itself by acquiring new student datas. The proposed algorithm exhibited better results, when compared to the classic ML algorithms such as Logistic Regression (LogR), Linear Regression (LR), kNN and Random Forest (RF).

The student performance is predicted based on the student behaviours like variation, duration and periodicity as presented by Ma et al [27]. It is challenging to design a prediction model with limited datasets and also it is of huge challenge to extract datas from a huge smartcard data samples manually. A unified framework to predict student performance belonging to multiple domains is proposed and it is termed as Dual Path Convolutional Neural Network (DPCNN). The proposed technique showcased better results than the existing methods.

To handle time series data in predicting final results of the graduate students, Mondal and Mukherjee et al [28] presented Recurrent Neural Network (RNN). The final results are estimated by the first and second term results with additional fifteen other parameters. This prediction model will help the teacher to understand the student's performance in early stages and can help them in improving the grades. The proposed RNN is compared with DNN and ANN. A prediction model based on deep learning techniques which addresses the sequential event prediction problem termed as GritNet is presented by Kim et al [29]. This proposed algorithm is based on Bidirectional Long Short Term Memory (BLSTM). The presented prediction model is applied on the Udacity students' data to predict graduation which performed better and providing accurate predictions.

3. PROPOSED METHODOLOGY

Given with the huge educational data, predicting the performance of the students is a huge challenge. This section proposed a new prediction model which has four major processes, 1) Collecting relevant data, 2) Processing the missing datasets, 3) Data Clustering and 4) Deep Learning Decision Support System (DLDSS) Prediction model. In this section, the author proposed a deep learning based framework to predict the student performance. The proposed framework details about the levels involved in predicting the student performance at the placement level. The four main stages of the proposed prediction framework model involve data collection and integrity, computing missing data subject and student wise, clustering, and DLDSS model. The detail explanation about all stages will be described in the next subsections.
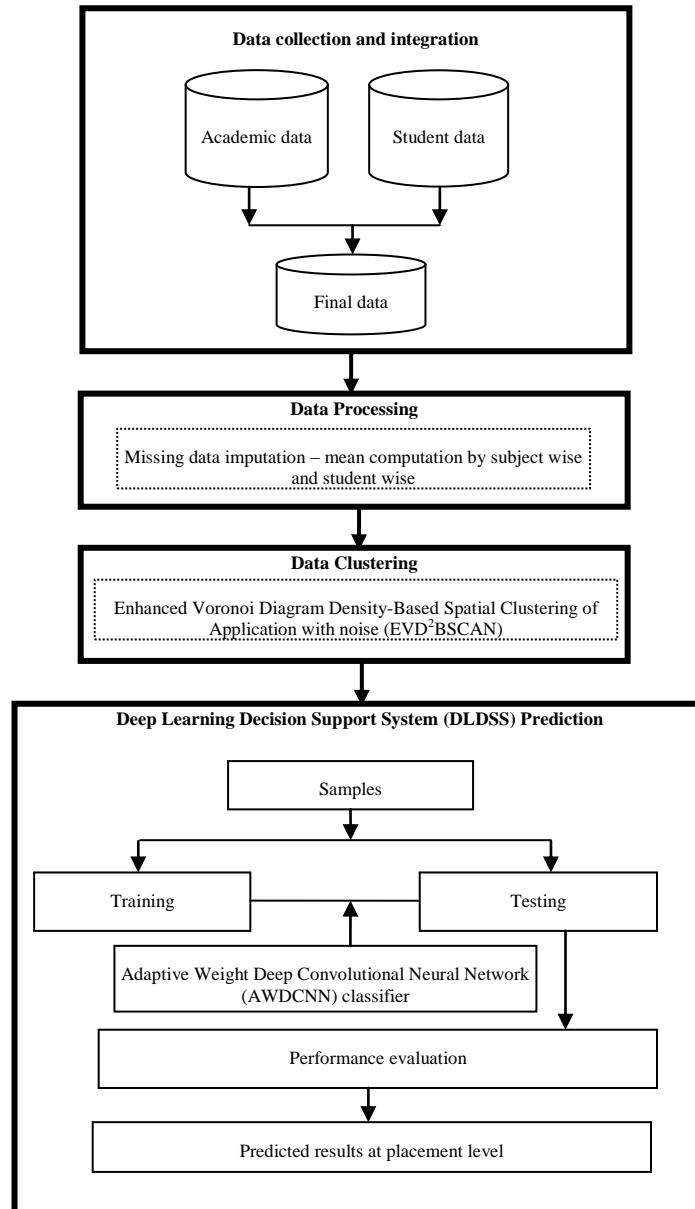
Figure 1. Flow Diagram Of Proposed Predictionmodel Based On Deep Learning Decision Support System (DLDSS)

3.1. DATASET COLLECTION

The database is collected from various colleges from different student groups considered for placements. The database includes, register number, student name, D.O.B, SSLC and HSC marks with school names, father name, mother name, medium of study, the percentage obtained in each course, marks obtained in individual subjects and in soft skills and their achievements during the course. The database was collected from 6324 students of colleges under Bharathiar University in the academic year 2017, completing their Under Graduate degree. The database includes student from different streams such as, Information Technology, Computer science, Commerce, Computer Applications, and others in UG. The R language is used to handle 75 columns and 632 rows of student's data. The irrelevant data in the database is removed and replaced by corrected data and for example, the student marks above 100 is considered as wrong data and it will be ignored from the computation. The missing data also is another concern to be considered while collecting a datasets to analyze student performance, since removal of false data may leads to missing data when ignored without adding new data to it.

3.2.   MISSING VALUE DATA IMPUTATION

The major issue in processing the database is the missing data and this is due to the missing instances, attributes without values, inadequate attributes, etc. These issues have to be sorted out before processing this datasets. This is eliminated in two techniques. Missing data can be filled by taking a subject average of SSLC, HSC and other term marks and this value can be used to fill the discontinuous datasets. Also, second method is to take the average of the particular student in a course and replacing the averaged mark in the missing section of the datasets. In filling this missing data, correlation and regression analysis is used [30] and the more of the technique is employed in the recent work [31].

3.3.   DATA CLUSTERING

Enhanced Voronoi Diagram Density-Based Spatial Clustering of Application with noise (EVD^{2}BSCAN) algorithm is proposed in this section. It is based on the concept that it starts the formation of the cluster by selecting core object. Then it computes the Density Mean (DM) of the growing cluster before allowing the expansion of an unprocessed core object. After that it computes the Density Mean (DM) including the ε -neighborhood of the unprocessed core object p. If the Density Variance (DV) of the growing cluster with respect to DM is less than a specified threshold value μ and the difference between the minimum and maximum objects lying in the ε -neighborhood of the objects, which are the objects of the growing cluster, including the ε neighborhood objects of the unprocessed core object, is less than a specified threshold value μ then only an unprocessed core object p is allowed for expansion otherwise the object is simply added into the cluster.

3.4.   DEEP LEARNING DECISION SUPPORT SYSTEM (DLDSS) PREDICTION MODEL

The Adaptive Weight Deep Convolutional Neural Network (AWDCNN) classifier is used in modeling the Deep Learning Decision Support System (DLDSS) model in predicting the student performance better than other existing models. An AWDCNN is used to estimate the student performance which is based on feed-forward NN [32] producing accurate prediction values. An AWDCNN prediction model have of an raw signal handling input layer, convolutional and pooling layers with sub-sampling processes, fully connected layers, and an output SoftMax Classifier layer. Figure 2 illustrate the processes involved in converting the raw input data into meaningful data using a neural network. The layer at the output together with the SoftMax classifier is considered to compute the performance of the students at placement level.
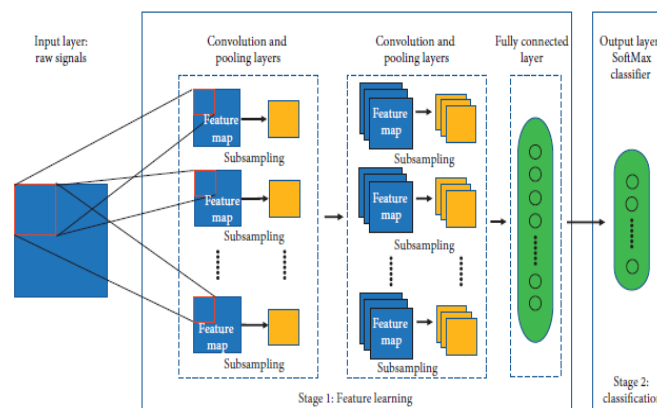
Figure 2. A Basic Architecture Of Awdcnn Classifeir

The most important layer in DCNN is the convolutional layer and with a set of weighted filters, it constructs a feature map of the student performance through convolution operations. These filter weights are updated frequently using GA. The data of different student attributes are extracted from different

filters with varying weights. The convolution operation used to extract the feature map is a mathematical concept where two matrices are point-multiplied, where the one input is the data of performance matrix and other is the filter matrix(feature matrix). To obtain non-linear relationship at the output characteristics map an activation function is primarily used. The sigmoid, tanh and softsign are the three types of saturated nonlinear functions and ReLU is a unsaturated type [33]. The most commonly used activation function is the ReLU due to its unsaturated nonlinear function speed while the gradients of training sequence descend. The ReLU is expressed as in equation (1), where the input is the prediction activation function of the student performance represented as x,

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} (1)$$

The pooling layer in the sub-sampling process verifies and selects the performance attributes of the students at perception domain. This layer effectively lowers the scale of the output attribute by extracting the maximum represented student performance attributes. Therefore the parameters required for translation invariance maintaining and modeling is greatly reduced. Based on the operation type, it is of two types, max pooling and average pooling. The max pooling is adapted in this proposed model due its nature of improved generalization capability and fast convergence as presented by Scherer et al [24]. The complete performance of the student is summarized by the connected layer, the last layer of the AWCNN, which performs based on the CNN and the SoftMax classifier is used to classify the student attributes. Let M be the vector representing the input student performance length and N vector representing the predicted output length. Then, a fully connected layer's total number of parameters is expressed as in equation(2)

$$Q = M * N + N (2)$$

To predict the student performance, the most widely adapted model is the Extended SoftMax regression algorithm. The label sample in the prediction training student performance is presented as $(x^{(1)}, y^{(1)}), \ldots, (x^{(k)}, y^{(k)})$, where the label y holds the value 0 or 1, the $x^{(i)} \in R^{n+1}$ predicted input student performance. Equation (3), express the logistic regression function as,

$$h_\theta(x) = \frac{1}{1 + e^{(-\theta^T)x}} (3)$$

where θ, the J(θ) is the minimum loss function and its model parameter after training is represented as 'θ' and is expressed in equation (4) as,

$$J(\theta) = -\frac{1}{k} \left[\sum_{i=1}^{k} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] (4)$$

The AWCNN units consist of the comparatively smooth functions of the internal weights and input. Based on the multilayer networks, the loss function gradient computation can be estimated by back-propagation of derivative chain rule layer by layer. GA is used to update the classifier weight. This gradient can only predict limited students performance therefore SGD is used to improve the accuracy.
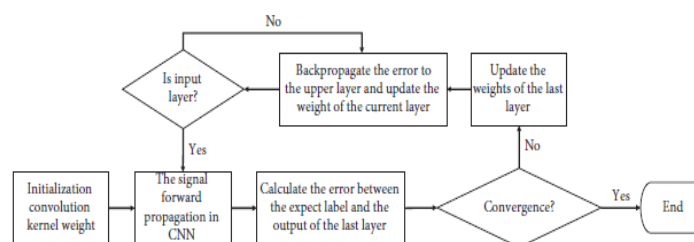
Figure 3. Awdcnn Training Process

Figure 3 shows the training process of an AWDCNN. The NN supported forward-propagation is used to extract the input performance dataset of the students and the multiple NN is used to compute the predicted student performance at the output layers. The obtained student performance through prediction is compared with the expected datasets or labels and the difference is transmitted backward layer to layer and then the corresponding filter weights are adjusted to minimize the error difference to produce the output close to the predictions after repeated iterations. This leads to the convergence at the end of AWDCNN training.

The weight update expression at the input and the output in the AWDCNN L-layer is expressed as,

$$\Delta w_{ij} = \alpha \delta_j X_i \quad (5)$$

If the last layer of the AWDCNN classifier is the L-layer, then

$$\delta_j = \left(T_j - Y_j\right) f_L'(X_i) \quad (6)$$

where the desired label is represented as T_j, the activation function reciprocal is expressed as $f_L'(X_i)$.

In case if the last layer of the AWDCNN classifier is not a L-layer, then

$$\delta_j = f_L'(X_i) \sum_{n=1}^{N_{L+1}} \delta_n \, w_{jn} \quad (7)$$

Where the no of student performance attributes and the weight between input and output is expressed as N_{L+1} and w_{jn} respectively in the L + 1 layer.

Network Optimization Strategy

The major concern in the AWDCNN classifier applications is the overfitting. To solve this issue, three methods like, weight regularization, batch normalization and dropouts are introduced [35-37].

1. In this method, certain neurons are declared not output temporarily with probability 'p' and the regular computations are performed to predict the student performance using the remaining active neurons with probability (1-p). This method leads to increased robustness and adaptability of the network model since random neurons are made inactive to predict the performance, which eventually means using different NN structure for each computation and iterations. For each iteration the weights are updated using GA.

2. Another method to reduce the effect of overfitting is to lower the values of the weights used at various levels of the layers. This in turn will reduce the computational complexity and regularize the weight distribution among the layers to predict the performance of the students. the commonly used regularization method is L2 and it adds cost in such a way that weight coefficient are squared comparing the network loss function and is expressed as,

$$J(w; X, y) = L(w; X, y) + \frac{\alpha}{2} w^T w \quad (8)$$

Where the network loss function is expressed as $L(w; X, y)$, the L2 method of regularization is $\frac{\alpha}{2} w^T w$, and the GA estimated model parameters are $w = (w_1, w_2, \ldots, w_n)$. The resultant normalised layer weights are termed as adaptive weight (aw). The weight coefficient is represented as 'α' and it is fixed to be 0.005.

The initial weights values 'w' is decided by the Genetic Algorithm(GA) and it may consider the optimal value at the initial iterations. The initial weights are said to be chromosomes and the initial set step is referred as initialize population. Here, m represents the initial population. To generate the initial weights of the model, random uniform function

is employed in the model with 6 set of weight w values ranging from 1to10.

Estimating the value of each chromosome by estimating the objective function is initiated. The objective function values are to be varied and should be maintained low and is also referred to as fitness value of the network. The fitness value computation is very important since the chromosomes which produce low fitness values are 'selected' and is allowed to propagate to the next iterations and those which produce higher fitness value will be discarded from the network models in the subsequent iterations.

The selection methods are used to select the right chromosomes to be propagated to the next layers and the most widely used GA method is the 'roulette wheel method'. Here the chromosomes which are capable of producing low fitness values are labelled with high fitness probability. Thereby, the chromosomes will be selected which has high fitness probability and this can be computed from equation (9).

$$FP = \frac{F_i}{\sum_{i=1}^{n} F_i} (9)$$

The chromosomes, in the crossover step are termed as genes. This is performed to convert the weight values into binary string form such as 0 or 1. Crossover is changing a single bit (0 or 1) or a group of string([0 1 1]) by the process of mating with the two parent chromosomes and the resultant binary string after the change is referred as 'offspring'.

The values of the strings termed as genes are altered using mutations i.e. the fitness value of 4 is [0 1 0 0] is converted to [0 1 0 1] and here the 4th bit is changed to vary the fitness value from four to five. Similarly this fitness value estimate is continued till when a chromosome achieves a fitness value equal to zero. Otherwise the iterations are continued till the optimal fitness values are obtained with new set of weight populations in network layers.

3. The normalization layer performs batch normalization using parameters such as c and β, which is of learnable network layer. The main purpose of this layer is to normalize the output of the previous layer with variance 1 and zero mean. This output is then fed into the input of the next network layer. The computation is expressed as follows(10-13) :

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x_i (10)$$
$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2 (11)$$
$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} (12)$$
$$y_i = \gamma \hat{x}_i + \beta (13)$$

The batch standardization methods provide many advantages when added after the convolutional layer, as follows,

(i) An enhanced initial learning rate is achieved due to the fast convergence.

(ii) The dependence on the network is greatly reduced in initialization of parameters.

(iii) The batch standardization, in regularization form is capable of lowering the dropouts mitigating the parameters in the overfitting scenario and can enhance the network ability to generalization.

The google Inc, proposed a network model named 'Inception', which an architecture type is supporting the AWDCNN classifier, developed from the network-in network architecture used in the early stages [38]. The common basic form is 1 x 1 convolutional layers and then a 3 x 3 layer. Here the channel-by-channel attributes and the spatial attributes are estimated independently which exhibits better optimization when compared to the traditional convolutional layers extracting the network attributes. This method will help the teachers and the students to understand their

performance in the early stages and could improve them in increasing their placement opportunities using this AWDCNN Classifier.
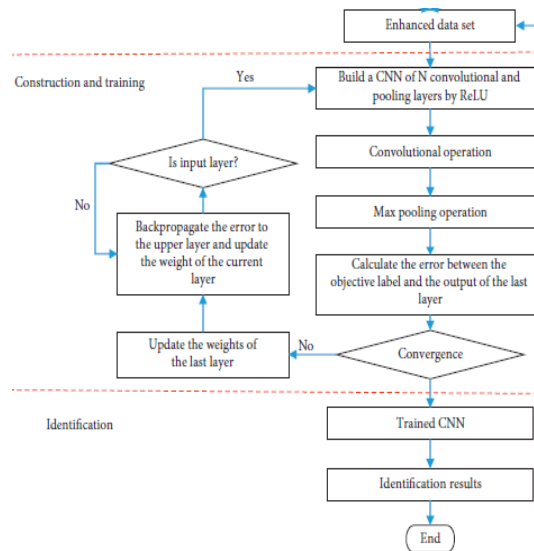
Figure 4. Flowchart Of Awdcnn Classifer

The AWDCNN classifier model training values are important in estimate the difference between the non-placement and placement students. Forward learning and back propagation method is used to attain better predictions in the AWDCNN classifiers. During the forward learning stage the N convolutional layers with the pooling layers are optimized to initialize the network parameters. In the back propagation phase, the weights are adjusted across the layers depending upon the layer output value and the expected value difference. These weights are adjusted using chain derivation rule till the difference (error gradient) becomes zero and this error gradient plays an important role in number of the iterations a network should run in order to get an normalized NN output. The regularization method is used in the layers to speed up the convergence process and to avoid overfitting. In order to improve the student performance accuracy, Inception module is used in the network.

4.   PERFORMANCE EVALUATION

The R language is used for statistical computing and handling graphical data and two different types of models use this R language to evaluate the student performance such as student-wise and subject-wise computations in reducing the missing data. This is similar to S language similar to the GNU project and is developed by Bell labs. This R language is a different implementation of S language. In order to measure the prediction model results, confusion matrix is formed with four categories: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) in Table 1 [39]. Let us assume positive 'p' and negative 'n' samples of testing set respectively. The main purpose of a classifier is to allot the samples with each class but sometimes the assigned classes may be wrong due to errors. The classifier performance is estimated by counting the TP, FP, TN and FN of a confusion matrix listed in table 1,

Table 1. Confusion Matrix

CONDITION		TRUE CLASS	
		POSITIVE	NEGATIVE
CLASSIFIER OUTPUT	POSITIVE	TP	FP
	NEGATIVE	FN	TN
	ROW SUM	p	n

The table 1 derives various performance measures as shown. Other performance measures that can be derived from the above table 1 which is used in learning are precision, recall, f-measure and accuracy respectively.

Table 2. Performance Comparison Metrics

Metrics	Prediction models(%)			
	ANFIS	SVM	DCNN	AWDCNN
Precision	85.32	86.25	88.91	91.25
Recall	82.54	87.14	89.14	92.35
F-measure	83.93	86.695	89.025	91.8
Accuracy	86.51	88.41	90.15	92.41
Error rate	13.49	11.59	9.85	7.59

Precision

It is defined as the ratio of number of true positives to the total number of false positives and true positive. It is presented as P in eq(14) as,

$$\text{Precision (P)} = TP/ (TP+FP) \quad (14)$$

Recall

It is defined as the ratio of number of true positives to the total number of false negatives and true positive. It is presented as R in eq(15) as,

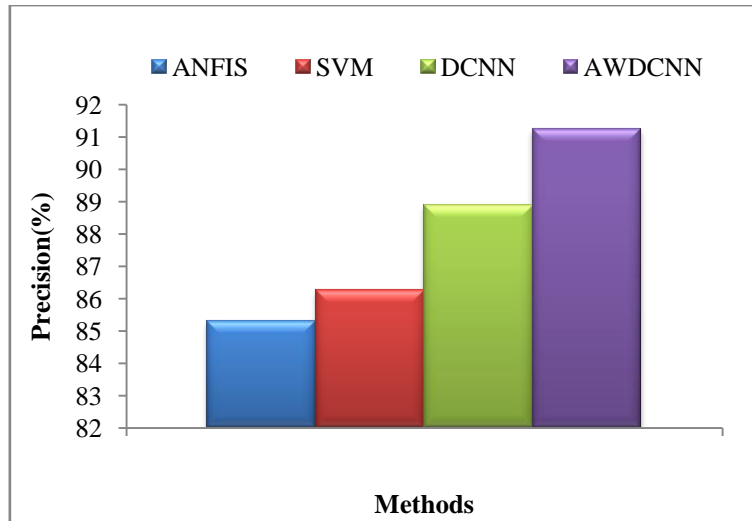$$\text{Recall (R)} = TP/ (TP+FN) \quad (15)$$

F-measure

F-measure is defined as the harmonic mean of precision and recall. It is given as,

$$\text{F-measure} = 2.(P.R)/ (P+R) \quad (16)$$

Accuracy

Accuracy is a measure of total proportion of correctly estimated instances to the set of instances as a whole.

$$\text{Accuracy} = (TP+TN)/ (TP+TN+FP+FN) \quad (17)$$

Figure 5. Precision Comparison Vs. Performance Prediction Methods

Figure 5 illustrates the outputs of various prediction models like SVM, DCNN, ANFIS and the proposed AWDCNN classifier considering the precision factor. From the presented simulated results it is evident that the newly presented classifier performs well than all the existing models with 91.25% of precision compared to ANFIS-5.93%, SVM-5% and DCNN-2.34%.
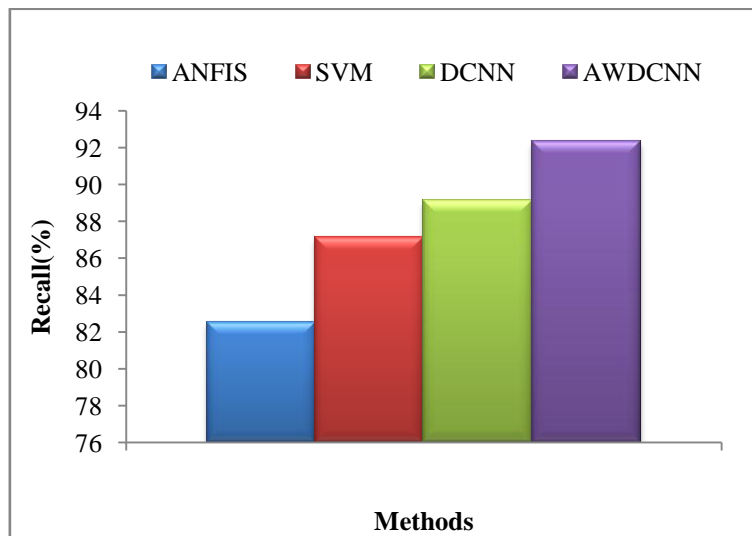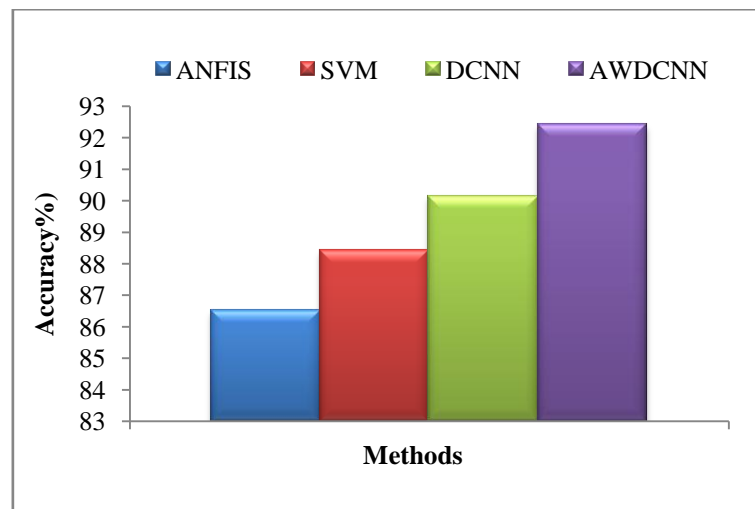

Figure 6. Recall Comparison Vs. Performance Prediction Methods

Figure 6 illustrates the output of various prediction models like SVM, DCNN, ANFIS and the proposed AWDCNN classifier considering the recall factor. From the presented simulated results it is evident that the newly presented classifier performs well than all the existing models with 92.35% of recall compared to ANFIS-9.87%, SVM-5.21% and DCNN-3.21%.
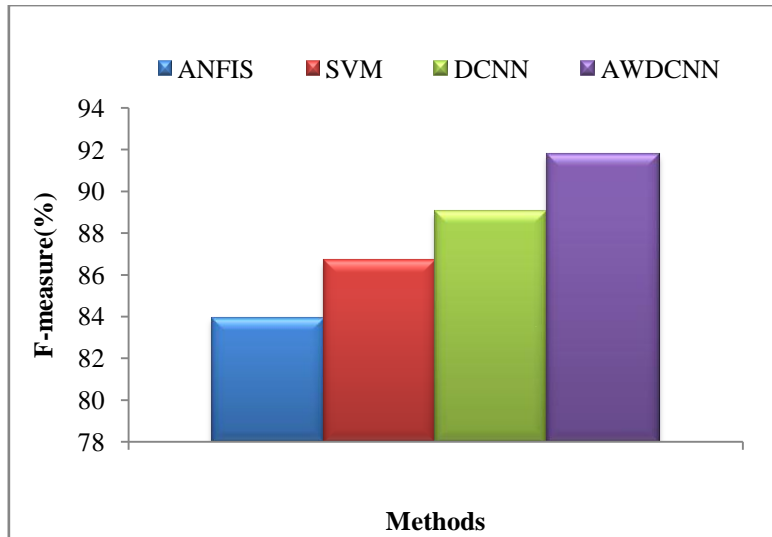
Figure 7. F-Measure Comparison Vs. Prediction Methods

The Figure 7 illustrates the outputs of various prediction models like SVM, DCNN, ANFIS and the proposed AWDCNN classifier with respect to F-measure. From the presented simulated results it is evident that the newly presented classifier performs well than all the existing models with 91.8% of F-measure values when compared to ANFIS-7.87%, SVM-5.105% and DCNN-2.775%.

Figure 8. Accuracy Comparison Vs. Prediction Methods

The Figure 8 illustrates the outputs of various prediction models like SVM, DCNN, ANFIS and the proposed AWDCNN classifier with respect to accuracy factor. From the presented simulated results it is evident that the newly presented classifier performs well than all the existing models with 92.41% of recall compared to ANFIS-5.9%, SVM-4% and DCNN-2.25%. The accuracy is increased due to the Deep Learning algorithms used in the student performance prediction model.

5. CONCLUSION AND FUTURE WORK

This work propose a Deep Learning Decision Support System (DLDSS) prediction model used to estimate the overall student performance in college placements revealing the potential of data mining with deep learning techniques. This enables the teachers to provide individual attention to the students and improve their capability. This proposes new prediction model includes four major processes:

1) Data collection and 2) Missing value data processing. 3) Data clustering and 4) DLDSS Prediction model. The Adaptive Weight Deep Convolutional Neural Network (AWDCNN) classifier is used in modeling the DLDSS prediction algorithm. An AWDCNN is used to estimate the student performance which is based on feed-forward NN producing accurate prediction values. An AWDCNN prediction model have of an raw signal handling input layer, convolutional and pooling layers with sub-sampling processes, fully connected layers, and an output SoftMax Classifier layer. The GA technique is used to perform weight regularization in the classifier and therefore it is referred as adaptive weights. These adaptive weights rather traditional fixed weights resulted in improved the student performance predictions. These models will definitely enhance the student performance, and acts as a supportive system in providing effective involvement towards the performance improvement of the students. In the future, this work can be extended to achieve more accuracy in improving the student performance prediction and acts as an effective tool for the instructor to enhance the teaching-learning process outcomes.

REFERENCES

1. Ngo, T., 2011. Data mining: practical machine learning tools and technique, by ian h. witten, eibe frank, mark a. hell. ACM SIGSOFT Software Engineering Notes, 36(5), pp.51-52.
2. Sahani, R., Rout, C., Badajena, J.C., Jena, A.K. and Das, H., 2018. Classification of intrusion detection using data mining techniques. In Progress in computing, analytics and networking (pp. 753-764). Springer, Singapore.
3. Romero, C. and Ventura, S., 2010. Educational data mining: a review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(6), pp.601-618.
4. Baker, R.S. and Inventado, P.S., 2014. Educational data mining and learning analytics. In Learning analytics (pp. 61-75). Springer, New York, NY.
5. Romero, C.; Ventura, S. Data mining in education. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2013, 3, 12–27.
6. Bakhshinategh, B.; Zaiane, O.R.; ElAtia, S.; Ipperciel, D. Educational data mining applications and tasks: A survey of the last 10 years. Educ. Inf. Technol. 2018, 23, 537–553.
7. Romero, C.; Ventura, S. Educational data mining: A review of the state of the art. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 2010, 40, 601–618.
8. Bousbia, N.; Belamri, I. Which Contribution Does EDM Provide to Computer-Based Learning Environments? In Educational Data Mining; Springer: Basel, Switzerland, 2014; pp. 3–28.
9. Romero, C.; Ventura, S. Educational data science in massive open online courses. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2017, 7, e1187.
10. Livieris, I.E., Mikropoulos, T.A. and Pintelas, P., 2016. A decision support system for predicting students' performance. Themes in Science and Technology Education, 9(1), pp.43-57.
11. Aziz, A.A., Ismail, N.H. and Ahmad, F., 2013. Mining Students' academic Performance. Journal of Theoretical & Applied Information Technology, 53(3), pp.485-495.
12. Algarni, A., 2016. Data mining in education. International Journal of Advanced Computer Science and Applications, 7(6), pp.456-461.
13. J. Whitehill, K. Mohan, D. Seaton, Y. Rosen, and D. Tingley. Delving deeper into mooc student dropout prediction. arXiv preprint arXiv:1702.06404, 2017.
14. F. Mi and D.-Y. Yeung. Temporal models for predicting student dropout in massive open online courses. In Proceedings of 15th IEEE International Conference on Data Mining Workshop (ICDMW 2015), pages 256–263, Atlantic City, New Jersey, 2015.

15. W. Wang, H. Yu, and C. Miao. Deep model for dropout prediction in moocs. In Proceedings of the 2nd International Conference on Crowd Science and Engineering (ICCSE 2017), pp. 26–32, Beijing, China, 2017.
16. Hegazi, M.O. and Abugroon, M.A., 2016. The state of the art on educational data mining in higher education. International Journal of Computer Trends and Technology, 31(1), pp.46-56..
17. Romero, C., Ventura, S., Pechenizkiy, S., & Baker, M. (2010). Handbook of Educational Data Mining. London: Chapman & Hall.
18. Kotsiantis, S. (2012). Use of machine learning techniques for educational proposes: a decision support system for forecasting students" grades. Artificial Intelligence Review, 37, 331–344.
19. Pandey, M. & Taruna, S. (2014). A multi-level classification model pertaining to the student"s academic performance prediction. International Journal of Advances in Engineering & Technology, 7(4), pp.13–29.
20. Sikder, M.F., Uddin, M.J. and Halder, S., 2016, Predicting students yearly performance using neural network: A case study of BSMRSTU. 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp. 524-529.
21. Hidayah, I. A. E. Permanasari, N. Ratwastuti, Student classification for academic performance prediction using neuro fuzzy in a conventional classroom, International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 221–225,2013.
22. Gray G., C. McGuinness, P. Owende, An application of classification models to predict learner progression in tertiary education, IEEE International Advance Computing Conference (IACC), pp. 549–554,2014.
23. Guo, B.; Zhang, R.; Xu, G.; Shi, C.; Yang, L. Predicting students performance in educational data mining. In Proceedings of the 2015 International Symposium on Educational Technology (ISET), Wuhan, China, 27–29 July 2015.
24. Asif, R.; Merceron, A.; Ali, S.A.; Haider, N.G. Analyzing undergraduate students' performance using educational data mining. Comput. Educ. 2017, 113, 177–194.
25. Turabieh, H., 2019, Hybrid Machine Learning Classifiers to Predict Student Performance. In 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS) ,pp. 1-6.
26. Xu, J.; Moon, K.H.; van der Schaar, M. A machine learning approach for tracking and predicting student performance in degree programs. IEEE J. Sel. Top. Signal Process. 2017, 11, 742–753.
27. Ma, Y., Zong, J., Cui, C., Zhang, C., Yang, Q. and Yin, Y., 2019, Dual Path Convolutional Neural Network for Student Performance Prediction. In International Conference on Web Information Systems Engineering (pp. 133-146). Springer, Cham.
28. Mondal, A. and Mukherjee, J., 2018. An Approach to predict a student's academic performance using Recurrent Neural Network (RNN). Int. J. Comput. Appl, 181(6), pp.1-5.
29. Kim, B.H., Vizitei, E. and Ganapathi, V., 2018. GritNet: Student performance prediction with deep learning. arXiv preprint arXiv:1804.07405.
30. Crawford, S. L. (2006). Correlation and regression. Circulation, 114(19), 2083-2088.
31. Kavipriya T., and Dr.P. Krishnapriya, "Data Pre-Processing to Fill the Missing Data Using Statistical Based Mean with Correlation Coefficient (SMCC)", Jour of Adv Research in Dynamical & Control Systems, 15-Special Issue, October 2017, pp.187-192.
32. T. Wiatowski and H. B¨olcskei, "A mathematical theory of deep convolutional neural networks for feature extraction," IEEE Transactions on Information Aeory, vol. 64, no. 3, pp. 1845–1866, 2017.
33. S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," vol. 1502, 2015, https://arxiv.org/abs/1502.03167.

34. D. Scherer, A. M¨uller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in Artificial Neural Networks—ICANN 2010, Springer, Berlin, Heidelberg, 2010.

35. N. Srivastava, G. Hinton, A. Krizhevsky et al., "Dropout: a simple way to prevent neural networks from overfitting," Ae Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

36. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," vol. 1607, 2016, https://arxiv.org/abs/1607.06450.

37. T. Van Laarhoven, "L2 regularization versus batch and weight normalization," vol. 1706, 2017, https://arxiv.org/abs/1706. 05350.

38. C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, Boston, MA, USA, October 2015.

39. Powers, David M W (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" (PDF). Journal of Machine Learning Technologies. 2 (1): 37–63.