Journal of Modern Technology and Engineering
Vol.9, No.2, 2024, pp.130-139
https://doi.org/10.62476/jmte9130

A MODEL PROPOSAL WITH DEEP LEARNING: ON STUDENT SUCCESS

Emrah Emirtekin*

Ege University, Izmir, Türkiye

Abstract. This study aims to investigate predicting student academic performance using interaction data from online learning environments and course and demographic information from student information systems using deep learning methods. Deep learning enables high-precision prediction in large and complex datasets, enabling more accurate and reliable student prediction. In this context, modelling student performance based on data from online learning environments and student information systems not only identifies students at risk of non-attendance, but also allows prediction of potential failures by the end of the semester, enabling timely intervention at an early stage in the process. This model can be used to grade student performance and provide feedback on an individual or group basis. In addition, it will make a significant contribution to tutors managing online courses by enabling them to monitor students' overall performance levels and organise learning processes more effectively. The dataset used in the study includes interaction data from 4,470 students enrolled in various online courses at Yaşar University during the autumn semester of 2019-2020, as well as course and demographic information from the student information system. The deep learning model developed in the study achieved 99.81% classification accuracy on training data and 98.36% on test data, predicting students' final grades.

Keywords: deep learning, machine learning, student success.

Corresponding author: Emrah, Emirtekin, Ege University, Izmir, Türkiye, e-mail: *emrah.emirtekin@ege.edu.tr*

Received: 12 April 2024; Revised: 24 May 2024; Accepted: 3 July 2024; Published: 30 August 2024.

1 Introduction

In today's world, the rapid advancement and development of technology has had a significant impact in many areas. One such area, whose importance has been further highlighted by the pandemic, is distance learning. Distance learning allows learners to participate in their courses online or in a blended format, regardless of time or location. These online courses are typically delivered through a Learning Management System (LMS). LMS platforms are widely used systems for presenting course content according to a specific plan, facilitating communication and interaction between students and teachers, and managing assessment and evaluation processes. They also serve as systems for recording and reporting data related to student activity in the course.

Various sources use the recorded data to determine academic performance, to structure future plans, to predict how students will perform in specific courses, and to identify factors affecting success or professional trends. Observing students' academic or course-related performance with these data is crucial for improving and designing the educational process (Quadri & Kalyankar, 2010).

How to cite (APA): Emirtekin, E. (2024). A model proposal with deep learning: On student success. *Journal of Modern Technology and Engineering*, *9*(2), 130-139 https://doi.org/10.62476/jmte9130

In order to predict student success, both machine learning (ML) and deep learning (DL) methods are currently widely used. DL models have gained importance due to their ability to uncover complex structures in high-dimensional data, where other models struggle, and their applicability in many domains (LeCun et al., 2015). As such, they have been widely used in various domains such as speech recognition, computer vision, and natural language processing (Amodei et al., 2016; Kendall & Gal, 2017; Young et al., 2018). The widespread use of DL models has been influenced by the growth and storage of data, as well as advances in technologies and algorithms for data processing (Najafabadi et al., 2015).

The literature contains numerous national studies on predicting students' academic success. ML models are predominantly used in these studies. However, fewer national studies use DL methods to predict student success in online courses. This study is important for creating a generalizable and dynamic model with higher performance scores than existing studies on online courses. A model that provides feedback on students' course success throughout the semester will contribute to students' self-assessment and enable teachers to monitor students' educational process individually or collectively.

The aim of this study is to develop a DL model for predicting students' end-of-semester course success based on course activities recorded in the LMS and demographic and course-related data from the student information system (SIS) for a given semester.

2 Literature

2.1 Deep Learning

ML can be defined as the software modeling of systems that learn on their own by making meaningful inferences from data or experiences through mathematical and statistical operations (Alpaydin, 2020). DL, on the other hand, is considered an advanced approach to Artificial Neural Networks (ANNs), which are a subset of machine learning, involving multiple layers (Deng & Yu, 2014). Many DL architectures are based on ANNs.

Today, various DL architectures are utilized, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Belief Networks (DBN), and Deep Autoencoders (LeCun et al., 2015). Different learning strategies are employed in training these models. These strategies include supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, the network is provided with information about what the output should be for a given input. The input is fed repeatedly into the network to adjust the weights between layers, thus training the network to determine appropriate weights. In unsupervised learning, the network does not receive information about the output for a given input. Instead, each input is assigned to a cluster in the output layer, effectively categorizing the input into clusters. Reinforcement learning also does not provide specific information about the output for a given input. Instead, the network receives feedback in the form of rewards for correct outputs and penalties for incorrect ones, guiding the training process (Sasakawa et al., 2008).

In DL, feature extraction and transformation involve numerous operations between layers in the network. Each layer processes the output from the previous layer and prepares it as input for the next layer. The operations performed in the layers of DL architectures are outlined below.

2.1.1 Input Layer

This is the first layer of the network, where raw features are provided to the network. The number of features input into the network affects the model's performance, training time, and memory requirements.

2.1.2 Convolution Layer

This layer is responsible for applying convolution operations to the input features, extracting important features from the raw data. The number of filters and their configurations in this layer impact the model's performance, training time, and memory usage.

2.1.3 ReLU Layer

This is the activation layer, also known as the rectified linear unit (ReLU) layer. Data is passed through an activation function, commonly $f(x) = max(0, x)$, to introduce non-linearity and enable faster learning in the network.

2.1.4 Pooling Layer

The pooling layer reduces the dimensionality of the input from the convolution layer, which helps decrease computational load, prevents overfitting, and reduces memory usage. Pooling is performed using filters that apply maximum or average pooling, with maximum pooling often being preferred for its effectiveness.

2.1.5 Fully Connected Layer

In this layer, neurons are fully connected to all neurons in the subsequent layer. Each neuron in this layer is connected to every neuron in the next layer, which is why it is called the fully connected layer.

2.1.6 DropOut Layer

The dropout layer is used to prevent overfitting in the network by randomly dropping neurons during training. This helps the model generalize better by avoiding over-reliance on specific neurons.

2.1.7 Normalization Layer

This layer plays a crucial role in significantly impacting the network's performance. It helps standardize the data by normalizing the variance among features to a certain range.

2.1.8 Classification Layer

The classification layer is used for classifying data into different categories. The output values in this layer correspond to the number of classes to be predicted. It usually produces probabilistic values, with the softmax classifier often being preferred for its superior performance in many cases.

2.2 Related Works

Studies utilizing deep learning techniques for predicting student success in the literature are summarized below:

(Özkan et al., 2018) conducted a study to predict student success using artificial intelligence-based learning methods. They used the "student performance" dataset from the UCI ML Repository, which includes 33 attributes for 395 students. The dataset was randomly divided into 90% training and 10% testing data. The study employed various algorithms, including C5, Boosted C5, Regression Trees, DVM, Logistic Regression (LR), RO, and DL algorithms. The DL model had an architecture with three hidden layers, each containing 256 neurons, and used ReLU as the activation function. The accuracy rates for classification models were: C5.0 at

86%, Boosted C5 at 82%, DVM at 78%, Logistic Regression at 82%, RO at 84%, and DL at 87%.

In another study by (Altun et al., 2019), multiple linear regression analysis and Artificial Neural Networks (ANN) were used to predict the graduation grades of 578 students in the Elementary School Teaching Department. The attributes considered were gender, marital status, age, and midterm exam scores from the first semester. The mean absolute error values were 94.3% for multiple linear regression and 94.43% for ANN. The ANN model comprised 10 input variables, two hidden layers (four neurons in the first layer, three neurons in the second layer), and one output layer. The sigmoid activation function was used, with a learning rate of 0.02, which provided the lowest error rate. Cross-validation was applied during both training and testing phases.

The study by (Nabil et al., 2021) aimed to predict students' academic success at an early stage using Educational Data Mining (EDM). The study specifically highlighted the high failure rates in courses such as "Programming" and "Data Structures". Predictive models using deep neural networks (DNN) and other machine learning models were developed using data from a public university. The DNN model proved to be the most effective in predicting student success and identifying students at risk of failure at an early stage, with an accuracy rate of 89%.

In the study conducted by (Giannakas et al., 2021), a Deep Neural Network (DNN) classification framework with two hidden layers was proposed to predict team-based academic performance in the field of software engineering at an early stage. The performance of the model was evaluated using different activation functions (Sigmoid, ReLU, Tanh) and optimization algorithms (Adagrad, Adadelta). Based on more than 30,000 data points collected from 74 teams, this framework was analyzed using the SHAP method to identify the most significant features of the model. The results showed that prediction accuracy was 76.73% with Adadelta and 82.39% with Adagrad. The overall learning performance of the model was calculated as 86.57%.

(Shoaib et al., 2024) developed an Artificial Intelligence Student Success Predictor to address the limitations of Campus Management Systems (CMS). This system is capable of predicting grades, student risks, and retention or dropout rates. The study implemented an ensemble model comprising Convolutional Neural Networks (CNN) and various classifiers (SVM, Random Forest, KNN) using a standardized dataset created from different data sources. The model achieved 93% accuracy for grade prediction, 93% for risk prediction, and 92% for retention prediction. This AI system aims to enhance academic decision-making processes by integrating with existing CMS infrastructures.

3 Method

3.1 Dataset

This section provides information about the dataset used in the study.

The data set used in this research is from (Emirtekin et al., 2020) study, which focuses on using ML techniques to predict students' success in online courses.

The dataset used in this study includes the interactions of 4,470 students enrolled in online courses at Yaşar University in the autumn semester of 2019-2020, recorded in the LMS and demographic information from the Student Information System (SIS). This dataset consists of a total of 17 features, including 11 numerical and 6 categorical at-tributes. The target feature for prediction is 'letterGrade', which includes 11 different grades (A, A-, B+, B, B-, C+, C, C-, D+, D, F). Details of the other features are given in Table 1. The dataset is split so that 85% is used to train the model and 15% is reserved for testing.

Table 1: Information about features

Source	Features	Type	Description
LMS	assignment	numeric	Homework submission
LMS	lesson	numeric	Visiting course pages
LMS	resource	numeric	Download education materials
LMS	syllabus	numeric	Visiting the syllabus
LMS	testANDquiz	numeric	Test and quiz
LMS	visit	numeric	Visit count
LMS	video	numeric	Watching course videos
SIS	courseCode	categorical	Course Code
SIS	courseLanguage	categorical	Course Language
SIS	gender	numeric	Gender
SIS	age	numeric	Age
SIS	faculty	categorical	Faculty
SIS	departman	categorical	Department
SIS	class	categorical	Class
SIS	letterGrade	categorical	Final letter grade
SIS	etcs	numeric	Work load
SIS	takeTCourse	numeric	The number of courses taken in term

3.2 Data Pre-Processing

In this section, the preprocessing stages of the features used in the DL model are outlined as follows:

Data Integration: Data obtained from different sources (such as LMS and SIS) have been integrated into a single dataset.

Missing Data Imputation: Missing values in numerical features have been completed using the mean values of the respective features.

Categorical Data Processing: Data in categorical features have been labeled with the "other" tag.

Normal Distribution and Scaling: Achieving normal distribution of data can affect the performance of certain algorithms. If the data is skewed to the right or left, model performance may be negatively impacted. Therefore, standardization has been applied to ensure normal distribution of numerical features and scaling of data has been performed.

Categorical Feature Transformation: DL models cannot work directly with categorical data; thus, categorical features were first encoded into numerical values using Label Encoder, and then represented in binary format through One Hot Encoding. As a result of these processes, the dataset's 17 features were expanded to 80 features.

Output Feature Processing: The output feature, letterGrade, which has multiple classes, was also encoded and rep-resented in binary format to be processed in the output layer.

Dataset Splitting: The dataset used in the DL model was divided into training and testing sets. The model was trained with the training data and evaluated with the test data. 15% of the dataset was allocated for testing, and 85% for training.

Stratified Sampling: Balanced representation of classes in both training and test datasets was ensured by applying stratified sampling.

3.3 Deep Learning Model (DLM)

This section provides information about the DLM developed within the scope of the research.

In a neural network with multiple layers, there are interconnected chains of neurons. Neural networks consisting of many hidden layers, along with input and output layers, are referred to as deep neural networks, and the process of training such a model is known as DL. In the development of the DLM, the Python programming language and the Keras library were utilized. Keras is a library for artificial neural networks developed in Python.

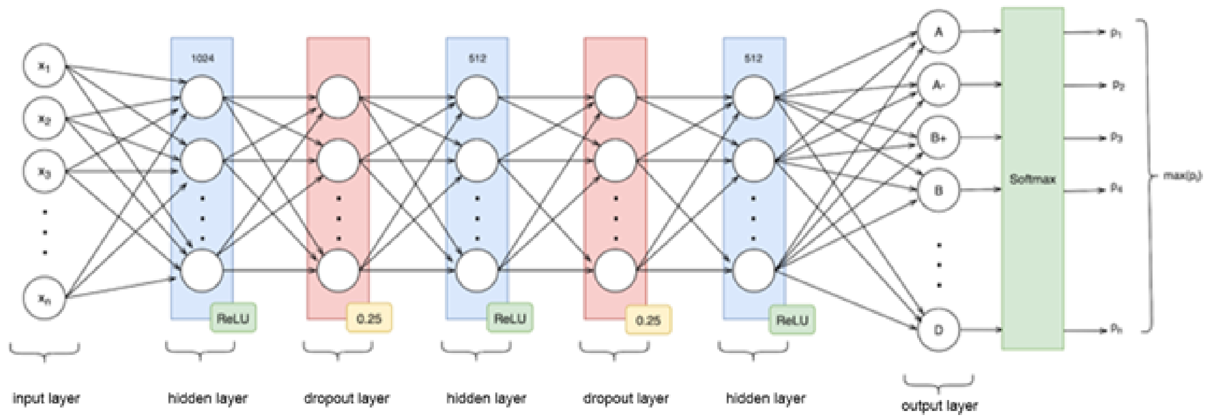


Figure 1: Deep Learning Model

```
model = Sequential()
model.add(Dense(1024, input_dim=79, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(512, activation='relu'))
model.add(Dense(11, activation='softmax'))
```

In the development of the model, the Sequential model from the Keras library was used. The DLM consists of an input layer, three hidden layers, two dropout layers, and an output layer. The input layer has 79 inputs, the first hidden layer has 1024 neurons, and the second and third hidden layers each have 512 neurons. To prevent overfitting in the neural network, dropout layers have been added after the first and second hidden layers, reducing the number of neurons in these layers by 25%. Finally, the output layer contains 11 neurons corresponding to the grades (A, A-, B+, B, B-, C+, C, C-, D+, D, F). The ReLU activation function is used in the hidden layers, while the Softmax function is applied in the output layer.

The training process of the model was initiated by presenting the first instance of the dataset to the network. Each input value is multiplied by the randomly assigned weights in the model to produce a new value. These values are then passed through an activation function to generate an output. This output value is used as the input for the next neuron. By computing all output values, the forward propagation process of the network is completed. Following this, the backpropagation process begins. The goal of backpropagation is to optimize the weights to minimize the overall error, thereby improving the network's learning. One complete pass of forward and backward propagation is referred to as an epoch. The number of inputs to be trained in one epoch is determined by the batch size. In this model, the number of epochs is set to 32, and the batch size is set to 5. The categorical crossentropy function is used as the loss function due to the multiple classes. The AdaDelta algorithm is chosen for optimizing the training process, as it helps to prevent the learning rate from decreasing too quickly.

4 Findings

This section presents the success status of the DLM and the metrics used for evaluating this success. The metrics include accuracy, precision, recall, F1 score, ROC Curve (Receiver Operating Characteristic Curve), and AUC (Area Under Curve). Additionally, the classification performance of the DLM is illustrated with graphs.

The evaluation of the DLM's success relies on the Confusion Matrix (CM). The CM (Table

2) is created by calculating the correctly classified inputs and the incorrectly classified inputs. In the CM, the rows represent the true classes of the examples in the test set, while the columns indicate the classification predictions made by the DLM.

Table 2. Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

4.1 Evaluation Metrics

4.1.1 Accuracy

Accuracy is calculated by dividing the number of correctly classified examples $(TN + TP)$ by the total number of examples $(FN + TN + TP + FP)$.

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FN + FP}$$

4.1.2 Precision

Precision is obtained by dividing the number of correctly classified positive instances (TP) by the number of instances predicted as positive $(FP + TP)$.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4.1.3 Recall

Recall is obtained by dividing the number of correctly classified positive instances (TP) by the total number of instances that are actually positive $(FN + TP)$.

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.1.4 F-Measure

There is an inverse relationship between precision and recall; increasing one can decrease the other. To achieve a balance between both metrics and obtain more accurate and sensitive results, the $F1$ score, which is the harmonic mean of precision and recall, is used.

$$\text{F-Measure} = \frac{2 * Recall * Precision}{Recall * Precision}$$

4.1.5 ROC-Curve

Accuracy, precision and recall may not always give accurate results on some datasets. Such datasets often contain unbalanced data. For datasets with this type of imbalance, evaluation metrics such as F1 score and ROC curve are used to better assess model performance.

4.1.6 Area Under Curve (AUC)

AUC, known as the Area Under the ROC Curve, measures the classification performance of a model. The size of the area under the curve is directly proportional to the classification success of the DLM. The larger the area, the higher the model's performance.

4.2 Model Performance

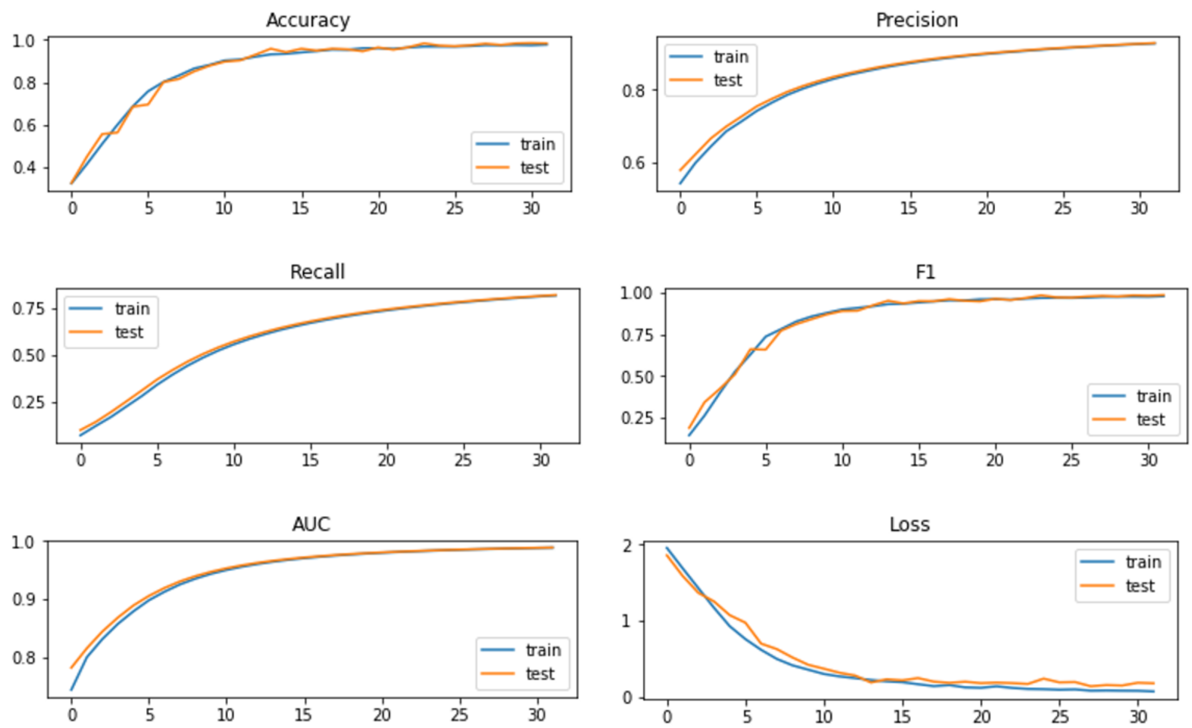
This section presents the values for the performance metrics of the DLM.

Table 3. Performance Metrics

Model	Accuracy	Precision	Recall	F1-Measures	AUC
DLM	0.983	0.928	0.824	0.985	0.989

5 Visualisation of Performance Metrics

This section visually presents the performance metrics (accuracy, precision, recall, $F1$ score, and AUC) and error values during the training and testing phases of the network.



6 Conclusion

In this study, a DLM was developed to classify end-of-term online course performance of students using a dataset comprising their activities from LMS and demographic and course information from SIS. The model achieved an accuracy of 98.3% in classifying students' end-of-term performance. The other performance metrics (Table 3) are as follows: precision: 92.8%, recall: 82.4%, $F1$ score: 98.5, and AUC: 98.9.

Previous studies have observed that accuracy alone does not always provide reliable results in evaluating the performance of DLM and ML models (Huang & Ling, 2005). Relying solely on accuracy to assess the performance of a DLM or ML model is insufficient. Therefore, other performance metrics are used to evaluate models. For models developed on imbalanced datasets, it is recommended to use the AUC metric, which represents the area under the ROC Curve, to evaluate model performance (Wu & Flach, 2005). The AUC metric for the DLM classified student performance with an accuracy of 98.9%, indicating high model performance.

When comparing the developed DLM with national studies in the literature, it was found that the developed model demonstrated higher performance in classifying students' success levels in terms of accuracy. Since the dataset used in this study is general and accessible, utilizing the features from this dataset (Table 1), the developed DLM, and the hyperparameters used in the model would be suitable for educational institutions to monitor students' individual or collective performance, plan for learning deficiencies, and enable self-assessment by students.

References

Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.

Altun, M., Kayıkçı, K., & Irmak, S. (2019). Sınıf Öğretmenliği Öğrencilerinin Mezuniyet Notlarının Regresyon Analizi ve Yapay Sinir Ağları Yöntemleriyle Tahmini/Estimation of Graduation Grades of Primary Education Students by Using Regression Analysis and Artificial Neural Networks. *e-Uluslararası Eğitim Araştırmaları Dergisi*, *10*(3), 29-43.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... & Chen, J. (2016, June). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International Conference on Machine Learning* (pp. 173-182).

Deng, L., Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, *7*(3–4), 197-387.

Emirtekin, E., Karatay, M., & Kışla, T. (2020). Online Course Success Prediction of Students with Machine Learning Methods. *Journal of Modern Technology & Engineering*, *5*(3).

Giannakas, F., Troussas, C., Voyiatzis, I., & Sgouropoulou, C. (2021). A deep learning classification framework for early prediction of team-based academic performance. *Applied Soft Computing*, *106*, 107355.

Huang, J., Ling, C.X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299-310.

Kendall, A., Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision?. In *Advances in Neural Information Processing Systems* (pp. 5574-5584).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Nabil, A., Seyam, M., & Abou-Elfetouh, A. (2021). Prediction of students' academic performance based on courses' grades using deep neural networks. *IEEE Access*, *9*, 140731-140746.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1.

Özkan, Y., Koçoğlu, F.Ö., & Erol, Ç.S. (2018). Prediction of Student Performance By Deep Learning Algorithm Öğrenci Performansının Yapay Zeka Derin Öğrenme Algoritması ile Öngörülmesi. *Preface Of the Editors*, 136.

Quadri, M. M., & Kalyankar, N. V. (2010). Drop out feature of student data for academic performance using decision tree techniques. *Global Journal of Computer Science and Technology.*

Sasakawa, T., Hu, J., & Hirasawa, K. (2008). A brainlike learning system with supervised, unsupervised, and reinforcement learning. *Electrical Engineering in Japan*, *162*(1), 32-39.

Shoaib, M., Sayed, N., Singh, J., Shafi, J., Khan, S., & Ali, F. (2024). AI student success predictor: Enhancing personalized learning in campus management systems. *Computers in Human Behavior*, *158*, 108301.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55-75.

Wu, S., Flach, P. (2005, August). A scored AUC metric for classifier evaluation and selection. In *Second Workshop on ROC Analysis in ML*, Bonn, Germany.