IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 7, 2020, accepted December 27, 2020, date of publication January 5, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049446

Predicting at-Risk Students at Different Percentages of Course Length for Early Intervention Using Machine Learning Models

MUHAMMAD ADNAN[1], ASAD HABIB[1], JAWAD ASHRAF[1], SHAFAQ MUSSADIQ[1], ARSALAN ALI RAZA[2], MUHAMMAD ABID[1], MARYAM BASHIR[1], AND SANA ULLAH KHAN[1]

[1]Institute of Computing, Kohat University of Science and Technology, Kohat 26000, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Islamabad 45550, Pakistan

Corresponding author: Muhammad Adnan (adnan@kust.edu.pk)

ABSTRACT Online learning platforms such as Massive Open Online Course (MOOC), Virtual Learning Environments (VLEs), and Learning Management Systems (LMS) facilitate thousands or even millions of students to learn according to their interests without spatial and temporal constraints. Besides many advantages, online learning platforms face several challenges such as students' lack of interest, high dropouts, low engagement, students' self-regulated behavior, and compelling students to take responsibility for settings their own goals. In this study, we propose a predictive model that analyzes the problems faced by at-risk students, subsequently, facilitating instructors for timely intervention to persuade students to increase their study engagements and improve their study performance. The predictive model is trained and tested using various machine learning (ML) and deep learning (DL) algorithms to characterize the learning behavior of students according to their study variables. The performance of various ML algorithms is compared by using accuracy, precision, support, and f-score. The ML algorithm that gives the best result in terms of accuracy, precision, recall, support, and f-score metric is ultimately selected for creating the predictive model at different percentages of course length. The predictive model can help instructors in identifying at-risk students early in the course for timely intervention thus avoiding student dropouts. Our results showed that students' assessment scores, engagement intensity i.e. clickstream data, and time-dependent variables are important factors in online learning. The experimental results revealed that the predictive model trained using Random Forest (RF) gives the best results with averaged precision = 0.60%, 0.79%, 0.84%, 0.88%, 0.90%, 0.92%, averaged recall = 0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91%, averaged F-score = 0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91%, and average accuracy = 0.59%, 0.79%, 0.84%, 0.88%, 0.90%, 0.91% at 0%, 20%, 40%, 60%, 80% and 100% of course length.

INDEX TERMS Predictive model, earliest possible prediction, at-risk students, machine learning, feed-forward neural network, random forest, early intervention.

I. INTRODUCTION

Rapid innovations in the design and development of online learning platforms such as Massive Open Online Course (MOOC), Virtual Learning Environments (VLEs), and Learning Management System (LMS) not only have overcome the limitations of space and time but have also made access to education easy and affordable. Evaluating and analyzing the students' data generated from online learning platforms can help instructors to understand and monitor students learning progress. The earlier the students' performance is detected in the VLEs, the better it is for the instructor to persuade and warn students for keeping them on the right track. Earlier studies report that the students learning variables stored in the database records can help instructors in predicting the performance of students in the future [1]–[3]. But predicting students' performance earlier in the course would be more helpful as compared to predicting students'

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

performance once they have completed the course and have given the final exam. However, developing a predictive model that can identify the exact learning behavior of students earlier in the course by analyzing their behavior data is a challenging task. In an online learning environment, where a large amount of data is generated every day, machine learning (ML) techniques could help in analyzing the variables that define the students and come up with the results that better describe their learning behavior, thus, ML may reveal information that is beneficial for both instructors and students [4], [5].

Identifying students at-risk of dropout and failure as early as possible during a course could help the instructors to execute timely and necessary interventions/persuasions to help students to remain steady during their studies [6].

Generally, in traditional classroom settings and online learning settings, a general approach is followed where the same guidelines are defined for all students ignoring individual discontentment. To provide personalized feedback and support right from the start of the semester, VLE designers require the development of a predictive model that makes rapid decisions about how and when to intervene students for support. Educational Data Mining (EDM) tools, techniques, and products have progressed significantly, helping educators to make education easy and effective [7]. However, these techniques lack in identifying at-risk students earlier in the course timeline, compelling instructors to perform significant manual work for students problem identification to keep them on track.

The emergence of Artificial Intelligence (AI), machine learning (ML), and deep learning (DL) techniques have facilitated and enabled researchers to develop series of predictive models to reveal hidden study patterns which explain the strength and weakness of online students [8], [9]. To reduce the dropout rates, researchers can use ML techniques to study different variables that significantly affect the students' dropout. Predictive models powers by ML techniques can present the accurate picture of students that are likely to quit their study thus facilitating instructors to come up with preventive measures before dropout behavior occurs. The prime objective of our research study is the earliest possible identification of students who are at-risk of dropouts by leveraging ML techniques to understand variables associated with the learning behavior of students and how they interact with the VLE.

By analyzing Open University Learning Analytics (OULA) dataset, it was observed that students are inconsistent in their online learning activities throughout course weeks resulting in high dropouts at the end of the course. Based on our observations, we developed a predictive model that can identify students at-risk of dropout at the very start of the course. The predictive model is capable of facilitating instructors to intervene students through persuasive messages that encourage students to keep themselves on the right track thus avoiding dropouts. The contribution of this study include:

- Developing and evaluating predictive models using several ML/DL algorithms to predict students' performance scores.
- Earliest possible identification of students in VLE who are at-risk of dropout during the course.
- Integrating personalized feedbacks with a predictive model to help instructors in interventing students at an optimal time.
- Discussing various persuasion techniques that may help students in increasing their study performance.

The rest of the paper is organized as follows: Section II presents background and related work. Section III is about the Open University Learning Analytics (OULA) dataset, selected for predicting at-risk students at a different percentage of course length. Section IV presents details about the experimental setup for training various predictive models. Section V discusses the experimental results. Various techniques related to interventing students through persuasion is discussed in section VI. Conclusion, limitations, and future work are presented in section VII.

II. BACKGROUND AND RELATED WORK

A. EDUCATIONAL DATA MINING (EDM)

EDM is an emerging discipline aimed at using statistical and ML methods to analyze large repository of educational data for a better understanding of student's behavior patterns and the learning environments. Several EDM studies have been carried out that leverages ML techniques to discover variables that significantly influence students' performance, dropout, engagement, and interaction in online learning platforms. Most of these studies target analyzing variables that are generated from students' online activities [10]–[12] whereas some studies also use demographics variables to observe their effect on students' study behavior [13], [14]. Earlier the prime variables to be considered for analysis were study time, study duration, learning content type, and variables derived from social interaction activities. As the online learning platform became more stable and interactive, variables such as assessment scores, assignment scores, clickstream, online forum interaction, and location were added in the analysis process [15]. Identification of significant variables becomes a challenge for researchers due to diversity in LMS, VLEs, MOOCs, courses offered, and course activity types.

In literature, the majority of studies target collecting variables and predicting students' performance at the end of the course. The results obtained from those studies were useful in identifying the significant variables that influence the students' performance the most; however, no solutions were provided to prevent students from dropout and failure. On the other hand, online learning platforms generate enormous data associated with student interaction, clickstreams, and courses, etc right from the start of the course. A comprehensive predictive model can be developed by analyzing variables data right from the start of the course that would be effective in preventing failures/dropouts and facilitating instructors to make effective interventions at the optimal time.

A study carried out by [16] implemented four ML algorithms for the early identification of students who have a high possibility of failure. The results revealed that the Support Vector Machine (SVM) was the most effective algorithm in the earlier identification of students with 83% accuracy. The study also indicated that the process of data preprocessing is very essential in increasing the performance of ML algorithms. Results of previous studies showed that the development of predictive models is possible earlier in the course however many challenges limit their application to a specific learning platform. One of the major challenges for researchers is making predictive models flexible capable of adjusting/adapting in different learning environments. Major reasons that are limiting predictive models to become flexible, general, and transferrable is because of the different course structures, instruction designs, and online platforms [17].

In the last few years, research studies used both statistical and predictive models to find insights in a large repository of data both formal and informal educational settings [13], [18], [19]. For example, several research studies [20]–[23] investigated the role of demographics variables towards contributing to successful learning performance or students' retention. A study carried out by [24] investigated and compared more than 120 different datasets related to economics and business undergraduate students' demographics and interaction behavior in online settings. The study observed the influence of a variety of variables such as educational background, clickstream data, assessment scores, entry test scores, and learning personality data on students' performance. While most of the studies targets discovering the impact of key variables on students performance, there are other studies [25]–[27] that encourage early intervention, informed support and timely feedback to guide at-risk students. Numerous studies carried out at Open University, UK [28], [29] tried to identify at-risk students by using several predictor variables. The studies mapped the study behavior of students to predict 1) low performance (when performance falls below a threshold value) 2) whether the students are successful or unsuccessful at the end of the course. The studies also indicated that demographic variables used along with students' behavior variables provided improved predictive models in terms of performance and accuracy.

Lee and Choi [30] carried out a study in which they tried to identify relevant variables that compel students to decide to quit the course. The variable responsible for student dropouts were classified into three categories. 1) students' demographic variables such as gender, background, relevant experience, skills, psychological attributes, earlier education, etc 2) variable related to course structure and requirement such as the number of assessments, institutional support, interaction, difficulty level, time duration, etc 3) environments/context factors such as technology used, location, external noise, work environment, home environment, etc.

A Time-series clustering approach was used by [31] for the earliest possible identification of at-risk students taking online courses. As compared to traditional aggregation approaches, the time-series clustering approach generated predictive models with higher accuracy. Sofer and Cohen used various learning analytics techniques on engagement variables generated from online courses to find their impact on students' performance achievements [32]. The results indicated that students achieving good academic positions have a higher engagement percentage compared to the students achieving lesser academic positions. As an example, students who completed the course were twice as much interactive as students who did not complete it.

A study carried out by [33] argued that clickstream variables representing students' online engagement are more accurate, objective, and comprehensive than self-reported data in measuring students' learning behavior. The clickstream data is more reliable as it is collected in an authentic learning environment while the learning behavior is happening as compared to self-reported data which is often generated from decontextualized and unreliable students' memories. Moreover, clickstream data are unobtrusive and did not require students full attention as they can be collected seamlessly without interrupting students learning process [34]. Finally, intuitively collected clickstream data can provide large-scale and timely measures of students learning behavior which could assist instructors in knowing about the students' online engagements daily.

Recently, a large number of research studies have been undertaken to analyze clickstream data generated from online learning platforms (MOOCs, LMS, VLE) to measure students' online engagements. While most of the studies try to explore the relationship between clickstream data and students' online engagements, limited studies have taken a step further to facilitate instructors in when and how to intervene students at the optimal time e.g., [35]–[38]

Shivangi Gupta and A. Sai Sabitha in their research study attempted to decipher those variables that are responsible for student retention in e-learning [39]. Decision Tree (DT) algorithm was used to determine the significant features to help MOOC learners and designers in improving course content, course design, and delivery. Several data mining techniques were applied to three MOOCs datasets to analyze the online students' in-course behavior. Finally, the authors claimed that the models they used could be useful in predicting significant features to minimize the attrition rate.

Akçapınar Gökhan developed an early warning system that used students' eBook reading data to predict students at-risk of academic failure [40]. To develop the best predictive model, 13 ML algorithms were used to train the model using data from different weeks of the semester. The best predictive model was selected according to the accuracy/Kappa metric and recommending optimal time for the instructors to intervene. The study revealed that all predictive models improved their performance results when more and more weekly data was used during the training process. The early warning system predictive models were successful in classifying low and high-performance students with an accuracy of 79% starting from the 3^{rd} week. When complete 15 weeks data were provided to various algorithms, Random Forest (RF) outperformed other algorithms whereas with the transformed data J48 outperformed all other algorithms. Moreover, upon using categorical data, the Naïve Bayes (NB) showed better performance.

Foretelling students' performance early in the course is a challenging problem in online learning environments due to diversity in course structure and MOOCs design. While the popularity of LMS/MOOCs is increasing rapidly, there is a need for an automated intervention system that can provide timely feedback to students. To integrate automated intervention system with LMS/MOOCs, researchers have implemented various ML algorithms that can support instructors in providing informed assistance to students during the learning process. ML algorithms such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees (DT), and Random Forest (RF), etc are commonly trained using daily, weekly, or monthly students log data to find students' learning patterns. Deep learning (DL) algorithms are now also used in creating predictive models as they can process raw data directly. Kőrösi, Gábor and Richard Farkas used the Recurrent Neural Network (RNN) algorithm trained on raw log student records to predict students' learning performance at the end of the course [41]. The results showed the dominance of RNN in providing superior performance as compared to standard baseline methods.

Alberto C. and John D. L. used multi-view genetic programming approach to develop classification rules to analyze students learning behavior, predict their academic performance, and trigger alerts at the optimal time to encourage at-risk students to improve their study performance [42]. The genetic programming technique was employed as it works nicely with multi-view learning. The prediction model learned and evolved is directly explainable without further adjustment. Moreover, while using genetic programming approach results in the natural evolution of classification rules with time as more and more data is available i.e. the classification rules evolved with the availability of new data. The early warning system built with comprehensible Genetic Programming classification rules specifically targets underperforming and underrepresented students. Comprehensible feedbacks are provided to students, instructors, and administration staff using three interfaces to provide timely support to students to keep them on the right track. The main drawback of this study was that the author did not mention explicitly the various semester stages at which the performance metrics such as accuracy, sensitivity, specificity, and Kappa, etc were calculated using multi-view genetic programming algorithm along with other machine learning algorithms.

A research study carried out by [43] used a ML algorithm called logistic regression to pinpoint students that are expected to drop out in an e-learning course. The algorithm uses students' historical grades as an input to model students' performance. As compared to the feed-forward neural network (FFNN), Support Vector Machine (SVM), a system for educational data mining (SEDM), and Probabilistic Ensemble Simplified Fuzzy ARTMAP - Adaptive Resonance Theory Mapping PESFAM techniques, the proposed drop out detection technique exhibited higher performance score in terms of precision, recall, specificity, and accuracy. The tutoring action plan based on logistic regression was able to reduce the dropout rate by 14%.

Lara J.A. *et al.*, proposed the use of knowledge discovery in databases (KDD) to extract information that might help teachers in knowing about the interaction of students with e-learning systems [44]. The proposed technique builds historical reference models of students that can be used to classify students in dropout or non-dropout classes. The proposed system called System for Educational Data Mining (SEDM) analyzes two groups of students for a single course i.e. dropout students who are not able to sit in the final examination and non-dropout students who are passed the course assessments and can sit in the final examination. SEDM was able to generate study patterns for the two groups that can be very helpful for instructors to improve students' study performance. Table 1 presents summary of the recent research studies that uses ML/DL techniques for creating predictive models, performing classification according to students' performance, dropouts prediction and performing early intervention

The studies discussed aforementioned are related to avoiding students' dropout, interventing students at the optimal time, predicting students at-risk of dropout, and classifying students into different performance groups, however, none of these studies predicts students at-risk of dropout at a different percentage of course length. The proposed predictive model could help educational institutions and instructors in the earliest possible identification of at-risk students thereby intervening students through proper persuasive techniques to encourage them to be on track and improve their study performance.

III. DATA DESCRIPTION

We used a freely available Open University Learning Analytics Dataset (OULAD), provided by the Open University, UK. Students' data is spread across 7 tables each containing students centered information such as students' demographics, students' Virtual Learning Environment (VLE) interaction, assessments, course registration, and courses offered. Tables relate to each other through key identifiers. Students' daily activities and VLE interaction are represented as clickstreams data (number of clicks) stored in the student VLE table. Students' assessment scores are stored in a dataset triplet called student-module-presentation. The OULAD was generated for the year 2013 and 2014 containing 7 courses, 22 module-presentations with 32,593 registered students. OULAD is freely accessible at https://analyse.kmi.open.ac.uk/open_dataset and has been certified by the Open Data Institute http://theodi.org/ [63].

TABLE 1. Summary of research studies using ML/DL techniques for student dropout and performance prediction along with the particular algorithms used, their performance, problem addressed and limitations.

Research Studies related to the use of ML/DL techniques in creating predictive models	Algorithm Used and Performance achieved	Problem addressed	Limitations
Chung, J. Y. and Lee, S. [6]	Random Forest (RF) with 95% accuracy	Students binary classification	Predictive model suffers from potential inaccuracy in calculating the weights of the features
Gray, C. C. and Perkins, D. [45]	1-Nearest-neighbor with 97% accuracy	Identifying possible failing students at week 3 of the fall semester	The predictive model was stable, only applicable to Bangor university students
Al-Shabandar, R. et al. [46]	Gradient Boosting Model with 95% accuracy	Identification of at-risk students with intensive earlier intervention in online courses	Temporal features were not considered. The predictive model was not validated with additional datasets
Lee, S. and Chung, J. Y. et al. [47]	Random Forest, boosted decision trees (BDT) with BDT having the highest accuracy of 99%	Improving the performance of dropout prediction model using the ML-based early warning system	A limited NEIS database was used. All features were not included in creating a predictive model.
Behr, A. et al. [48]	Random Forest with AUC (area under the curve) of 0.86	Binary classification, modeling student's dropout	Students' satisfaction (wishes and needs) features were not considered.
Martins, L. C. B. et al. [49]	Gradient Boosting Machine, Deep Learning, Distributed Random Forest. The deep learning model achieved the highest True Positive Rate of 71.1%	Earliest possible prediction of students' attrition	Data about the first semester of the study was not included.
Hussain, M. et al. [4]	Artificial neural network (ANN) and Support vector machine (SVM) achieved the highest accuracy of 75%	Predicting students' difficulties in online learning	Students' dropout prediction was not performed. Model accuracy was low.
Mduma, N. et al. [50]	Linear regression with a ROC score of .88	Identification of at-risk students using a machine learning method	The under-sampling approach with a penalized model was not used.
Figueroa-Cañas, J. and Sancho-Vinuesa [51]	Decision Trees (DT) with more than 90% accuracy	Identifying dropout-prone students earlier in online statistical course	The methodology used both the training set and validation set of the students enrolled in the same academic year
Ortigosa, A. et al. [52]	C5.0 algorithm with more than 85% accuracy	Identification of real-life challenges of the early dropout prevention system	Comprehensive real-world performance of the prediction model and the effectiveness of retention action is needed.
Imran, A. S. et al. [53]	A feed-forward deep neural network with accuracy >90%	predicting and explaining student dropout	Prediction is done after the course completion
Wu, Nannan et al. [54]	CLMS-Net. Combination of Convolutional Neural Network, Long Short-Term Memory network, and Support Vector Machine. Accuracy = 91.55%	Predicting dropout in MOOCs	The validity of the predictive model on more datasets are needed
Rosé, Carolyn P et al. [55]	Traditional machine learning models. Machine learning models do not generalize well	Understanding why ML models alone are not the solution	ML models are not interpretable and actionable
Mubarak, A. A. et al. [56]	LSTM, ANN, SVM, Logistic Regression. LSTM with highest accuracy of 93%	Using a deep neural network to predict learning analytics in MOOCs courses videos	The current model only employs' learners interaction patterns with videos. A complete learning activity pattern of learners is missing.
Liao, S. N. et al. [57]	Support Vector Machine with AUC = .70	Identifying students at-risk of performing poorly in courses	The study did not predict at-risk students earlier in the course.
Sekeroglu, B et al. [58]	Support Vector Regression (SVR), LSTM, SVM, Gradient Boosting Classifier (GBC), ANN. ANN with the highest accuracy of 87.78%.	Students performance classification and prediction using ML techniques	The study does not address the earliest possible identification of at-risk students. The dataset size was very small
Mao, Ye [59]	Bayesian Knowledge Tracing (BKT), Intervention-BKT (IBKT), LSTM. LSTM and LSTM+SK achieved the highest accuracy of 74%	Developing students model for intervention	The model hypermeters were tuned manually. Moreover, intervention techniques were introduced after the course completion
Iqbal, Z et al. [60]	Collaborative Filtering (CF), Matrix Factorization (MF), and Restricted Boltzmann Machines (RBM) techniques. RBM showed better performance having RMSE = 0.3, MSE = 0.09, MAE = 0.23	Grade prediction of students using ML techniques.	Features related to students' motivation were not included. A limited dataset was used. The early prediction was not rendered.
Fwa, H. L. and Marshall, L. [61]	Hidden Markov Model (HMM) Leave-One-Out Cross-Validation (LOOCV)	Modeling programming students using unsupervised ML techniques	Techniques related to student engagement were not used and implemented. The early prediction was not rendered.
Xu, Jie et al. [62]	Linear Regression Logistic Regression Random Forest kNN EPP. Ensemble-based Progressive Prediction showed the best result having the lowest mean square error	Tracking and predicting students performance using ML techniques	Courses prediction to the students was not carried out. No intervention technique was discussed.

When properly analyzed and modeled, OULAD can provide a very suitable platform for an early forecast of at-risk students.

A. DATA PREPROCESSING

To enhance the performance efficiency of the predictive models, all missing variables instances in the form of nulls, or noise were removed or replaced by their mean values from the OULAD. As an example, the date values were missing in the assessments table which represents the date the assessments were taken/submitted. As the date is an important variable in the early prediction of at-risk students, all the date instances having N/A, null, or missing values were replaced by the date mean value.

TABLE 2. Student-Assessment-Clickstream triplet.

Students Demographics	Students relative assessment performance score during the different length of the course modules	Late assignment/assessments during the different length of the course modules	Sum of clicks per course module during the different length of the course modules	Average clicks per course module during the different length of the course modules	Final Performance
code_module, code_presentation, id_student, gender, region, highest_education, imd_band, age_band, num_of_prev_attempts, studied_credits, disability	AS20, AS40, AS60, AS80, AS100	LS20,LS40, LS60, LS80, LS100	SC20, SC40, SC60, SC80, SC100	AC20, AC40, AC60, AC80, AC100	Final result having values Withdrawn, Fail, Pass, Distinction

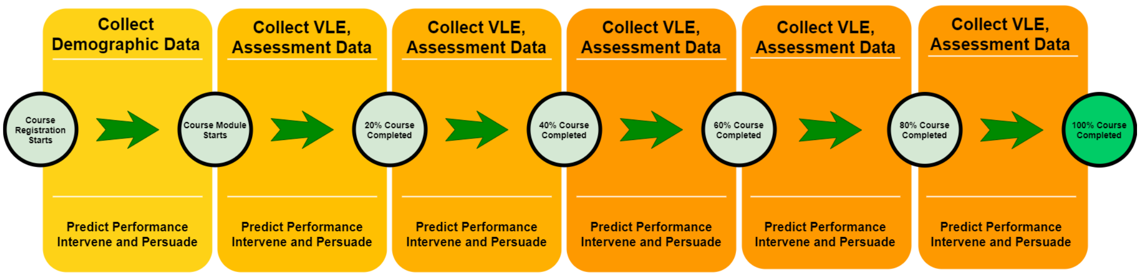


FIGURE 1. Predicting and intervening at-risk students at different percentages of the course length.

B. FEATURE ENGINEERING

For the earliest possible prediction of students' performance, we divided the course length into 5 parts i.e. 20%, 40%, 60%, 80%, and 100% of course completed. We also assumed that demographic data solely can also be used to predict students' upcoming performance in assessments and final exams. The students' future performance prediction was determined by modeling the predictive models using only demographics data, using demographics and 20% course completion data, demographics, and 40% course completion data, and so on. To predict students' performance at different times of course module, several new variables were created from the existing variables. Relative Score (RS) variables were created to represent student relative performance at 20%, 40%, 60%, 80%, and 100% of course module completion (RS20, RS40, RS60, RS80, RS100). Variables indicating the number of late submissions were created when 20%, 40%, 60%, 80%, and 100% of the course module was completed (LS20, LS40, LS60, LS80, LS100). Variables representing the raw assessment scores were also created at 20%, 40%, 60%, 80%, and 100% of the course module completion (AS20%, AS40%, AS60%, AS80%, AS100%). Variables representing students' VLE interaction in the form of clickstreams were created for the different percentage of course module length. Two types of variables namely sum_clicks and mean_clicks were created to indicate the sum of clicks and average clicks at 20%, 40%, 60%, 80%, and 100% of the course module completion (SC20%, SC40%, SC60%, SC80%, SC100%, AC20%, AC40%, AC60%, AC80%, AC100%). Students' demographics table was merged with the students' assessment table to get demographics and assessment data in one table. Moreover, the VLE information i.e. students clickstream data was also merged with demographics data to know students' interaction with VLE learning contents during a course module. More information about the triplet student-assessment-clickstream table is detailed in table 2. The triplet student-assessment-clickstream table contains 31 columns with 32593 rows.

IV. EXPERIMENTAL SETUP FOR PREDICTIVE MODELING

For predicting the performance of at-risk students at a different percentage of the course length, the variables about students' demographics, VLE interaction, and assessments were used. This workflow is shown in figure 1 where the length of the course is divided into 6 periods i.e. course starts, 20%, 40%, 60%, 80%, and 100% of the course studied.

Six ML algorithms and one DL algorithm were selected for training/testing the predictive models during different stages of the course. For modeling Open University Learning Analytics Dataset (OULAD), these algorithms were designated for classifying students' performance into four categories i.e. Withdrawn (students who were not able to complete the course), Fail (students who completed the course but were not able to secure passing marks), Pass (completed courses with a passing score), Distinction (completed courses with excellent grades). The Python 3.7.8 scripts were used for the construction of predictive models. The Python libraries used were TensorFlow, Keras, sklearn, numpy, and seaborn.

A. EVALUATION CRITERIA

Before training and testing predictive models at various stages of course module length, the dataset at hand was split into training and testing set using K-fold cross-validation technique where the value of k was set to 10. By using the k-fold cross-validation technique, the dataset is divided into k sets where k-1 sets are used for model training and the remaining

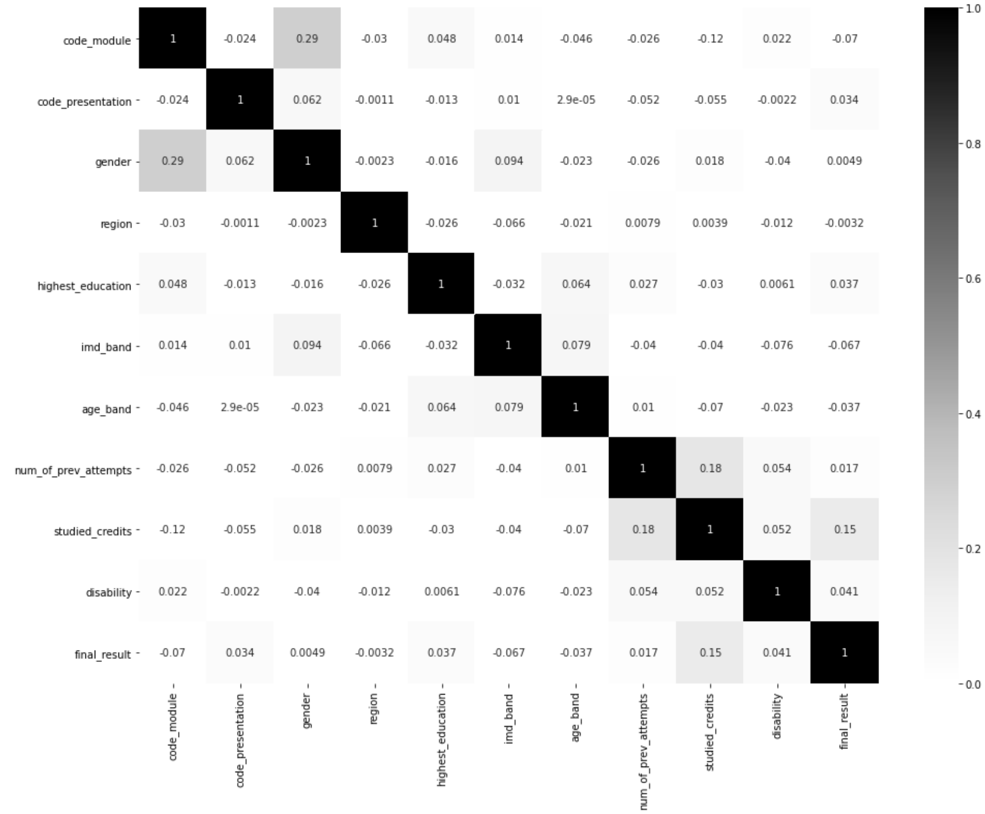


FIGURE 2. Heatmap showing correlation between demographics variables and final result.

1 set is using for model testing i.e. measuring the prediction performance of the model on the unseen data not used during the model training. The metrics selected for measuring the performance of predictive models include the following:

1) ACCURACY

The accuracy is calculated by dividing the number of correct classes predicted by the total number of classes i.e. accuracy = (True Positives + True Negatives)/all.

2) PRECISION

Determines the fraction of true positives among true positives and false positives predicted i.e. precision = true positives/(true positives + false positives). If a small percentage of students (1%) are withdrawing or failing the course, we could build a predictive model that always accurately predicts whether students are getting withdrawn, fail, pass, or distinction. This predictive model would be 99% accurate but 0% useful and reliable.

3) RECALL

Recall ensures that the predictive model is not overlooking a few VLE students who are getting Withdrawn, Fail, Pass, or Distinction grades. Suppose if only 1% of students are getting the Distinction position and the rest are getting Withdrawn, Fail, or Pass grades, then the predictive model would correctly predict Withdrawn, Fail, or Pass grades with 99% accuracy. The predictive model will have an accuracy of 99% and students having a Distinction position will likely be categorized among Withdrawn, Fail, and Pass students. Recall ensures that we are not overlooking those 1% of students having a Distinction position. Recall = TP/(TP + FN).

4) F-SCORE

Determines the harmonic mean of recall and precision of a predictive model. F1 measure is good for classification problems where the target labels are imbalanced. F-measure = 2 * (Precision * Recall)/(Precision + Recall).

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. PHASE 1: USING ONLY DEMOGRAPHICS DATA FOR CONSTRUCTING PREDICTIVE MODELS

Before training the 7 predictive models using only demographics data, a heatmap was constructed to know the demographics variables correlations with the final result. As shown in figure 2, we noticed that there is no significant positive or negative correlation between demographics variables and the final result. Only studied_credtis and num_of_prev_attempts have a weak positive correlation between them. All the demographics variables were considered for training RF, SVM, K-NN, ET, AdaBoost classifier, Gradient boosting classifier, and ANN models.

TABLE 3. Performance score of the 7 predictive models when trained only on demographics data.

Precision	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Distinction	0.1531	0.0383	0.1154	0.1479	0.0136	0.0122
Fail	0.2439	0.1786	0.2839	0.2484	0.1042	0.1120
Pass	0.4586	0.6135	0.5051	0.4604	0.7078	0.7162
Withdrawn	0.4291	0.2595	0.3299	0.3928	0.4884	0.4914
Averaged	0.3820	0.7527	0.3844	0.3709	0.5968	0.6017
Recall	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.1691	0.0376	0.1512	0.1666	0.3151	0.3482
Fail	0.2725	0.2679	0.2641	0.2633	0.3843	0.3815
Pass	0.4372	0.4444	0.4193	0.4285	0.4448	0.4498
Withdrawn	0.4122	0.1789	0.4286	0.3999	0.4554	0.4649
Averaged	0.3745	0.3541	0.3664	0.3644	0.4442	0.4500
F-score	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.1605	0.0177	0.1307	0.1565	0.0260	0.0236
Fail	0.2574	0.1507	0.2736	0.2556	0.1639	0.1731
Pass	0.4475	0.4255	0.4580	0.4438	0.5462	0.5525
Withdrawn	0.4202	0.1585	0.3724	0.3961	0.4709	0.4775
Averaged	0.3777	0.4641	0.3716	0.3672	0.4967	0.5020
Accuracy	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.1691	0.0757	0.1510	0.1665	0.3306	0.3557
Fail	0.2724	0.2764	0.2641	0.2633	0.3838	0.3809
Pass	0.4370	0.4063	0.4194	0.4284	0.4448	0.4497
Withdrawn	0.4121	0.3270	0.4283	0.4000	0.4551	0.4646
Averaged	0.3745	0.3541	0.3664	0.3644	0.4442	0.4500

Table 3 presents the results of 7 predictive models when trained only on the demographics data using K-fold cross-validation techniques where the value of k was set to 10. The final result variable was set as the target variable which the predictive models will try to predict whereas all other demographics variables acted as input the predictive models. The values of precision, recall, accuracy, and f-score of the predictive models for various positions of students' final results when trained only on the demographics data indicates the very low performance of predictive models. Moreover, the performance of all predictive models for the Fail position is very bad. In early intervention systems, where identification of at-risk students is vital, the performance results of predictive models for Fail students become more crucial as such the students at-risks can be intervened earlier in the course for improving study behavior.

B. PHASE II: USING DEMOGRAPHICS AND CLICKSTREAM DATA FOR CONSTRUCTING PREDICTIVE MODELS

To improve the performance of predictive models, clickstream data (students' interaction with VLE in the form of numbers of clicks during the course timeline) along with demographics was considered to train the predictive models. From the heatmap in figure 3, we noticed that the correlation of the final result with all other variables is similar and there is still no significant negative/positive correlation of demographics and clickstream variables with the final result. The correlation between sum_clicks100 and mean_clicks100 is moderate but still far away from significant. Next, we considered all the demographics and clickstream variables for training and testing the predictive models.

Table 4 shows the performance results of the predictive models developed using demographics and clickstream data. In the case of Pass class, the RF, ET, AdaBoost, and Gradient Boost classifiers are showing satisfactory results whereas, for the Fail and Distinction class, the performance scores are still very low. Though the performance scores of the predictive models are better than when trained only on demographics data, but much farther from being acceptable.

C. PHASE III: CONSIDERING DEMOGRAPHICS, CLICKSTREAM, AND ASSESSMENT SCORES FOR DEVELOPING PREDICTIVE MODELS

The assessment scores were further added to demographics and clickstream for possible predictive model performance improvement. From the heatmap in figure 4, we observed a moderate correlation between assessment scores and the final result. Interestingly the average score variable (AS100) has a moderate negative correlation with the final result which implies an increase in the final result score when the average assessment score decreases. As expected, a strong positive correlation (.87) was noted between the average score and the relative score variable. Moreover, the mean clicks (MC100) and sum clicks (SC100) variables were also having a significant positive correlation with the assessment score variables. Finally, a weak correlation was observed between the late submission variable (LS100) and the final result. Table 5 presents the performance scores of the predictive models when trained on the demographics, clickstream, and assessment data. A substantial improvement was noted in the performance of predictive models for Pass, Withdrawn, Distinction, and Fail classes when assessment data was added for constructing the predictive models. The performance results of SVM and K-NN models were still very low with the accuracy of .32 and .38.

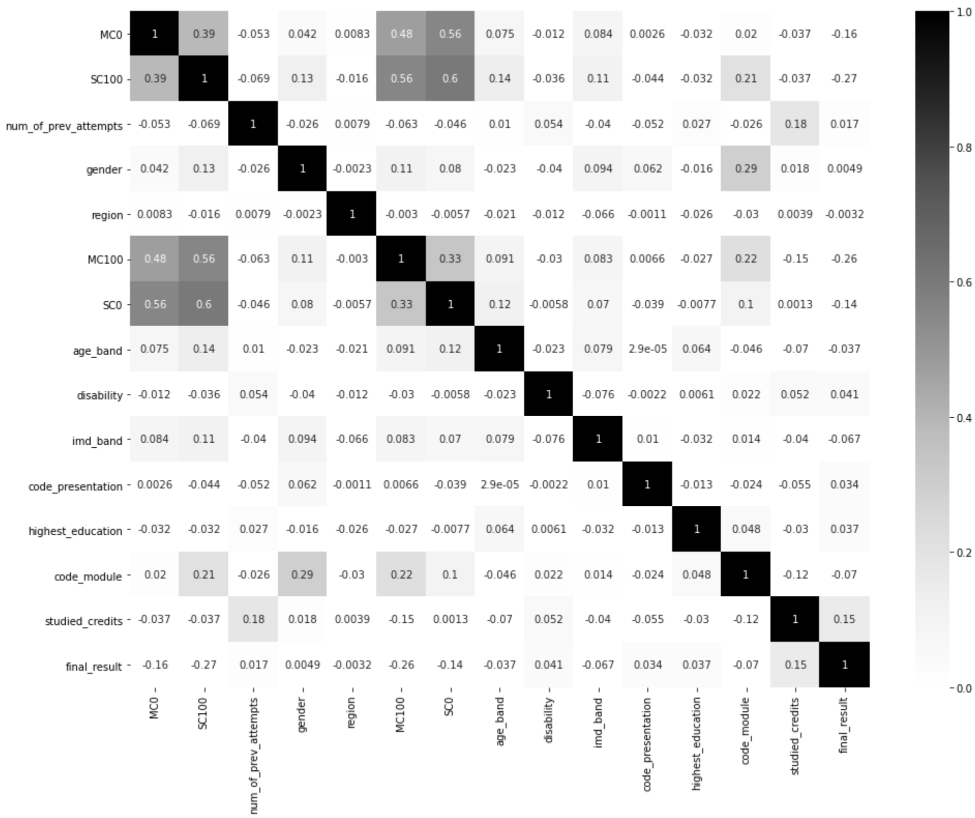


FIGURE 3. Heatmap showing correlation between demographics plus clickstream variables and final result.

TABLE 4. Performance score of the 7 predictive models when trained demographics plus clickstream data.

Precision	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Distinction	0.0602	0.1222	0.1440	0.0792	0.0760	0.0674
Fail	0.2759	0.4993	0.3313	0.2884	0.2479	0.2863
Pass	0.9085	0.2440	0.6962	0.8690	0.8846	0.9133
Withdrawn	0.7182	0.5654	0.6039	0.7004	0.7337	0.7396
Averaged	0.7611	0.7154	0.5607	0.7229	0.7457	0.7727
Recall	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.4770	0.0663	0.2045	0.3935	0.4560	0.5071
Fail	0.4840	0.3318	0.3524	0.4605	0.4485	0.5192
Pass	0.6244	0.3012	0.5820	0.6178	0.6338	0.6320
Withdrawn	0.7147	0.6411	0.6780	0.6987	0.6819	0.7189
Averaged	0.6336	0.3871	0.5373	0.6175	0.6247	0.6451
F-score	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.1068	0.0795	0.1689	0.1317	0.1300	0.1187
Fail	0.3514	0.2918	0.3414	0.3546	0.3186	0.3689
Pass	0.7401	0.2374	0.6339	0.7221	0.7384	0.7470
Withdrawn	0.7164	0.5159	0.6387	0.6995	0.7064	0.7290
Averaged	0.6773	0.4528	0.5455	0.6542	0.6681	0.6886
Accuracy	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.4778	0.1880	0.2054	0.3937	0.4590	0.5215
Fail	0.4840	0.2526	0.3526	0.4605	0.4462	0.5191
Pass	0.6244	0.5661	0.5818	0.6178	0.6338	0.6321
Withdrawn	0.7147	0.5036	0.6779	0.6986	0.6819	0.7188
Averaged	0.6336	0.3872	0.5373	0.6175	0.6247	0.6451

D. FEATURE ENGINEERING

To further improve the performance results, a merge operation was performed where Distinction-Pass classes were combined into Pass class and Withdrawn-Fail classes were merged into Fail class as these classes are of the same types and portray similar information. The goal of performing the feature engineering technique was to improve the performance of the predictive models especially for the Fail class as

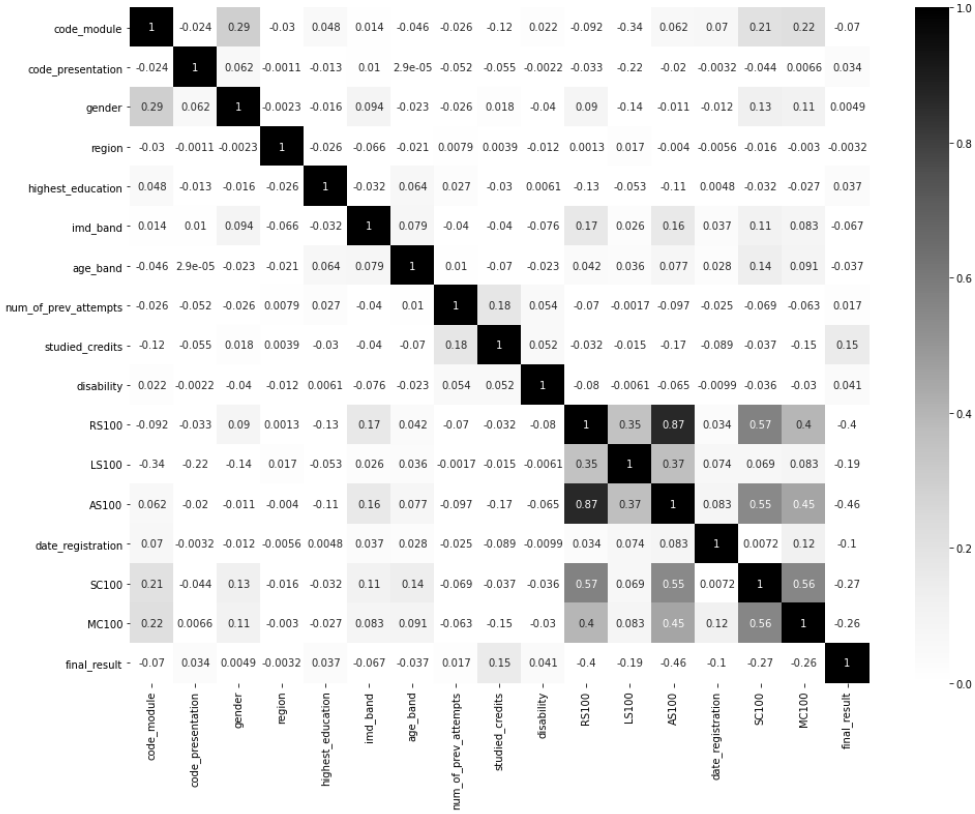


FIGURE 4. Heatmap showing correlation between demographics, clickstream, and assessment variables with final result.

TABLE 5. Performance score of the 7 predictive models when trained demographics, clickstream, and assessment variables.

Precision	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Distinction	0.4589	0.0505	0.2242	0.4818	0.4512	0.5091
Fail	0.3691	0.4907	0.3680	0.3780	0.3517	0.3693
Pass	0.9027	0.4426	0.8333	0.8978	0.8434	0.8967
Withdrawn	0.8304	0.6754	0.7553	0.8148	0.7171	0.8409
Averaged	0.7757	0.7382	0.6942	0.7660	0.7012	0.7788
Recall	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.6586	0.0406	0.3848	0.6528	0.5355	0.6668
Fail	0.5807	0.3396	0.4568	0.5647	0.4575	0.5938
Pass	0.7570	0.6008	0.7064	0.7614	0.7456	0.7651
Withdrawn	0.7492	0.7351	0.7258	0.7479	0.6980	0.7483
Averaged	0.7235	0.4888	0.6518	0.7210	0.6620	0.7292
F-score	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.5403	0.0450	0.2828	0.5541	0.4893	0.5772
Fail	0.4511	0.3045	0.4075	0.4527	0.3855	0.4553
Pass	0.8234	0.3972	0.7646	0.8240	0.7914	0.8256
Withdrawn	0.7876	0.6852	0.7402	0.7799	0.7015	0.7919
Averaged	0.7415	0.5441	0.6685	0.736936	0.6760	0.7464
Accuracy	**RandomForest**	**SupportVectorMachine**	**KNearestNeighbor**	**ExtraTreeClassifier**	**AdaBoostClassifier**	**GradientBoosting**
Distinction	0.6579	0.3252	0.3842	0.6527	0.5363	0.6675
Fail	0.5800	0.2485	0.4568	0.5648	0.4425	0.5936
Pass	0.7570	0.6460	0.7064	0.7615	0.7455	0.7651
Withdrawn	0.7493	0.7043	0.7259	0.7480	0.6989	0.7484
Averaged	0.7235	0.4888	0.6518	0.7210	0.6620	0.7292

the students belonging to the Fail class are at-risk and need informed guidance. Table 6 a decent increase in the predictive model performances after performing feature engineering. On average all the predictive models achieved greater than 80% performance score for precision, recall, F-score, and accuracy. Overall, RF outperformed all other baseline models whereas the SVM showed the lowest performance. The performance scores of GradientBoosting, AdaBoost, and

TABLE 6. Predictive models performance after performing feature engineering.

Precision	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Fail	0.8943	0.8220	0.8710	0.8899	0.8902	0.8931
Pass	0.9493	0.7916	0.9254	0.9516	0.9396	0.9457
Averaged	0.9220	0.9032	0.8985	0.9212	0.9150	0.9195
Recall	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Fail	0.9517	0.8823	0.9288	0.9536	0.9427	0.9484
Pass	0.8892	0.8330	0.8651	0.8854	0.8844	0.8877
Averaged	0.9202	0.8079	0.8966	0.9190	0.9135	0.9179
F-score	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Fail	0.9221	0.8266	0.8990	0.9206	0.9157	0.9199
Pass	0.9182	0.7382	0.8942	0.9172	0.9111	0.9157
Averaged	0.9201	0.8307	0.8966	0.9189	0.9134	0.9178
Accuracy	RandomForest	SupportVectorMachine	KNearestNeighbor	ExtraTreeClassifier	AdaBoostClassifier	GradientBoosting
Fail	0.9517	0.8155	0.9288	0.9536	0.9427	0.9484
Pass	0.8892	0.7993	0.8651	0.8854	0.8844	0.8877
Averaged	0.9202	0.8079	0.8966	0.9190	0.9135	0.9179

ExtraTree classifiers were almost similar and closer to the RF. Ultimately, the RF classifier was selected for training and testing the predictive model for the different duration (0%, 20%, 40%, 60%, 80%, 100%) of the course module.

E. PHASE IV: TRAINING PREDICTIVE MODEL USING DEEP FEED FORWARD NEURAL NETWORK (DFFNN)

After training the predictive models using traditional ML methods, next, we used a deep learning technique called Deep Feed Forward Neural Network (DFFNN) to train the predictive model using OULAD. DL techniques differ from the techniques used in traditional ML algorithms in that they structure algorithms in layers that can learn and make sensible judgments on their own. DL algorithms process data by using graphs of neurons in the input, hidden, and output layers. Like traditional ML algorithms, DFFNN was trained repeatedly using demographics data, demographics + clickstream data, demographics + clickstream + assessment data, and lastly on all the variables of OULAD, setting the final result as the target variable and the rest of the variables as the predictor variables. Predictive models based on DFFNN were trained by first initializing the neurons' edges weights to numbers close to zero using TensorFlow Dense Class. Utilizing the TensorFlow Sequential class, the first observation of OULAD was fed to the DFFNN by applying the forward propagation technique. Applying the Rectified Linear Unit (ReLU), the neurons during the forward propagation process were activated in such a way that the impact of each neuron activation was restricted by the weights of the edges. For training predictive models, the Keras API offers fully-configurable components that can be plugged simultaneously with as little constraints as possible. In particular, the Keras API provides the feed-forward neural networks initialization schemes, activation functions, cost function optimizers, and regularization schemes components that can be combined to create predictive models.

The structure (input, hidden, and output layers) of DFFNN was modified according to the number of variables provided. For example, when only demographics variables were used, the eight categorical variables ('code_module', 'code_presentation', 'gender', 'region', 'highest_education', 'imd_band', 'age_band', 'disability') were converted into dummy variables using one-hot encoding technique and one numerical variable ('num_of_prev_attempts') was scaled using MinMax scaler. After performing a one-hot encoding and scaling operation, a total of 47 variables were created ready to be fed to the DFFNN. Therefore, based on input variables, 47 neurons were used in the input layer, 24 were used in the first hidden layer, 12 were used in the second hidden layer and four neurons were used at the output layer for representing Withdrawn, Fail, Pass, and Distinction classes. Similarly, as new clickstream and assessment variables were added to the demographics variables, new DFFNN layers structures were created for a smooth training process. The hypermeters used for training the predictive model using DFFNN were loss='categorical_crossentropy', optimizer='adam', metrics='accuracy', activation function at hidden layers = ReLU, activation function at the output layer = softmax, epochs = 100, training set = 85%, and testing set = 15%. Tables 7, 8, 9 and 10 present the precision, recall, f-score, support, average accuracy, macro average, and weighted average scores of the prediction models when trained using demographics variables, demographics + clickstream variables, demographics + clickstream + assessment variables, and all variables. The results indicated that the predictive model generated mediocre scores when trained only on demographics data which entails that demographics data only cannot be used for the earliest possible prediction of at-risk students. When trained only demographics + clickstream variables, the predictive model performance score improved which suggests that clickstream variables are an important indicator of students' performance. Upon adding the assessment variables with demographics and clickstream variables, the accuracy score improved 8% meanwhile the precision, recall, and f-score remained low for Distinction, Fail, and Withdrawn classes. Finally, when all the predictor variables were used in the training process, a slight improvement was noticed in the metrics variables

TABLE 7. DFFNN performance results when trained only on demographics variables.

	precision	recall	f1-score	support
Distinction	0.56	0.01	0.02	434
Fail	0.39	0.12	0.18	1093
Pass	0.44	0.70	0.54	1819
Withdrawn	0.42	0.45	0.44	1543
accuracy			.43	4889
macro avg	0.45	0.32	0.30	4889
weighted avg	0.43	0.43	0.38	4889

TABLE 8. DFFNN performance results when trained on demographics and clickstream variables.

	precision	recall	f1-score	support
Distinction	0.77	0.02	0.04	480
Fail	0.52	0.20	0.29	1075
Pass	0.62	0.92	0.74	1832
Withdrawn	0.67	0.76	0.71	1502
accuracy			0.63	4889
macro avg	0.64	0.48	0.45	4889
weighted avg	0.63	0.63	0.56	4889

TABLE 9. DFFNN performance results when trained on demographics, clickstream and assessment variables.

	precision	recall	f1-score	support
Distinction	0.66	0.47	0.55	440
Fail	0.55	0.36	0.44	1081
Pass	0.74	0.90	0.81	1831
Withdrawn	0.76	0.80	0.78	1537
accuracy			0.71	4889
macro avg	0.68	0.63	0.64	4889
weighted avg	0.70	0.71	0.70	4889

TABLE 10. DFFNN performance results when trained on all variables.

	precision	recall	f1-score	support
Distinction	0.64	0.49	0.55	447
Fail	0.55	0.37	0.44	1040
Pass	0.76	0.91	0.82	1886
Withdrawn	0.76	0.80	0.78	1516
accuracy			0.72	4889
macro avg	0.68	0.64	0.65	4889
weighted avg	0.70	0.72	0.70	4889

with only a 1% increase in the average accuracy score. Figure 5a, 5b, 5c, and 5d present the confusion matrices for the predictive models when the predictions were evaluated using the testing set. The diagonal elements are the correctly predicted classes where it can be noticed that the correct predictions increased upon adding more and more data during the predictive model training process.

1) USING FEATURE ENGINEERING TO IMPROVE DFFNN PERFORMANCE

To further improve the performance of DFFNN, feature engineering technique was applied to the final result variable where Pass-Distinction classes were combined into Pass class and Withdrawn-Fail classes were merged into Fail class. Once again the DFFNN based predictive model was trained with 81 neurons at the input layer, 24 neurons at hidden layer 1, 12 neurons at hidden layer 2, and 2 neurons at the output layer. The hyperparameters used were loss=‘binary_crossentropy’, optimizer=‘adam’, metrics=‘accuracy’, activation functions at hidden layers = ReLU, activation function at output layer = sigmoid, and epochs = 100. Table 11 presents the performance score of DFFNN with much improved scores than the multiclass classification arrangement. Overall, an 18% improvement was noticed in the average accuracy score. Figure 6 illustrates the confusion matrix for the predictive model when the predictions were evaluated using the testing set considering only Pass and Fail classes. After the merged operation, the predictive model accuracy substantially improved but we also noticed that as compared to the RF-based predictive model accuracy, it is still 2% less, therefore for predicting students’ performance at different lengths of course module, the RF method was finally selected.

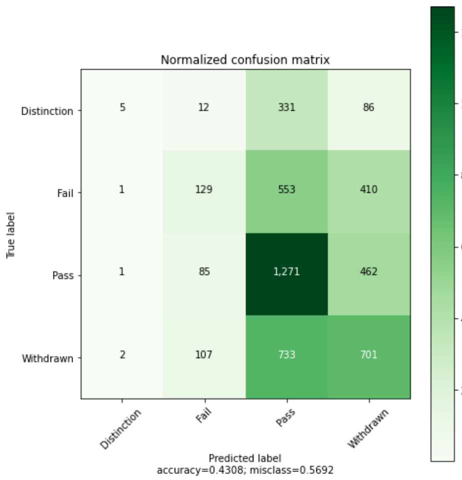
TABLE 11. DFFNN performance after feature engineering step for Fail and Pass classes.

	precision	recall	f1-score	support
Fail	0.98	0.83	0.90	2572
Pass	0.84	0.98	0.90	2317
accuracy			0.90	4889
macro avg	0.91	0.90	0.90	4889
weighted avg	0.91	0.90	0.90	4889

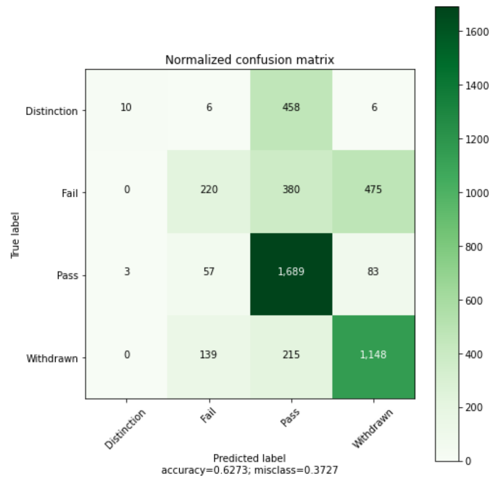
F. PHASE V: CONSTRUCTING RF PREDICTIVE MODEL AT DIFFERENT PERCENTAGE OF COURSE MODULE LENGTH

Table 12 shows the result of the RF predictive model when trained and tested repeatedly on demographics data, 20% course data, 40% course data, 60% course data, 80% course data, 100% course data. Random Forest (RF) uses ensembling techniques by constructing a multitude of decision trees (DT) for classification and regression tasks. During the RF training process, multiple DTs are constructed by using different subsamples of training data, and the mean scores are used to improve prediction accuracy and avoid over-fitting.

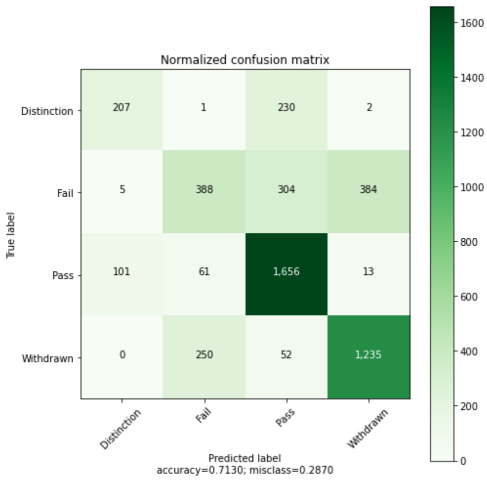
RF predictive model, when trained only on demographics variables, gave the inferior performance scores with averaged precision = .60, recall = .59, f-score = .59, and accuracy = .59 for both Fail and Pass classes. When trained on 20% of course data, the RF predictive model performance improved with averaged precision = .79, recall = .79, f-score = .79 and accuracy = .79 for both Fail and Pass classes. It was also observed that the performance score for Pass class was high as compare to Fail class when more clickstream and assessment data was provided to the RF predictive model. More improvement was observed in the performance score of the RF predictive model when clickstream data and assessment data were provided at 40% of course completion with averaged precision = .84, recall = .84, f-score = .84, and accuracy = .84. The performance score results revealed that RF predictive model performance improved when more and more clickstream and assessment data was provided which infers that the model was learning which variables have a significant



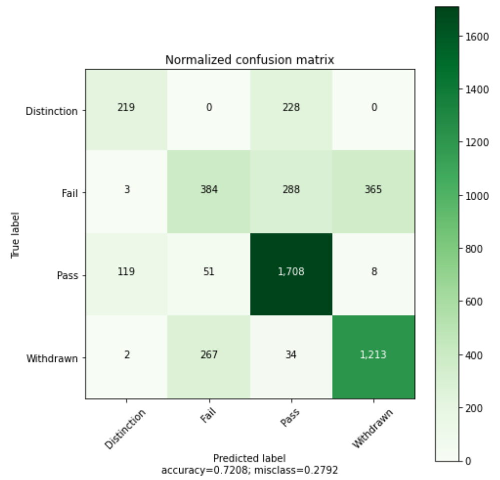
(a) Normalized confusion matrix showing the DFFNN prediction results when tested on demographics variables



(b) Normalized confusion matrix showing the DFFNN prediction results when tested on demographics, and clickstream variables



(c) Normalized confusion matrix showing the DFFNN prediction results when tested on demographics, clickstream, and assessment variables



(d) Normalized confusion matrix showing the DFFNN prediction results when tested on all variables

FIGURE 5. DFFNN confusion matrices when tested on different variables of OULAD.

TABLE 12. Performance scores of RF predictive model at different percentage of course length.

Precision	No Clickstream, Assessement	20% Course	40% Course	60% Course	80% Course	100% Course
Fail	0.6593	0.7501	0.7992	0.8519	0.8811	0.8936
Pass	0.5279	0.8408	0.8874	0.9201	0.9351	0.9473
Averaged	0.6033	0.7980	0.8456	0.8869	0.9084	0.9207
Recall	**No Clickstream, Assessement**	**20% Course**	**40% Course**	**60% Course**	**80% Course**	**100% Course**
Fail	0.6097	0.8404	0.8881	0.9225	0.9382	0.9499
Pass	0.5809	0.7504	0.7980	0.8475	0.8755	0.8884
Averaged	0.5972	0.7929	0.8408	0.8841	0.9066	0.9190
F-score	**No Clickstream, Assessement**	**20% Course**	**40% Course**	**60% Course**	**80% Course**	**100% Course**
Fail	0.6334	0.7927	0.8413	0.8858	0.9087	0.9209
Pass	0.5529	0.7930	0.8403	0.8822	0.9043	0.9169
Averaged	0.5989	0.7929	0.8408	0.8840	0.9065	0.9189
Accuracy	**No Clickstream, Assessement**	**20% Course**	**40% Course**	**60% Course**	**80% Course**	**100% Course**
Fail	0.6096	0.8404	0.8881	0.9225	0.9382	0.9499
Pass	0.5806	0.7505	0.7980	0.8475	0.8755	0.8884
Averaged	0.5972	0.7929	0.8408	0.8841	0.9066	0.9190

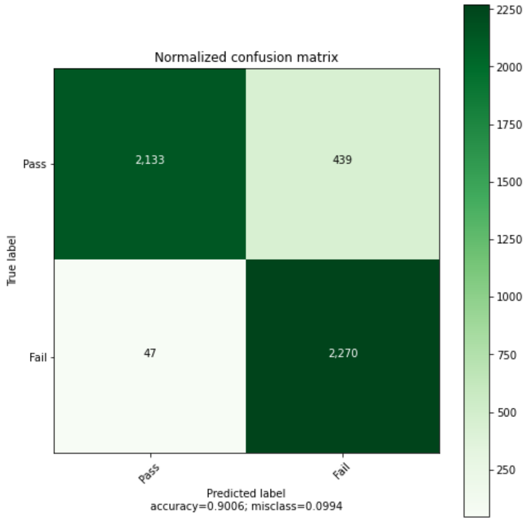


FIGURE 6. Normalized confusion matrix showing the DFFNN prediction for Fail and Pass classes.

relationship with the students' final result. At 60% of course completion, an overall 4% improvement in the performance score of the RF predictive model was observed with average precision = .88, recall = .88, f-score =.88 and accuracy = .88. At 80% of course completion, an increase of 2% in the performance score was observed with averaged precision = .90, recall = .90, f-score = .90 and accuracy = .90 for both Fail and Pass classes. Finally, at 100% course completed, we observed the highest RF predictive model performance score with averaged precision = .92, recall = .91, f-score = .91 and accuracy = .91 for both Fail and Pass classes.

The precision, recall, f-score, and accuracy scores improved when more and more course data was provided to the RF predicted model. Overall the precision score improved from .60 to .92, the recall score improved from .59 to .91, the F-score improved from .59 to .91 and the accuracy score improved from .59 to .91. The performance metric results concluded that the performance of the RF predictive model improved with the length of the course. Even at 20% of course length, the RF predictive model showed a decent performance score with precision for Fail class = .75, Pass class = .84, recall for Fail class = .85, Pass class = .75, f-score for Fail class = .79, Pass class = .79, and accuracy for Fail class = .84, Pass class = .75 which indicated the model could be very useful in interventing at-risk students as early as 20% of course length. As a result, the RF predictive model at 20% of course completion can assist instructors in interventing students and assisting them during their studies. Moreover, the performance scores further improved with 40%, 60%, 80%, and 100% of the course length which indicated that the RF predictive model can support instructors with higher accuracy to intervene students and provide the needed feedback. Intervening students at 100% of course length is useless as the student would have completed the course but the target

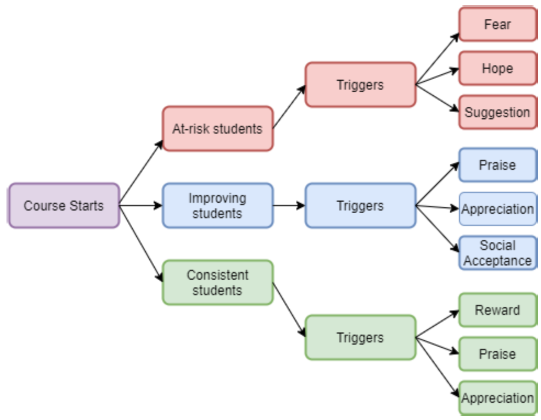


FIGURE 7. Different types of triggers for students having different performance state.

of the predictive model is to support instructors as early as possible in the course timeline to help students at-risk of failure or dropouts.

VI. INTERVENING STUDENTS THROUGH PERSUASIVE TECHNIQUES

Fogg Behavior Model (FBM) suggests that three factors (ability, motivation, and triggers) must be present at the same time to increase people's attitude positively [64]. For intervening and persuading students to improve their study behavior, the optimal time selection is important. Based on the satisfactory results (with 79% accuracy, precision, recall, f-score) of the RF predictive model, students can be intervened after 20% of the course length. Furthermore, if comprehensive details are included in the students' demographic data, then intervention and persuasion can be carried out at the start of the course. Figure 7 summarizes the types of triggers presented to students having fragile, improving, and consistent study performance. The factors of fear, hope, and suggestion are added to the triggers directed to at-risk students. Similarly, for triggers intended for improving and consistent students, the factors of praise, reward, appreciation, and social acceptance are added. The optimal time for sending triggers to students depends upon the results of the predictive model during different stages of the course. As an example, the trigger having hope factor for at-risk students for improving their study behavior might be as follows:

According to our predictive model, your assessment score achieved a 50% success rate. You made a commendable effort this week. Your class position can be further improved if you cautiously follow the next week lectures. If you submit all your assessments on time next week, you will achieve a cumulative score of 60%.

Similarly, trigger having a fear factor for at-risk students for improving their study behavior might be as follows:

According to our predictive model, you are consistently getting a low assessment score. It would be great if nobody is dropped from the course due to a low assessment score.

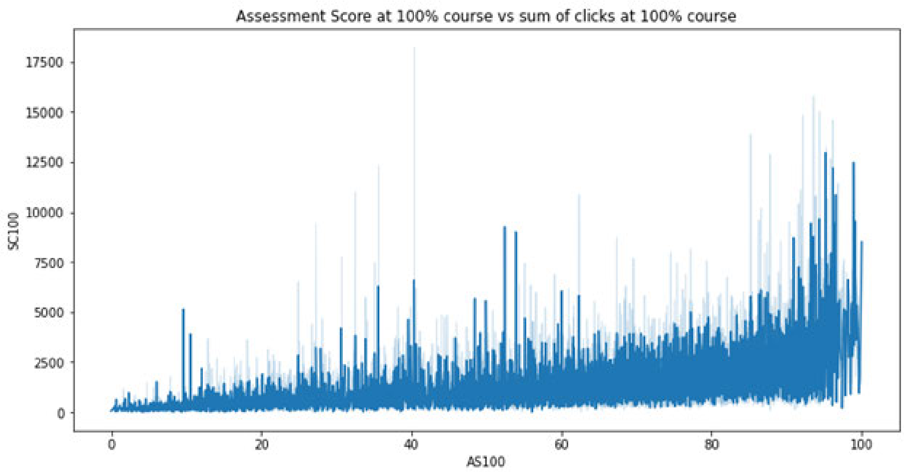


FIGURE 8. Assessment score at 100% course vs sum of clicks at 100% course.

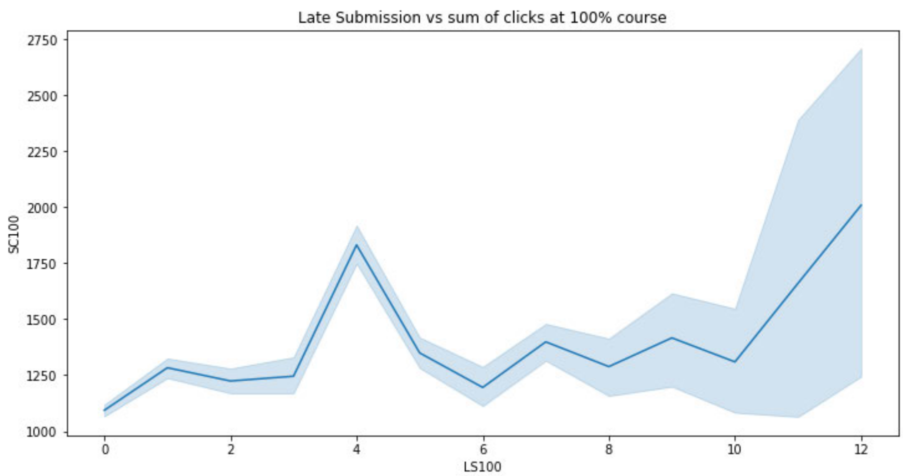


FIGURE 9. Late submission vs sum of clicks at 100% course.

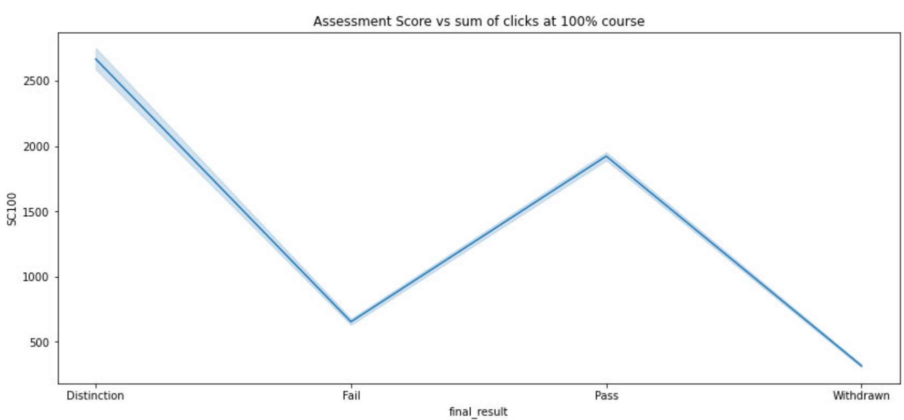


FIGURE 10. Assessment score vs sum of clicks at 100% course.

VII. CONCLUSION, LIMITATION, AND FUTURE WORK

Predicting and interventing students during different stages of course length provides benefits to both students and instructors. It provides instructors an opportunity to assist students at-risk of dropout and make an intervention at the optimal time to improve their study behavior. In the

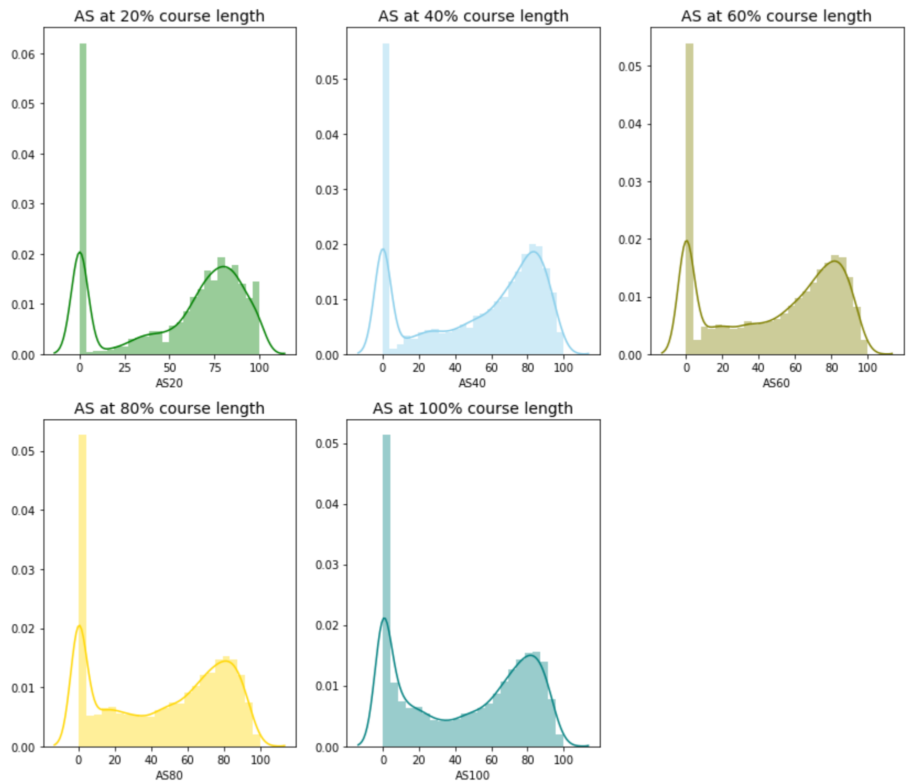


FIGURE 11. Assessment performance at different percentage of course length.

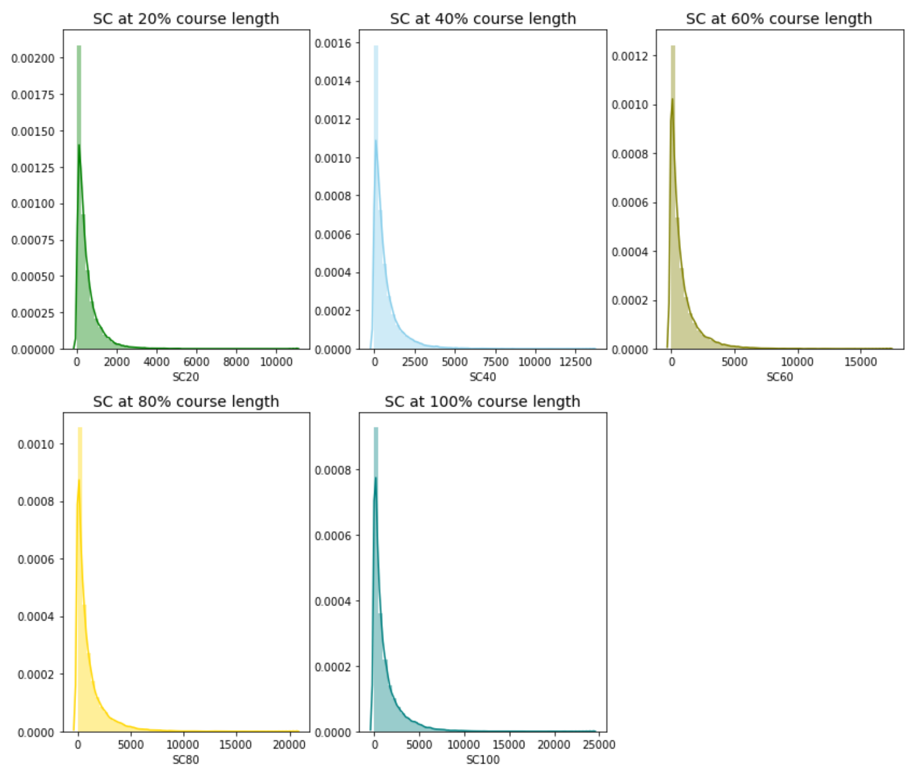
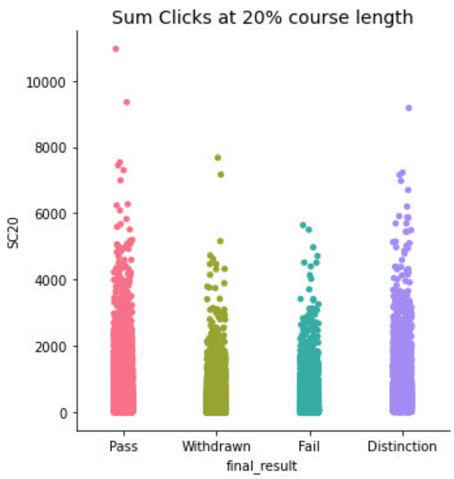
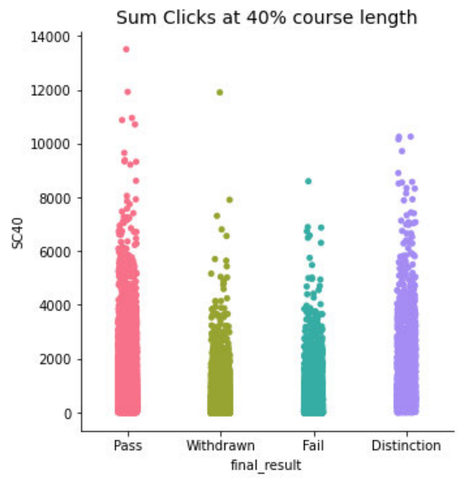


FIGURE 12. Sum of clicks at different percentage of course length.

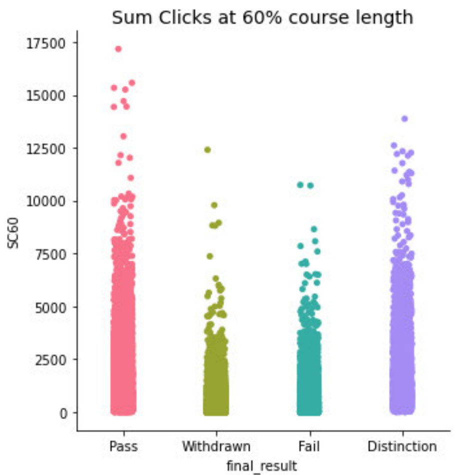
present study, we proposed various predictive models trained on several ML and DL algorithms for predicting students' performance based on demographics variables, demographics + clickstream variables, and demographics + clickstream



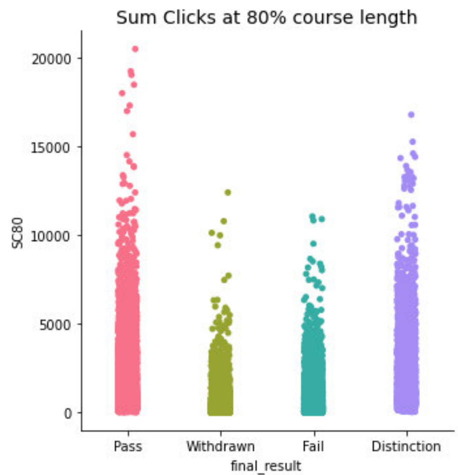
(a) Sum of clicks at 20% of course length



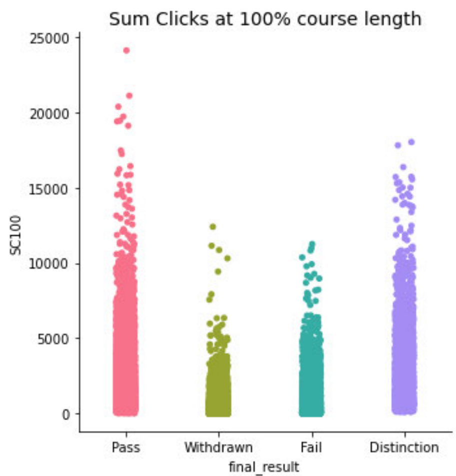
(b) Sum of clicks at 40% of course length



(c) Sum of clicks at 60% of course length



(d) Sum of clicks at 80% of course length



(e) Sum of clicks at 100% of course length

FIGURE 13. Students' VLE interaction vs final grade positions at different percentage of course length.

+ assessment variables. RF predictive model with the highest performance scores was finally selected for predicting students' performance at the different lengths of course. Such a predictive model can facilitate instructors to make timely

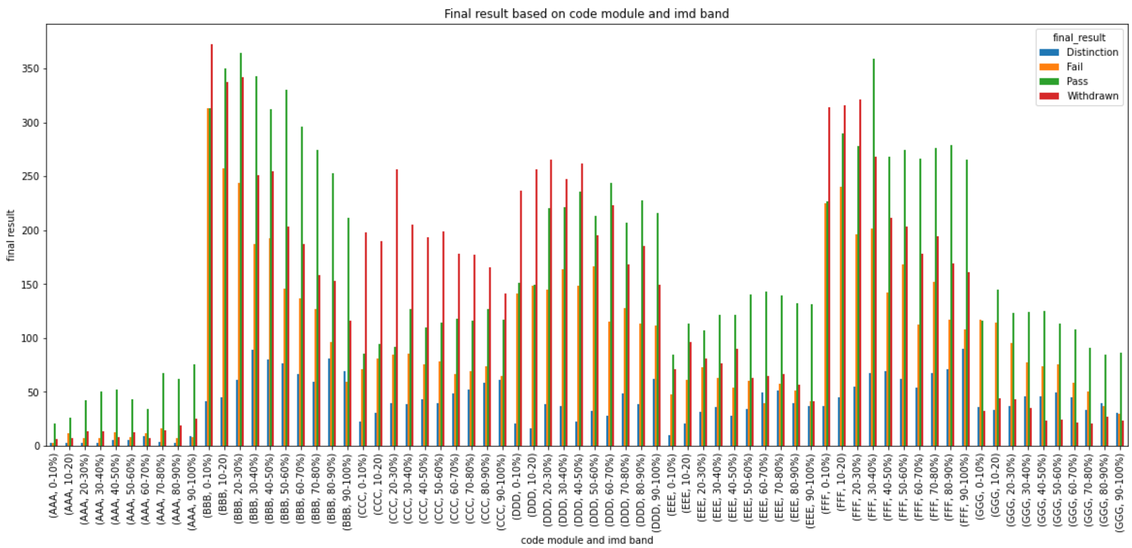


FIGURE 14. **Relationship of Code module and imd_band with the final result where highest performance is observed in course BBB, FFF and lowest performance in course AAA.**

interventions and persuade at-risk students to improve their study performance. Out of many variables, clickstream, and assessment variables were having the most significant impact on the final result of the students.

This study revealed that techniques such as feature engineering momentously improve the performance of predictive models. During the course module timeline, the performance of students was predicted at the very beginning when only demographics variables were available. Subsequently, the students' performance was predicted at 20%, 40%, 60%, 80%, and 100% of the course length. Even at 20% of the course length, the RF predictive model was producing promising results with 79% average precision score, 79% average recall score, 79% average f-score, and 79% average accuracy score. At 60% of the course length, the RF predictive model performance improved significantly with 88% of average precision, recall, f-score, and accuracy. Finally, at 100% course completion, the highest RF predictive model performance scores were observed with an average precision of 92%, average recall of 91%, average f-score of 91%, and average accuracy of 91 percent. We also observed that the performance scores for Fail class individually after the feature engineering process were better than the Pass class performance scores. Getting a higher performance score for the Fail class was due to an imbalanced class problem with 17,208 Fail students and 15,385 Pass students.

Overall, the results of the RF predictive model demonstrated effectiveness in the earliest possible prediction of the performance of at-risk students. Such data-driven studies can assist VLE administrators and instructors in the formulation of the online learning framework which can contribute to the decision-making process. We also deemed that more in-depth studies are required to evaluate various online activities in the OULAD. Particularly, how various early intervention techniques can be implemented in the online learning environment to encourage students to keep on the right track. In the future, we plan to examine the activity-wise significance having a prominent influence on the students' performance by modeling textual variables relating to students' feedbacks by utilizing deep learning models and natural language processing techniques.

APPENDIX

OULAD EXPLORATORY DATA ANALYSIS (EDA)

Before training the predictive models, EDA was performed on OULAD to understand relationships among different variables. Figure 8 presents the comparison of clickstream data with assessment score at the end of the course indicating that with a higher assessment score the sum clicks per student increased.

Figure 9 shows that students having more late assessment submitted were interacting more with the VLE but the relationship is not very significant. From the line chart, it can be observed that students having 12 late assessment submissions were involved more than any of the remaining late submissions.

Figure 10 reveals that students getting Distinction (completed courses with excellent grades) and Pass (completed courses with a passing score) grades in the final result were involved more with the VLE as compare to the student getting Fail (students who completed the course but were not able to secure passing marks) and Withdrawn (students who were not able to complete the course) score.

The five subplots in figure 11 shows students' assessment performance at 20%, 40%, 60%, 80%, and 100% of course length. The subplots reveal similar students' assessment scores at a different percentage of course length. Students showing a lower assessment performance at the beginning of the course have a similar score during the rest of the course length.

Similarly, the subplots in figure 12 shows clickstream pattern at different stages of the course length. It can be observed that the clickstream pattern remains the same for most students during a different percentage of the course length. However, we can also observe that VLE interaction at the beginning of the course (at 20% and 40% of course length) is higher but remain almost the same after the 50% of course completion (at 60%, 80%, and 100% of course length).

Similarly, the subplots in figure 13 shows students similar clickstream patterns at a different percentage of course length for the Pass, Withdrawn, Fail, and Distinction cases.

Finally, figure 14 presents a relationship between the final result and code module/imd_band where it can be observed that students showed the lowest performance in code module AAA and highest performance in code module BBB and FFF.

REFERENCES

[1] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an 'early warning system' for educators: A proof of concept," *Comput. Edu.*, vol. 54, no. 2, pp. 588–599, Feb. 2010.

[2] C. Romero, S. Ventura, and E. García, "Data mining in course management systems: Moodle case study and tutorial," *Comput. Edu.*, vol. 51, no. 1, pp. 368–384, Aug. 2008.

[3] S. Valsamidis, S. Kontogiannis, I. Kazanidis, T. Theodosiou, and A. Karakos, "A clustering methodology of Web log data for learning management systems," *J. Educ. Technol. Soc.*, vol. 15, no. 2, pp. 154–167, 2012.

[4] M. Hussain, W. Zhu, W. Zhang, S. M. R. Abidi, and S. Ali, "Using machine learning to predict student difficulties from learning session data," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 381–407, Jun. 2019.

[5] O. E. Aissaoui, Y. E. A. El Madani, L. Oughdir, and Y. E. Allioui, "Combining supervised and unsupervised machine learning algorithms to predict the learners' learning styles," *Procedia Comput. Sci.*, vol. 148, pp. 87–96, Jan. 2019.

[6] J. Y. Chung and S. Lee, "Dropout early warning systems for high school students using machine learning," *Children Youth Services Rev.*, vol. 96, pp. 346–353, Jan. 2019.

[7] S. A. Salloum, M. Alshurideh, A. Elnagar, and K. Shaalan, "Mining in educational data: Review and future directions," in *Proc. Joint European-US Workshop Appl. Invariance Comput. Vis.* Cairo, Egypt: Springer, 2020, pp. 92–102.

[8] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A systematic review of deep learning approaches to educational data mining," *Complexity*, vol. 2019, May 2019, Art. no. 1306039.

[9] K. S. Rawat and I. Malhan, "A hybrid classification method based on machine learning classifiers to predict performance in educational data mining," in *Proc. 2nd Int. Conf. Commun., Comput. Netw.* Chandigarh, India: National Institute of Technical Teachers Training and Research, Department of Computer Science and Engineering, 2019, pp. 677–684.

[10] S. M. Jayaprakash, E. W. Moody, E. J. M. Lauría, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *J. Learn. Analytics*, vol. 1, no. 1, pp. 6–47, May 2014.

[11] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura, "Early dropout prediction using data mining: A case study with high school students," *Expert Syst.*, vol. 33, no. 1, pp. 107–124, Feb. 2016.

[12] S. Palmer, "Modelling engineering student academic performance using academic analytics," *Int. J. Eng. Edu.*, vol. 29, no. 1, pp. 132–138, 2013.

[13] Z. Papamitsiou and A. Economides, "Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence," *Edu. Technol. Soc.*, vol. 17, no. 4, pp. 49–64, 2014.

[14] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1432–1462, Mar. 2014.

[15] N. Z. Zacharis, "A multivariate approach to predicting student outcomes in Web-enabled blended learning courses," *Internet Higher Edu.*, vol. 27, pp. 44–53, Oct. 2015.

[16] E. B. Costa, B. Fonseca, M. A. Santana, F. F. de Araájo, and J. Rego, "Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses," *Comput. Hum. Behav.*, vol. 73, pp. 247–256, Aug. 2017.

[17] D. Gašević, S. Dawson, T. Rogers, and D. Gasevic, "Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success," *Internet Higher Edu.*, vol. 28, pp. 68–84, Jan. 2016.

[18] A. Bozkurt and I. E. Aydin, "Cultural diversity and its implications in online networked learning spaces," in *Supporting Multiculturalism Open Distance Learning spaces*. Hershey, PA, USA: IGI Global, 2018, pp. 56–81.

[19] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning Analytics*. New York, NY, USA: Springer, 2014, pp. 61–75.

[20] L. Cen, D. Ruta, L. Powell, B. Hirsch, and J. Ng, "Quantitative approach to collaborative learning: Performance prediction, individual assessment, and group composition," *Int. J. Comput.-Supported Collaborative Learn.*, vol. 11, no. 2, pp. 187–225, Jun. 2016.

[21] A. Mueen, B. Zafar, and U. Manzoor, "Modeling and predicting students' academic performance using data mining techniques," *Int. J. Mod. Edu. Comput. Sci.*, vol. 8, no. 11, p. 36, 2016.

[22] S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," *Comput. Edu.*, vol. 61, pp. 133–145, Feb. 2013.

[23] F. Marbouti, H. A. Diefes-Dux, and K. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," *Comput. Edu.*, vol. 103, pp. 1–15, Dec. 2016.

[24] D. T. Tempelaar, B. Rienties, and B. Giesbers, "In search for the most informative data for feedback generation: Learning analytics in a data-rich context," *Comput. Hum. Behav.*, vol. 47, pp. 157–167, Jun. 2015.

[25] R. F. Kizilcec, M. Pérez-Sanagustín, and J. J. Maldonado, "Self-regulated learning strategies predict learner behavior and goal attainment in massive open online courses," *Comput. Edu.*, vol. 104, pp. 18–33, Jan. 2017.

[26] J. Kuzilek, M. Hlosta, D. Herrmannova, Z. Zdrahal, and A. Wolff, "Ou analyse: Analysing at-risk students at the open university," *Learn. Analytics Rev.*, vol. 8, pp. 1–16, Mar. 2015.

[27] A. Wolff, Z. Zdrahal, A. Nikolov, and M. Pantucek, "Improving retention: Predicting at-risk students by analysing clicking behaviour in a virtual learning environment," in *Proc. 3rd Int. Conf. Learn. Analytics Knowl.*, 2013, pp. 145–149.

[28] M. Hlosta, D. Herrmannova, L. Vachova, J. Kuzilek, Z. Zdrahal, and A. Wolff, "Modelling student online behaviour in a virtual learning environment," 2018, *arXiv:1811.06369*. [Online]. Available: http://arxiv.org/abs/1811.06369

[29] Y. Cui, F. Chen, and A. Shiri, "Scale up predictive models for early detection of at-risk students: A feasibility study," *Inf. Learn. Sci.*, vol. 121, nos. 3–4, pp. 97–116, Feb. 2020.

[30] Y. Lee and J. Choi, "A review of online course dropout research: Implications for practice and future research," *Educ. Technol. Res. Develop.*, vol. 59, no. 5, pp. 593–618, Oct. 2011.

[31] J.-L. Hung, M. C. Wang, S. Wang, M. Abdelrasoul, Y. Li, and W. He, "Identifying at-risk students for early interventions—A time-series clustering approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 45–55, Nov. 2015.

[32] T. Soffer and A. Cohen, "Students' engagement characteristics predict success and completion of online courses," *J. Comput. Assist. Learn.*, vol. 35, no. 3, pp. 378–389, Jun. 2019.

[33] P. H. Winne, "Improving measurements of self-regulated learning," *Educ. Psychologist*, vol. 45, no. 4, pp. 267–276, Oct. 2010.

[34] L. Sha, C.-K. Looi, W. Chen, and B. H. Zhang, "Understanding mobile learning from the perspective of self-regulated learning," *J. Comput. Assist. Learn.*, vol. 28, no. 4, pp. 366–378, Aug. 2012.

[35] R. Baker, B. Evans, Q. Li, and B. Cung, "Does inducing students to schedule lecture watching in online classes improve their academic performance? An experimental analysis of a time management intervention," *Res. Higher Edu.*, vol. 60, no. 4, pp. 521–552, 2019.

[36] A. Cicchinelli, E. Veas, A. Pardo, V. Pammer-Schindler, A. Fessl, C. Barreiros, and S. Lindstädt, "Finding traces of self-regulated learning in activity streams," in *Proc. 8th Int. Conf. Learn. Anal. Knowl.*, Mar. 2018, pp. 191–200.

[37] J. M. Lim, "Predicting successful completion using student delay indicators in undergraduate self-paced online courses," *Distance Edu.*, vol. 37, no. 3, pp. 317–332, Sep. 2016.

[38] J. Park, K. Denaro, F. Rodriguez, P. Smyth, and M. Warschauer, "Detecting changes in student behavior from clickstream data," in *Proc. 7th Int. Learn. Anal. Knowl. Conf.*, Mar. 2017, pp. 21–30.
[39] S. Gupta and A. S. Sabitha, "Deciphering the attributes of student retention in massive open online courses using data mining techniques," *Edu. Inf. Technol.*, vol. 24, no. 3, pp. 1973–1994, May 2019.
[40] G. Akçapçnar, M. N. Hasnine, R. Majumdar, B. Flanagan, and H. Ogata, "Developing an early-warning system for spotting at-risk students by using eBook interaction logs," *Smart Learn. Environ.*, vol. 6, no. 1, p. 4, Dec. 2019.
[41] G. Kőrösi and R. Farkas, "Mooc performance prediction by deep learning from raw clickstream data," in *Proc. Int. Conf. Adv. Comput. Data Sci.* Valletta, Malta: Springer, 2020, pp. 474–485.
[42] A. Cano and J. D. Leonard, "Interpretable multiview early warning system adapted to underrepresented student populations," *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 198–211, Apr. 2019.
[43] C. Burgos, M. L. Campanario, D. D. L. Peña, J. A. Lara, D. Lizcano, and M. A. Martínez, "Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout," *Comput. Electr. Eng.*, vol. 66, pp. 541–556, Feb. 2018.
[44] J. A. Lara, D. Lizcano, M. A. Martínez, J. Pazos, and T. Riera, "A system for knowledge discovery in e-learning environments within the European higher education area–application to student data from open university of Madrid, UDIMA," *Comput. Edu.*, vol. 72, pp. 23–36, Mar. 2014.
[45] C. C. Gray and D. Perkins, "Utilizing early engagement and machine learning to predict student outcomes," *Comput. Edu.*, vol. 131, pp. 22–32, Apr. 2019.
[46] R. Al-Shabandar, A. J. Hussain, P. Liatsis, and R. Keight, "Detecting at-risk students with early interventions using machine learning techniques," *IEEE Access*, vol. 7, pp. 149464–149478, 2019.
[47] S. Lee and J. Y. Chung, "The machine learning-based dropout early warning system for improving the performance of dropout prediction," *Appl. Sci.*, vol. 9, no. 15, p. 3093, Jul. 2019.
[48] A. Behr, M. Giese, and K. Theune, "Early prediction of university dropouts—A random forest approach," *J. Nat. Stat.*, vol. 1, pp. 743–789, Feb. 2020.
[49] L. C. B. Martins, R. N. Carvalho, R. S. Carvalho, M. C. Victorino, and M. Holanda, "Early prediction of college attrition using data mining," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1075–1078.
[50] N. Mduma, K. Kalegele, and D. Machuve, "Machine learning approach for reducing students dropout rates," *Int. J. Adv. Comput. Res.*, vol. 9, no. 42, 2019, doi: 10.19101/IJACR.2018.839045.
[51] J. Figueroa-Ca nas and T. Sancho-Vinuesa, "Predicting early dropout student is a matter of checking completed quizzes: The case of an online statistics module," in *Proc. LASI-SPAIN*, 2019, pp. 100–111.
[52] A. Ortigosa, R. M. Carro, J. Bravo-Agapito, D. Lizcano, J. J. Alcolea, and O. Blanco, "From lab to production: Lessons learnt and real-life challenges of an early student-dropout prevention system," *IEEE Trans. Learn. Technol.*, vol. 12, no. 2, pp. 264–277, Apr. 2019.
[53] A. S. Imran, F. Dalipi, and Z. Kastrati, "Predicting student dropout in a MOOC: An evaluation of a deep neural network model," in *Proc. 5th Int. Conf. Comput. Artif. Intell. (ICCAI)*, 2019, pp. 190–195.
[54] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-net: Dropout prediction in MOOCs with deep learning," in *Proc. ACM Turing Celebration Conf.*, May 2019, pp. 1–6.
[55] C. P. Rosé, E. A. McLaughlin, R. Liu, and K. R. Koedinger, "Explanatory learner models: Why machine learning (alone) is not the answer," *Brit. J. Educ. Technol.*, vol. 50, no. 6, pp. 2943–2958, 2019.
[56] A. A. Mubarak, H. Cao, and S. A. M. Ahmed, "Predictive learning analytics using deep learning model in MOOCs' courses videos," *Edu. Inf. Technol.*, vol. 6, pp. 1–22, Jul. 2020.
[57] S. N. Liao, D. Zingaro, K. Thai, C. Alvarado, W. G. Griswold, and L. Porter, "A robust machine learning technique to predict low-performing students," *ACM Trans. Comput. Edu.*, vol. 19, no. 3, pp. 1–19, Jun. 2019.
[58] B. Sekeroglu, K. Dimililer, and K. Tuncal, "Student performance prediction and classification using machine learning algorithms," in *Proc. 8th Int. Conf. Educ. Inf. Technol.*, Mar. 2019, pp. 7–11.
[59] Y. Mao, "Deep learning vs. Bayesian knowledge tracing: Student models for interventions," *J. Educ. Data Mining*, vol. 10, no. 2, pp. 1–27, 2018.
[60] Z. Iqbal, J. Qadir, A. Noor Mian, and F. Kamiran, "Machine learning based student grade prediction: A case study," 2017, *arXiv:1708.08744*. [Online]. Available: http://arxiv.org/abs/1708.08744
[61] H. L. Fwa and L. Marshall, "Modeling engagement of programming students using unsupervised machine learning technique," *GSTF J. Comput.*, vol. 6, no. 1, pp. 1–6, 2018.
[62] J. Xu, K. H. Moon, and M. van der Schaar, "A machine learning approach for tracking and predicting student performance in degree programs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 5, pp. 742–753, Aug. 2017.
[63] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open university learning analytics dataset," *Sci. Data*, vol. 4, no. 1, Dec. 2017, Art. no. 170171.
[64] B. J. Fogg, "Fogg behavior model," Behav. Des. Lab., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2019. [Online]. Available: https://behaviormodel.org/

MUHAMMAD ADNAN received the bachelor's degree in computer science from the University of Peshawar and the master's degree in information technology from the School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad, where he is currently pursuing the Ph.D. degree. He is currently a Lecturer with the Institute of Information Technology, Kohat University of Science and Technology. His research area includes mobile learning, machine learning, adaptive learning, and ubiquitous systems.

ASAD HABIB received the Ph.D. degree from the Nara Institute of Science and Technology, Nara, Japan. He is currently an Assistant Professor with the Institute of Information Technology, Kohat University of Science and Technology. He is involved in the area of natural language processing, human–computer interaction, artificial intelligence, learning technologies, mobile learning, and adaptive interface design.

JAWAD ASHRAF received the Ph.D. degree from the Department of Computer Science, University of Leicester, U.K. He is currently with the Institute of Information Technology, Kohat University of Science and Technology, where he is working on partner-based scheduling algorithm for grid workflows in advance reservation environment, K-shortest path variant for routing in advance reservation environment, and novel workflow job selection technique.

SHAFAQ MUSSADIQ received the M.S. degree in computer science and the Ph.D. degree from the School of Electrical Engineering and Computer Science, NUST, Islamabad. She is currently an Assistant Professor with the Institute of Information Technology, Kohat University of Science and Technology, Kohat. Her research work includes machine learning, deep neural networks, multimedia technologies, and big data in education.

ARSALAN ALI RAZA is currently pursuing the Ph.D. degree in computer science with the Kohat University of Science and Technology, Kohat, Pakistan. He is currently a Lecturer in computer science with the COMSATS Institute of Information Technology, Pakistan. His research interest includes natural language engineering, data mining, machine learning, and artificial intelligence.

MUHAMMAD ABID received the Ph.D. degree from the Institute of Computing, Kohat University of Science and Technology, Kohat. He is currently working on various projects related to automation and transparency of educational institutes. His research area includes natural language processing, machine learning, and data mining.

MARYAM BASHIR is currently pursuing the Ph.D. degree with the Institute of Computing, Kohat University of Science and Technology, Pakistan. Her Ph.D. thesis is on trajectory data mining. She is currently a Visiting Lecturer with the Government Post Graduate College, Kohat.

SANA ULLAH KHAN received the M.S. degree in computer science from the Islamia College University Peshawar in 2014, where he is currently pursuing the Ph.D. degree in the field of medical image processing. He is also a permanent Faculty Member with the Institute of Computing, Kohat University of Science and Technology (KUST), Kohat, serving as a Lecturer for the past three years. He has authored various research articles of journals and conferences. His research interest includes computer vision, medical imaging, machine learning, cloud computing, and deep learning.