Intelligent System for Personalised Interventions and Early Drop-out Prediction in MOOCs

ALJ Zakaria
University Sidi Mohamed Ben Abdellah
Faculty of Science Dhar-Mahraz
Fez, Morocco

BOUAYAD Anas
University Sidi Mohamed Ben Abdellah
Faculty of Science Dhar-Mahraz
Fez, Morocco

Cherkaoui Malki Mohammed Ouçamah
University Sidi Mohamed Ben Abdellah
Faculty of Science Dhar-Mahraz
Fez, Morocco

***Abstract*—In this paper, we propose an approach to early detect students at high risk of drop-out in MOOC (Massive Open Online Course); we design personalised interventions to mitigate that risk. We apply Machine Learning (ML) algorithms and data mining techniques to a dataset extracted from XuetangX MOOC learning platforms and sourced from the KDD cup 2015. Since this dataset contains only raw student log activity records, we perform a hybrid feature selection and dimensionality reduction techniques to extract relevant features, and reduce models complexity and computation time. Besides, we built two models based on: Genetic Algorithms (GA) and Deep Learning (DL) with supervised learning methods. The obtained results, according to the accuracy and the AUC (Area Under Curve)-ROC (Reciever Operator Characteristic) metrics, prove the pertinence of the extracted features and encourage the use of the hybrid features selection. They also proved that GA and DL are outperforming the baseline algorithms used in related works. To assess the generalisation of the approach used in this work, The same process is performed to a second benchmark dataset extracted from the university MOOC. Then, a single web application hosted on the university server, produces an individual weekly drop-out probability, using time series data. It also proposes an approach to personalise and prioritise interventions for at-risk students according to the drop-out patterns.**

Keywords*—*MOOC; drop-out; dimensionality reduction; features selection; personalised intervention

I. Introduction

Following the emergence of new digital technologies aiming to modernize the traditional education system. Massive Open Online Courses (MOOCs) have gained popularity in recent years [1]. In 2020, 16,300 courses were offered by 950 universities, and the number of enrollment has reached more than 180 million learners worldwide [2]. MOOCs have become an ideal source of self development that bridges the gap between industry requirements and skills acquired in the university [3]. Despite these benefits that bring a substantial improvement to the student learning experience. MOOCs are facing many problems today. Among the most cited are the high drop-out and the low completion rate. The average completion rate for a MOOC is 12.6% [4]. We can also cite the weak interactions and the absence of tutor support to the significant number of enrolled participants. This excessive attrition rate in MOOCs, has prompted researchers to consider the use of learning analytic for early prediction of learners at risk of drop-out [5].

Learning analytic consists of analysing the log trace and the data collected while students interact with the MOOC courses [6]. Then, using supervised machine learning, mathematical models can automatically detect student at-risk of drop-out based on their previous behaviour and interactions during the course. The use of learning analytic has shown an encouraging potential and reduced visibly the attrition rates in MOOCs [7], [8]. However, it is limited, given the enormous number of enrolment either by 1) the late detection of students at-risk , by 2) the absence of prioritization which will considerably reduce the number of students that the instructor must address each time. Finally 3) the prediction models provide no clues for the monitor to propose a personalized intervention for each droppers pattern. Thus, a system that is capable of detecting at-risk students and providing a customised intervention is therefore needed.

The aim of this paper is to build an intelligent system using DL and GA in the field of learning analytic, and able to overcome these three limitations. This paper will describe the process followed to address the problem related to early prediction student drop-out, prioritizing student needing intervention according to a weekly temporal model prediction based on student drop-out probability. In the light of the obtained results, a timely personalized intervention can be designed and delivered to retain students at-risk. The obtained system gives the possibility to meet the challenges of identifying on a large scale students at high risk of dropping- out, while also satisfying the requirement to be able to support early intervention.

This paper is organised as follows: the next section present a brief overview of related work. The Section 4 is devoted to describe the different components of the used dataset and the extracted features. Section 5, is dedicated to presenting the methods used to predict student drop-out. Section 6, is dedicated to discussing the obtained results. Finally, we conclude the paper and give some perspectives on future work.

II. Related Work

Many researchers have recently focused on MOOCs student drop-out. Time consideration is very significant when tackling this problem. Early detection plays a masterful role in reducing the attrition rate. In fact, several studies have proved that 75% of drop-outs occur in the first weeks [9]. Similarly, Gitinabard et al. [10] analyse students logs and forum data of an annual MOOC lessons. They apply Logistic Regression and Support Vector Machine (SVM) to predict drop-out in the first weeks of each course. The students were flagged as drop-out with the precision of 70% after the third week. Berens et al. [11] developed an early predicting system using demographic data and a boosting algorithm combining several ML algorithms: Linear Regression, Neural Network, Decision Tree, AdaBoost. The system provides an accuracy of 58.2% for the first semester and 81.5% for the fourth semester. Authors in [12] used deep learning and achieved an accuracy of 0.92% in the first week. In [13] ALJ et al. used several baseline machine algorithms and obtain an AUC ROC score of 90%. The performance of machine learning algorithms highly depends on the used dataset used. ML algorithms are unable to provide effective parameter setting method. Therefore, feature selection of parameters is another research content of this paper. A hybrid features selection method is proposed. Besides, Genetic Algorithms (GA) are used for parameters tuning.

GA are generally used for optimization problems and tuning classifiers in multiple fields such as emotion recognition [14] and medicine [15], and feature selection [16]. Despite GA are not quite used in the field of student drop-out, they are producing significant results.

This study contributes to the current state-of-the-art of the field in two main directions. First, by developing a comprehensive approach for studying and detecting early drop-out in a data perspective. Second, by designing a prioritized and personalized intervention for student at-risk of drop-out.

III. Data

A. Data Description

This study use a dataset provided from KDD cup [17], an annual Data Mining and Knowledge Discovery competition organised by ACM Special Interest Group. In KDD cup 2015, the dataset used in the competition contains users trace log extracted from XuetangX which is one of the biggest Chinese MOOC learning platforms. The aim was to predict users drop-out using different data mining techniques and machine learning algorithms.

The detailed description of the five parts of dataset is as follows:

1) The first part of the dataset contains information about the start and the end date of each course according to the Table I.

TABLE I. Course Informations

Fields	Type	Description
Course-ID	Nominal	Course Identifier
Course-S	Date	Course Starting Date
Course-E	Date	Course Ending Date

2) Information about the modules of each course, sub-modules and also the category of modules and their start date according to the Table II.

TABLE II. Module Informations

Fields	Type	Description
Course-ID	Nominal	Course Identifier
Module-ID	Nominal	Module Identifier
Module-cat	Nominal	Module Category
Module-child	Nominal	Sub-Module
Module-S	Date	Module Starting Date

3) Informations about enrolments: an enrolment is a (Student, Course) entry according to the Table III.

TABLE III. Enrolment Information

Fields	Type	Description
Enrolment-ID	Nominal	Enrolment Identifier
Student-ID	Nominal	Student Identifier
Course-ID	Nominal	Course Identifier

4) The fourth part of the dataset contains a log trace of every enrolment, log timestamps, and the source and the type of the event according to the Table IV.

TABLE IV. Log Information

Fiels	Type	Description		
Enrolment-ID	Nominal	Enrolment Identifier		
Student-ID	Nominal	Student Identifier		
Course-ID	Nominal	Course Identifier		
Event-Time	Timestamps	Time when the event occurs		
Source	Nominal	Source of Event (Server / Browser)		
Event	Nominal	Event Type	e1	Problem
			e2	Access
			e3	Video
			e4	Wiki
			e5	Discussion
			e6	Navigate
			e7	Page Close

5) The last part of the dataset contains information about the real value of enrolment result according to the Table V

TABLE V. Enrolment Result

Fields	Type	Description		
Enrolment-ID	Nominal	Enrolment Identifier		
Result	Boolean	Student Result	0	Success
			1	Drop-out

The dataset captures a trace log of 79186 students and 120543 enrolment, because every student can enrol in multiple courses. If a user leaves no records for course C in the log during the next 10 days, it is defined as drop-out from the course.

We notice that for all courses the drop-out represented by 1 in the table of enrolment results exceeds 65%. The majority class is drop-out with (95581) 79% compared to success (24961) 21% of enrolments. In this case, a class imbalance problem is faced. In order to balance the dataset, we will oversample the minority class in order to increase its cardinality to be equal to the majority class.

Oversampling: this technique duplicates copies of some points from the minority class to increase its cardinality to be equal to the majority class. New samples can be generated by random repetition or using more sophisticated methods such as SMOTE or ADASYN.

Synthetic Minority Oversampling Technique (SMOTE) [18] uses the KNN algorithm to generate new synthetic data points that combine features of the data point and its K closest neighbours. However, it still has some weaknesses regarding the oversampling logic used. On the one hand, it does not consider generating new samples from neighbours which may come from the other class. The synthetic observations can overlap with other observations of the majority class. On the other hand, generating multiple synthetic observations risks introducing additional noise in the dataset, this could potentially bias the model.

ADASYN [19] for Adaptive Synthetic, a version of SMOTE that has been improved. Instead of generating the same number of synthetic observations for each observation of the minority class, ADASYN adapts the oversampling to the distribution density of the observations of the minority class. Concretely, it produces more synthetic samples in regions of feature space where the density of minority observations is low and fewer samples in regions with higher density.

The selection of the technique to use remains strictly linked to the data set used. For our example, SMOTE gives better results. Finally, we end up with two datasets Unbalanced Dataset (UB) and Balanced Dataset (BD).

B. Feature Engineering

The information available on the KDD cup 2015 dataset lacks personal information (e.g. age, sex, nationality) and information regarding the course (e.g. prerequisites, difficulty level). The logging trace remains the most potent source of information. Our feature extraction method is based on counting the log of every enrolment; an enrolment is a (student, course) entry. The extracted features can be divided into two parts:

Enrolment History Features : It contains features about the history of interaction with the MOOC, such as the number of successful courses, the number of failed courses, the cumulative number of days spent on the MOOC during old registrations, and the cumulative number of logs of each event present on the catalogue in Table IV.

Current Enrolment Features: It contains features about the number of days spent on the MOOC during the current enrolment and the count of logs of each event.

We also extract the count of minutes spent for every enrolment; after examining the log trace, we notice that all sessions start with the **Navigate** event and sometimes with the **Access**. We calculate the difference of time expressed in minutes between one of the two events and the end of the session expressed with the **Page_close**. The accumulation of minutes is recorded for each enrolment in the variable m according to the following algorithm:

ALGORITHM 1
Algorithm of Connected Minutes

```
Data : Raw log data
Result: Connected minutes per enrolment
@ConnectedMin = 0; @BeginDay = ”00:00:00”
@EndDay = ”23:59:59”; @TBegin = ””; @TEnd = ””;
while not at the end of enrolment log rows do
    read current
    if @Evente=’Nagivate’ or (@Evente =’Access’ and @TimeBegin =”” ) then
        @TimeBegin= Time-Event;
    end if
    if @Evente=’Page close’ then
        @TimeEnd= Time-Event;
    end if
    @Diff=DateDiff(MIN,@TBegin,@TEnd);
    if @Ddiff ¡ 0 then
        @x=DateDiff(MIN,@hdebut,@EndDay);
        @y=DateDiff(MIN,@BeginDay,@hfin);
        @Ddiff= @x+@y;
    end if
    @ConnectedMin= ConnectedMin+@Ddiff;
end while
```

Finally we end up with features presented in the Table VI.

TABLE VI. EXTRACTED FEATURES

N	Features
1	Enrolment Identifier
2	Count of student previous enrolments
3	Count of student previous succeeded enrolments
4	Count of student previous drop-out enrolments
5	Count of log for the current enrolment
6	Count of log for all previous enrolments
7	Count of days between first and last log
8	Count of days between first and last log for all previous enrolments
9	Count of log for the event : Problem
10	Count of log for the event : Problem for all previous enrolments
11	Count of log for the event : Video
12	Count of log for the event : Video for all previous enrolments
13	Count of log for the event : Navigate
14	Count of log for the event : Navigate for all previous enrolments
15	Count of log for the event : Page-close
16	Count of log for the event : Page-close for all previous enrolments
17	Count of log for the event : Access
18	Count of log for the event : Access for all previous enrolments
19	Count of log for the event : Discussion
20	Count of log for the event : Discussion for all previous enrolments
21	Count of log for the event : Wiki
22	Count of log for the event : Wiki for all previous enrolments
23	Count of logs in the first 10 days of course
24	Count of logs in the second 10 days of course
25	Count of logs in the last 10 days of course
26	Count of active minutes
27	Count of active days
	Enrolment result : Success 0 /Drop-out 1

C. Feature Selection

When solving a classification problem, processing extracted data vectors is an important step. Indeed, the performance of the classifier highly depends on the correct choice of the content of these vectors. However, the problem becomes difficult to re- solve and very expensive in terms of training time and resources owing to the large dimension of these vectors. Consequently, it is useful, and sometimes necessary to reduce the dimensionality of these vectors to be compatible with resolution methods, even if this reduction may lead to a slight loss of information. Reducing the number of explanatory variables has a double advantage. On one hand the model will be easily interpretable due to the few number of variables. On the other hand, the prediction error will be reduced by removing the non-informative variables.

Feature selection is a process allowing to select a subset of features considered as the most relevant from a starting set using various criteria and different methods. The process is working as follows:

1) From the initial set of variables, the selection process determines a subset of variables that he considers to be the most relevant.

2) The subset is then evaluated with the classifier to assess the performance and relevance of the selection.

3) Depending on the result of the evaluation, a criterion for stopping the process determines whether the subset of variables can be used in the learning process, otherwise another subset of variables is generated and tested. The stopping criteria can be a predefined number of features to keep or a fixed number of iterations or even a criterion related to the evaluation function.

Methods used for selection can be classified into three main categories: Filter, Wrapper and Embedded.

1) Filter: The filter approach was the first method for selecting features [20]. It is considered a pre-processing step before the learning phase; the evaluation of features is usually done independently of the classifier. We define an importance score for each feature that reflects its quality as a predictor. We also define a score of similarity between two characteristics. The objective is to select the variables with the highest importance score and the lowest similarity scores to reduce redundancy. The main advantage of filtering methods is their computational efficiency and robustness against overfitting. Unfortunately, these methods do not consider interactions between characteristics and tend to select characteristics involving redundant rather than complementary information. The filter method used in this work is the Chi-squared test.

2) Wrappers: The main drawback of filter approaches is ignoring the influence of the selected variables on the learning algorithm's performance. To solve this problem, Kohavi and John introduced the concept of a Wrapper for selecting features [21]. Wrappers use the accuracy of the learning algorithm as an evaluation function to estimate the relevance of the variable. The Wrapper methods are generally considered to be better than those of filtering. They can select proper small subsets of features. However, features selected by this method are only suited to the classification algorithm and are not necessarily valid if we change the classifier. Also, the complexity of the learning algorithm makes the Wrapper methods very expensive in terms of computation time. It has been demonstrated by [21] that Wrapper methods produce better performance than some filtering methods. This paper will use Recursive Feature Elimination with both Random Forest (RFE-RF) and Gradient Boosting (RFE-GB).

Recursive Feature Elimination (RFE) is a selection algorithm based on backward elimination, in which recursive elimination aim to select a subset of optimum features. The learning is performed first with all the p variables, the least discriminant variable is removed, then the learning is performed on the p-1 remaining variables. This process is iterated until the number of desired variables is obtained.

3) Embedded: Embedded methods incorporate the selection of variables into the learning process. Embedded methods can use all the dataset as a training set which is an advantage that can improve the result. In addition, Guyon and Elisseeff [22] specify that Embedded approaches surpass Wrapper approaches concerning the computation times and the robustness against over-fitting. In this study, we are using both regularization: Ridge and lasso as Embedded methods.

Regularisation in ML adds a penalty term to the different coefficients of the model. The main purpose of Regularization is to avoid overfitting by improving the generalisation of the model. It improves the performance of models on new data. the main types of regularisation are Ridge and lasso:

Lasso regression (L1): Adds the squared magnitude of coefficients as a penalty term to the loss function.

$$L + \lambda \sum_{i=0}^{n} x_i^2$$

Ridge regression (L2): Adds the absolute value of magnitude of coefficients as a penalty term to the loss function.

$$L + \lambda \sum_{i=0}^{n} |x|$$

It is therefore used for variable selection. While L1 set the coefficient of unnecessary variables to 0, L2 is approaching them to zero.

In this study, we will implement a hybrid approach that combines all the methods seen previously. Each method will participate to elect if the variable will be selected or not. The scores obtained for each method will be then aggregated and normalised so that they are between 0 for the lowest rank and 1 for the highest. Variables with a high average will be selected according to score obtained in the Table VII.

TABLE VII. Features Ranking

N	Lasso	REF-RF	REF-GB	Ridge	Chi-2	R-Lasso	Mean
1	0.0	1.0	0.7	0.0	0.0	0.48	0.36
2	0.00	0.71	0.05	0.07	1.00	0.01	0.31
3	0.00	0.29	0.75	0.21	1.00	0.91	0.53
4	0.00	0.57	0.35	0.14	1.00	0.00	0.34
5	0.48	1.00	0.90	0.39	0.97	0.00	0.62
6	0.06	1.00	0.2	0.00	1.00	0.00	0.38
7	0.00	1.00	1.00	0.05	1.00	0.89	0.66
8	0.00	1.00	0.45	0.00	1.00	0.01	0.41
9	0.00	1.00	1.00	0.01	0.99	0.02	0.50
10	0.00	1.00	0.15	0.00	1.00	0.14	0.38
11	0.00	1.00	0.80	0.00	1.00	0.34	0.52
12	0.00	1.00	0.60	0.00	1.00	0.00	0.43
13	0.00	1.00	0.55	0.01	1.00	0.62	0.53
14	0.00	0.00	0.00	0.00	1.00	0.00	0.17
15	0.00	1.00	1.00	0.02	1.00	0.13	0.53
16	0.00	1.00	0.25	0.00	1.00	0.00	0.38
17	0.00	1.00	0.95	0.00	0.99	0.28	0.54
18	0.00	1.00	0.40	0.00	1.00	0.01	0.40
19	0.00	0.14	0.10	0.00	1.00	0.00	0.21
20	0.00	0.86	0.65	0.00	1.00	0.00	0.42
21	0.00	1.00	0.30	0.04	1.00	0.02	0.39
22	0.00	0.43	0.50	0.00	1.00	0.00	0.32
23	0.00	1.00	1.00	0.40	0.99	1.00	0.73
24	0.00	1.00	0.85	0.39	0.99	0.87	0.68
25	1.00	1.00	1.00	0.38	0.99	1.00	0.90
26	0.17	1.00	1.00	0.00	0.67	1.00	0.64
27	0.00	1.00	1.00	1.00	1.00	1.00	0.83

According to the results found on the Table VII. The dimensionality of the dataset will be reduced to the eight following features:

- Count of student previous succeeded enrolments
- Count of log for the current enrolment
- Count of days between first and last log
- Count of log for the event : Access
- Count of logs in the second 10 days of course
- Count of logs in the last 10 days of course
- Count of active minutes
- Count of active days

In the next section, in order to prove the relevance of our selection, results obtained with this set of features will be compared with the results obtained using the initial set of variables.

IV. Methodology

The variable enrolment result has two alternative outcomes 1 for drop-out and 0 for success. Thus, the problem can be modelled as a binary classification. Besides, since the data used for training and testing is already labelled, the models are built with supervised learning. Fig. 1 present the methodology used in this paper.

Supervised learning begins with the training process. During training, the algorithm optimises the mapping function through a pair consisting of an input vector and the desired output value. The goal is to create a model that correctly classifies new unseen data. The predicted outputs are then compared to the accurate observation on the validation set to compute the model's performance and generalisation ability. In this study, we will implement both GA and DL to create our models.

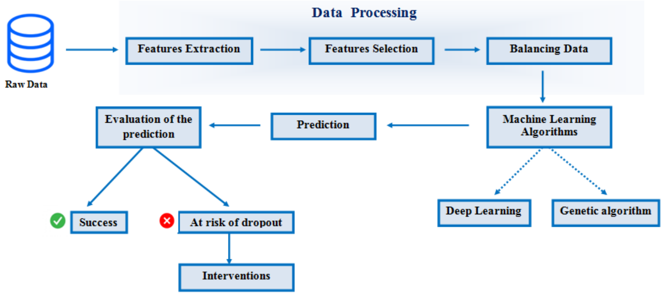


Fig. 1. Methodology.

A. Deep Learning

DL is based on the models of NN with many hidden layers, called DNN. While a traditional NN can only handle a single hidden layer as show in the Fig. 2. DL data processing is carried through multiple layers to compute the output. Each layer is made of many artificial neurons imitating the biological neurons in a very simplified way. Each connection in the network is characterised by a coefficient or a synaptic weight that mainly describes the behaviour of the network.

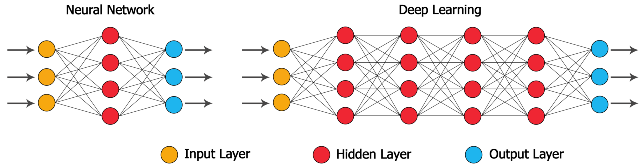


Fig. 2. Deep Learning.

During the learning process, weights are calculated in order to determine whether to amplify or dampen the output. The weights are adapted to minimise the difference between the network output and the expected output.

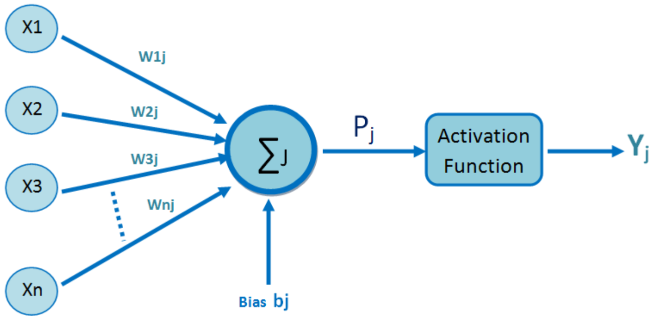


Fig. 3. Artificial Neuron.

As Fig. 3 shows, the neurons receive the information produced by other nodes through the input connections. Each neuron J performs a weighted sum of the n input values. Weights assigned to the neuron's inputs are stored in a matrix W. The value wij represents the weight of the input connection Xi of neuron j. Then to this sum a bias bj is added. This total represents the biassed post-synaptic potentials formulated as follows:

$$P_j = \sum_{i=1}^{N} w_{i,j} X_i + b_j$$

Finally, an activation function f transforms this biased potential to obtain the activation value of the neuron. This values is then transmitted to other neurons. Among the commonly used activation functions we can find Rectified linear unit (ReLU), Sigmoid and Hyperbolic tangent.

$$X_j = f(P_j)$$

B. Genetic Algorithms

GA are stochastic optimization methods belonging to the family of evolutionary algorithms. They are commonly used for resolving complex optimization and search problems. GAs are inspired by the Darwinian mechanisms of the natural evolution of biological populations and rely on derived techniques such as selection, mutation and crossing. GA use the principle of survival of individuals considered to be the strongest or best suited to the environment by combining the strengths of each individual to create the next generation considered to be a better solution to the problem. This process is repeated several times until finding individuals have genetic information that corresponds to the best solution to the problem. In general, the process of a GA as presented in Fig. 4 is based on the following phases:

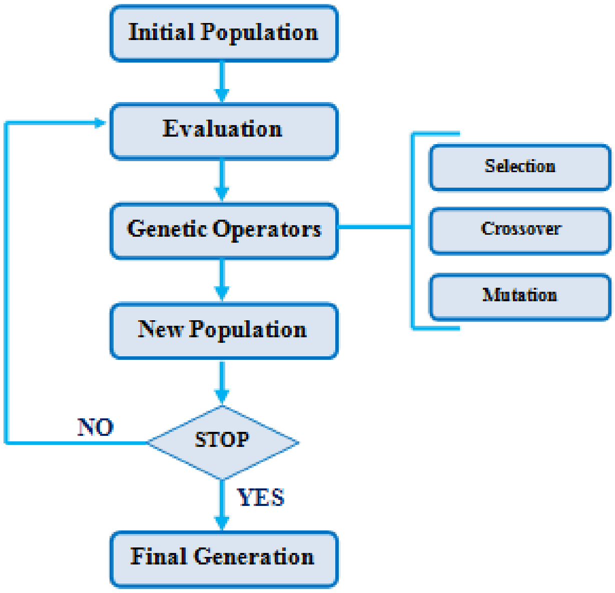


Fig. 4. Architecture of a Genetic Algorithm.

(1) Initialization: This GA mechanism must produce a non-homogeneous population of individuals who will serve as a parent for future generations. The choice of the initial population is important because it can make the convergence to the global optimum more or less rapid. The initial population must be distributed over the entire research area.

(2) Evaluation: Evaluate each capacity to the target variable with high accuracy. The LR algorithm was used to build prediction models. Individuals selected by the GA search were used as an input for LR, and the results from LR are used again with different variable sets in order to enhance the prediction score.

(3) Genetic Operators: Operators guarantee the possibility of diversifying populations over the generations and exploring the solution space. We apply the following operations during a GA cycle: Selection, Crossing and Mutation.

(a) Selection: The selection consists of choosing the individuals serving to create the next generation, the individuals who will survive . The selection of individuals is carried out most often on the basis of the evaluation function. Several selection operators are used such as the roulette wheel selection [23], Rank in the population [24] or tournament selection [25].

(b) Crossover: Crossing is responsible for constructing an individual solution for the problem from the mixture of many other solutions. In crossing, the chromosomes exchange sequences of genes between them. This process is applied to each pair of chromosomes selected with a certain probability of P. The pairs of chromosomes are copied without modification into the next generation with the probability 1 - P. The higher P, the more new individuals appear in the population.

We present the best-known ones among the most used crossover methods: the one-point crossover and the multi-point crossover.

One-Point Crossover : It is about randomly choosing a crossing point for each pair of chromosomes and performing a swap of the sets of sequences of this point between the two parents, giving birth to two new offspring as shown in Fig. 5.

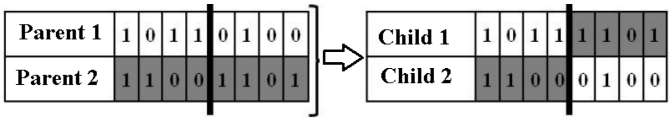


Fig. 5. Single-Point Crossover.

Multi-Point Crossover : In this case, several crossing points are selected and the swap is done on the different parts of the sequences surrounded by these points between the genes of the parents as shown in Fig. 6.

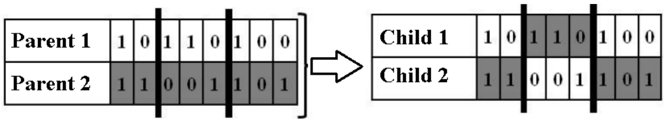


Fig. 6. Multipoint Crossover.

(c) Mutation : Mutation is defined as the unexpected change in the value of a gene in a chromosome. Fig. 7 illustrates an example of a mutation applied to the fourth position of a binary chromosome. The mutation plays the role of noise which prevents the evolution from stopping. It allows the extension of space exploration and guarantees that the global optimum can be reached. This operator, therefore, avoids a convergence towards the local optimum.

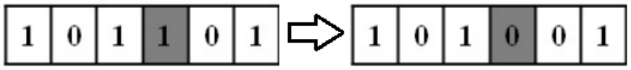


Fig. 7. Exemple of Mmutation.

The following section will be dedicated to presenting the results and scores obtained by applying the methods presented in this section on the data obtained in the fourth section .

V. Results

After performing a hybrid feature selection method in Section 4, we end up with two datasets: a first dataset containing all extracted features (All) and a second one containing only the seven selected features (selected). To assess the relevance of this selection, we will compare the Accuracy and AUC-ROC scores for the two datasets using DL and GA models.

TABLE VIII. Models Scores

Model	Features	Accuracy	AUC ROC
GA	All	0.926	0.898
	Selected	0.933	0.894
DL	All	0.943	0.876
	Selected	0.938	0.887

According to the results obtained in Table VIII, we notice that the scores remained almost the same even after eliminating several features. It is explained by the fact that some eliminated features had no role in predicting the target variable. In other cases, they may introduce noise. The score has been improved for the GA model after eliminating the unnecessary variables.

We also notice that the two algorithms used in this work outperform the basic algorithms used in previous work. It is explained by GA evolutionary and self-correcting character and DL methods.

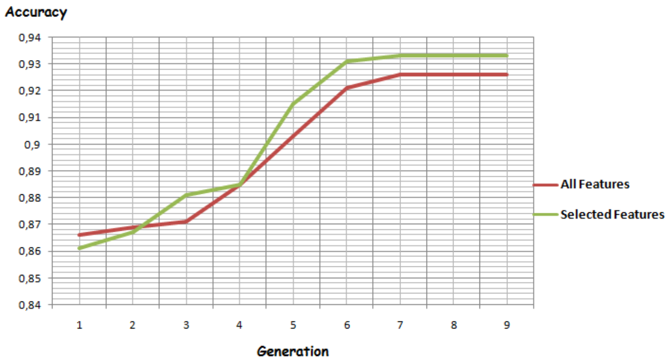


Fig. 8. Accuracy Evolution over Generations.

Fig. 8 provides information about how quickly the GA converges to the optimal solution. After only seven generations, the algorithm found an optimal solution for the problem. This rapid evolution of accuracy over generations depends highly on the value of mutation rate. In practice; it consists of a high mutation rate at the start of the algorithm to allow better solutions for space exploration. Then a decrease in this rate allows the convergence of the algorithm.

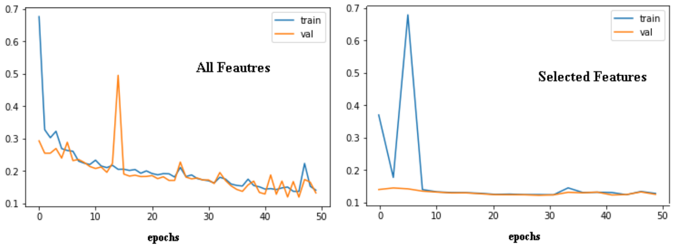


Fig. 9. Loss Evolution over Epochs

Learning curve graphs presented in Fig. 9 are commonly used for the NN model. It plots the variation of loss or accuracy over epochs. The data is divided into two parts training and validation sets. Learning curve graphs are firstly used to examine the model convergence; we expect that the loss decreases and the accuracy increase as the number of epochs increases. We also expect that the model will converge after training for several epochs. Secondly, It is used to diagnose if the model has over-fitted or under-fitted the learning set.

Fig. 9 show that we obtain an accuracy of 90% after 40 epochs for the dataset using all features. The same accuracy is obtained after only 10 epochs for the dataset using selected features, which is explained by the fact that a model containing fewer features will be less complex and require fewer epochs to converge. Fig. 9 show that for both datasets, we can safely stop the training process at 50 epochs without fearing over-fitting or under-fitting.

After proving the relevance of models and the set of features used in this article, and since the temporal aspect is present in our dataset, the next step in this work is to use ARIMA to predict independent variables for future weeks for every enrolment. After obtaining these values, we use them as an observation for the models to predict the value of the target variable Y. With the function predict.proba() in python sickit-learn library, we can find the weekly drop-out percentage.

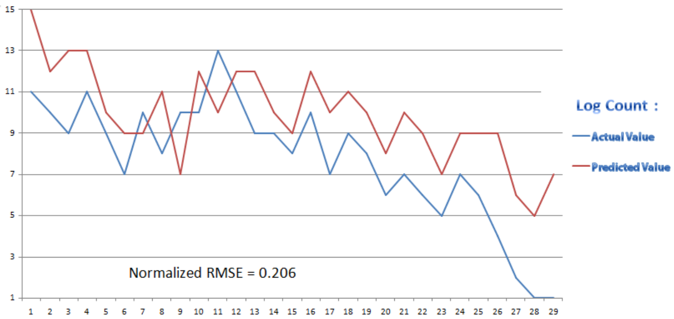


Fig. 10. Log Count Error Prediction

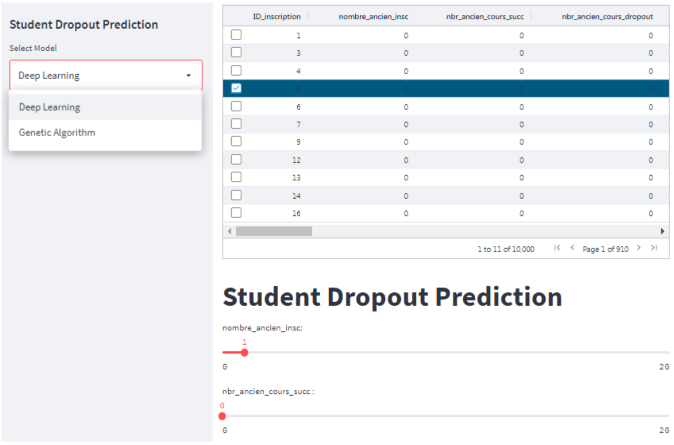
We can judge the accuracy of the ARIMA model predictions through the normalized RMSE (Root Mean Sqaured Error) value. Fig. 10 shows an example of the error between the actual value of the number of logs per day and the predicted value. The normalized RMSE value is suitable for all variables and indicates the accuracy of the ARIMA model and stationary time series of the variables.

At our university, we extracted student logs from the university MOOC to build a dataset similar to the KDD cup dataset. This dataset has undergone the same process mentioned in this article, starting with feature extraction and feature engineering and ending with models' predictions score. Table IX presents the different model scores for the university dataset.

TABLE IX. Models Scores University Data

Model	Features	Accuracy	AUC ROC
GA	All	0.931	0.876
	Selected	0.921	0.887
DL	All	0.888	0.875
	Selected	0.898	0.886

An intelligent system was hosted in the university server based on a single web page using Python and Streamlit framework. The system inputs the enrolment entry (student, course), fetches the corresponding logs and aggregates them by week. In addition, the ARIMA method is used to predict new observations for the following weeks using the previous ones. The system offers the possibility to choose the model used to predict the drop-out rate each week, as shown in Fig. 11 and Fig. 12.


Student Dropout Prediction

Select Model

Deep Learning

Deep Learning
Genetic Algorithm

	ID_inscription	nombre_ancien_insc	nbr_ancien_cours_succ	nbr_ancien_cours_dropout
□	1	0	0	0
□	3	0	0	0
□	4	0	0	0
☑	5	0	0	0
□	6	0	0	0
□	7	0	0	0
□	9	0	0	0
□	12	0	0	0
□	13	0	0	0
□	14	0	0	0
□	16	0	0	0

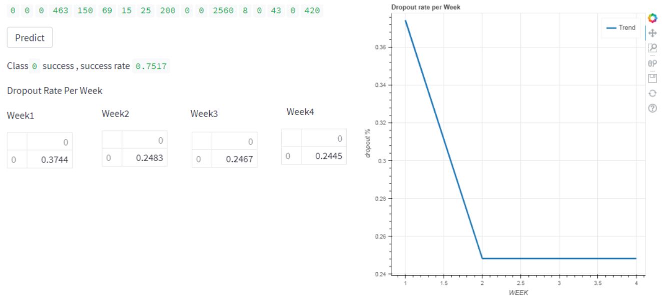
1 to 11 of 10,000 Page 1 of 910

Student Dropout Prediction

nombre_ancien_insc: 1 (0 – 20)

nbr_ancien_cours_succ : 0 (0 – 20)

Fig. 11. Web Page Interface.


0 0 0 463 150 69 15 25 200 0 0 2560 8 0 43 0 420

Predict

Class 0 success , success rate 0.7517

Dropout Rate Per Week

Week1

	0
0	0.3744

Week2

	0
0	0.2483

Week3

	0
0	0.2467

Week4

	0
0	0.2445

Fig. 12. Results Web Page.

In order to determine student drop-out profiles and patterns, We performed a correlation analysis between extracted features and the student's final performance (completion or drop-out). We found that when the pattern is: Access — Video — Assignments — Discussion is respected, The accuracy of retention is increased visibly. It means that the expected behaviour of the student is to access the course link represented by the event (Access), watch the course content through videos, and then visit the Assignments page. Finally, discuss the ambiguous and misunderstood points on the discussions page. Similarly, the student behaviour on assignments pages, the count of the viewing Video events, Discussion, and wiki page measured weekly, provided an indicator of student engagement and persistence and were an excellent early predictor of the drop-out rate and performance.

When we used K-mean clustering, we found three main clusters for droppers: According to our analysis of weeks 1 and 2, data indicates three dominant clusters.

- **cluster 1** student with little time spent watching videos. This cluster corresponds to students who didn't complete the course videos. This grouping was a strong indicator of drop-out 95% of these students did not complete the course.
- **cluster 2** concerns students who complete watching videos but have a few visits and time spent on the assignments page. In other words, these students have completed most of the course but didn't take quizzes and exercises. This

grouping was also a strong indicator of drop-out: 80% dropped out in the fourth week.
The reasons which explain the behaviour of the students of clusters 1 and 2 can be linked to several causes extracted from related works:
  - **causes linked to student:**
    - The lack of students motivation and engagement: is considered one of the most influential factors preventing students from completing a MOOC. For instance, [26] surveyed 134 students who had not completed the MOOC courses and found that the majority of students had the intention to complete their study but they were unable to do so due to low motivation and poor feedback.
    - The students lack the abilities/skills and prerequisites to follow the course. According to previous studies [27], [28], [29], demonstrating the effect of students' academic skills and abilities and their prior experience on the drop-out rate in MOOCs.
  - **causes linked to the course:**
    - It can also be linked to the course length and difficulty. According to [30], [31], [32] the complexity or the difficulty of the course content was found in many studies to be associated with students drop-out rates
  - **causes linked to instructors:**
    - The lack of instructor supervision: The poor feedback provided by the instructors due to the massive number of enrolled students per course has been reported to be an significant predictor of students drop-out in MOOC courses [33] and [34].

- **cluster 3** concerns students who have spent an average time on the videos and assignments page but have few visits to the discussion page. It can be interpreted that those students are not social enough to communicate with others students or ask questions about ambiguous points in courses or exercises.
  Isolation and lack of interactivity in MOOCs directly affect students drop-out. A survey [29] about MOOCs drop-out showed from the droppers' comments that they mentioned feeling isolated and unmotivated to continue due to low interaction and communication with students and instructors. They complained that the instructor did not praise or motivate them after the quizzes. They also stated that instructors did not engage learners in discussion or facilitate brainstorming.

After detecting patterns of the student at high risk of drop-out, the following section outlines examples of interventions:

- **Interventions for cluster 1 and 2:** For students lacking the necessary prerequisites to take the course, it is wise to detect them through a survey or quizzes at the beginning of each course. Then send them courses and exercises containing the prerequisites they lack to follow the course. According to the literature, there is a myriad of interventions aiming to increase students motivation and engagement by creating interest in the course topics [35]. Some interventions dealt with demotivation through an email mechanism [36]. Other interventions try to get absent students back into the course and collect their reasons for leaving [37].
  The lack of instructor supervision and the poor instructor feedback is due to the considerable enrolment number per course in MOOCs. The solution here is to focus on students flagged at high risk of drop-out on a particular week according to the drop-out rate given by the system.
  Regarding the course content, the course must appear helpful for the students in real life. It should contain a lot of application and practical exercises. The skills learned in the course must apply to real-world problems, particular career goals, or later life roles.
- **Interventions for cluster 3:** Multiple research has found that social connectedness to school is linked to higher rates of student academic success [38], teachers and peers can serve as sources for facilitating this social connection. The intervention proposed for this cluster is a weekly peer-support group meeting that focuses on enhancing students' academic and interpersonal skills combined with daily interactions. It will improve outcomes for students flagged as a potential drop-out. We could form a peer-support group of three to four participants, and the 5th is the student flagged as a drop-out. This methodology is more suitable for blended learning; it has been tested in the university, and the results obtained improved classroom behaviour, increased academic engagement, and positive peer and teacher interactions.

The cost of a students drop-out is very high in terms of wasted time, effort and money. When a student decides to leaves, connection with that student is lost, and generally nothing is done to determine the reasons behind. Institutions can implement this system or similar, to anticipate and reduce the number of drop-out.

Several other drop-out patterns might be detected, and such predefined intervention strategies can be learned from expert teachers or from historical data [12]. The current study is just a first step toward an ultimate automated personalized intervention system.

From an algorithmic perspective, the experiment in this study showed that deep learning is outperforming other baseline algorithms either in prediction accuracy or in generating more accurate drop-out probabilities. Moreover, deep learning showed more robustness against over-fitting.

VI. Conclusion

The excessive drop-out rate in MOOCs encourage to use data mining techniques and ML algorithms in order to predict students at risk of drop-out. In the light of the obtained results we can conclude that for classification problems based on raw activity records, features extraction and data preparation is a necessary step before building models. The hybrid features selection algorithm adopted in this work is effective. One of our main contributions was obtaining competitive prediction results with a minimum number of variables.

According to the result obtained we can also conclude that our proposed models using GA and DL are producing very competitive results in this problem. Models used in this study are outperforming the obtained result using the baseline algorithms in previous works. This study was very useful, and optimises the drop-out prediction in the university MOOC, because it is not only focusing on early detecting students at risk of drop- out, but it also personalise intervention and seeks for the reasons behind, in order to increase retention rate in the MOOC. As a perspective, the methodology used in this article must be tested on other benchmark datasets in order to assess its relevance.

Acknowledgment

We would like to gratefully acknowledge the organizers of KDD Cup 2015 as well as XuetangX for making the datasets available.

References

[1] D. Lizcano, J. A. Lara, B. White, and S. Aljawarneh, "Blockchain-based approach to create a model of trust in open and ubiquitous higher education," *Journal of Computing in Higher Education*, vol. 32, no. 1, pp. 109–134, 2020.

[2] D. Shah, "The second year of the mooc: A review of mooc stats and trends in 2020," *The Report by class central [Electronic resource]. URL: https://www. classcentral. com/report/the-second-year-of-the-mooc/(accessed: 09.11. 2021)*, 2020.

[3] F. Dalipi, A. S. Imran, and Z. Kastrati, "Mooc dropout prediction using machine learning techniques: Review and research challenges," in *2018 IEEE global engineering education conference (EDUCON)*. IEEE, 2018, pp. 1007–1014.

[4] C. Bossu and T. Heck, "Engaging with open science in learning and teaching," *Education for Information*, vol. 36, no. 3, pp. 211–225, 2020.

[5] R. Yu, H. Lee, and R. F. Kizilcec, "Should college dropout prediction models include protected attributes?" in *Proceedings of the eighth ACM conference on learning@ scale*, 2021, pp. 91–100.

[6] A. F. Wise, S. Knight, and S. B. Shum, "Collaborative learning analytics," in *International handbook of computer-supported collaborative learning*. Springer, 2021, pp. 425–443.

[7] C. F. de Oliveira, S. R. Sobral, M. J. Ferreira, and F. Moreira, "How does learning analytics contribute to prevent students' dropout in higher education: A systematic literature review," *Big Data and Cognitive Computing*, vol. 5, no. 4, p. 64, 2021.

[8] J. Figueroa-Cañas and T. Sancho-Vinuesa, "Changing the recent past to reduce ongoing dropout: an early learning analytics intervention for an online statistics course," *Open Learning: The Journal of Open, Distance and e-Learning*, pp. 1–18, 2021.

[9] P. M. Moreno-Marcos, C. Alario-Hoyos, P. J. Muñoz-Merino, and C. D. Kloos, "Prediction in moocs: A review and future research directions," *IEEE transactions on Learning Technologies*, vol. 12, no. 3, pp. 384–401, 2018.

[10] N. Gitinabard, F. Khoshnevisan, C. F. Lynch, and E. Y. Wang, "Your actions or your associates? predicting certification and dropout in moocs with behavioral and social features," *arXiv preprint arXiv:1809.00052*, 2018.

[11] J. Berens, K. Schneider, S. Görtz, S. Oster, and J. Burghoff, "Early detection of students at risk–predicting student dropouts using administrative student data and machine learning methods," *Available at SSRN 3275433*, 2018.

[12] W. Xing and D. Du, "Dropout prediction in moocs: Using deep learning for personalized intervention," *Journal of Educational Computing Research*, vol. 57, no. 3, pp. 547–570, 2019.

[13] Z. Alj and M. O. C. Malki, "Predicting students drop-out in mooc from learning behavior using machine learning," *Journal of Uncertain Systems*, p. 2250011, 2022.

[14] R. Munoz, R. Olivares, C. Taramasco, R. Villarroel, R. Soto, T. S. Barcelos, E. Merino, and M. F. Alonso-Sánchez, "Using black hole algorithm to improve eeg-based emotion recognition," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[15] R. Olivares, R. Munoz, R. Soto, B. Crawford, D. Cárdenas, A. Ponce, and C. Taramasco, "An optimized brain-based algorithm for classifying parkinson's disease," *Applied Sciences*, vol. 10, no. 5, p. 1827, 2020.

[16] Y. Xue, H. Zhu, J. Liang, and A. Słowik, "Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification," *Knowledge-Based Systems*, vol. 227, p. 107218, 2021.

[17] M. LLC. (1999) MS Windows NT kernel description. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

[18] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[19] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[20] X. Geng, T.-Y. Liu, T. Qin, and H. Li, "Feature selection for ranking," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 407–414.

[21] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[22] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[23] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 69–93.

[24] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 94–101.

[25] B. L. Miller, D. E. Goldberg *et al.*, "Genetic algorithms, tournament selection, and the effects of noise," *Complex systems*, vol. 9, no. 3, pp. 193–212, 1995.

[26] C. Gütl, R. H. Rizzardini, V. Chang, and M. Morales, "Attrition in mooc: Lessons learned from drop-out students," in *International workshop on learning technology for education in cloud*. Springer, 2014, pp. 37–48.

[27] H. B. Shapiro, C. H. Lee, N. E. W. Roth, K. Li, M. Çetinkaya-Rundel, and D. A. Canelas, "Understanding the massive open online course (mooc) student experience: An examination of attitudes, motivations, and barriers," *Computers & Education*, vol. 110, pp. 35–50, 2017.

[28] M. Yamba-Yugsi and S. Luján-Mora, "Cursos mooc: factores que disminuyen el abandono en los participantes," *Enfoque UTE*, vol. 8, pp. 1–15, 2017.

[29] K. S. Hone and G. R. El Said, "Exploring the factors affecting mooc retention: A survey study," *Computers & Education*, vol. 98, pp. 157–168, 2016.

[30] G. R. El Said, "Understanding how learners use massive open online courses and why they drop out: Thematic analysis of an interview study in a developing country," *Journal of Educational Computing Research*, vol. 55, no. 5, pp. 724–752, 2017.

[31] T. Eriksson, T. Adawi, and C. Stöhr, "“time is the bottleneck”: a qualitative study exploring why learners drop out of moocs," *Journal of Computing in Higher Education*, vol. 29, no. 1, pp. 133–146, 2017.

[32] S. Zheng, M. B. Rosson, P. C. Shih, and J. M. Carroll, "Understanding student motivation, behaviors and perceptions in moocs," in *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, 2015, pp. 1882–1895.

[33] S. Halawa, D. Greene, and J. Mitchell, "Dropout prediction in moocs using learner activity features," *Proceedings of the second European MOOC stakeholder summit*, vol. 37, no. 1, pp. 58–65, 2014.

[34] D. F. Onah, J. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: behavioural patterns," *EDULEARN14 proceedings*, vol. 1, pp. 5825–5834, 2014.

[35] T. NeCamp, J. Gardner, and C. Brooks, "Beyond a/b testing: sequential randomization for developing interventions in scaled digital learning environments," in *Proceedings of the 9th International Conference on learning analytics & knowledge*, 2019, pp. 539–548.

[36] I. Borrella, S. Caballero-Caballero, and E. Ponce-Cueto, "Predict and intervene: Addressing the dropout problem in a mooc-based program," in *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, 2019, pp. 1–9.

[37] J. Whitehill, J. Williams, G. Lopez, C. Coleman, and J. Reich, "Beyond prediction: First steps toward automatic intervention in mooc student stopout," *Available at SSRN 2611750*, 2015.

[38] L. Bond, H. Butler, L. Thomas, J. Carlin, S. Glover, G. Bowes, and G. Patton, “Social and school connectedness in early secondary school as predictors of late teenage substance use, mental health, and academic outcomes,” *Journal of adolescent health*, vol. 40, no. 4, pp. 357–e9, 2007.