Intelligent Decision Technologies 1–14
IOS Press

A semi-supervised self-trained two-level algorithm for forecasting students' graduation time

I.E. Livieris [a,*], V. Tampakas [a] N. Karacapilidis [b] P. Pintelas [c]
[a] *Department of Computer & Informatics Engineering (DISK Lab), Technological Educational Institution of Western Greece, Greece, GR 263-34.*
[b] *Department of Mechanical Engineering & Aeronautics, University of Patras, GR 265-00, Greece.*
[c] *Department of Mathematics, University of Patras, GR 265-00, Greece.*

Abstract. During the last decades, educational data mining has become a significant tool for the prediction of students' progress and performance. In this work, we present a new semi-supervised self-trained two-level classification algorithm for predicting students' graduation time. The proposed algorithm has three major features: Firstly, it identifies with high accuracy the students at-risk of not completing their studies; secondly, it classifies the students based on their expected graduation time; thirdly, it meaningfully relates the explicit classification information of labeled data with the information hidden in the unlabeled data. Our preliminary numerical experiments indicate that the proposed algorithm exhibits reliable predictions based on the students' performance during the first two years of their studies.

Keywords: Data mining, machine learning, educational data, prediction model, semi-supervised learning, self-labeled algorithms, two-level classification algorithm, student academic performance.

1. Introduction

Higher education constitutes a critical factor in human resources development, increasing people's knowledge and competencies and ensuring nations' economic prosperity. The main objective of a higher education institute and one of its biggest challenges is to provide quality education to its students. In 2001, a *National Research Council report* [10] illustrated the immediate need to develop innovative methodologies to assist higher institutes, which will further improve the quality of their studies, facilitate students' timely graduation and limit their dropout. To achieve a higher level of studies' quality, one should consider three key aspects: the first two put emphasis on refining teaching and knowledge acquisition methods, while the third one concerns the development of efficient systems for monitoring students' progress and identifies key aspects of their success.

Nowadays, student retention of successful and on-time graduation constitutes an enduring issue in higher education [2]. More specifically, recent studies have shown that only a minority of students has successfully completed a four-year bachelor program on time [17,39,49]. Some of the causes which significantly affect students' progress are the credits lost in transfer, the inability to register for required courses and the remediation sequences that usually do not work. As the cost of higher education (fees, living expenses, etc) has been significantly increased during the past decade, prolonged graduation time becomes a crucial factor in discouraging students, ultimately leading them to dropout.

A crucial step towards effective intervention is the development of a system which can accurately predict students' graduation time through the continu-

*Corresponding author. E-mail: livieris@teiwest.gr.

ous monitoring of their academic progress. Indeed, the ability to accurately predict students' future performance is essential for effectively carrying out necessary pedagogical interventions to ensure students' on-time graduation. Furthermore, by analyzing students' progress, appropriate strategies and actions could be better planned by the educational institutions, in order to decrease the students' mean graduation time and limit dropout.

Over the last decades, a large amount of students' records have been maintained and accumulated by the educational institutes. The application of machine learning and data mining techniques on these records is admittedly valuable, offering a first step in extracting useful and novel information in order to gain a deeper insight in the prediction of students' progress and performance. Along this line, many researchers have conducted studies in order to cluster students based on their academic progress and identify the key features which affect their performance. Nevertheless, most of these studies examine the efficiency of supervised classification methods, while semi-supervised methods [23,24,40] have been rarely applied to the educational field. Generally speaking, Semi-Supervised Learning (SSL) methods are prominent machine learning techniques which attempt to achieve strong generalization by efficiently combining the explicit classification information of labeled data with the information in the unlabeled data [32,44,55,56].

Furthermore, although the prediction of students performance in a course has been extensively studied in the literature [7,22–26,28,37,42] the early prediction of their graduation time is a completely different task which imposes a set of new challenges. This is mainly due to four factors: firstly, students attend many courses during their studies but not all courses are equally informative about their graduation; secondly, predictions need to be made based on the evolution of students' progress, not only on their most recent accomplishments; thirdly, students differ in terms of knowledge backgrounds and specializations; fourthly, students usually select the courses with different sequence. By also taking into account the expanding volume of data, from the increasing student enrollments and by the continually shifting performance during their studies, we conclude that the task of predicting students' graduation time becomes a very complicated, attractive and challenging task.

In this work, we present a new semi-supervised algorithm for the accurate prediction of the students' graduation time, which is based on the adaptation of a novel two-level classification scheme [22,25] as a base learner in the self-training framework. Our prediction model identifies students at-risk of not graduating within six years of studies or dropout and classifies the students based on their expected graduation time. The preliminary numerical experiments indicate that the proposed algorithm exhibits reliable predictions based on courses' characteristics, students' demographic information and their performance in the courses attended during the first two years of their studies. In Greece, average university studies are designed to last four years (8 academic semesters) except engineering, architecture and medicine which last longer. Nevertheless, since there is no limit in graduation time, a student may graduate in more than four years. In this study, we set the maximum graduation time to six years.

The remainder of this paper is organized as follows: The next section presents a survey of recent studies concerning the application of data mining in education. Section 3 defines the semi-supervised classification problem and the self-training approach. Section 4 presents a detailed description of the data collection and data preparation used in our study and the proposed semi-supervised two-level classification algorithm. Section 5 presents a series of experiments carried out in order to examine and evaluate the accuracy of the proposed algorithm against the most popular SSL classification algorithms. Finally, Section 6 provides concluding remarks and sketches future work directions.

2. Related studies

During the last decades, the development and adoption of machine learning systems for predicting students' performance has gained popularity, addressing many issues in the educational domain and providing useful outcomes about the learning process and students' behavior. In the literature, a variety of research studies has been conducted to predict students' academic performance either to determine students at-risk or to facilitate curriculum planning. Some extensive reviews [3,8,12,31,35] present the chronicles of recent educational data mining advances and developments and analyze the outcomes produced by a machine learning approach. Additionally, they describe in detail the most accurate prediction models utilized for gaining significant insights on students' behavior, interactions and progress, and summarize the diverse

factors which influence students' future performance. A number of studies have been carried out in recent years; some useful outcomes of them are briefly presented below.

Nagy et al. [29] developed an intelligent student advisory framework to provide pieces of consultations to a first year university student to pursue a certain education track where he/she will likely succeed in, aiming to decrease the high rate of academic failure among students. This framework acquired information from a database containing the students' academic achievements before enrolling to higher education together with their first year grade after enrolling in a certain department. After acquiring the relevant information, the intelligent system provides recommendations for a certain department for a new student utilizing both classification and clustering techniques. Furthermore, the authors presented a case study using students' data collected from Cairo Higher Institute for Engineering, Computer Science and Management department during the period 2000-2012, to prove the efficiency of the proposed framework.

Iam-On et al. [16] present a prediction model for forecasting students' dropout prediction in Mae Fah Luang University using ensemble of mixed-type data clusterings. Their main goal was to disclose interesting patterns, which could contribute to predicting student performance and dropout, based on their pre-university characteristics, admission details and initial academic performance at university. The dataset utilized in their study consists of students' demographic detail, academic performance and enrollment records. Their numerical results revealed that their proposed approach is more effective than several benchmark transformation techniques, across different classifiers.

Saa [37] studied the identification of multiple factors which theoretically influence and affect low academic students' performance in higher education and concluded to some interesting results. More specifically, the author stated that the students' performance is not totally dependent on their academic efforts; there are many other personal and social factors that have equal or greater influences as well. Moreover, he developed a qualitative model that classifies students based on their academic achievements and provides reliable predictions.

Along this line, Yassein et al. [52] utilized machine learning and data mining techniques to deeply analyze students' data and identify features affecting student performance in selected courses in Najran University in Saudi Arabia. More specifically, they studied the relationship between both practical work and assignments in several courses and students' success rate. Their results revealed the strong relationship between these factors; in addition, it was found that a large number of given assignments acts negatively on course academic performance.

Xu et al. [49] developed a novel machine learning method based on students' progressive performance for predicting students performance in degree programs. The dataset used in their study contained 1169 undergraduate students over three years from Mechanical and Aerospace Engineering department at UCLA. Their proposed method adopts a latent factor model-based course clustering method developed to discover relevant courses for constructing base predictors, while an ensemble-based progressive prediction architecture was developed to incorporate students' evolving performance into the prediction. Their experimental results demonstrated the effectiveness of their proposed method, achieving superior performance to benchmark approaches.

Burgos et al. [7] utilized knowledge discovery techniques to analyse historical student course grade data in order to predict whether or not a student will drop out of a course. They utilized logistic regression models and the numerical experiments conducted with data on over 100 students for several distance learning courses, confirmed the efficacy of their predictive models. Moreover, their predictive models have been integrated in a special-purpose tutoring action planning to enhance the quality of the educational content thus limiting students' dropout of a course. The application of the plan managed to reduce the dropout rate by 14% with respect to previous academic years in which no dropout prevention mechanism was applied.

In more recent works, Livieris et al. [23,24] evaluated the performance of several semi-supervised learning algorithms for predicting the students performance in the final examinations. They also presented a case study utilizing a dataset concerning the performance of 3716 secondary school students over ten years. Their preliminary numerical experiments indicated that semi-supervised methods can develop reliable prediction models which achieve very good classification accuracy by utilizing a few labeled along with many unlabeled data. Based on the previous works, Tampakas et al. [41] presented a two-level classification algorithm for predicting students' graduation time. Additionally, they presented some interesting experimental results indicating that their proposed algorithm exhibits reliable predictions based on the students' per-

formance in their courses during the first two years of their studies.

3. A review of semi-supervised classification via self-training approach

This section provides the definitions and necessary notations for the semi-supervised classification problem and briefly describes the semi-supervised self-training algorithm.

3.1. Semi-supervised classification

Suppose that X denotes the domain of instances, in which a single instance $x \in X$ is represented as $x = \{x_1, x_2, \ldots, x_d, y\}$ with x belonging to a class y and a d-dimensional space in which x_i is the value of the i-th feature. Then, let us assume that the training set $L \cup U$ consists of a labeled set L of N_L instances where y is known and of an unlabeled set U of N_U instances where y is unknown with $N_L \ll N_U$. Moreover, there exists a test set T of N_T unseen instances where y is unknown, which has not been utilized in the training stage. Notice that the basic aim of the semi-supervised classification is to obtain an efficient learning hypothesis utilizing the instances in the training set.

3.2. Self-training algorithm

Self-training is a wrapper based semi-supervised approach which constitutes an iterative procedure of self-labeling unlabeled data and it is generally considered to be a simple and effective SSL algorithm. According to Ng and Cardie [30] "*self-training is a single-view weakly supervised algorithm*" which is based on its own predictions on unlabeled data to teach itself. Nowadays, it has been established as one of the most popular and frequently utilized semi-supervised algorithms due to its simplicity of implementation and its classification efficacy [24,27,38].

In the self-training framework, an arbitrary classifier is initially trained with a small amount of labeled data, which constitutes its training set. Subsequently, at each iteration the classifier's training set is augmented gradually with classified unlabeled instances which have achieved a probability value over a defined threshold c and thus are considered sufficiently reliable to be added to the training set. In other words, it is iteratively enlarging its training labeled set with its own most confident predictions and retrained, aiming to increase its accuracy by exploiting unlabeled points. Notice that the way in which the confidence predictions are measured is dependant on the type of utilized base learner [45]. A high-level description of Self-training algorithm is presented in Algorithm 1.

Algorithm 1: Self-training

Input: L – Set of labeled instances.
U – Set of unlabeled instances.
c – Confidence level.
C – Base learner.

Output: Trained classifiers C.

repeat
  Train C on L.
  Apply C on U.
  Select instances for which the classifier's confidence is more than c per iteration (x_{MCP}).
  Remove x_{MCP} from U and add to L.
until some stopping criterion is met or U is empty.

Clearly, this model accepts that its own predictions tend to be correct without making any specific assumptions for the input data. Therefore, since the success of the self-training algorithm is heavily depended on the newly-labeled data based on its own predictions, its weakness is that erroneous initial predictions will probably lead the classifier to generate incorrectly labeled data [55].

Moreover, in the original description of the self-training approach [51], the iterative process was repeated until all the instances from U are labeled and added to L. Nevertheless, Blum and Mitchell [6] established a limit to the number of iterations which has been adopted by many researchers [24,43–45].

4. Research methodology

The main goal of the research described in this paper is the development of a prediction model for the early identification of students at-risk of not completing their studies within six years and accurately classifying the students who are about to successfully graduate. We have adopted a two-stages methodology, where the first stage concerns data collection and data preparation, while the second one deploys the proposed semi-trained two-level classification algorithm.

4.1. Dataset

We have utilized a dataset concerning 282 student records from the School of Health & Social Welfare of Technological Institute of Western Greece over four years (2010-2013). The data consists of demographic information as well as information of the students' performance in courses of the first two years of their studies. It is worth to note that the Bachelor's degree program consists of four (4) academic years (eight semesters). Each record comprised 127 variables divided in two groups: the "*Demographic-based group*" and the "*Performance-based group*".

The Demographic-based group represents attributes concerning students' gender, age, home location and type of high school, which are presented in Table 1. Notice that, the reason why most researchers utilize this demographic information is because the students with different demographic features usually exhibit different styles of learning process [5].

Attribute	Values
Genger	male/female
Age	integer
Home location	nominal
High school type	technical/general/evening

Table 1
Demographic-based group attributes

Attribute	Values
Type of course	core/laboratory/clinical
Number of times examined	integer
Final grade in the course	integer

Table 2
Performance-based group attributes

The Performance-based group represents attributes concerning courses characteristics and students' progress in several courses. More analytically, the Bachelor's program in the first two years includes twenty five (25) core courses, twelve (12) laboratory courses and four (4) clinical ones. For each course, we register its type (core/laboratory/clinical), the number of times the student has taken exams to pass the course during the first three years of his/her studies and the final grade (Table 2). It is noted that in case, the student had not successfully passed the course, the grade assigned is -1.

Finally, the students were classified utilizing a four-level classification scheme, based on the years needed to complete their studies, namely $\{$*4 years*, *5 years*, *6 years*, *Fail*$\}$; the class distribution is illustrated in Figure 1.

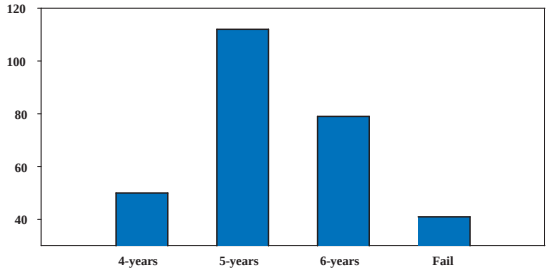

Fig. 1. Class distribution

4.2. A semi-supervised self-trained two-level classification algorithm

The proposed algorithm is based on the adoption of a two-level classification scheme as a base learner in the self-training framework. Generally, two-level classification schemes are heuristic pattern recognition tools which are supposed to yield better classification accuracy than single-level ones at the expense of a certain complication of the classification structure [18,22,25,50].

Next, we present a detailed description of the two-level classifier which constitutes the base learner in the proposed self-training framework. The first level of classification scheme utilizes a classifier to distinguish the students who are likely to "*Graduate*" or "*Fail*". More analytically, this classifier predicts whether the student will manage to complete his/her studies within six years. Clearly, the primary goal of the classifier in this level is to identify the students who are at-risk of not completing their studies. In the rest of this work, we refer to this classifier as A-level classifier. In case the verdict (or prediction) of the A-level classifier is "*Graduate*", a second-level classifier is utilized to conduct a more specialized decision and distinguish between "*4 years*", "*5 years*" and "*6 years*" to finish his/her studies. We refer to this classifier as B-level classifier. An overview of our two-level base learner is depicted in Figure 2.

Moreover, it is worth to mention that the corresponding training labeled sets L_{A} and L_{B} of the A-level and B-level classifier, are generated by the original training set L as follows: Let (x,y) be an instance contained in the training labeled set L, where x stands for the vector of attributes while y stands for the out-

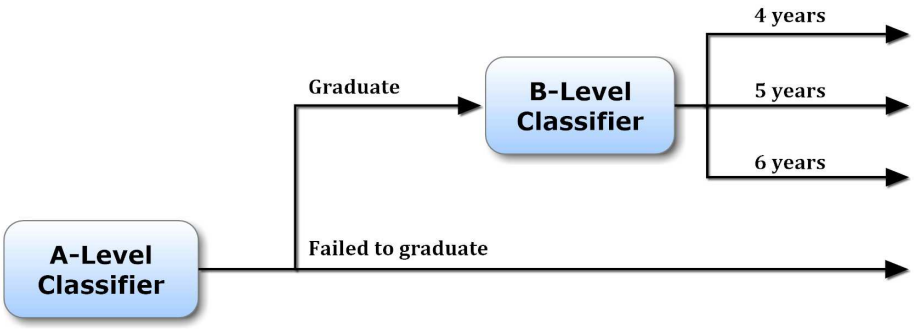

Fig. 2. An overview of the two-level classifier

put variable. In case, $y =$ "*Fail*" then it is immediately imported to the training set $L_{\rm A}$. In contrast, in case $y \neq$ "*Fail*" then the instances $(x,$"*Graduate*"$)$ and (x,y) are imported in the training sets $L_{\rm A}$ and $L_{\rm B}$, respectively.

In fact, $L_{\rm A}$ is composed of all the instances in which the respective output variable y is "*Fail*" and the rest instances of the training set with the output variable y changed into "*Graduate*"; while $L_{\rm B}$ is constituted by all the instances of the training set in which the respective output variable y is "*4 years*", "*5 years*" and "*6 years*". A high level description is presented in Algorithm 2.

Algorithm 2: Train2Level

Input: C_A – User selected A-level base learner.
C_B – User selected B-level base learner.
L – Set of training labeled instances.

Output: Trained classifiers C_A and C_B.

/* Construction of training sets L_A and L_B */
1: Set $L_A = \emptyset$ and $L_B = \emptyset$.
2: **for** each $(x,y) \in L$ **do**
3: **if** $(y ==$ "Fail") **then**
4: $L_A = L_A \cup \{(x,y)\}$.
5: **else**
6: $L_A = L_A \cup \{(x,$"Graduate"$)\}$.
7: $L_B = L_B \cup \{(x,y)\}$.
8: **end if**
9: **end for**

/* Training phase */
10: Train classifier C_A on labeled set L_A.
11: Train classifier C_B on labeled set L_B.

Remarks: For each instance (x,y) in the training labeled set L, x stands for the vector of attributes while y stands for the output variable i.e. $y \in \{$"4 years", "5 years","6 years","Fail"$\}$.

In the sequel, we describe the proposed semi-supervised algorithm which incorporates the presented two-level classifier as a base learner in the self-training framework. Initially, the two-level classifier is trained on the training labeled set L. Next, the base learner is applied to the unlabeled set U in order to classify the unlabeled instances, aiming to enlarge L with its own most confident predictions in the following way:

Suppose that (x_u, y_u) is an unlabeled instance contained in U. If the instance is predicted as "*Fail*" by the A-level classifier (Step 5), then in case its estimated confidence is over a predefined threshold c_1, then the instance $(x_u,$"Fail"$)$ is added to the labeled set L (Steps 6-9), otherwise it is omitted. If the unlabeled instance is predicted as "*Graduate*" by the A-level classifier (Step 11), then if the estimated confidence of A-level classifier is over the predefined threshold c_1 (Step 12), then the instance is considered to be labeled by the B-labeled classifier. More specifically, in case the estimated confidence of B-level classifier is over another predefined threshold c_2, then the instance is considered sufficiently reliable to be added to the labeled training set L, labeled by the B-level classifier (Steps 13-17); otherwise it remains in the unlabeled set U. It is worth noticing that in case an unlabeled instance is not labeled by the two-level base learner then it still remains in the unlabeled set U and it is re-examined in the next iteration. Moreover, the way the confidence predictions are measured in Steps 6, 11 and 13, depends on the type of the used base learner (see [4,14,45] and the references there in).

A high level description of the proposed semi-supervised Self-Trained Two-Level algorithm (STTL) algorithm is presented in Algorithm 3.

Algorithm 3: Self-Trained Two-Level algorithm (STTL)

Input: L – Set of labeled instances (Training labeled set).
U – Set of unlabeled instances (Training unlabeled set).
C_A – User selected A-level base learner.
C_B – User selected B-level base learner.
c_1 – Confidence level of A-level base learner.
c_2 – Confidence level of B-level base learner.

Output: Trained two-level classifier.

repeat
  Train2Level(C_A, C_B, L). (Algorithm 2).
  for each $(x_u, \hat{y}_u) \in U$ **do**
    $\hat{y}_A = C_A(x)$. (Apply C_A on x)
    if ($\hat{y}_A ==$ "Fail") **then**
      if ($Conf(C_A, x_u) > c_1$) **then**
        $L = L \cup \{(x_u, \hat{y}_A)\}$.
        $U = U - \{(x_u, \hat{y}_u)\}$
      end if
    end if
    if ($\hat{y}_A ==$ "Graduate") **then**
      if ($Conf(C_A, x_u) > c_1$) **then**
        $\hat{y}_B = C_B(x)$. (Apply C_B on x)
        if ($Conf(C_B, x_u) > c_2$) **then**
          $L = L \cup \{(x_u, \hat{y}_B)\}$.
          $U = U - \{(x_u, \hat{y}_u)\}$
        end if
      end if
    end if
until some stopping criterion is met or U is empty.

Remarks: For each instance $(x_u, \hat{y}_u)$ in the unlabeled set U, $\hat{y}_A$ and $\hat{y}_B$ stand for the predicted output variable by the A-level and B-level classifier on x_u, respectively; while $Conf(C_A, x_u)$ and $Conf(C_B, x_u)$ stand for the estimation confidence of A-level and B-level over instance x_u, respectively.

5. Experimental results

In this section, we report on a series of experiments carried out to explore the efficiency of the proposed semi-supervised self-trained two-level classification algorithm. The parameters of the proposed algorithm STTL were set $c_1 = 0.95$ and $c_2 = 0.8$ while all base learners were used with their default parameter settings included in the WEKA 3.9 Machine Learning Toolkit for minimizing the effect of any expert bias.

Our experimental results are obtained in three distinct phases: In the first phase, we evaluate the classification performance of the proposed self-labeled two-level classification algorithm against the classical Self-training algorithm; in the second phase, we explore the classification efficiency of the proposed algorithm utilizing different learners as A-level and B-level classifiers; while in the third phase, we evaluate its performance with that of the most popular and commonly used self-labeled algorithms and supervised algorithms.

The following supervised classification algorithms were utilized as A-level and B-level classifiers: Naive Bayes (NB) algorithm was the representative of the Bayesian networks [11] while the Multi-Layer Perceptron (MLP) [36] was the representative of the artificial neural networks which has been established as well-known learning algorithm for building and training a neural network [19]. From the support vector machines, we have selected the Sequential Minimal Optimization (SMO) algorithm since it is one of the fastest training methods [33] while *k*NN algorithm [1] was selected as instance-based learner with Euclidean distance as distance metric. From the decision trees, C4.5 algorithm [34] was chosen for our study and RIPPER (JRip) algorithm [9] was selected as typical rule-learning technique since it is probably the most frequently utilized method for producing classification rules. Studies have shown that the above classifiers constitute some of the most effective and widely used algorithms [48] for classification problems.

The implementation code was written in Java using the WEKA Machine Learning Toolkit [14] and the classification accuracy was evaluated using the stratified 10-fold cross-validation i.e. the data was separated into ten folds so that each fold had the same distribution of grades as the entire data set. Furthermore, the training partition was divided into labeled and unlabeled subsets. Similar to [44,47] in the division process, we do not maintain the class proportion in the labeled and unlabeled sets since the main aim of semi-supervised classification is to exploit unlabeled data for better classification results. Hence, we use a random selection of examples that will be marked as labeled instances, and the class label of the rest of the instances will be removed. Moreover, we ensure that every class has at least one representative instance. In order to study the influence of the amount of labeled data, three different ratios of the training data were used: 20%, 30% and 40%.

5.1. First phase of the experiments

In the sequel, we evaluate the performance of the proposed algorithm STTL against Self-training using

several base learners in terms of accuracy. Accuracy constitutes probably the most frequently used measure for assessing the overall effectiveness of an algorithm and it is defined as the ratio of correct predictions of a classification model, namely

$$A_C = \frac{\text{number of Correctly classified students}}{\text{total number of students}}$$

Table 3 summarizes the accuracy of the proposed two-level algorithm and Self-training, relative to all labeled ratio. Notice that for each utilized base-learner, the highest accuracy is highlighted in bold. Clearly, the aggregated results present that the proposed algorithm is by far the most effective one, independent of the ratio of labeled instances. Moreover, the self-trained two-level algorithm presents the highest classification accuracy in all cases except the one using SMO as base learner for ratio 20% and 30%.

Base	Ratio = 20%		Ratio = 30%		Ratio = 40%	
learner	Self	STTL	Self	STTL	Self	STTL
NB	35.44%	**47.68%**	31.49%	**47.70%**	37.24%	**48.52%**
MLP	55.33%	**60.08%**	56.70%	**60.53%**	60.27%	**64.88%**
SMO	**56.76%**	49.82%	**58.53%**	54.37%	59.61%	**60.75%**
C4.5	66.69%	**67.33%**	66.00%	**68.30%**	66.69%	**69.40%**
JRip	52.14%	**59.17%**	55.31%	**60.02%**	55.63%	**64.73%**
*k*NN	64.85%	**66.17%**	73.39%	**77.92%**	75.86%	**81.48%**

Table 3

Performance evaluation of STTL against Self-training algorithm

5.2. Second phase of experiments

Next, we focus our interest on the experimental analysis on evaluating the performance of the proposed algorithm utilizing different classification algorithms as A-level and B-level classifiers. Therefore, we consider the following three performance metrics.

$$A_{NG} = \frac{\text{number of students correctly predicted Not to Graduate}}{\text{total number of not graduated students}}$$

$$A_G = \frac{\text{number of students correctly predicted to Graduate}}{\text{total number of graduated students}}$$

$$A_{CGT} = \frac{\text{number of graduated students Correctly predicted their Graduation Time}}{\text{total number of graduated students}}$$

The first two metrics A_{NG} and A_G evaluate the performance of A-level classifier, while the third metric A_{CGT} evaluates the performance of B-level classifier. Moreover, since the number of students who failed in the examinations is about 15%, it is crucial for a prediction model to correctly identify them. As one of the two main goals of this study is to identify the students at-risk, it is significant to achieve the highest possible predictive accuracy for the student who failed to graduate within six years of studies. Therefore, similar to [25,41], we present an additional performance metric

$$F_{1.5} = \frac{(1+1.5^2)nFiF}{(1+1.5^2)nFiF + 1.5^2 nFiG + nGiF},$$

where *nFiF* stands for the number of students who failed to graduate and correctly identified, *nFiG* stands for the number of students who failed and identified as graduated and *nGiF* stands for the number of students who graduated and identified as failed.

It is worth mentioning that the performance metric $F_{1.5}$ takes into account the accuracy for students who failed and graduated and weights more the accuracy for students who failed than the students who successfully graduated [46]. From an educator's perspective, it is better to misidentify a "graduate" student than a "failed" student. Misidentifying a "graduate" student as a potential fail may encourage him/her to work harder and improve his/her performance. In the contrary, misidentifying a "failed" student as a potential graduated may prevent him/her from taking the proper actions to enhance its performance [25].

Table 4 presents the performance evaluation of A-level and B-level classifiers utilizing various classification algorithms, relative to all labeled ratio. Notice that the best score for each performance metric is highlighted in bold. Clearly, *k*NN exhibits by far the best overall performance as A-level classifier. More analytically, it reports the highest classification accuracy of correctly identified students which failed to graduated while it presents the second highest performance of correctly identified students which successfully graduated, relative to all labeled ratio. Furthermore, although C4.5 reported the highest performance regarding A_G metric, it exhibits poor performance for the performance metrics A_{NG} and $F_{1.5}$. Therefore, it constitutes one of the worst prediction model for identifying students at-risk of failing to graduate in this study. With respect to the performance metric A_{CGT}, *k*NN illustrates the highest percentage of correctly classifying students which have successfully graduated, followed by C4.5.

Tables 5, 6 and 7 summarize the performance of the proposed semi-supervised two-level classifier using various A-level and B-level classifiers, utilizing 20%, 30% and 40% as labeled data ratio, respectively.

Classifier	Ratio = 20%				Ratio = 30%				Ratio = 40%			
	A_{NG}	A_G	$F_{1.5}$	A_{CGT}	A_{NG}	A_G	$F_{1.5}$	A_{CGT}	A_{NG}	A_G	$F_{1.5}$	A_{CGT}
NB	26.09%	84.42%	26.62%	55.28%	32.61%	89.45%	34.95%	58.29%	43.48%	81.91%	40.75%	60.30%
MLP	60.87%	96.98%	66.18%	65.33%	60.87%	95.98%	65.23%	67.34%	60.87%	96.98%	64.31%	68.84%
SMO	39.13%	**98.49%**	46.99%	63.32%	47.83%	**99.50%**	56.52%	67.34%	50.00%	**99.50%**	58.63%	72.36%
C4.5	52.17%	93.47%	55.52%	**72.36%**	56.52%	96.48%	61.90%	73.87%	60.87%	93.97%	63.41%	78.39%
JRip	52.17%	96.48%	57.99%	67.34%	52.17%	97.49%	58.87%	70.35%	56.52%	95.48%	61.01%	72.86%
*k*NN	**73.91%**	97.49%	**77.54%**	72.31%	**71.74%**	97.98%	**76.33%**	**79.90%**	**82.61%**	97.48%	**84.30%**	**85.43%**

Table 4

Performance evaluation of each classifier for performance metrics A_{NG}, A_G, $F_{1.5}$ and A_{CGT}

		B-Level					
		NB	MLP	SMO	C4.5	JRip	*k*NN
A-Level	NB	47.68%	49.85%	53.85%	53.55%	49.07%	63.62%
	MLP	55.87%	60.08%	59.18%	64.08%	61.72%	65.37%
	SMO	53.43%	52.23%	49.82%	62.05%	46.12%	61.72%
	C4.5	52.15%	59.15%	59.27%	65.35%	59.63%	62.02%
	JRip	53.45%	56.82%	54.27%	67.03%	59.17%	66.60%
	*k*NN	58.02%	65.80%	59.70%	65.77%	56.78%	**66.17%**

Table 5

Performance evaluation of self-trained two-level classification algorithm (Ratio = 20%)

		B-Level					
		NB	MLP	SMO	C4.5	JRip	*k*NN
A-Level	NB	47.70%	50.17%	55.55%	57.17%	56.37%	66.57%
	MLP	55.88%	60.53%	62.83%	67.78%	66.05%	71.47%
	SMO	58.33%	55.62%	54.37%	68.13%	58.48%	63.68%
	C4.5	53.03%	58.07%	61.97%	68.30%	60.82%	64.52%
	JRip	53.43%	59.18%	58.73%	71.40%	60.02%	73.00%
	*k*NN	58.02%	66.70%	63.77%	70.65%	64.98%	**77.92%**

Table 6

Performance evaluation of self-trained two-level classification algorithm (Ratio = 30%)

		B-Level					
		NB	MLP	SMO	C4.5	JRip	*k*NN
A-Level	NB	48.52%	53.27%	57.23%	63.65%	59.58%	65.35%
	MLP	56.70%	64.88%	64.43%	68.55%	71.80%	77.88%
	SMO	56.72%	63.62%	60.75%	72.23%	63.25%	72.28%
	C4.5	55.02%	61.25%	66.15%	69.40%	64.22%	73.57%
	JRip	56.65%	64.05%	65.28%	71.90%	64.73%	77.43%
	*k*NN	62.50%	70.63%	66.12%	77.55%	68.50%	**81.48%**

Table 7

Performance evaluation of self-trained two-level classification algorithm (Ratio = 40%)

As before, the best classification performance for each labeled ratio is highlighted in bold. kNN exhibited the highest average classification accuracy as A-Level classifier, relative to all labeled ratio, followed by MLP and JRip. More specifically, it presented 62.04%, 67.01% and 71.13% average classification accuracy utilizing 20%, 30% and 40% as labeled data ratio, respectively. Furthermore, as regards B-level classifier, kNN exhibited the best performance reporting 64.25%, 69.53% and 74.67% average classification accuracy utilizing 20%, 30% and 40% as labeled data ratio, respectively.

In the area of machine learning, the statistical comparison of classification algorithms over multiple datasets is fundamental and it is frequently performed by means of a statistical test [24,26]. Since our motivation stems from the fact that we are interested in evaluating the rejection of the hypothesis that all the algorithms perform equally well for a given level based on their classification accuracy and highlighting the existence of significant differences between our proposed algorithm and the classical self-labeled algorithms, we utilized the non-parametric Friedman Aligned Ranking (FAR) [15] test. Moreover, the Finner test [13] is applied as a post-hoc procedure in order to find out which algorithms present significant differences.

Classifier	FAR	Finner Post-Hoc Test	
		p_F-value	Null Hypothesis
kNN	16.7222	-	-
MLP	36.7222	0.055409	accepted
Jrip	47.0556	0.004583	rejected
C4.5	57.2222	0.000175	rejected
SMO	73.6667	0	rejected
NB	95.6111	0	rejected

Table 8

FAR test and Finner post-hoc test for A-level classifier

Tables 8 and 9 present the information of the statistical analysis performed by nonparametric multiple comparison procedures for all classification algorithms utilized as A-level and B-level classifiers, respectively. Notice that the control algorithm for the post-hoc test is determined by the best (e.g. lowest) ranking obtained in each FAR test. Moreover, the adjusted p-value with Finner's test (p_F) was presented based on the corresponding control algorithm at the $\alpha = 0.05$ level of significance. The post-hoc test rejects the hypothesis of equality when the value of p_F is less than the value of a.

Classifier	FAR	Finner Post-Hoc Test	
		p_F-value	Null Hypothesis
kNN	14.6667	-	-
C4.5	24.1667	0.362857	accepted
Jrip	58.5556	0.000033	rejected
SMO	66.7222	0.000001	rejected
MLP	68.3889	0.000001	rejected
NB	94.5	0	rejected

Table 9

FAR test and Finner post-hoc test for B-level classifier

Clearly, kNN demonstrates the best overall performance as A-level and B-level classifier, reporting the highest probability-based ranking by statistically presenting better results, relative to all utilized classification algorithms.

Based on the previous discussion, we conclude that the best classification performance of the proposed self-trained two-level classifier was exhibited in case kNN was utilized as A-level and B-level classifier.

5.3. Third phase of experiments

In the sequel, in order to illustrate the efficiency of the proposed self-trained two-level algorithm, we evaluate its performance against other state-of-the-art self-labeled algorithms, i.e. [51], Co-training [6], Tri-training [54], SETRED [20], Democratic-Co learning [53] and Co-Forest [21], in terms of classification accuracy. The configuration parameters of all self-labeled algorithms used in the experiments are presented in Table 10.

Self-labeled algorithm	Parameters
Co-training	Maximum number of iterations = 40. Initial unlabeled pool = 75.
Tri-training	No parameters specified.
SETRED	MaxIter = 40. Threshold = 0.1.
Democratic-Co learning	Classifiers = kNN, C4.5, NB.
Co-Forest	Number of Random Forest classifiers = 6. Threshold = 0.75.

Table 10

Parameter specification for all self-labeled algorithms employed in the experimentation

Table 11 summarizes the accuracy of the classical self-labeled methods and the proposed self-trained two-level classification algorithm, utilizing 20%, 30% and 40% as labeled data ratio. Notice that the highest accuracy is highlighted in bold for each labeled ratio. It should be mentioned that the proposed algorithm STTL uses *k*NN as A-level and B-level classifier. The interpretation of Table 11 shows that the proposed algorithm STTL is by far the most effective, presenting the highest classification accuracy in all cases, followed by Tri-training (*k*NN) and Co-training (*k*NN).

More specifically, the STTL reported 66.17%, 77.92%, 81.48% for labeled ratio 20%, 30% and 40%, respectively while Tri-training (*k*NN) exhibited 65.55%, 75.86%, 78.04% in the same situations.

Algorithm	Ratio = 20%	Ratio = 30%	Ratio = 40%
Co-train (NB)	37.23%	39.73%	42.48%
Co-train (MLP)	44.35%	47.49%	53.88%
Co-train (SMO)	46.15%	47.52%	51.77%
Co-train (C4.5)	62.39%	67.38%	67.41%
Co-train (JRip)	51.06%	53.90%	55.27%
Co-train (*k*NN)	64.84%	73.35%	78.35%
Tri-train (NB)	40.38%	45.70%	47.49%
Tri-train (MLP)	58.89%	60.97%	61.02%
Tri-train (SMO)	57.84%	60.31%	62.11%
Tri-train (C4.5)	65.95%	66.70%	68.82%
Tri-train (JRip)	58.10%	59.59%	59.95%
Tri-train (*k*NN)	65.55%	75.86%	78.04%
SETRED	62.97%	64.10%	62.97%
Co-Forest	58.77%	73.10%	73.88%
Democratic	60.63%	67.32%	71.45%
STTL	**66.17%**	**77.92%**	**81.48%**

Table 11

Performance evaluation of STTL against state-of-the-art self-labeled algorithms

Finally, in order to illustrate the classification performance of the proposed algorithm, we evaluate its performance against the most popular state-of-the-art supervised algorithms, namely NB, MLP, SMO, JRip, C4.5 and *k*NN. Additionally, we evaluate STTL against its component Supervised Two-Level (STL) algorithm [41] which utilizes *k*NN and C4.5 as A-level and B-level classifier, respectively. Notice that all supervised classifiers were trained with 100% of the training set while STTL was trained using only 40% of the training set as labeled data. The aggregated results presented in Table 12 illustrate that the proposed algorithm STTL presented better classification accuracy than all supervised single classifiers and its component STL.

6. Conclusions

In this work, we presented a new semi-supervised self-trained two-level classification algorithm STTL for the accurate prediction of the students' graduation time. The presented experimental results illustrated that the proposed algorithm identifies students at-risk of not graduating and accurately classifies the students' based on their graduation time. Our work could provide valuable hints and insights for better educational support by offering customized assistance according to students' predicted performance, hence it can be used as a reference for decision making in the graduate program admission process.

Our future work focuses on augmenting our experiments by applying the proposed algorithm on additional data from several departments in order to extract useful information about key factors which may affect and influence students' performance. Since our experimental results are quite encouraging our future research concerns the adaptation of our methodology to other self-labeled methods, such as co-training and tri-training.

Finally, it is worth mentioning that the students' attributes utilized in our work do not constitute a conclusive list. An extension can introduce new attributes and other criteria which were not in the current database. Moreover, a study on the significance of each attribute or the application of feature selection techniques as a preprocessing step may further increase the performance of the proposed algorithm. We certainly intent to investigate it in the near future.

Acknowledgments

The authors would like to thank the *Technological Institute of Western Greece* for granting the corresponding data collection utilized in this study.

References

[1] D. Aha. *Lazy Learning*. Dordrecht: Kluwer Academic Publishers, 1997.

[2] S. Aud, M. Planty, and W.J. Hussar. *Condition of education 2011*. Government Printing Office, 2011.

Classifier	NB	MLP	SMO	JRip	C4.5	10NN	STL	SSTL
Accuracy	47.80%	62.10%	64.97%	66.58%	69.73%	68.89%	78.73%	**81.48%**

Table 12

Performance evaluation of STTL against state-of-the-art supervised algorithms

[3] R.S. Baker and K. Yacef. The state of educational data mining in 2009: A review future visions. *Journal of Educational Data Mining*, 1(1):3–17, 2009.

[4] D. Baumgartner and G. Serpen. Large experiment and evaluation tool for WEKA classifiers. In *International Conference on Data Mining*, volume 16, pages 340–346, 2009.

[5] U. bin Mat, N. Buniyamin, P.M. Arsad, and R. Kassim. An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention. In *2013 IEEE 5th Conference on Engineering Education (ICEED)*, pages 126–130. IEEE, 2013.

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *11th annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[7] C. Burgos, M.L. Campanario, D. de la Peña, J.A. Lara, D. Lizcano, and M.A. Martínez. Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout. *Computers & Electrical Engineering*, 2017.

[8] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.

[9] W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, pages 115–123, 1995.

[10] National Research Council. *Building a workforce for the information economy*. National Academies Press, 2001.

[11] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[12] A. Dutt, M.A. Ismail, and T. Herawan. A systematic review on educational data mining. *IEEE Access*, 29:103–130, 2017.

[13] H. Finner. On a monotonicity problem in step-down multiple test procedures. *Journal of the American Statistical Association*, 88(423):920–923, 1993.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA data mining software: An update. *SIGKDD Explorations Newsletters*, 11:10–18, 2009.

[15] J.L. Hodges and E.L. Lehmann. Rank methods for combination of independent experiments in analysis of variance. *The Annals of Mathematical Statistics*, 33(2):482–497, 1962.

[16] N. Iam-On and T. Boongoen. Improved student dropout prediction in thai university using ensemble of mixed-type data clusterings. *International Journal of Machine Learning and Cybernetics*, 8(2):497–510, 2017.

[17] N. Johnson. Three policies to reduce time to degree. *Complete College America*, 2011.

[18] L.I. Kuncheva. "Change-glasses" approach in pattern recognition. *Pattern Recognition Letters*, 14:619–623, 1993.

[19] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein. A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters*, 20(1):7–14, 1999.

[20] M. Li and Z.H. Zhou. SETRED: Self-training with editing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 611–621. Springer, 2005.

[21] M. Li and Z.H. Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1088–1098, 2007.

[22] I.E. Livieris, K. Drakopoulou, Th. Kotsilieris, V. Tampakas, and P. Pintelas. DSS-PSP - A decision support software for evaluating students' performance. In *Engineering Applications of Neural Networks (EANN)*, volume 744, pages 63–74. Springer, 2017.

[23] I.E. Livieris, K. Drakopoulou, T. Mikropoulos, V. Tampakas, and P. Pintelas. An ensemble-based semi-supervised approach for predicting students' performance. In *Research on e-Learning and ICT in Education*, pages 25–42. Springer, 2018.

[24] I.E. Livieris, K. Drakopoulou, V. Tampakas, T. Mikropoulos, and P. Pintelas. Predicting secondary school students' performance utilizing a semi-supervised learning approach. *Journal of Educational Computing Research*, pages 1–23, 2018.

[25] I.E. Livieris, T. Kotsilieris, V. Tampakas, and P. Pintelas. Improving the evaluation process of students' performance utilizing a decision support software. *Neural Computing and Applications*, pages 1–12, 2018.

[26] I.E. Livieris, V. Tampakas, N. Kyriakidou, T. Mikropoulos, and P. Pintelas. Forecasting students' performance using an ensemble ssl algorithm. In *IEEE 1st International Conference on Technology and Innovation in Learning, Teaching and Education*, pages 1–8, 2018.

[27] P. Lodhi, O. Mishra, and S. Jain. Self-learning knowledge base using naïve bayes classifier. *Intelligent Decision Technologies*, pages 1–8, 2018.

[28] T. Minami and Y. Ohura. Toward learning support for decision making: Utilization of library and lecture data. In *Intelligent Decision Technologies*, pages 137–147. Springer, 2012.

[29] H.M. Nagy, W.M. Aly, and O.F. Hegazy. An educational data mining system for advising higher education students. *World Academy of Science, Engineering and Technology International Journal of Information Engineering*, 7(10):175–179, 2013.

[30] V. Ng and C. Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 94–101. Association for Computational Linguistics, 2003.

[31] A. Peña-Ayala. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4):1432–1462, 2014.

[32] N.N. Pise and P. Kulkarni. A survey of semi-supervised learning methods. In *Proceedings of the 2008 International Conference on Computational Intelligence and Security*, volume 2, pages 30–34. IEEE Computer Society, 2008.

[33] J. Platt. Using sparseness and analytic QP to speed training of support vector machines. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in neural information processing*

systems, pages 557–563, MA: MIT Press, 1999.

[34] J.R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco, 1993.

[35] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE on Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 40(6):601–618, 2010.

[36] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362, Cambridge, Massachusetts, 1986.

[37] A.A. Saa. Educational data mining & students' performance prediction. *International Journal of Advanced Computer Science and Applications*, 7(5):212–220, 2016.

[38] M. Sigdel, I. Dinç, S. Dinç, M.S. Sigdel, M.L. Pusey, and R.S. Aygün. Evaluation of semi-supervised learning for classification of protein crystallization imagery. In *SOUTHEASTCON 2014, IEEE*, pages 1–6. IEEE, 2014.

[39] M.C. Stetser and R. Stillwell. Public high school four-year on-time graduation rates and event dropout rates: School years 2010-11 and 2011-12. First Look. NCES 2014-391. *National Center for Education Statistics*, 2014.

[40] V. Tam, E.Y. Lam, S.T. Fung, W.W.T. Fok, and A.H.K. Yuen. Enhancing educational data mining techniques on online educational resources with a semi-supervised learning approach. In *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering*, pages 203–206. IEEE, 2015.

[41] V. Tampakas, I.E. Livieris, E. Pintelas, N. Karacapilidis, and P. Pintelas. Prediction of students' graduation time using a two-level classification algorithm. In *IEEE 1st International Conference on Technology and Innovation in Learning, Teaching and Education*, pages 1–7, 2018.

[42] K. Tourtoglou and M. Virvou. An intelligent recommender system for trainers and trainees in a collaborative learning environment for uml. *Intelligent Decision Technologies*, 6(2):79–95, 2012.

[43] I. Triguero, S. García, and F. Herrera. SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification. *IEEE Transactions on Cybernetics*, pages –, 2014.

[44] I. Triguero, S. García, and F. Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, 2015.

[45] I. Triguero, J.A. Sáez, J. Luengo, S. García, and F. Herrera. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41, 2014.

[46] C.J. van Rijsbergen. *Information retrieval*. London: Butterworths, second edition, 1979.

[47] Yu Wang, X. Xu, H. Zhao, and Z. Hua. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems*, 23(6):547–554, 2010.

[48] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[49] J. Xu, K.H. Moon, and M. van der Schaar. A machine learning approach for tracking and predicting student performance in degree programs. *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[50] L. Xu and A. Krzyzak C.Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on systems, man, and cybernetics*, 22(3):418–435, 1992.

[51] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[52] N.A. Yassein, R.G.M. Helali, and S.B. Mohomad. Predicting critical courses affecting students performance: A case study. *Journal of Information Technology & Software Engineering*, 7(5):1–5, 2017.

[53] Y. Zhou and S. Goldman. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 594–602. IEEE, 2004.

[54] Z.H. Zhou and M. Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[55] X. Zhu and A.B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[56] D. Zinovev, E. Varutbangkul, D.S. Raicu, J.D. Furst, and S.G. Armato III. Semi-supervised learning approaches for predicting semantic characteristics of lung nodules. *Intelligent Decision Technologies*, 3(4):207–217, 2009.