Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 7140797, 11 pages
https://doi.org/10.1155/2020/7140797

Research Article

Intelligent Course Plan Recommendation for Higher Education: A Framework of Decision Tree

Xiaoliang Chen,[1,2] **Jianzhong Zheng**,[1] **Yajun Du**,[1] **and Mingwei Tang**[1]

[1]*School of Computer and Software Engineering, Xihua University, Chengdu 610039, China*
[2]*Department of Computer Science and Operations Research, University of Montreal, Montreal, QC H3C3J7, Canada*

Correspondence should be addressed to Xiaoliang Chen; xdxlchen@gmail.com

Received 3 July 2019; Revised 3 September 2019; Accepted 13 September 2019; Published 23 January 2020

Academic Editor: Ricardo López-Ruiz

The framework of outcomes-based education(OBE) has become a central issue for global university education, which is benefited to drive the education development by a series of assessments for historical teaching data, especially student course score, and employment information. The issue of how to timely update the talent training plans for computer major in a university has received considerable critical attention. It is becoming extremely difficult to ignore the requirement of fast shortened update cycle in IT area. One of the main obstacles is that the teaching inertia and the fixed awareness of a major training plan may delay the feedback of teaching problems. There is still a contradiction between the plan rationality and the real-time needs of contemporary IT enterprises. Hence, this paper puts forward a novel data-based framework to evaluate the relevance between the major courses, employment rate, and enterprise needs through the decision tree expression, thus providing reliable data support for systematic curriculum reform. On top of that, A recommendation algorithm is proposed to automatically generate the computer course group that satisfies the staff requirements of IT enterprises. Finally, teaching and employment data of Xihua University in China is applied as an example to undertake course optimization and recommendation. The consequences have an obvious positive effect on student employment and enterprise feedback.

1. Introduction

A few prestigious universities have long recognized the importance of getting their students well-prepared for their future careers by offering appropriate professional skill training from the academic courses [1–3]. Correspondingly, a growing number of computer colleges around the world have started to update their talent training plans for undergraduates and offer them to current IT enterprises [4, 5]. Hence, the course plan plays an extremely important role in helping qualified computer teachers fully understand their work, especially course duration and the depth of teaching.

It is widely acknowledged that traditional experiential curriculum setting has brought about a lot of convenience to the educational planner. These plans for course setting and class hour allocation have been acclaimed for their advantages within a range of short time. However, course updates that rely on completely artificial analysis have created some potential problems almost every college is exposed to. For example, some course plans of computer major have become noticed with staying the same for many years without doing anything meaningful update. Static plans do not comply with the need for the rapid development of IT enterprises.

Instead, taking internal relations from educational big data has become increasingly popular among the educators. There is an urgent need to address the data-based course plan optimization problems by a series of simple operations. On the other hand, convenient operations are the basis for wide application in other majors and universities. Hence, we first put forward a novel data-based framework to optimize course configuration and course hours through the decision tree expression. Use of decision tree studies is a well-established approach in data science, but there is rarely applied for curriculum optimization. Secondly, a course group recommendation algorithm is therefore proposed to be timely obtain the course group configuration that has employment advantages.

Decision tree analysis [6–9] is a popular data mining technique that has many various potential applications such as energy and power systems [10, 11], medical and health [12, 13], traffic and transportation [14], accounting and finance

[15], etc. This method has already made an important contribution to the field of university education. For example, the solutions of teacher evaluation [16], student performance analysis [17–21], scholarship grantee selection [22], etc. have benefited a lot from decision tree technologies. Moreover, most studies in the field have simply focused on student roles. Some examples of such researches include online persistent predication [23] and student performance predication [19, 20]. The former determines the situation of students in web-supported courses by investigating the logs of students in school. The latter analyzes the health, social activity, relationships, and academic performance, most related to and affect the performance of students. The convenience content material for learners is also predicted by decision trees [24].

However, previous studies of decision tree applications in higher education have not dealt with the crucial tasks of course group optimization and recommendation. Few educators have been able to draw on any systematic research into course plan of a major. Hence, we first proposed a framework of course plan recommendation that is illustrated in Figure 1.

Some of educators are faced with a dilemma: how to judge whether the running course plan will benefit the student's employment and how to seek support teaching data to update the course plan. The framework shown in Figure 1 can be more useful for quickly setting a reasonable course plan in computer education, which provides a novel understanding for course plan configuration and reports a feasible data-based intelligent solution for the dilemma.

Traditional course plan is adjusted according to the experience of education experts. However, the fact is completely divorced from the needs of IT enterprises in real time. Our framework suggests another path that can be illustrated in Figure 1. Firstly, the module of teaching process data such as scores and student performance should be detailed and recorded into the data system. On top of that, the module of enterprise needs should be recorded and updated in real time. The presented data system for teaching and enterprise information will be analyzed by the classifiers. The aim is to obtain the weight of the teaching curriculum for actual employment, which can be utilized to guide guide the reform of the course plan. Finally, basic differential analysis ensures the effectiveness of the new plan. The final data-based approach has obvious advantages in real time performance compared with the experience-based one.

The rest of the paper is organized as follows. Section 2 briefly introduces the basics of decision tree analysis. Section 3 elaborates a case study on how to use the framework shown in Figure 1. Section 4 deals with the recommendation of course groups according to decision tree. The proposed methods are applied to the major "computer science and technology" of Xihua University in China. Finally, we summarize our conclusion in Section 5.

2. Decision Tree Analysis

Developing a structured and really efficient decision tree contributes a lot to approximate discrete value functions and process data classification. First of all, the decision tree takes the attribute (the column of the structured table) as the node in the tree, uses the attribute value (data item value or binary relation data point) as the branch from the tree, and applies the amount of information contained in the attribute to hierarchically divide them. Decision tree goes beyond the limits of a data table so that researchers can process the work of classification and trend prediction. Attribute selection and the generation of nodes is not arbitrary. Instead, attributes with greater importance should be prioritized as the upper nodes of the tree. Information gain [6] is invoked as a measure to select attributes.

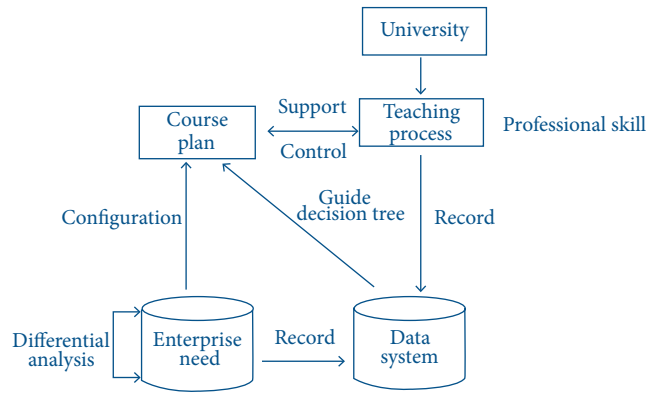


Figure 1: Framework of course plan recommendation.

Input: evaluation matrix $D = S \times P(d_{ij})^{m \times n}$, where S is the training sample set and P is the attribute set.

Output: decision tree *DeTree*.

1 initial *DeTree* as a root Node $n = (S, P)$ with the whole sample in S and attribute in P;
2 **for each** *leaf node* $n' = (S', P')$ *in DeTree with* $S' \subseteq S$ *and* $P' \subseteq P$ **do**
3 **if** $P' = \emptyset$ or $\forall S_i \in S'$,s_i, *belongs to a same classification* **then**
4 stop and output *DeTree*;
5
6 **else**
7 1. calculate the information gain $Gain(p_i \in P, S) = H(S) - E(p_i)$, for all attributes about n';
8 2. select $p_i \in P'$ with $Max(Gain)$ as the category test attribute about node n';
9 **for each** *test attribute* $p_i \in P'$ **do**
10 construct a branch labeled with the attribute value;
11 **end**
12
13 **end**
14 **end**

Algorithm 1: ID3: decision tree construction algorithm.

Input: evaluation matrix $D = S \times P(d_{ij})^{m \times n}$, where S is the training sample set and P is the attribute set.

Output: decision tree *DeTree*.

```
initial DeTree as a root node n = (S, P) with the whole sample in S and attribute in P;
for each leaf node n' = (S', P') in DeTree with S' ⊆ S and P' ⊆ P do
    if P' = ∅ or ∀S_i ∈ S', S_i belongs to a same category then
        stop and output DeTree;

    else
        1. calculate the information gain ratio GainRatio(p_i) = (H(S) − E(p_i))/E(p_i) for all attributes about n';
        2. select p_i ∈ P' with Max(GainRatio) as the category test attribute about node n';
        for each test attribute p_i ∈ P' do
            construct a branch labeled with the attribute value;
        end

    end
end
```

Algorithm 2: C4.5: decision tree construction algorithm.

We briefly describe the concepts used in this article. Parameter $H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i$ is called the information entropy of source $S = \{S_1, S_2, \ldots, S_k\}$, where $p_i = P(s_i)$ is a probability distribution with respect to S_i. The expected information required for S_i is denoted as $I(S_i) = I(m_i, n_i) = -(m_i/(m_i + n_i))\log_2(m_i/(m_i + n_i)) - (n_i/(m_i + n_i))\log_2(n_i/(m_i + n_i))$ if S_i possesses m_i and n_i positive and negative examples, respectively. The expected entropy required for the root of attribute p_i can be described as $E(p_i) = \sum_{i=1}^{k}(((m_i + n_i)/|S|)I(S_i))$. The information gain of attribute p_i can be calculated by a simple formula $Gain(p_i, S) = H(S) - E(p_i)$. The details of ID3 and C4.5 are present in Algorithms 1 and 2, respectively.

3. A Case Study of Framework Applications in Computer Major

The approach of decision tree was used to search for a systematic curriculum that is more adaptable to the needs of enterprises. A case experiment allows education researchers all over the world to find the best reasonable talent training program by considering their actual school situation.

3.1. Experiment Data Set. We collect the real data of comprehensive course scores and the latest employment information of the students in Xihua University of China from 2014 to 2017. Desensitization data sets with the major Computer Science and Technology were used to support this study and are available at [https://github.com/1007105767/StudentsData]. These datasets are cited at relevant places within the text as references [25].

The data set contains five big data formate (*.csv) tables, where three of them (Data1_2014_students.csv, Data2_2015_students.csv, and Data3_2016_ students) preserve the student's course scores. Each of them contains approximately 40 thousand records. On top of that, two of them (2016_ Employment basic data.csv and 2017_Employment basic data.csv) record the employment information for graduates in the last two years.

The following case study confirms the convenience of curriculum update by using decision tree analysis. Two subsystems of Xihua University are designed to record student-related information and produce the focused five data tables [25].

(1) *Educational Information Management System (EIMS)*: accurately record the score of each course for each student.

(2) *Graduate Information Management System (GIMS)*: record the work, salary, and feedback of undergraduates and the situation of employers.

GIMS collects relevant graduate data regularly from enterprises and archive their storage by structured electronic table.

This experiment randomly selected 200 undergraduates of computer major as the data source, the aim of which is to analyze the relevance and influence between the current curriculum for the school students and the employments at the next two years. We expect that the result can provide data support to implement the benign fine-tuning of the professional curriculum.

3.2. Target Data Acquisition

3.2.1. Step A: Course Data Setting. Course data are derived from the talent training program (version 2015) of the major Computer Science and Technology. The key aspects of the course relations is presented in the flow chart that is shown in Figure 2. According to the school's educational administration system, all courses as a major are classified according to the following three modules: culture and morality, professional, and practical training.

It is extended that a more granular module division has been playing an increasingly important role in helping researchers get more useful information from data analysis. Hence, this work choose 30 courses in the system according to the degree of importance in computer science. All of them are divided into the following six modules:

(1) Natural science and language(NSL, 110 hours): physics B, College English, Professional English.

(2) Mathematics and algorithms(MA, 264 hours): Advanced Mathematics, Linear Algebra, Probability Theory and Mathematical Statistics, Discrete Mathematics, Numerical Computation Methods, and Data Structures and Algorithms.

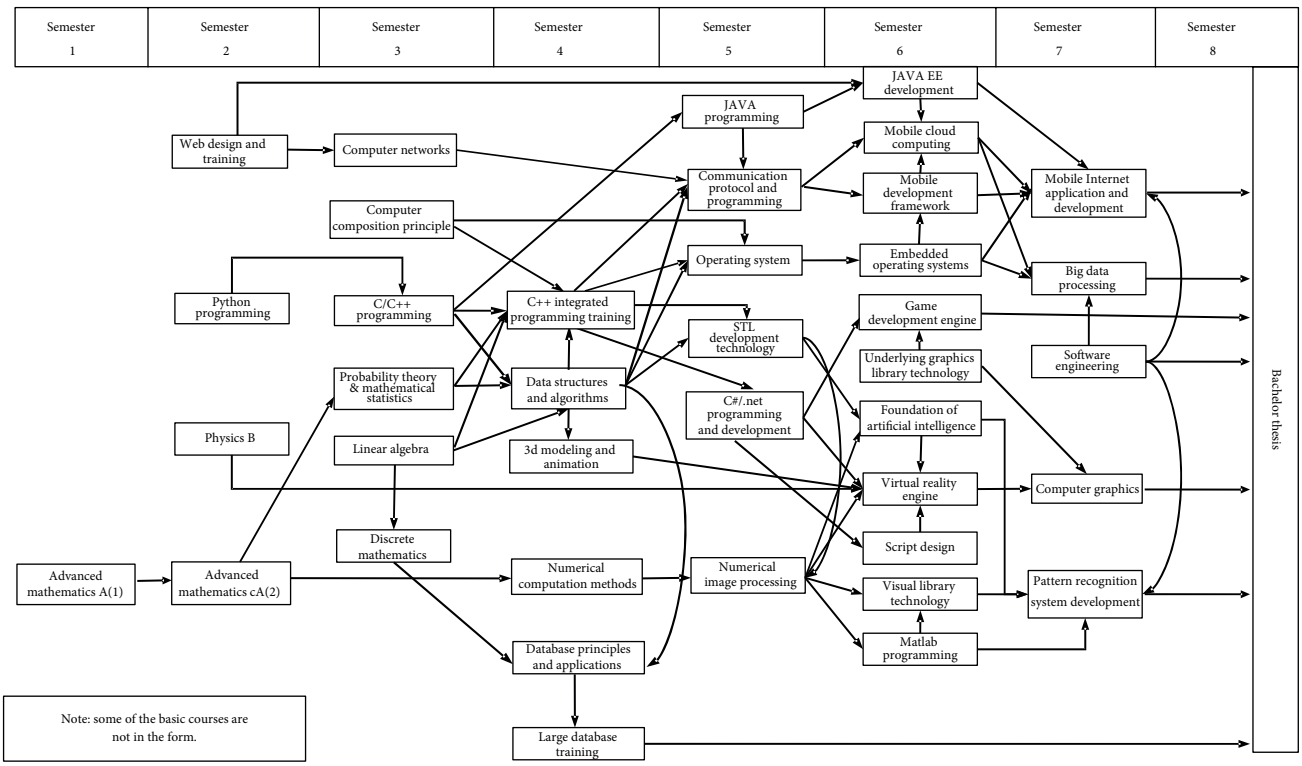


Figure 2: Course configuration flow chart of computer science and technology in Xihua University.

(3) Programming language(PL, 288 hours): Python, C/C++, JAVA, C#/.net, Matlab, and Script.

(4) System and database(SD, 156 hours): Operating System, Database Principles, and Large Database Training.

(5) Software development skills(SDS, 144 hours): JAVA EE Development, Mobile Internet Application and Development, Game Development Engine, Virtual Reality Engine, Pattern Recognition System Development, and Big Data Processing.

(6) Software development theory(SDT, 180 hours): Software Engineering, Mobile Development Framework, Computer Graphics, Foundation of Artificial Intelligence, Embedded Operating Systems, and Numerical Image Processing.

Table 1: Graduate transcript.

No.	ID	Physics B	Python	C/C++	⋯
1	3320150991101	81	92	95	⋯
2	3320150991102	73	86	75	⋯
3	⋯	⋯	⋯	⋯	⋯

Table 2: Graduate employment feedback.

No.	ID	City	Type	M-income
1	3320150991101	ChenDu	State-owned	5200
2	3320150991102	DeYang	private-owned	4800
3	3320150991103	Unknown	None	0
4	3320150991104	ShenZhen	Joint venture	7000
5	⋯	⋯	⋯	⋯

3.2.2. Step B: Graduate Evaluation Data Setting. Graduate evaluation data are divided into two parts: the grades and the employment feedback. This work records the transcripts of each subject, the nature of employment, monthly income, and other information of randomly 200 undergraduates at university through inquiry GIMS. These data are summarized into two kinds of forms that are shown in Tables 1 and 2, respectively.

Original data need to be preprocessed in order to obtain a better accuracy. This work has started to implement noise elimination, supplement missing data, and eliminate error data. First of all, the data type in Table 1 is correct excepted for a few empty items. The futile data are derived from missing an examination of some courses. We fill in the default with score 0. On top of that, Table 2 has a small number of meaningless items, but lack of monthly income information. We use the average income with respect for the employment area to fill the blank. Finally, Table 2 also has a little amount of erroneous items. For example, a monthly income of 230 is obviously impossible. We abandon the record and add a new one as a data supplement from the database.

3.3. Decision Tree Construction and Analysis. University educators should be well prepared for their course scheme before teaching. Hence, this study uses ID3 and C4.5 to analyze the acquired data by building an easily identifiable graphical

Table 3: Graduate data rating.

No.	ID	NSL	MA	PL	SD	SDS	SDT	Feedback
1	3320150991101	B	A	A	A	B	A	Good
2	3320150991102	C	A	B	B	C	B	Moderate
3	3320150991103	B	B	B	B	C	C	Moderate
4	3320150991104	B	B	A	A	A	A	Good
5	...	...	...	...	...	...	...	...

data association. The proposed analysis steps are possible to be accepted in the current education.

3.3.1. Step C (ID3): Decision Tree Size Setting. Decision tree analysis is faced with a dilemma: which size of a tree (the number of nodes) is reasonable? A score has 100 discrete values if it is set to be an integer type and its interval from 1 to 100. Furthermore, the score has 1000 values if we retain one decimal place, which will result in an excessively large decision tree and cause serious difficulties for further analysis. Hence, reasonable discrete discrimination is necessary. This work converts the scores in Table 1 into the three-level letters ("A", "B", "C"), and updates the employment feedback in Table 2 into the two-level ("Good", "Moderate") evaluation model.

According to the course modules described in Step A, Table 1 will update as per the following rules. The course group rating is "A," "B," and "C" if exceeding 2/3 of the total course number in the group satisfy score ≥ 85, $75 \leq$ score < 85, and score < 75, respectively. On the other hand, Table 2 will update as per the following rules.

(1) Rating "Good" if the nature of an enterprise is state-owned, M-income ≥ 4000 in a municipal level city, M-income ≥ 5000 in a second-tier provincial capital, or M-income ≥ 7000 in a first-tier cities such as BeiJin and ShangHai.

(2) Ratings "moderate" if none of the cases are labeled with "good".

3.3.2. Step D (ID3): Decision Tree Analysis. Decision tree approaches comply with typical greedy idea. Each stage seeking its subtrees in the next layer is built on the one that is currently constructed. As a result, a beneficial decision tree can be constructed by a series of top-down stepwise recursive steps according to ID3 and C4.5 algorithms. In this case, course grouping and the discrete grading of experimental data have been completed. Then, we merge the hierarchical information from Tables 1 and 2. Therefore, the evaluation matrix $D = \left(d_{ij}\right)^{m\times n}$ shown in Table 3 can be calculated, where $m = 200$ and $n = 6$ are the number of randomly selected student samples (sample set S) and the number of course groups (attribute set P).

The core step of ID3 algorithm is to calculate the information gain $Gain(p_i \in P, S) = H(S) - E(p_i)$ for each attribute in matrix D. The attribute that obtains the greatest gain will be constructed as a new node n' in the focused decision tree. All data are subsequently divided into two categories according to the if-then rule around node n'. The process iterative execution until all data belong to the same category. The specific calculation results are as follows.

(1) Calculate the information entropy required for classification.

(2) Calculate the expected information and expected entropy to determine the root attribute.

(3) Calculate the information gain $Gain(p_i, S)$ for attributes.

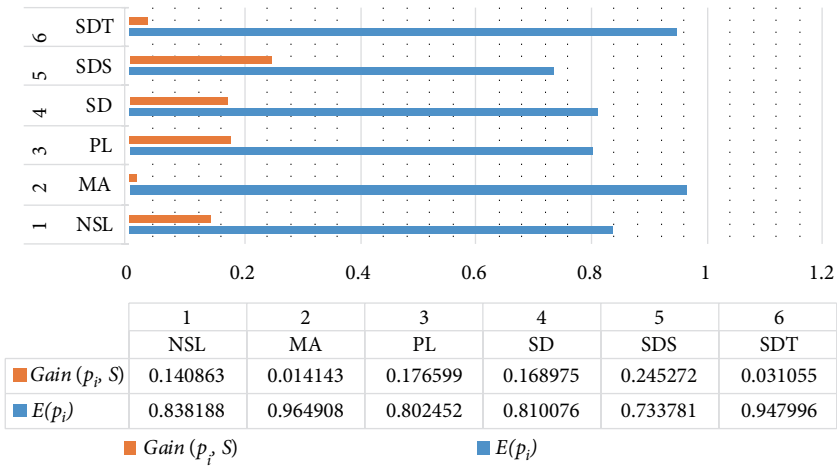
(4) Generate decision tree.

The final information gain for attributes is represented in Figure 3. The characteristic of category test attributes required individuals to pick up the maximum one. Hence, the course group "SDS" with the maximum information gain 0.245272 represented in Figure 3 is chosen as a category test attribute according to the first iteration. Consequently, "SDS" is considered as the root of the tree, the three attribute values "A", "B", and "C" of which become its three branches, respectively. Finally, the decision tree shown in Figure 4 is constructed by executing the loop that is represented at the second line in ID3 algorithm.

Building the decision tree was undertaken to evaluate the course rationality of the demand proposed by enterprises. The insights gained from the tree will be of assistance to make a global judgment on the operation of the current curriculum. First of all, the root "SDS" (Software Development Skills) is the first group obtained by gain computation, which indicates that the actual demand for undergraduates is to have a variety of software development skills.

Further course analysis revealed that Java EE Development, Pattern Recognition System Development, and Big Data Processing, etc. should help students with the most for their future career. On top of that, the left sub-tree of the root "SDS" further illustrates the significance since only the left branch has a path (PL →'B'→ MA → 'B' → SDT → 'B' →'Good'), in which three evaluations are "B" and the result is still "Good". This study has found that the support degrees of total 144 class hours in group "SDS" are obviously not enough. Therefore, it is necessary to increase the hours for the group "SDS" in order to improve the employment rate and salary.

Secondly, the result must be "Good" if the groups "PL" and "NSL" are scored as "A", which supports the secondary employment relevance of the groups. The findings reported here shed new light on curriculum reform. IT Enterprises expect their future careers who are proficient in multiple languages since the source of the project may come from different



	1 NSL	2 MA	3 PL	4 SD	5 SDS	6 SDT
$Gain(p_i, S)$	0.140863	0.014143	0.176599	0.168975	0.245272	0.031055
$E(p_i)$	0.838188	0.964908	0.802452	0.810076	0.733781	0.947996

$Gain(p_i, S)$ $E(p_i)$

Figure 3: Information gain for attribute set.

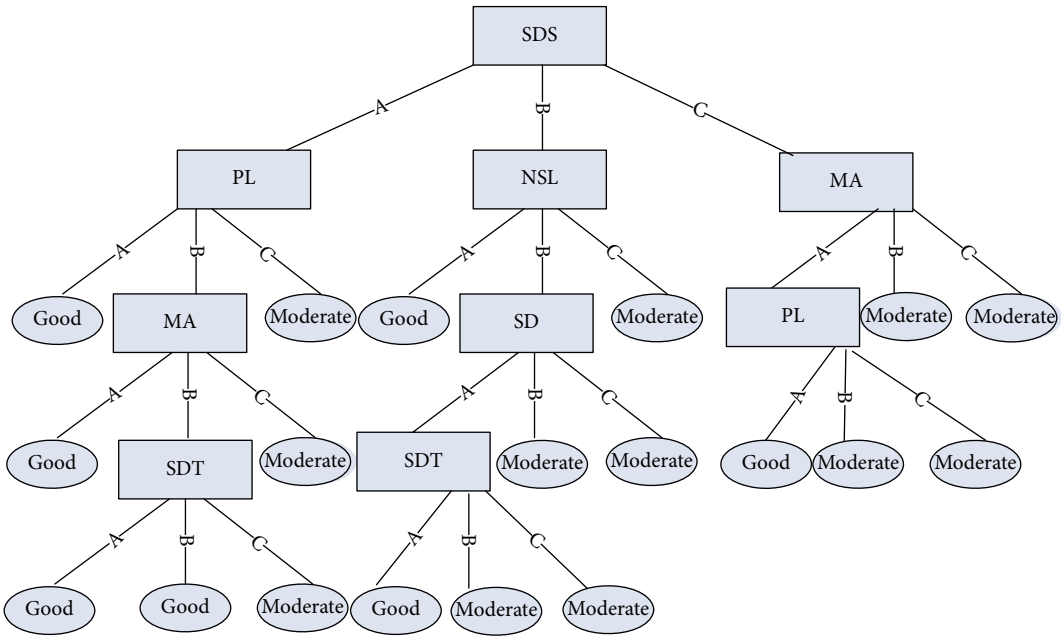


Figure 4: Decision tree by using ID3 algorithm.

developmental requirements. On the other hand, the English language score, e.g., CET exam, and the English professional capability are generally used as the first level for a company to filter employees.

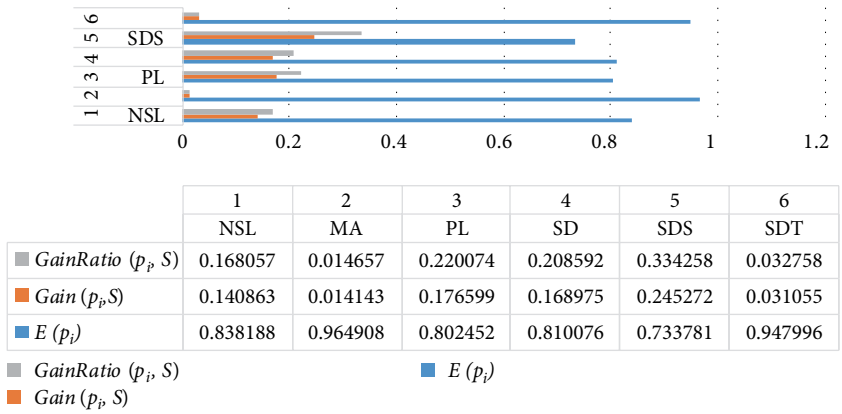
Thirdly, group "MA" (Mathematical Basis and Algorithms) is generally measured as "A" whatever the groups "SDS" (Software Development Capabilities) and the "PL" (Programming Languages) are evaluated as "Good" or "Moderate". The fact shows that a good theoretical foundation and an excellent learning ability have been playing an increasingly important role in helping enterprises to upgrade their technologies. Finally, groups "SD" and "SDT" are closest to the leafs of the tree. Besides, their evaluation discriminants were then shown according to the judgment after the remaining course that has not been evaluated as "Good". Hence, the importance of "SD" and "SDT" is slightly reduced, time schedules of which have already been in conformity with employment needs.

The following conclusions can be drawn from the present study. From the course schedule and average class hours, the "PL" has a total of 288 class hours (average 48 hours/class), and the "MA" has a total of 264 class hours (average 44 hours/class). The former can appropriately raise to an average of 52 hours/course, and the latter should be improved according to 48 hours/course.

It has been universally accepted that better accurate results should be adopted by running C4.5 compared with ID3. They are identical in both parts of the data processing that is represented in Steps A and B. Hence, the rest of this section only describes Steps C and D for C4.5 algorithm.

Table 4: Graduate data rating for average scores.

No.	ID	NSL	MA	PL	SD	SDS	SDT	Feedback
1	3320150991101	81	92	95	90	86	90	Good
2	3320150991102	65	90	78	80	60	82	Moderate
3	3320150991103	73	86	75	79	70	69	Moderate
4	3320150991104	80	88	96	90	91	93	Good
5	...	...	...	...	...	...	...	...



	1 NSL	2 MA	3 PL	4 SD	5 SDS	6 SDT
GainRatio (p_i, S)	0.168057	0.014657	0.220074	0.208592	0.334258	0.032758
Gain (p_i,S)	0.140863	0.014143	0.176599	0.168975	0.245272	0.031055
E (p_i)	0.838188	0.964908	0.802452	0.810076	0.733781	0.947996

Figure 5: Information gain for attribute set.

3.3.3. Step C(C4.5): Decision Tree Size Setting. The expected entropy $E(p_i)$ in the computation of information gain $Gian(p_i)$ plays an increasingly important role in helping people to evaluate a division. However, $Gian(p_i)$ is not stable enough since the value of $E(p_i)$ may present obviously fluctuate in some cases. Hence, information gain ratio $GainRatio(p_i)$ in C4.5 algorithm is applied to weaken the fluctuations, which will result in a further valuable new tree. On top of that, the branch breakpoint of the tree are set as follows in order to get more accurate results.

(1) Assume that $[a, b]$ is the attribute value interval, which is divided into n equal points $x_1, x_2, \ldots, x_n$ according to the demand. For example, the fractional interval $[0, 100]$ can be divided according to the rounding manner.
(2) Calculate the information gain of intervals $[a, x_i]$ and $[x_i, b]$ for any point $x_i (1 \le i \le n)$, respectively.
(3) Take the point with the largest information gain as the breakpoint.
(4) Continue the calculation according to the information gain ratio.

In this section, the method of three-level letter rating in Table 3 is revised as showed in Table 4, in which the average score of a course group is considered as an evaluation result.

Labelling attribute scores in the branch weight of a tree is being adopted by using C4.5 algorithm to analyze the data in Table 4. However, excessive sample score points can result in a tree size that is not conducive to analysis. For example, there exist the values from the lowest score of 36 to the highest score of 96 in the data set. Hence, the fractional intervals are converted into fractional breakpoints. We can obtain the breakpoint 88 of the interval [85, 96] by using the first four records in Table 4 since the gain of 88 is the largest one within the totally 12 points in the interval. Then, a novel tree is built by using the breakpoints as the attribute values.

3.3.4. Step D (C4.5): Decision Tree Analysis. The specific calculation results are as follows.

(1) *Calculate the Information Entropy Ratio Required for Classification.* The group Mathematics and Algorithm (MA, 264 hours) is illustrate as an example. First of all, the gain $Gain(p_i, S)$ and the entropy $E(p_i))$ of "MA" are 0.014143 and 0.964808, respectively, which are recorded in Figure 3. Hence, we have the following ratio $GainRatio(p_i, S) = 0.014657$. Information gain ratio for any items in attribute set is calculated, which values are shown in Figure 5. C4.5 method is loaded to construct a decision tree by using the gain ratios. As a result, this study obtain the tree that is represented in Figure 6.

Parameters where significant differences have been found include the category classification accuracy by comparing the decision trees shown in Figures 4 and 6. Firstly, the root "SDS" of the course group shown in Figure 6 is similar to that represented by Figure 5. Hence, the most obvious finding to emerge from the analysis is that the actual software

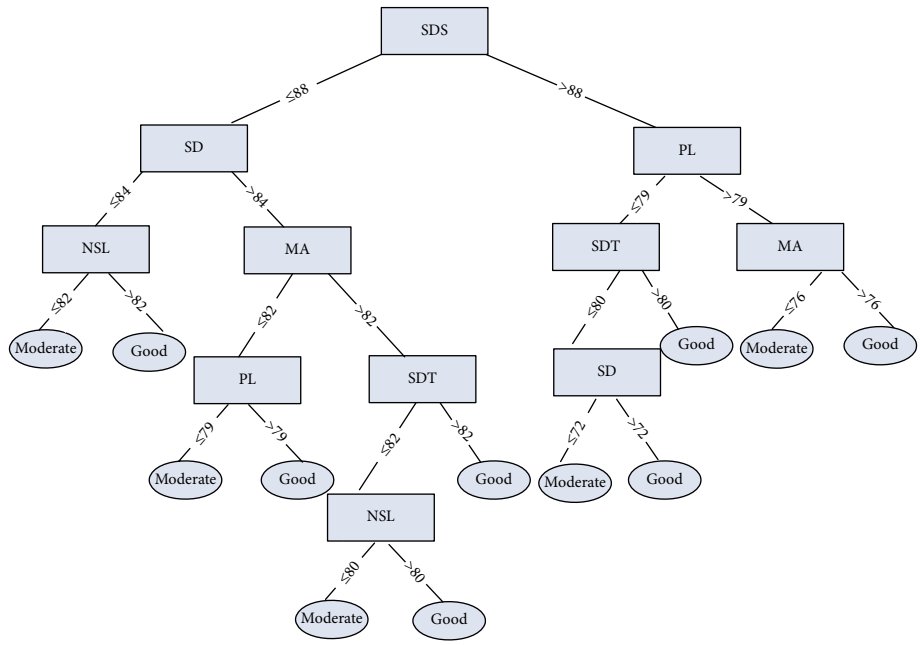


Figure 6: Decision tree by using C4.5 algorithm.

development ability of undergraduates for sure plays an important role in enterprise development. On top of that, the root "SDS" has two branches "PL" and "SD" compared with the three sub-trees "PL", "NSL", and "MA" shown in Figure 4.

According to the characteristics of decision trees, only the leaf nodes of a tree represent the final result of a classification mission. Hence, the downward branch of "PL" and "SD" in Figure 6 does not make the ultimate decision of "Good" or "Moderate" for a course group. A possible explanation for this might be that the situations of programming language and database systems are not enough to make a judgment. Further information about other course groups is needed. The result matches those observed in actual prestigious IT companies in China. A perfect example can be found in Huawei Technology Corporation, who will recruit new employees by testing multiple aspects, especially a series of technology groups.

Thirdly, another important finding was that the evaluator of a company can benefit a lot from taking the factor of height difference between the left and right sub-trees. This finding may partly be explained by the example shown in Figure 6, in which the height of the left (res. right) subtree of the root is 5 (res. 4). This structure emphasizes the importance of "SDS" since the higher (> 88) the root attribute score is, the less of the subsequent judgment. Otherwise, more steps are needed in order to decide whether the skills of a graduate are in conformity with the employment needs.

Finally, the weight value (breakpoint) can be invoked as a criterion of importance. For example, breakpoint scores, i.e., 79, 80, 76, and 72, of the right branch from the root "SDS" are generally lower than the left branch, i.e., 84, 83, 82, and 79, respectively. One accepted finding is the extreme importance of software development skills. On top of that, the node "PL" only appeared in the left branch of the root. Thus, companies believe that some undergraduates possess the basic solid theoretical knowledge if they have proficient development skills. Course score of programming languages will be further investigated when software development skills of a graduate are not satisfactory.

The following conclusions can be drawn from the present study. Firstly, the path (SDS → SD → NSL →"Good") shown in Figure 6 illustrates some of the main characteristics of the tree decision for graduate assessment. A graduate can be decided as "good" if he/she has practical skills and excellent language communication skills even if "SD" is weak. On top of that, the breakpoint of English skill (NSL) at the level 6 is lower than that at the level 3 on the tree. Language communication ability can form a certain compensation from SDT and MA by considering the breakpoints in the path (SD → MA → SDT → NSL →"Good"). This path illustrates the fact that an excellent theoretical foundation is suitable for some company positions such as algorithm engineer.

3.4. Stratified k-Fold Cross-Validation. K-fold cross-validation [26] is a statistical method for accuracy estimation and model selection. It divides a target data set into k subsets. One of them is considered as a test set and the remaining $k - 1$ subsets as a training set. Hence, k distinct scores can be obtained. The final average accuracy score is determined as the correct rate of the classification model. However, this method may not work for the data set in this paper. Among the data, the best two classes with some consecutive numbered students have more than 85% evaluations are labeled with "Good". K-fold cross-verification may select a test set containing all the records with labeled "Good", which may lead to a very low validation accuracy. The data features in this article do not apply to simple K-fold cross-validation methods.

The extended method stratified k-fold cross-validation [27] avoids the unpleasant feature of data, in which the test

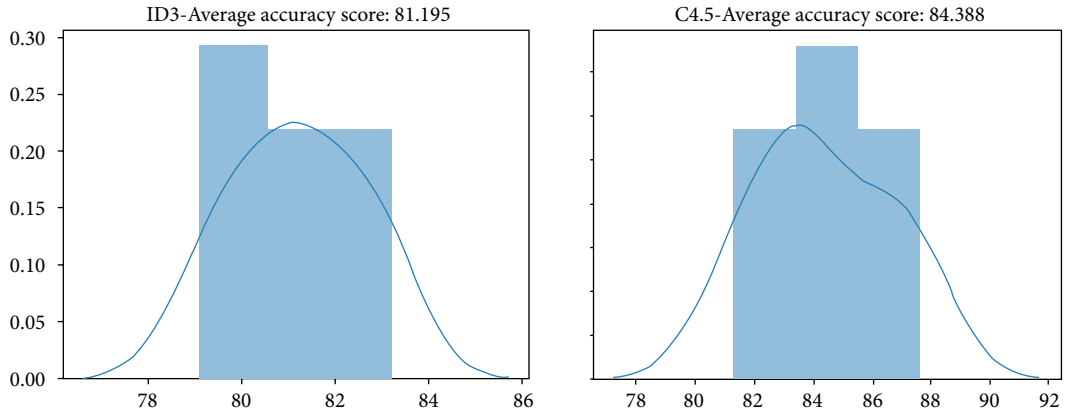


Figure 7: The average accuracy score of ID3 and CD4.5 for focused course classification problem.

set will take 1/10 (k=10) from each subset so that the proportion of each subset category ("Good" or "Moderate") is the same as the ratio for the entire data set. As a result, it is impossible to appear unfortunate test sets with only labeled "Good" or "Moderate". As shown in Figure 7, by performing on the two models ID3 and C4.5 involved in the education problem of this manuscript, the average accuracy is 81.195 and 84.388 by performing ID3 and C4.5 to the educational course data sets, respectively. From the perspective of accuracy, the decision tree shown in Figure 6 has advantages.

4. Course Grouping Recommendation Algorithm

This section proposes an algorithm based on decision trees to recommend an appropriate course scheme. Logical details on the course recommendation task can be shown in Algorithm 3. When the algorithm input a well-trained decision tree model, it will filter out the nondominant course nodes through the depth-first traversal of the tree. Simultaneously, the dominant nodes from the root of a decision tree to a leaf node are saved into a set during the depth-first traversal process. Finally, we will get sets of dominant course with the same number of leaf nodes. These sets represent different recommended course groups. If the teaching reform department of the university can increase the teaching hour and difficulty of these courses through reform and the related explanations of the course in the same group can be illustrated by teachers as much as possible, better employment quality will appear. Continuous curriculum improvement according to the mechanism has a positive effect on the quality of student employment.

Perfect examples can be found in such well-recognized scenarios of new student career planning and career skill group recommendation. Students will clearly realize which combination of course may lead to a beneficial employment environment. Real-time scores and employment data highlight the importance of the following algorithm. The recommended course groups and their expected scores are calculated and shown in Figure 8 if the tree shown in Figure 6 is considered as the input of CGR algorithm.

Input: Decision tree *DeTree*.
Output: Course optimal paths.
for each *node n* in *DeTree* **do**
  if n_i *is not be traversed* **then**
    add processed token "T";
    if n_i is an leaf **then**
      (1) output the set of path that satisfy $(n_1(d_1), n_2(d_2), \ldots, n_i(d_i))$ with $d_j \in D, 1 \le j \le i$;
      (2) delete any node $n_j(d_j)$ labeled operators "≤";
    else
      add n_i into the current path set;
    end
  else
    traverse the next n_i according to Depth-first search (DFS).
  end
end

Algorithm 3: CGR: course grouping recommendation algorithm.

The findings of this research provide insights for current students about their job search skills in different career positions. Experiment results show that IT companies pay attention to the following course groups: Group 1 (Sales and Product Manager), Group 2 (Software Engineer and Technical Management), Group 3 (Software Engineer and Research), Groups 4 and 5 (Algorithm Engineer and Research), Group 6 (Software Engineer), and Group 7 (Technical Management and Other Administrative Post). Thus, their expected staffs in the next two years should be well-equipped with appropriate skills for different careers or jobs in a company.

5. Conclusion

This research is undertaken to design a data-based framework and provide a conventional solution for the problem of timely

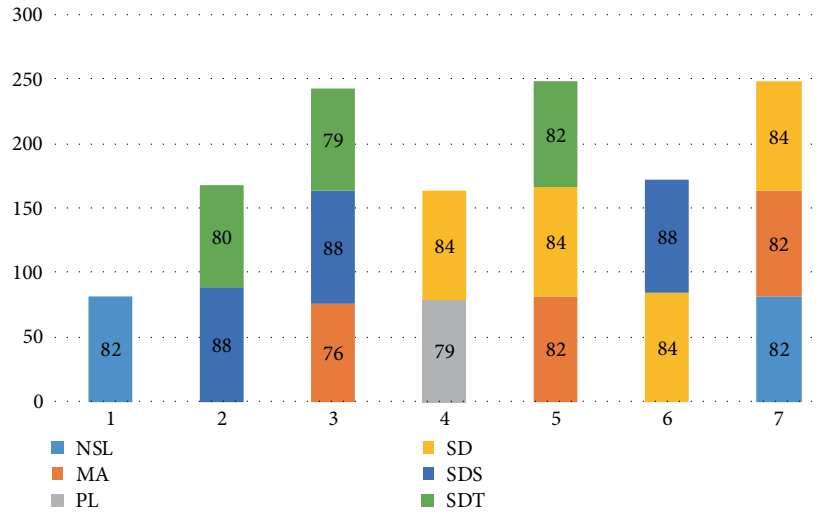


Figure 8: Recommended course groups by CGR algorithm.

course update of the talent training plans for a target major. The second aim of this study is to alleviate the contradiction between the talent training plan and the real-time needs of contemporary IT enterprises. The investigation of the framework has shown that a simple 4-stage data processing and a subsequent recommendation algorithm CGR can get a course update with time and social advantages. Experimental results show that the decision tree can play a positive role in the data-based teaching reform.

On the other hand, empirical recommendations of course may not necessarily serve as an effective way to increase the employment rate and income for some undergraduates. Hence, algorithm CGR is investigated after the basis of decision tree analysis. Feedback data are collected in 2018. These groups are valued far more than the old one (version 2014). As a result, the employment rate increased by 3.4 percent and the average monthly salary increased by 1,400 of 2018 computer undergraduates in Xihua University. These findings contribute in university education to our understanding of intelligent education and provide a basis for agile course plan recommendation. To sum up, these results add to the rapidly expanding field of elementary education planning to intelligent science applications.

An issue that was not addressed in this study was how to obtain a course plan containing new required course according to historical data. Hence, greater efforts are needed to make a recommendation by considering additional supplementary data such as advanced technology progress.

Data Availability

The data used to support the findings of this study (article ID 7140797 submitted to DDNS) are included within the article by reporting a website link shown in Line 1, Page 6 in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of Interest.

Acknowledgments

Fruitful discussion with Yue Wu and Yongquan Fan is gratefully acknowledged. The authors thank anonymous reviewers for many useful discussions and insightful suggestions. This work is supported by the the Chunhui Plan Cooperation and Research Project, Ministry of Education of China (No. Z2015100), the National Natural Science Foundation (Grant Nos. 61902324, 61872298, 61802316, 61602389, 61472329, 61532009), the Civil Aviation Administration of China (No. PSDSA201802), the Chengdu Science and Technology Bureau (Nos. 2016-XT00-00015-GX, 2017-RK00-00026-ZF), Science and Technology Department of Sichuan Province (Nos. 2016JY0244, 2017RZ0009, 2018GZ0096, 2019GFW131, 2020JY, 2020GFW), Scientific Research Fund of Sichuan Provincial Education Committee (Nos. 15ZB0134, 17ZA0360), Sichuan Science and Technology Innovation Seedling Project (No. 192523), the Key Scientific Research Fund of Xihua University (No.z1412616), and the University-sponsored Overseas Education Project of Xihua University.

References

[1] F. J. Garca-Peffalvo and A. J. Mendes, "Exploring the computational thinking effects in pre-university education," *Computer in Human Behavior*, vol. 80, pp. 407–411, 2018.

[2] P. Premand, S. Brodmann, R. Almeida, R. Grun, and M. Barouni, "Entrepreneurship education and entry into self-employment among university graduates," *World Development*, vol. 77, pp. 311–327, 2016.

[3] M. E. Dulama and O. R. Ilovan, "How powerful is feedforward in university education – a case dtudy in romanian geography education on increasing learning efficiency," *Educational Sciences: Theory and Practice*, vol. 16, no. 3, pp. 827–848, 2016.

[4] A. Yadav, S. Gretter, S. Hambrusch, and P. Sands, "Expanding computer science education in schools: understanding teacher experiences and challenges," *Computer Science Education*, vol. 26, no. 4, pp. 235–254, 2016.

[5] A. Repenning, R. Grover, K. Gutierrez et al., "Scalable game design: a strategy to bring systemic computer science education to schools through game design and simulation creation," *ACM Transactions on Computing Education*, vol. 15, no. 2, pp. 1–31, 2015.

[6] J. R. Quinlan, "Decision trees and decision making," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 339–346, 1990.

[7] S. Ruggieri, "Efficient C4.5 classification algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 438–444, 2002.

[8] S. Sathyadevan and R. R. Nair, "Comparative analysis of decision tree algorithms: ID3, C4. 5 and random forest.," *Computational Intelligence in Data Mining*, vol. 1, pp. 549–562, 2015.

[9] C. J. Mantas, J. Abellán, and J. G. Castellano, "Analysis of Credal-C4.5 for classification in noisy domains," *Expert Systems with Applications*, vol. 61, pp. 314–326, 2016.

[10] Z. Yu, F. Haghighat, B. C. Fung, and H. Yoshino, "A decision tree method for building energy demand modeling," *Energy and Buildings*, vol. 42, no. 10, pp. 1637–164, 2010.

[11] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Transactions on Power Systems*, vol. 25, no. 3, pp. 1611–1619, 2010.

[12] H. C. Chen and S. Bennett, "Decision-tree analysis for predicting first-time pass/fail rates for the NCLEX-RN in associate degree nursing students," *Journal of Nursing Education*, vol. 55, no. 8, pp. 454–457, 2016.

[13] H. Kaur and S. K. Wasan, "Empirical study on applications of data mining techniques in healthcare," *Journal of Computer science*, vol. 2, no. 2, pp. 194–200, 2006.

[14] Z. Yuan and C. Wang, "An improved network traffic classification algorithm based on hadoop decision tree," in *Proceeding of the 2016 IEEE International Conference of Online Analysis and Computing Science*, pp. 53–56, IEEE, China, 2016.

[15] D. Zhang, X. Zhou, S. C. Leung, and J. Zheng, "Vertical bagging decision trees model for credit scoring," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7838–7843, 2010.

[16] T. C. Chang and H. Wang, "A Multi criteria group decision-making model for teacher evaluation in higher education based on cloud model and decision tree," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 12, no. 5, pp. 1243–1262, 2016.

[17] G. S. Abu-Oda and A. M. El-Halees, "Data mining in higher education: university student dropout case study," *International Journal of Data Mining and Knowledge Management process*, vol. 5, no. 1, pp. 15–27, 2015.

[18] P. Kaur, M. Singh, and G. S. Josan, "Classification and prediction based data mining algorithms to predict slow learners in education sector," *Procedia Computer Science*, vol. 57, pp. 500–508, 2015.

[19] R. R. Kabra and R. S. Bichkar, "Performance prediction of engineering students using decision trees," *International Journal of Computer Applications*, vol. 36, no. 11, pp. 8–12, 2011.

[20] A. Hamoud, A. S. Hashim, and W. A. Awadh, "Predicting student performance in higher education institutions using decision tree analysis," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 2, pp. 26–31, 2018.

[21] A. Topirceanu and G. Grosseck, "Decision tree learning used for the classification of student archetypes in online courses," *Procedia Computer Science*, vol. 112, pp. 51–60, 2017.

[22] E. Sugiyarti, K. A. Jasmi, B. Basiron, and M. Huda, "Decision support system of scholarship grantee selection using data mining," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15, pp. 2239–2249, 2018.

[23] A. Hershkovitz and R. Nachmias, "Online persistence in higher education web-supported courses," *The Internet and Higher Education*, vol. 14, no. 2, pp. 98–106, 2011.

[24] F. Elghibari, R. Elouahbi, and F. El Khoukhi, "Data mining for detecting e-learning courses anomalies: an application of decision tree algorithm," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 3, pp. 980–987, 2018.

[25] "Student basic data set of Xihua University from2014 to 2017," https://github.com/1007105767/StudentsData.

[26] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of *k*-fold cross validation in prediction error estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 569–575, 2009.

[27] N. A. Diamantidis, D. Karlis, and E. A. Giakoumakis, "Unsupervised stratification of cross-validation for accuracy estimation," *Artificial Intelligence*, vol. 116, no. 1-2, pp. 1–16, 2000.