International Journal of Innovative Computing, Information and Control
Volume **14**, Number **6**, December **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 2129–2140

THE PREDICTIVE MODELING FOR LEARNING STUDENT RESULTS BASED ON SEQUENTIAL RULES

Huu-Quang Nguyen[1], Thi-Thiet Pham[2], Van Vo[2], Bay Vo[3,4,*]
and Thanh-Tho Quan[5]

[1]Office of Academic Affairs
[2]Faculty of Information Technology
Industrial University of Ho Chi Minh City
12 Nguyen Van Bao Street, Ward 4, Go Vap District, Ho Chi Minh City, Vietnam
{ nguyenhuuquang; phamthithiet; vttvan }@iuh.edu.vn

[3]Division of Data Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[4]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
19 Nguyen Huu Tho Street, Tan Phong Ward, District 7, Ho Chi Minh City, Vietnam
[*]Corresponding author: vodinhbay@tdtu.edu.vn

[5]Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
68 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
qttho@hcmut.edu.vn

Received April 2018; revised July 2018

Abstract. *Nowadays, learning activities at universities in Vietnam are mostly in the form of credit-based mode. That is, to graduate students have to complete the subjects specified in the curriculum including the compulsory and optional ones. Therefore, to achieve their best performance, students would need guidelines on study direction in the compulsory subjects and choose the optional courses appropriate to their interests and abilities. Based on these practical requirements, the paper proposes a tool to assist students in predicting their own academic performance in order to improve their academic ability and be more scientifically grounded. In addition, the tool also helps students choose the subjects for the next semester in a reasonable manner. This tool is based on a set of sequential rules derived from the learning result of the students. To evaluate the performance of the proposed model, this tool was tested from real students records in the Faculty of Information Technology in Ho Chi Minh City University of Industry.*
Keywords: Sequence database, Sequential pattern, Student assistance tool, Sequential rule, Prefix-tree

1. **Introduction.** In Vietnam universities, students study mostly in the form of credit. To be considered for graduation, students must complete and accumulate the grade points for the subjects specified in the curriculum including many compulsory and optional ones. Therefore, they must have a sense of direction in the compulsory subjects and choose the optional courses which are appropriate to their interests and abilities in order to achieve the best results.

For compulsory and optional subjects, students must complete a number of prerequisite subjects in order to enroll in the following subjects. Therefore, they need to have a specific orientation in how to select and arrange the courses reasonably to be able to complete the program within the specified time. This orientation should be implemented as soon as the students are still freshmen.

DOI: 10.24507/ijicic.14.06.2129

The classification of subjects that are appropriate, necessary and attractive to students; to determine subjects which make students interested in learning better in the following subjects in the study path is also a big problem for the educational managers at the university.

In this research, we address this issue by considering *sequential pattern mining* [1-6], which is one of the major directions in the field of data mining and is being studied by many computer scientists. Sequential pattern is the exploitation of common patterns related to time or other events, with the requirement that common patterns are substrings in the sequence database whose occurrence is greater than the minimum threshold support specified by the user. This is an important part of data mining and is widely used in many fields such as medicine, education, training, economics, science and society.

Sequence database is a large set of sequences where each sequence is a list of the sorted events. In the real world, people are constantly gathering data and archiving to get a huge database, making many industries increasingly interested in exploiting sequential patterns from the database. However, this process encountered two major challenges. The first is the mining time, and the second is the problem of extracting sequential rules. The sequential rules extend the usability and meaning of the sequential patterns, representing the latent knowledge of sequence database. This set of sequential rules extracted from the sequence database is the basis for building our forecasting system, used as an assistance tool for students as previously discussed.

From the above-mentioned issues, it is necessary to have a tool such that students can predict the learning outcomes of themselves in next subjects and from that can also help students adjust their better learning ability as well as choice of optional subjects more scientific.

Some contributions are also introduced for building the course recommendation systems by Sunita and Lobo [7], Huynh et al. [8], Amer and Jamal [9], Campagni et al. [10], etc. Sunita and Lobo [7] proposed an architecture for course recommender system and how the data flows through this system. The Moodle tool which is the learning management system for collecting the data and the Weka tool are used to build the system in [7]. Huynh et al. [8] introduced several methods which can be used for building the course recommendation systems. Their study also compared and analyzed their performance by using a real educational dataset. Amer and Jamal [9] applied association rule mining algorithms to getting recommendation system. Campagni et al. [10] proposed a model to mine the students' results in three steps including: (1) mining frequent sequence patterns based on the SPAM (Sequential PAttern Mining) algorithm, (2) select the most common pattern and then (3) cluster analysis based on the results of previous steps. The results of the study have shown what type of exams will make it difficult for students. The research also demonstrates that the proposed methodology can be applied to all the student learning outcomes. However, the above systems were not applied in practice.

Therefore, this paper proposes the predictive modeling as a tool that allows students who are studing at Faculty of Information Technology of Industrial University of Ho Chi Minh City to predict their learning results based on sequential rules.

The rest of this paper is organized as follows. Section 2 reviews some works related to mining the content of paper. The proposed modeling and the pseudo-code for the proposed modeling are discussed in Section 3. Section 4 presents a case study of the learning student results based on sequential rules. Section 5 presents the experimental results, and conclusion and future work are presented in Section 6.

2. **Related Works.** Sequential pattern mining plays an important role in the area of data mining research, and has a broad range of applications, e.g., analysis of DNA sequence patterns, customer purchase behavior analysis, and network access mode of analysis, the analysis of scientific experiments, disease treatment early diagnosis, and prediction of natural disasters. Sequential pattern mining is firstly proposed by Agrawal and Srikant in 1995 [1]. After that, the same authors further based on the Apriori property to develop a generalized and refine algorithm, called GSP (Generalized Sequential Patterns) [2]; but it incorporates time constraints, sliding time windows, and taxonomies in sequence patterns. These algorithms were deployed on multiple repositories and achieved good performance on transactional items. There have also been many algorithms after that, which are proposed to improve the effect of mining sequential patterns such as SPADE (Sequential PAttern Discovery using Equivalence classes) algorithm [3]. SPADE divides the candidate sequences into distinct groups such that each group can be completely stored in the main memory. The SPADE algorithm outperforms GSP by a factor of two at lower support values. However, the searching in the format of dataset is done by the id-list interaction and needs to scan database three times in each mining so the SPADE algorithm consumes much more time to transform the horizontal dataset to vertical dataset and also require additional storage space several times larger. SPAM algorithm [4] can speed up the mining process of sequential patterns by using a lexicographic sequence tree and bitmap representation. SPAM outperforms SPADE by about a factor on small datasets and better than an order of magnitude for reasonably large datasets but the SPADE algorithm is more space-efficient than the SPAM because the whole algorithm of SPAM with its data structure fits in the main memory. Prefix-Span algorithm [5] only examines the prefix subsequences and only projects their corresponding postfix subsequences into projected databases. In each projected database, sequential patterns are grown by exploring only local frequent patterns. By using Pattern-Growth in the sequential pattern mining, Pei et al. [5] conducted a systematic study of the exploitation of sequential patterns in large databases and developed an approach of pattern growth for efficient extraction and expansion of sequential patterns. In this approach, a sequence database is recursively referenced to a smaller set of reference databases, and sequential patterns were developed in each reference database by exploring localized fragments. This mining method can be extended to exploit sequential patterns with user-defined constraints. The PrefixSpan performed better than the SPADE algorithm. However, the prefixspan algorithm required high memory space for creating and processing of huge number of projected sub-databases. PRISM (PRIme-encoding based Sequence Mining) algorithm [6] used the primal block encoding approach to represent candidates' sequences, and join operations over the primal blocks to determine the frequency for each candidate. Experiments in [6] have also shown that PRISM algorithm is one of the best methods for mining sequential patterns. It outperforms existing methods by an order of magnitude or more, and has a very low memory footprint.

Sequential rules express temporal relationships between sequential patterns in the sequence database [11]. Sequential rules can be considered as natural extension of sequential patterns, similar to association rules that is natural extension of frequent itemsets.

In the field of sequential rule mining, there are studies for the full set of sequential rules from sequential patterns such as Lo et al. [12], and Fournier-Viger et al. [13,14]. However, if the sequential rule mining is based on other measures such as in Pham et al. [15], or lift [16] or conviction [17], the above approach is inappropriate. The method to mine the full set of sequential rules proposed by Spiliopoulou [18] is appropriate because it is mainly based on the brute force method. However, it consumes computational costs for generating and test candidates.

Fournier-Viger et al. [13] presented an algorithm for mining sequential rules. First, the algorithm looks for $1 * 1$ rules and then recursively develops them by scanning sequences containing them to find the elements that can extend their left or right parts to generate the set of popular sequential rules. Experimental results of the author group based on actual data sets have demonstrated significant performance time and memory.

Van et al. introduced the IMSR_PreTree (Improve Mining Sequential Rules) algorithm [11] to generate common sequential rules based on the prefix tree. In order to find a common sequence of samples, the authors used a longitudinal approach for counting and computation of support based on PRISM algorithm. Research has shown that the proposed algorithm can significantly reduce search space in the mining process by trimming seedlings to provide non-critical rules in the early stages.

Methods for mining non-redundant sequential rules, which remove a significant number of redundant sequential rules in the process of mining sequential rules from the set of sequential patterns, have been developed in [19,20]. Pham et al. [19] used the prefix-tree structure that stored the set of sequential generator patterns and closed sequential patterns to mine non-redundant sequential rules. The approach of Tran et al. [20] uses a dynamic bit vector data structure and adopts a prefix tree to compress the data in the mining process to mine non-redundant sequential rules directly from sequence databases.

3. **The Proposed Model for Mining the Learning Student Results.** The proposed predictive model for mining the learning student results is introduced in Figure 1.

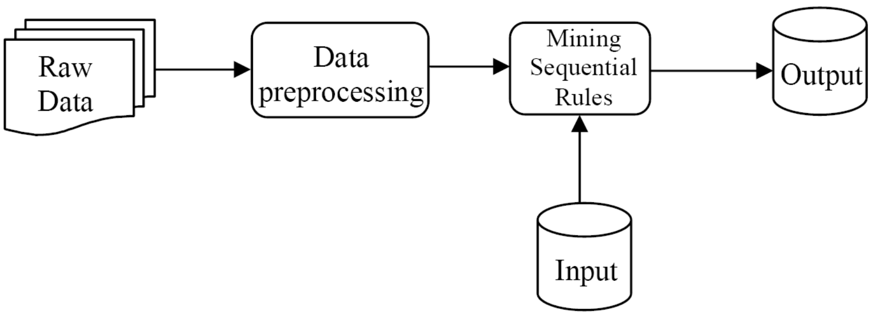

FIGURE 1. The proposed predictive model for mining the learning student results

3.1. **Data preprocessing.** Data preprocessing is a data mining technique that involves transforming raw data into an understandable format for specific purpose. In real world application, data are generally incomplete (for example in these cases: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), noisy (containing several errors or outliers) and inconsistent (containing discrepancies in codes or names).

Data preprocessing is one of the most required steps in a data mining process which deals with the preparation and transformation of the initial dataset. In general, data preprocessing methods are divided into following categories: data cleaning, data integration, data transformation and data reduction.

In our case, the database is collected from the results of students who have studied credit learning mode at the Faculty of Information Technology, HCMC University of Industry (from 2010 to January 2017). The preprocessing processes such as filtering of required attributes, clearing of unusual data, and exclusion of outlier elements improve data quality. Otherwise, data transformation is used to be consistent with the proposed algorithm for sequential mining.

3.2. **Input and output data.** The input data of our proposed approach is a collection of one or more subjects' students studied and result of the corresponding subjects.

The output data is the predicted result of the subjects that student will study in the incoming time. Based on this prediction, the students can assess and adjust their studying as well as the basics of optional subjects.

3.3. **Mining sequential rules.** There are several algorithms to mine sequential rules such as Lo et al. [12], Pham et al. [15], Spiliopoulou [18], and Fournier-Viger et al. [21]. We selected IMSR_PreTree algorithm [11] which was proposed by Van et al. in 2014 to apply in mining sequential rules in student results because it can be used for mining all sequential rules as some algorithms introduced in Section 2. Experimental results [11] showed that IMSR_PreTree is faster than the others above and it is very effective in mining large databases.

IMSR_PreTree [11] uses a prefix-tree structure. Every node in the prefix-tree is always the prefix of its child nodes (only sequence-extended nodes). Consequently, it is possible to generate rules directly without scanning all frequent sequences for prefix checking. So the IMSR_PreTree algorithm [11] is used for the popular sequential rule mining phase in the proposed model.

Figure 2 shows the IMSR_PreTree algorithm [11]. It creates rules from the set of sequential patterns stored in the prefix tree. The creating rules' process of this algorithm will remove the samples for minor rules so as to minimize the tree traversal.

```
Input: FS – the set of frequent sequences stored in prefix-tree, minConf
Output: All significant rules
Algorithm:
GENERATE_RULES_FROM_PREFIX-TREE()
      1. For each node r at level 1 in prefix-tree
      2.    GENERATE_RULES_FROM_ROOT(r)
GENERATE_RULES_FROM_ROOT(r)
      3. Let pre = r.sequence
      4. For each node nseq ∈ r.Seq_Extended_Set
      5.    For each node l ∈ sub-tree nesq
Step 1 6.        Try to from rule = pre->post,
                                pre++post = l.sequence
      7.         If sup(l.sequence)/sup(pre) ≥ minConf
      8.             Output rule
      9.         Else Stop generating rules from l.children with prefix pre
                 //according to Theorem 1
      10. For each node nesq ∈ r.Seq_Extended_Set
      11.     GENERATE_RULES_FROM_ROOT(nesq)
Step 2
      12. For each node nesq ∈ r.Items_Extended_Set
      13.     GENERATE_RULES_FROM_ROOT(nesq)
```

FIGURE 2. IMSR_PreTree algorithm [11]

This procedure is performed using the following steps.

Step 1: Generating rules with the prefix that is a sequence at the root. Let *pre* be the sequence at root node (line 3). We find that sequence *pre* is the prefix of all the sequences in sequence-extended sub-trees of the root r. Hence, we generate only rules from the sequences in these sub-trees in turn. We generate rule from each sub-tree with prefix *pre* (lines 3-9). In this process (lines 5-9), we traverse the sub-tree in a depth-first manner so that the pruning technique can be applied.

Step 2: Because all extended nodes of the current root are the prefix of the sub trees at the next level, we call this procedure recursively for every extended-node of the root (lines 10-13). This process is recursively repeated until reaching the last level of the tree.

For example: Given a sequence database as Table 1 with distinct items being $\{A, B, C\}$.

Table 1. A sequence database

SID	Sequence
1	⟨(AB)(B)(B)(AB)(B)(AC)⟩
2	⟨(AB)(BC)(BC)⟩
3	⟨(B)(AB)⟩
4	⟨(B)(B)(BC)⟩
5	⟨(AB)(AB)(AB)(A)(BC)⟩

With ***minSup*** = 50%, the set of sequential patterns from this sequence database in Table 1 that consists of sequential patterns has the support value $Sup \geq minSup$, i.e., $Sup \geq 50\% \times 5 \approx 3$. Totally all 21 sequential patterns are found and shown in Table 2.

Table 2. Sequential patterns

Size	⟨Sequential pattern⟩: Support value
1	⟨(A)⟩: 4, ⟨(B)⟩: 5, ⟨(C)⟩: 4, ⟨(AB)⟩: 4, ⟨(BC)⟩: 3
2	⟨(A)(B)⟩: 3, ⟨(A)(C)⟩: 3, ⟨(AB)(B)⟩: 3, ⟨(AB)(C)⟩: 3 ⟨(B)(A)⟩: 3, ⟨(B)(B)⟩: 5, ⟨(B)(C)⟩: 4, ⟨(B)(AB)⟩: 3, ⟨(B)(BC)⟩: 3
3	⟨(A)(B)(B)⟩: 3, ⟨(A)(B)(C)⟩: 3, ⟨(B)(B)(B)⟩: 4, ⟨(B)(B)(C)⟩: 4, ⟨(AB)(B)(B)⟩: 3, ⟨(AB)(B)(C)⟩: 3, ⟨(B)(B)(BC)⟩: 3

From the set of sequential patterns in Table 2 and *minConf* = 70%, after applying IMSR_PreTree algorithm in Figure 2, there are 18 sequential rules, which satisfy *minConf*, generated as shown in Table 3 (only generating sequential rules from the sequential patterns which have the length of it greater than 1).

4. **The Case Study.** In this section, we present a case study showing how the mined sequential rules can be used to help students in their study paths. According to the curriculum of each discipline in the university, students must complete certain prerequisite subjects in order to register for the next subject. So, students always want to acquire a guideline which can assist them in selecting and arranging the subjects as appropriate to be able to complete the program within the specified time.

For example, in our educational program in three first semesters, there are many subjects that are not used for sequential predictions. The subjects in Table 4 only cover the subjects to predict the rules. Table 4 shows the results from Semester 1 to 3 of student ID 16026741. Table 5 describes the educational program in Semester 4, and it includes the compulsory and optional subjects. We consider a case as follows. The student ID 16026741 in the beginning of the semester 4 wants to get the good guidance to register the optional subjects. According to the educational program in Table 5, in this semester the student has to complete 3 compulsory subjects and on the other hand he must choose 2 optional subjects in 4 subjects. The optional subjects in the 4th semester are devided

TABLE 3. The set of sequential rules generated from the set of sequential patterns

Sequential pattern	Sequential rule, *conf* = sup(X++Y)/sup(X)×100%	Confidence $conf \geq minConf$?
⟨(A)(B)⟩: 3	⟨(A)⟩ → ⟨(B)⟩, 3/4×100% = 75%	Yes
⟨(A)(C)⟩: 3	⟨(A)⟩ → ⟨(C)⟩, 3/4×100% = 75%	Yes
⟨(AB)(B)⟩: 3	⟨(AB)⟩ → ⟨(B)⟩, 3/4×100% = 75%	Yes
⟨(AB)(C)⟩: 3	⟨(AB)⟩ → ⟨(C)⟩, 3/4×100% = 75%	Yes
⟨(B)(A)⟩: 3	⟨(B)⟩ → ⟨(A)⟩, 3/5×100% = 60%	No
⟨(B)(B)⟩: 5	⟨(B)⟩ → ⟨(B)⟩, 5/5×100% = 100%	Yes
⟨(B)(C)⟩: 4	⟨(B)⟩ → ⟨(C)⟩, 4/5×100% = 80%	Yes
⟨(B)(AB)⟩: 3	⟨(B)⟩ → ⟨(AB)⟩, 3/5×100% = 60%	No
⟨(B)(BC)⟩: 3	⟨(B)⟩ → ⟨(BC)⟩, 3/5×100% = 60%	No
⟨(A)(B)(B)⟩: 3	⟨(A)⟩ → ⟨(B)(B)⟩, 3/4×100% = 75% ⟨(A)(B)⟩ → ⟨(B)⟩, 3/3×100% = 100%	Yes Yes
⟨(A)(B)(C)⟩: 3	⟨(A)⟩ → ⟨(B)(C)⟩, 3/4×100% = 75% ⟨(A)(B)⟩ → ⟨(C)⟩, 3/3×100% = 100%	Yes Yes
⟨(B)(B)(B)⟩: 4	⟨(B)⟩ → ⟨(B)(B)⟩, 4/5×100% = 80% ⟨(B)(B)⟩ → ⟨(B)⟩, 4/5×100% = 80%	Yes Yes
⟨(B)(B)(C)⟩: 4	⟨(B)⟩ → ⟨(B)(C)⟩, 4/5×100% = 80% ⟨(B)(B)⟩ → ⟨(C)⟩, 4/5×100% = 80%	Yes Yes
⟨(AB)(B)(B)⟩: 3	⟨(AB)⟩ → ⟨(B)(B)⟩, 3/4×100% = 75% ⟨(AB)(B)⟩ → ⟨(B)⟩, 3/3×100% = 100%	Yes Yes
⟨(AB)(B)(C)⟩: 3	⟨(AB)⟩ → ⟨(B)(C)⟩, 3/4×100% = 75% ⟨(AB)(B)⟩ → ⟨(C)⟩, 3/3×100% = 100%	Yes Yes
⟨(B)(B)(BC)⟩: 3	⟨(B)⟩ → ⟨(B)(BC)⟩: 3/5×100% = 60% ⟨(B)(B)⟩ → ⟨(BC)⟩: 3/5×100% = 60%	No No

TABLE 4. An example of student ID 16026741 result from Semester 1 to 3

Student ID	**Year**	**Semester**	**Subject ID**	**Result**	**Optional**
16026741	2016	1	Calculus A1	B	No
16026741	2016	1	Foundations of Computing	C+	No
16026741	2016	1	Foundations of Programming	B	No
16026741	2016	1	Studying Skills in IT	C	No
16026741	2016	2	Programming Techniques	A+	No
16026741	2016	2	Computer System	C+	No
16026741	2016	2	Discrete Structures	B+	No
16026741	2016	2	Calculus A2	A+	No
16026741	2017	3	Graph Theory	A+	No
16026741	2017	3	Object-Oriented Programming	B	No
16026741	2017	3	Data structures and Algorithms	C+	No
16026741	2017	3	Computer Networks	B	No

into 2 groups, one has 3 credits and the other has 2. Students must choose the optional subjects in each group based on the column optional and group.

If students do not get help from our proposed tool, they do not know how to choose which subjects he will get the best result depending on his abilities. The prediction accuracy of student performance is useful in many ways in universities. For example,

Table 5. The subjects for semester 4 of educational program

#	Semester	Subject ID	Subject Name	Credits	Optional (Yes/No)	Group
...	...	...	...	...	...	
18	4	4401	Computer statistics and applications	3	No	
19	4	4433	System analysis and design	4	No	
20	4	4436	Database system	3	No	
21	4	4555	Event-driven programming with .Net technology	3	*Yes*	*1*
22	4	4556	Event-driven programming with Java technology	3	*Yes*	*1*
23	4	4130	Probability statistics	2	*Yes*	*2*
24	4	4138	Logic	2	*Yes*	*2*
...	...	...	...	...	...	

Table 6. The example of the subjects' result predicted based on the result which student learnt

The result which student learnt (Subject [mark])	The subjects' result is predicted (Subject [mark])	Advice
Foundations of Computing [C+]	Database system [C+]	
	System analysis and design [B]	
Calculus A1 [B]	Probability statistics [B]	
	Logic [A]	Choose
Foundation of Programming [B]	Event-driven programming with .Net technology [B]	Choose
	Event-driven programming with Java technology [C]	

identifying the good students for scholarships or identifying the weak students who are likely to fail is also important for their right guidance. Real world data of students such as engineering students can be used in sequential pattern mining algorithms, strategic programs or tools can be planned for improving or maintaining student's performance during their period of studies in the university.

Students who want to know the predictions of future subjects can be chosen of the subjects they have studied as well as their respective scores. The predicted data will be processed and displayed based on the sequential rules.

Table 6 presents the set of sequential rule is generated from the set of sequential patterns with compulsory and optional subjects. With the result of Foundations of Computing marked C+ and using the helping tool with sequence patterns data mining, we can predict the results of database system and system analysis and design C+, B respectively. Similarity, for the result B of Foundation of Programming, we can consult the student to choose the event-driven programming with .Net technology (B) subject rather than event-driven programming with Java technology (C). Thus, using the helping tool with sequence patterns mining will support students to adjust their learning attitude to achieve higher results.

5. **Experimental Evaluation.** The results of student database are collected from the results of students who have studied credit at the Faculty of Information Technology, HCMC University of Industry, implemented on Windows 7 operating system with a Core I7 processor and 8 GBs of main memory.

The collected raw data is comprised of two main types of data.

(1) Course data in the curriculum from the 2010 to 2017. This includes the course contents: Course Code, Subject Name, and Semester. This data is consistent after comparing the courses between different courses to obtain the most common data among the courses. Then remove the non-specialized subjects to obtain the data as shown in Table 7.

Table 7. The subject for information technology students

Semester	**Subjects ID**	**Subjects Name**
1	1131	Calculus A1
1	1539	Introduction to Computing
...	...	...
5	5486	Database Security
5	5553	Application Development (with assignment)
...	...	...
8	8811	Compiler
8	8923	Specialized Project

(2) The point data is converted into literal point scales according to the scoring method of Ho Chi Minh City University of Industry. This data consists of 8 points scored on a literal scale including F, D, D+, C, C+, B, B+ and A.

Based on the information of the subject and the point scale used in Ho Chi Minh City University of Industry, the author discoursed the data to convert the data into integer form as shown in Table 8 (the discrete data is from 1 to 688).

Table 8. The discrete data of subjects' result

Subjects	**Subject ID**	**F**	**D**	**D+**	**C**	**C+**	**B**	**B+**	**A**
Calculus A1	1131	1	2	3	4	5	6	7	8
Introduction to Computing	1539	9	10	11	12	13	14	15	16
...									
Database Security	5486	361	362	363	364	365	366	367	368
Application Development	5553	369	370	371	372	373	374	375	376
...									
Complier	8811	673	674	675	676	677	678	679	680
Capstone Project	8923	681	682	683	684	685	686	687	688

(3) Student's score data includes information on the year, semester, subject, grade (literal scale), as well as discrete data as shown in Table 9.

The output is a sequential rule that can be exploited by the IMSR_PreTree algorithm based on student learning outcomes.

Table 10 shows the archived results of the mining sequential rules from the results of student database.

The collected data is the learning result of 433 students and in the case we use *minSup* = 20%, 59 sequential patterns are mined and 3 sequential rules satisfied *minConf* = 25%.

Table 9. The data of students' result after discretized

Students ID	**Year**	**Semester**	**Subjects ID**	**Subjects Name**	**Mark**	**Discrete Data**
11245051	2013	2	5425	Database management systems	B	222
11245051	2013	3	5428	Human Computer interaction	C	236
11245051	2013	3	6574	Network Administration and Technical Support	C	420
11245051	2013	3	6919	Assignments Module 1	B	438
11245051	2013	3	3481	Requirement analysis and management	A	104
...						

Table 10. The archived results of the mining sequential rules from the results of student database

minSup (%)	**Number of sequential patterns**	**Time (seconds)**	**minConf (%)**	**Number of sequential rules**	**Time (seconds)**
20	59	0.89178	50	2	0.00609
			25	3	0.00248
			0	3	0.00283
10	413	8.55645	50	20	0.054329
			25	277	0.052238
			0	282	0.055689
0	9674	348.4767	50	1040	22.815
			25	5328	22.812
			0	13372	29.0797

With $minSup$ = 10% and $minConf$ = 25%, there are 413 sequential patterns and 277 sequential rules. The chosen values of $minSup$ and $minConf$ will depend on the specific purpose, in the case to get the possible cases, we choose $minSup = 0$ and $minConf = 0$ for our experimental evaluation.

Based on the sequential rules extracted from the input data, the authors have developed the predictive application for student learning results as shown in Figure 3.

Figure 3 describes the predicting tool which predicted result of the student ID 16026741 in the 4th semester based on the list of subjects' result that student learnt in 03 semester before. The predicted results include the following subjects and the corresponding prediction scores. The compulsory subjects are system analysis and design – B, database system – C+. The optional subjects are event-driven programming with .Net technology – B, event-driven programming with Java technology – C, probability statistics – B, logic – A.

In this case, the prediction is the list of compulsory subjects that help the student know how to adjust his learning attitude to achieve higher results than predicted. Based on the prediction of optional subjects, the system helps the student choose subjects which likely have higher results. For example, in the group of optional subjects with the predictive

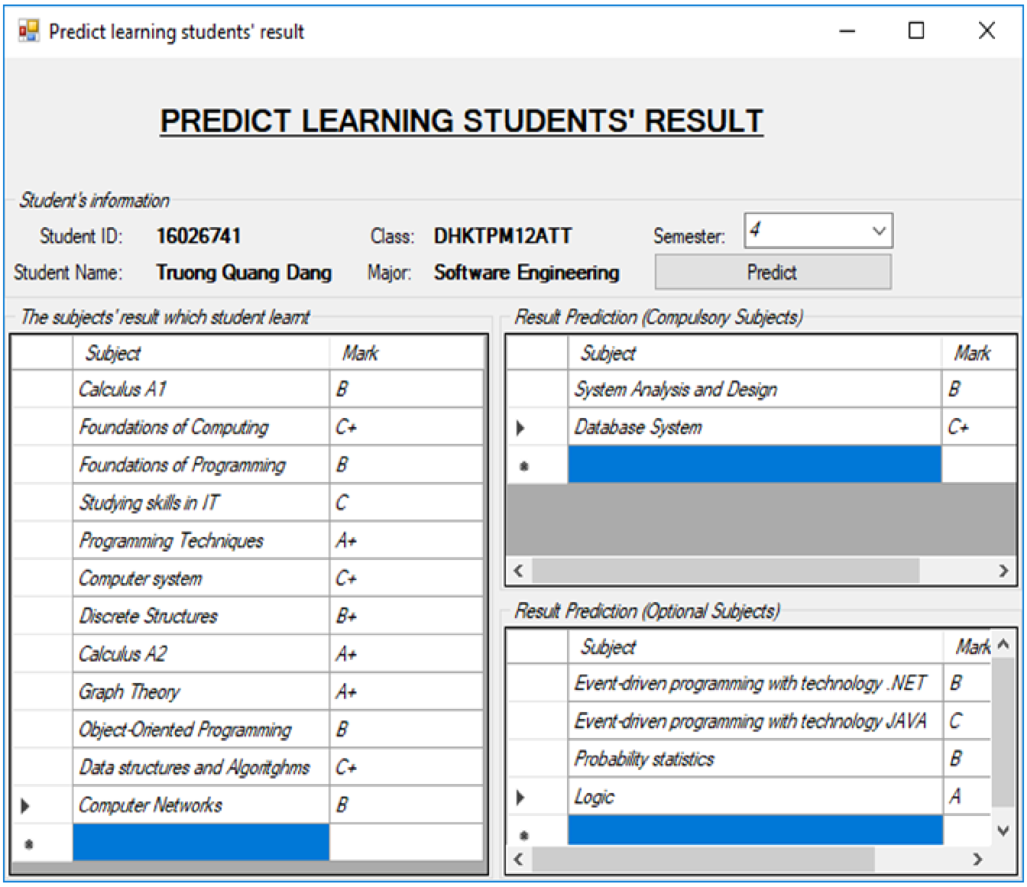

FIGURE 3. Predictive tool for the learning student results based on sequential rules

result of probability statistics subject being B mark and logic subject being A mark, the student should choose subject A rather than choose subject B.

With the input data being fixed sets, using the C# programming language and SQL server management system to store data, the application predicts student subjects' result of the next subjects depending on their old learning results that the students have completed before.

The displayed data in this application will be the result of the prediction that students may use this prediction data as a basis for selecting the appropriate subjects during the optional coursework registration process the next period as well as adjusting the learning of themselves.

6. **Conclusions.** The study was conducted on the basis of the results of learning by credit students at the Faculty of Information Technology of Industrial University in Ho Chi Minh City. Based on a sequential set of exploits, the authors have developed an application to predict student learning outcomes to assist students in predicting their own learning outcomes. This ability helps students to study and have more scientific basis in the selection of subjects to be registered each semester. The current database is small (433 records) and focuses only on specialized subjects so the number of sequential rules is not much. For the next work, the authors will collect more data and expand it to other subjects besides specialized subjects. In the future researches, the authors will proceed to exploit the set of sequential rules by using non-redundant rule sets, and then compare with the full set of sequential rules to develop the predictive model of students' results.

Acknowledgment. This research is funded by Industrial University of Ho Chi Minh City under grant number 182.CNTT03.

REFERENCES

[1] R. Agrawal and R. Srikant, Mining sequential patterns, *Proc. of the 11th International Conference on Data Engineering*, pp.3-14, 1995.

[2] R. Agrawal and R. Srikant, Mining sequential patterns: Generalizations and performance improvements, *Proc. of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, Avignon, France, pp.3-17, 1996.

[3] M. J. Zaki, SPADE: An efficient algorithm for mining frequent sequences, *The Journal of Machine Learning Research*, vol.42, nos.1-2, pp.31-60, 2000.

[4] J. Ayres, J. Flannick, J. Gehrke and T. Yiu, Sequential pattern mining using a bitmap representation, *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.429-435, 2002.

[5] J. Pei et al., Mining sequential patterns by pattern-growth: The PrefixSpan approach, *IEEE Trans. Knowledge and Data Engineering*, vol.16, no.10, pp.1424-1440, 2004.

[6] K. Gouda, M. Hassaan and M. J. Zaki, PRISM: A primal-encoding approach for frequent sequence mining, *Journal of Computer and System Sciences*, vol.76, no.1, pp.88-102, 2010.

[7] B. A. Sunita and L. M. R. J. Lobo, Course recommender system in e-learning, *International Journal of Computer Science and Communication (IJCSC)*, vol.3, no.1, pp.159-164, 2012.

[8] L. T. N. Huynh, H.-H. Nguyen and T.-N. Nguyen, Methods for building course recommendation systems, *International Conference on Knowledge and Systems Engineering (KSE 2016)*, pp.163-168, 2016.

[9] A.-B. Amer and A. Jamal, An automated recommender system for course selection, *International Journal of Advanced Computer Science and Applications*, vol.7, no.3, pp.166-175, 2016.

[10] R. Campagni, D. Merlini and R. Sprugnoli, Sequential patterns analysis in a student database, *ECML-PKDD Workshop: Mining and Exploiting Interpretable Local Patterns (I-Pat 2012)*, Bristol, 2012.

[11] T. T. Van, B. Vo and B. Le, IMSR_PreTree: An improved algorithm for mining sequential rules based on the prefix-tree, *Vietnam Journal of Computer Science*, vol.1, no.2, pp.97-105, 2014.

[12] D. Lo, S. C. Khoo and L. Wong, Non-redundant sequential rules theory and algorithm, *Information Systems*, vol.34, nos.4-5, pp.438-453, 2009.

[13] P. Fournier-Viger, U. Faghihi, R. Nkambou and E.-M. Nguifo, CMRules: Mining sequential rules common to several sequences, *Knowledge-Based Systems*, vol.25, no.1, pp.63-76, 2012.

[14] P. Fournier-Viger, C.-W. Wu, V. S. Tseng, L. Cao and R. Nkambou, Mining partially-ordered sequential rules common to multiple sequences, *IEEE Trans. Knowledge and Data Engineering*, vol.27, no.8, pp.2203-2216, 2015.

[15] T.-T. Pham, J. Luo, T.-P. Hong and B. Vo, An efficient algorithm for mining sequential rules with interestingness measures, *International Journal of Innovative Computing, Information and Control*, vol.9, no.12, pp.4811-4824, 2013.

[16] M. J. Berry and G. S. Linoff, *Data Mining Techniques for Marketing, Sales and Customer Support*, John Wiley & Sons, 1997.

[17] S. Brin, R. Motwani, J. Ullman and S. Tsur, Dynamic itemset counting and implication rules for market basket data, *Proc. of the 1997 ACMSIGMOD International Conference on the Management of Data*, pp.255-264, 1997.

[18] M. Spiliopoulou, Managing interesting rules in sequence mining, *Proc. of European Conference on Principles of Data Mining and Knowledge Discovery*, pp.554-560, 1999.

[19] T. T. Pham, J. Luo, T. P. Hong and B. Vo, An efficient method for mining non-redundant sequential rules using attributed prefix-trees, *Engineering Applications of Artificial Intelligence*, vol.32, pp.88-99, 2014.

[20] M. T. Tran, B. Le, B. Vo and T. P. Hong, Mining non-redundant sequential rules with dynamic bit vectors and pruning techniques, *Applied Intelligence*, vol.45, no.2, pp.333-342, 2016.

[21] P. Fournier-Viger, C.-W. Wu, V. S. Tseng and R. Nkambou, Mining sequential rules common to several sequences with the window size constraint, *Proc. of the 25th Canadian International Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol.7310, pp.299-304, 2012.