Integration of the PageRank Algorithm, Sequence Processing, and CPT+ for Webpage Access Prediction

Nguyen Thon Da, Tan Hanh, Pham Hoang Duy

Abstract: In this article, we provide a novel model to address the issue of webpage access prediction. In particular, the main approach we propose aims to reduce execution time by reducing the sequence space. This solution combines calculation of PageRank values of sequences in sequence databases and analysis of sequences from these shortened sequence databases. To evaluate the solution, we chose K-fold validation with K = 10 by randomizing the dataset 10 times; then the system calculated the average PageRank values of sequences. Next, with acceptable accuracy (when the size of datasets was reduced by up to 30% by PageRank calculation), we performed next access page prediction by analysing 1000 sequences. Experimental results for the real FIFA dataset show that our new proposed approach is much better than previous approaches in terms of prediction execution time.

Keywords: Webpage access prediction, sequence prediction, CPT+, PageRank algorithm.

I. INTRODUCTION

The problem of predicting a user's browsing behaviour on a website has a significant role in our lives these days [1]. Knowing the user's browsing history on the site grants us valuable information as to which of the most frequently accessed pages will be accessed next. Recently, a few works have proposed approaches for webpage access prediction. Some widely used data mining methods that have been applied to achieve this goal are association rule mining, clustering, and Markov classification [2]. There are two main categories for sequential rule mining: the first comprises algorithms for mining sequential rules appearing in a single sequence of events and the second consists of the algorithms that discover rules in sets of sequences (sequence databases) [3].

The problem of discovering rules in sets of sequences consist of finding all sequential rules from a sequence database such that their support and confidence are, respectively, higher than or equal to the user-defined Thresholds minSup and minConf. A few characteristic works are [3–6]. The original goal of association rule mining was to address the market basket problem [2].

With Markov models, each event solely depends on the previous events and these are built using only part of the information contained in training sequences. Therefore, these models do not use all the information contained in training sequences to perform predictions, and this can severely reduce their accuracy [7].

The clustering methods partition data objects into a number of homogeneous groups based on their similarity and do not classify user sessions directly but can help build better classification models if data objects are properly clustered. The drawbacks associated with them hinder their improvements when it comes to webpage access prediction and state space complexity [2].

Da et al. [8] presents a detailed survey of methods of solving issues related to next page access prediction.

Recently, Compact Prediction Tree (CPT), a good model for sequence prediction, has been presented. It relies on a tree structure and a more complex prediction algorithm to offer considerably more accurate predictions than many state-of-the-art prediction models. Nevertheless, a significant drawback of CPT is its high time and space complexity [7]. In Gueniche et al.[7], the authors address this issue by proposing three novel strategies to reduce the CPT's size and prediction time and increase its accuracy. Although webpage prediction using CPT+ has many advantages, better solutions are needed to achieve higher-accuracy predictions and to improve time performance. Because of this, we propose a novel method that is an integrated model combining web usage mining, the PageRank algorithm, sequence processing, and CPT+ for next page access prediction.

The remainder of the paper is organized as follows. In Section I, we present fundamental knowledge about sequence databases, a model that integrates the PageRank algorithm with CPT+, and a model that integrates sequence processing with CPT+. Then, in Section II we propose a novel approach to webpage access prediction that combines the use of PageRank calculation and analysis of sequences. In the subsequent two sections, we present an experimental evaluation and an experimental study. Finally, we conclude with our solution.

Revised Manuscript Received on March 09, 2020.

***** Corresponding Author

Nguyen Thon Da*, Department of Information Systems, University of Economics and Law, VNU-HCM, HCM City, Vietnam, Asia. Email: dant@uel.edu.vn

Tan Hanh, Department of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, Asia. Email: tanhanh@ptit.edu.vn

Pham Hoang Duy, Department of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, Asia. Email: duyph@ptit.edu.vn

II. BACKGROUND

A. Sequence database

In the context of CPT+ for predicting next page access, a sequence database is a set of sequences where each sequence is a list of pages. For instance, the table shown below contains four sequences. The first sequence, named S1, contains five webpages. This sequence means that Webpage page1 was followed by page2, page3, page4, and page6, respectively.

Table 1. A simple sequence database

ID	Sequences
S1	page1, page2, page3, page4, page6
S2	page4, page3, page2, page5
S3	page5, page1, page4, page3, page2
S4	page5, page7, page1, page4, page2, page3
S5	page7, page4, page8, page2

B. Compact Prediction Tree

CPT is a recently proposed prediction model which compresses training sequences without information loss by exploiting similarities between subsequences [7]. CPT contains significant structures: a PT (prediction tree), an LT (lookup table), and an inverted index. Like a prefix tree, the PT is a compact representation of the training sequences. The LT is an associative array which allows any training sequences in the PT to be located with a constant access time, and the inverted index is a set of bit vectors that indicates, for each item i from the alphabet Z, a set of sequences containing i..

For example, Fig. 1 illustrates the creation of the three structures by insertion of sequences.

$S_1 = \langle A, B, C, D, F \rangle$

$S_2 = \langle D, C, B, E \rangle$

$S_3 = \langle E, A, D, C, B \rangle$

$S_4 = \langle E, G, A, D, B, C \rangle$

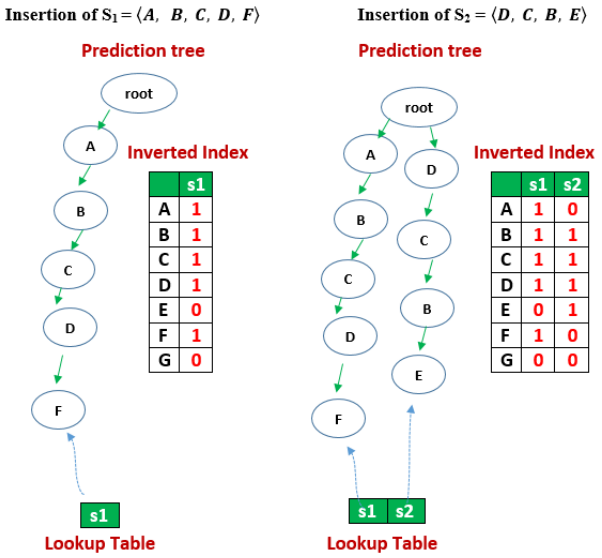

Fig. 1. Insertion of sequences S1 and S2 into Compact Prediction Tree

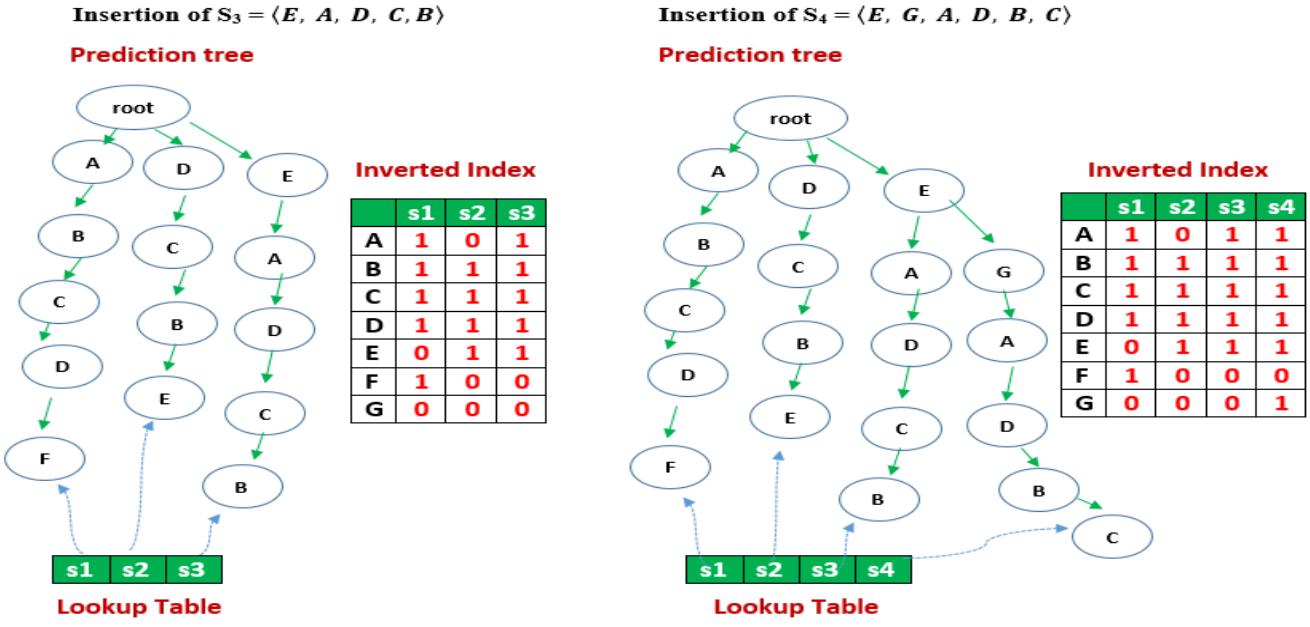

Fig. 2. Insertion of sequences S3 and S4 into Compact Prediction Tree

Prediction of the next items in a sequence S is performed by identifying the sequences similar to Py(S), where Py(S) is the suffix of S of size y. Each item found in the consequent of a similar sequence of s is stored in a data structure called a Count Table (CT), which stores the support (frequency) of each of these items, an estimation of P(e|Py(s)). The most supported item or items in the CT are returned by CPT.

C. Compact Prediction Tree + (CPT+)

Compact Prediction Tree + (CPT+) [7], an improved version of the CPT model, is a model used for predicting sequences.

In particular, the model is used for performing sequence predictions. In the context of predicting next page access, sequence prediction consists of predicting the next page of a sequence based on a set of training sequences. In this case, it can be used to predict the next webpage that a user will visit based on webpages visited previously by one or more users. CPT+ uses three strategies: Frequent Subsequence Compression (FSC), Simple Branches Compression (SBC), and Prediction with improved Noise Reduction (PNR), to improve the CPT.

FSC (Strategy 1) identifies these frequent subsequences and replaces each of them with a single item (see Fig. 3).

SBC (Strategy 2) is an intuitive compression strategy that reduces the size of the prediction tree. This strategy consists of replacing each simple branch with a single node representing the whole branch (see Fig. 4).

PNR (Strategy 3) is a recursive procedure. To perform a prediction, PNR considers a minimum number of subsequences obtained from the suffix of S of size y. PNR first removes noise from each subsequence. Next, the CT is updated using these subsequences. When the number of updates reaches the threshold a prediction is performed. This strategy is a generalization of the noise reduction strategy used by CPT.

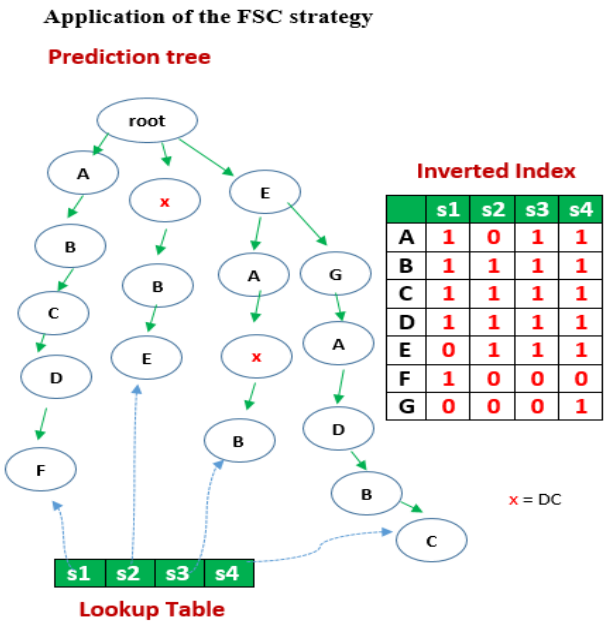

Fig. 3. Application of FSC strategy

Figure 4 illustrates the resulting prediction tree after applying FSC to the tree shown in Fig. 3. The frequent subsequence $\langle D, C \rangle$ has been replaced by a new symbol x, thus reducing the number of nodes in the prediction tree. The SBC strategy replaces the simple branches A, B, C, D, and F by the single node ABCDF; the simple branches B and E by the single node BE, and the simple branches G, A, D, B, and C by the single node GADBC. To identify and replace simple branches, the prediction tree is traversed from the leaves using the inverted index.

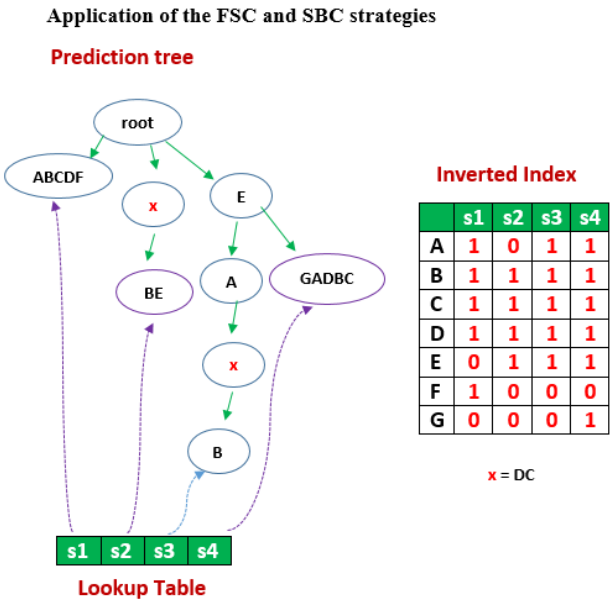

Fig. 4. Application of FSC and SBC strategies

D. The integrated model of PageRank and CPT+

This proposal has been introduced in the article [9]. The integrated model of PageRank and CPT+ is described below. Firstly, a graph database is created from a sequence database. As shown in Figure 5, each node in the graph database is a number representing a page n the sequence database.

Secondly, the PageRank algorithm is used to calculate the PageRank value for every node in the graph database. Following this, for each sequence containing nodes, the average value of the nodes' PageRank values is calculated. In the next step, sequences are given the PageRank values that are calculated in the previous step.

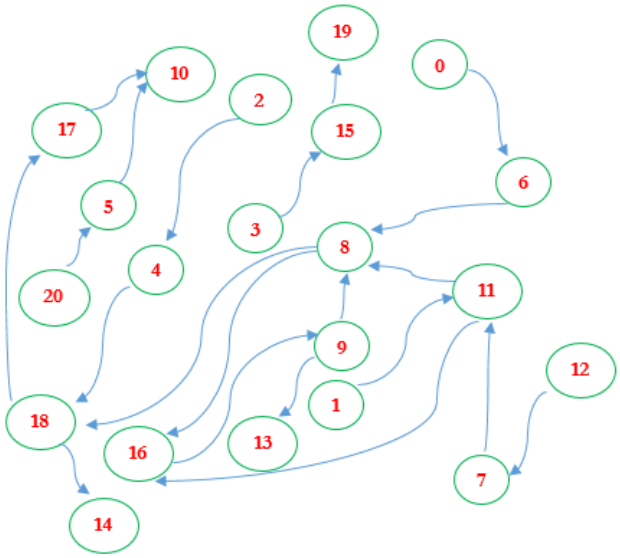

Fig. 5. Illustration of a graph database

Every node has a page rank and then the average of every sequence in the original sequence database is calculated. Finally, sequences are sorted in descending order depending on the PageRank average values.

E. The integrated model of PageRank sequence processing and CPT+

This work has been done in the paper [10]. The pseudo code for this approach is presented as below.

Input:

+ arr_sequence: an array containing sequences in the sequence database

+ arr_query: an array containing items to prepredict the next items of sequence *query*.

Output:

A novel sequence database in which redundant sequences are removed.

Pseudo Code:

```
1. // Find sequences which contain the sequence query
2. Allocate an array seq that has n items
3. k := 0 //k: number of items in the sequence query
4. str_contains_query = " "
5. // str_contain_query is sequence that contains the sequence query
6. For i = 0 to (k – 1) do
7. If (arr_sequence[i] contains at least one item belonging to query)
Then
8. Begin
9. If (query ⊆ arr_contains_query[i] and it is not in the last position
of ⊈
10. arr_contain_query[i] Or (query ⊆ arr_contains_query[i] and it
is in the
11. last position of arr_contains_query[i] And Card{query ⊆
12. arr_contains_query[i]} > 1)) Then
13. Begin
14. SD_OK + = arr_contains_query[i] // selected valid
sequences.
15. End
16. End
```

For example, consider a sequence $s = \langle A, D \rangle$ and a sequence database:

$s_1 : \langle E, C, F, A, D \rangle$

$s_2 : \langle D, C, G \rangle$

$s_3 : \langle A, B, C, D, F \rangle$

$s_4 : \langle D, C, B, E \rangle$

$s_5 : \langle E, A, D, C, B \rangle$

$s6 : \langle E, G, A, D, B, C \rangle$

$s_7 : \langle C, B, E, A, D \rangle$

$s_8 : \langle D, C, B, E \rangle$

$s_9 : \langle A, C, B, G, F \rangle$

$s_{10} : \langle E, B, G, A, D \rangle$

We remove redundant sequences from the sequence database to get a new sequence database of small size.

$s_1 : \langle A, B, C, D, F \rangle$

$s_3 : \langle E, A, D, C, B \rangle$

$s_4 : \langle E, G, A, D, B, C \rangle$

From the three sequences above, a CPT [11] can be created as shown in Fig. 6.

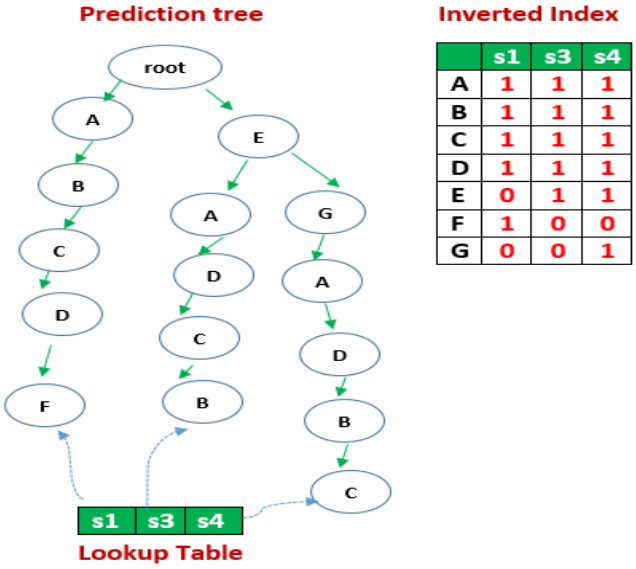

Fig. 6. A diagram of CPT

III. PROPOSED MODEL: INTEGRATION OF PAGERANK ALGORITHM AND SEQUENCE PROCESSING FOR WEBPAGE ACCESS PREDICTION

Let s_{query} be the sequence that we need to find the next page accessed. We propose four main steps to predict a possible webpage called p_{next}.

We propose three major steps to improve the time performance for webpage access prediction as below.

Step 1: By using K-fold validation, We run randomized K-fold validation 10 times to get 10 datasets (each dataset contains two parts: 90% of data for training, 10% remaning of data for testing) the PageRank values for each sequence Then, we use PageRank algorithm to get rid of redundant sequences from them. **Step 2**: We improve the accuracy by using PageRank and CPT+.

$$SDB1 = g1\ (SDB) \quad (1)$$

where SDB1 is the sequence database created by the function *g1* [9].

Step 3: We improve the time execution by using sequence processing and CPT+.

$$SDB2 = f_2\ (s_{query}, SDB1) \quad (2)$$

where SDB2 is the sequence database created by the function f_2 [10].

Step 4: We perform webpage access prediction by using CPT+.

$$p_{next} = G(s_{query}, SDB2) \quad (3)$$

where p_{next} is the possible predicted webpage. It is created by using the function G [7] where the input values are the sequence query s_{query} and the sequence database *SDB2*.

Our novel proposed model is better than the method [9] due to the fact that in Step 2, *SDB1* has a large size if we use [7] to perform the prediction.

Therefore, to reduce the size of *SDB1*, we use Step 3 to obtain the sequence database *SDB2*. Thus, *SDB2* << *SDB1* (in terms of size or number of sequences in the sequence database), and the execution time for prediction will be quicker.

IV. EXPERIMENTAL EVALUTION

To evaluate the accuracy of our proposed approach, a set of experiments was performed. Our test environment consists of a computer with an Intel i7 third-generation processor with 31.5 GB of available RAM running a 64-bit version of Ubuntu 16.04.5 LTS (Xenial Xerus) using a Java 8.1 environment with Eclipse Neon.3.

A. Dataset

We used the FIFA dataset collected from the website http://www.philippe-fournier-viger.com/spmf/datasets/FIFA.txt. It contains 20,450 sequences and 2,990 distinct links. This dataset includes click stream data recorded on the 1998 FIFA World Cup website, holds 1,352,804,007 webpage requests, and is a set of individual requests containing metadata. The author of paper [11] converted requests into sequences to obtain the final dataset.

B. Evaluation Framework

To evaluate out approach, we used the SPMF framework [12] to calculate the accuracy of the sequence databases created.

$$Accuracy = \frac{|successes|}{|sequences|} \quad (4)$$

where *accuracy* (4) is a measure that evaluates the accuracy of a given predictor and is the number of successful predictions divided by the total number of test sequences [11].

C. Experimental Results

Using the cross-checking algorithm K-fold validation (with K = 10), the FIFA dataset was divided randomly and the algorithm was run 10 times'. With each random time, data were separated into two parts, with 90% of the data used for training and the remaining 10% for testing (predicting). In all 10 considered cases, when the size of the training datasets was reduced by 30%, the values of prediction accuracy were higher than those of the original dataset (90% of the FIFA dataset', the first part of the dataset in K-fold validation). Therefore, we chose 70% as the size of the training datasets for next page access prediction. As mentioned above, the remaining 10% of data were used for testing (predicting), and 1000 sequences belonging to the testing datasets were chosen randomly for next page access prediction. Due to the limitation on the number of pages in this article, we only display one of the experimental results, which is illustrated in detail in the Appendix; that is the result obtained from 100 random sequences (collected from testing data) performing on training data in the first randomization (among 10 randomizations in the algorithm K-fold validation) for sequence prediction.

As shown in the Appendix, the first sequence is S1 = <266, 28, 211>, predicted on the training dataset in the first randomization. The best page predicted from S1 was page 80. The value of 100 is the accuracy of the shortened dataset obtained by the algorithm introduced in [10]. It indicates that, in the first randomization, the accuracy of prediction for the created dataset is higher than that for the shortened training dataset (70% of the size of the training data created from the proposed algorithm introduced in [9]). The values 7896 and 876 (in milliseconds) are the execution time values when predicting the sequence S1 in two ways: the first by performing the prediction using the shortened training dataset and the second by performing the prediction using the shortened training dataset but adding the sequence analysis solution (proposed in [10]). We investigated 1000 test cases in this way for 10 training datasets (mentioned above).

Table II – Statistical Results of Execution Time Comparison

Training Datasets	Mean (milliseconds)	Standard Deviation	Difference (milliseconds)	t-statistic (T-test statistic)	p-value (probability value)
70% of the size of the training FIFA dataset, using the PageRank algorithm [9]	6666.56	4136.234	3323.78	25.411	< 0.001
70% of the size of the training FIFA dataset, using the PageRank algorithm [9] and sequence analysis [10]	3342.78	4201.432			

The experimental results illustrated in Table II show that the combination of using the PageRank algorithm with sequence analysis to reduce the size of datasets resulted in significant shortening of the execution time and thus improved the system's efficiency. As shown in Table II, the computation of prediction with the reduced dataset of sequences is nearly twice as fast as that of the original dataset in the case of the FIFA dataset. The paired sample t-test found the difference in prediction execution time to be statistically significant.

Table III – Statistical Results of Accuracy Comparison

Training Dataset	Mean (%)	Standard Deviation	Difference (%)	t-statistic (T-test statistic)	p-value (probability value)
70% of the size of the FIFA dataset, using the PageRank algorithm [9]	99.95440	.006862	-0.00196	-.267	.789 (> 0.001)
70% of the size of the FIFA dataset, using the PageRank algorithm [9] and sequence analysis [10]	99.95636	.232628			

The experimental results illustrated in Table III show that the combination of using the PageRank algorithm and sequence analysis to reduce the size of datasets resulted in a significant reduction of accuracy. As shown in Table III, the accuracy of prediction achieved with the reduced dataset of sequences is nearly the same as that of the original dataset in the case of the FIFA dataset. In this case, the paired sample t-test found no significant difference in accuracy to be statistically significant. Thus our solution is better than those of the works [9, 10] in terms of the execution time required for prediction.

V. CONCLUSION

The experimental results show that the combination of two solutions (calculation by PageRank and analysis of sequences) is worthwhile As described, the datasets are shortened twice, the first time through calculating PageRank values to remove redundant sequences and the second time through eliminating useless sequences in datasets by shortening Compact Prediction Trees by obtaining a grid of redundant branches in terms of width. However, in a few rare cases, the execution time of the new approach could be a little slower than those of previous approaches. For instance, predicting sequences S5, S3, S7, and 25 other similar sequences with our new approach has an execution time that is slower than that of the previous approach (using only PageRank calculation to shorten sequences in sequence databases). However, the accuracy of our proposal is often higher than that of previous approaches. Thus, in general, our proposed approach is useful and meaningful for next page access prediction and better than previous approaches in terms of prediction execution time.

VI. APPENDIX

Table IV– Accuracy and Execution Time when Computing Sample Webpage Access Prediction for a Shortened FIFA Dataset

No.	Sequence of pages from test training (10% size of original FIFA dataset, first randomization) (input)	Best predicted pages (output)	Accuracy of sequence databases (training datasets)	Prediction time (milliseconds)	
1	S1 <266, 28, 211>	80	100	7896	876
2	S2 <162, 122,384>	341	99.642	7222	578
3	S3 <1, 47, 45>	30	99.983	7122	9863
4	S4 <96, 1506, 98>	118	100	6435	279
5	S5 <47, 32, 155>	30	99.973	6402	5903
6	S6 <117, 96, 243>	494	100	6980	685
7	S7 <30, 44, 10>	98	99.966	7245	12031
8	S8 <493, 171, 233>	28	100	5742	897
9	S9 <210, 285, 303>	475	98.788	5684	329
10	S10 <80, 312, 3>	28	100	5319	289
11	S11 <46, 1, 155>	30	99.954	6194	7226
12	S12 <124, 29, 51>	161	100	7006	1720
13	S13 <161, 193, 225>	192	99.942	7666	2109
14	S14 <173, 245, 11>	94	100	5201	352
15	S15 <63, 109, 28>	80	100	5590	1363

16	S16 <104, 138, 249>	297	99.832	6116	663
17	S17 <196, 37, 156>	86	99.776	6152	538
18	S18 <49, 47, 135>	30	99.941	5713	1870
19	S19 <1420, 423, 1478>	425	100	5808	245
20	S20 <30, 63, 109>	80	100	5362	980
21	S21 <68, 70, 18>	98	100	6890	9764
22	S22 <97, 155, 135>	98	100	6502	9201
23	S23 <163, 128, 11>	94	100	6554	414
24	S24 <425, 373, 404>	648	100	9570	289
25	S25 <68, 101, 323>	136	100	7301	950
26	S26 <99, 151, 140>	161	100	6409	1648
27	S27 <98, 164, 118>	376	100	6780	851
28	S28 <155, 13, 131>	98	99.984	5870	12175
29	S29 <42, 120, 119>	161	100	6108	826
30	S30 <21, 36, 37>	99.967	99.967	5944	12330
31	S31 <996, 1027, 969>	746	100	6938	329
32	S32 <135, 90, 131>	98	99.969	6319	14207
33	S33 <366, 70, 532>	98	100	7129	928
34	S34 <100, 102, 297>	313	100	6130	1799
35	S35 <900, 1040, 418>	192	100	6135	560
36	S36 <162, 308,103>	341	99.819	6802	613
37	S37 <109, 182, 28>	80	100	6316	1988
38	S38 <205, 239, 173>	246	100	6542	437
39	S39 <470, 471, 412>	30	100	7418	714
40	S40 <248, 233, 211>	109	100	6169	1118
41	S41 <30, 676, 3, 182>	80	100	6000	1179
42	S42 <182, 51, 72, 28>	80	100	6726	2182
43	S43 <94, 165, 111, 43>	292	100	6282	253
44	S44 <8, 50, 57, 10>	98	99.968	7474	12582
45	S45 <97, 98, 118, 117>	72	100	6393	10091
46	S46 <57, 297, 274, 100>	102	99.894	6271	948
47	S47 <86, 297, 100, 274>	102	100	5961	1272
48	S48 <72, 27, 153, 42>	161	99.853	6157	675
49	S49 <99, 121, 253, 168>	161	100	7193	1689
50	S50 <137, 116, 165, 180>	322	100	7126	472
51	S51 <182, 51, 72, 28>	80	100	6605	1200
52	S52 <1, 45, 46, 32>	30	99.982	6253	8162
53	S53 <131, 8, 2, 50>	98	100	7150	12690
54	S54 <163, 173, 25, 40>	94	100	6595	393
55	S55 <295, 279, 137, 116>	292	100	5899	476
56	S56 <57, 18, 98, 21>	118	99.981	7758	12650
57	S57 <24, 98, 37, 59>	118	99.981	8262	11578
58	S58 <315, 319, 18, 399>	98	100	7958	505
59	S59 <232, 4, 266, 510>	109	100	7992	1635
60	S60 <98, 7, 228, 356>	109	100	8139	797
61	S61 <28, 220, 209, 80>	489	100	7541	464

62	S62 <51, 325, 182, 349>	370	100	7366	900
63	S63 <70, 181, 68, 97>	118	100	9721	11637
64	S64 <97, 90, 31, 10>	98	99.981	8360	11001
65	S65 <63, 1001, 1081, 182>	80	100	6493	490
66	S66 <82, 36, 118, 59>	98	99.98	7333	10049
67	S67 <98, 360, 215, 103>	384	100	6573	276
68	S68 <46, 33, 155, 98>	118	99.967	6386	4296
69	S69 <15, 32, 155, 147>	30	99.973	6339	5244
70	S70 <26, 243, 81, 306>	109	100	12491	1168
71	S71 <27, 287, 29, 162, 122>	384	100	7893	337
72	S72 <70, 97, 98, 118, 117>	72	100	7291	10556
73	S73 <135, 2, 50, 131, 13>	98	99.966	6512	12129
74	S74 <264, 137, 116, 165, 180>	322	100	6621	488
75	S75 <33, 155, 147, 135, 131>	98	99.971	7392	4991
76	S76 <202, 165, 328, 292, 116>	322	100	6742	660
77	S77 <147, 131, 44, 90, 8>	98	99.968	5925	13031
78	S78 <13, 155, 2, 147, 135>	98	99.954	7056	11803
79	S79 <90, 155, 131, 13, 21>	98	99.966	7223	11581
80	S80 <4, 266, 510, 199, 362>	109	100	5903	949
81	S81 <205, 77, 163, 239, 173>	11	100	6056	346
82	S82 <100, 221, 86, 274, 102>	297	100	6875	1155
83	S83 <117, 59, 70, 181, 118>	72	100	6642	9583
84	S84 <155, 2, 18, 147, 135>	98	99.983	6669	10792
85	S85 <73, 119, 99, 121, 40>	161	100	7283	1794
86	S86 <81, 356, 176, 171, 170>	28	100	6586	960
87	S87 <90, 155, 8, 30, 135>	98	99.965	6405	12008
88	S88 <98, 202, 328, 181, 323>	376	100	7755	1470
89	S89 <124, 27, 51, 29, 121>	161	100	7859	1820
90	S90 <131, 8, 13, 44, 50>	98	99.968	6729	12910
91	S91 <27, 140, 120, 509, 73>	133	100	5619	389
92	S92 <68, 216, 36, 181, 70>	136	100	7744	944
93	S93 <98, 100, 138, 102, 507>	242	100	8722	410
94	S94 <140, 133, 99, 148, 936>	16	100	5489	258
95	S95 <437, 674, 472, 411, 289>	679	100	5263	294
96	S96 <2, 57, 14, 37, 30>	98	99.964	7303	11386
97	S97 <135, 13, 18, 44, 17>	98	99.976	6896	6272
98	S98 <12, 177, 200, 81, 234>	109	100	5394	848
99	S99 <131, 17, 2, 18, 13>	98	99.954	6290	6369
100	S100 <30, 366, 97, 611, 470>	118	100	6492	372

ACKNOWLEDGEMENT

This research was funded by Vietnam National University - Ho Chi Minh City (VNU-HCM) under grant number C2019-34-06.

REFERENCES

1. M. Deshpande and G. Karypis, "Selective Markov models for predicting Web page accesses," *ACM Trans. Internet Technol. (TOIT)*, vol. 4, no. 2, pp. 163–184, 2004.
2. F. Khalil, J. Li, and H. Wang, "An integrated model for next page access prediction," *IJ Knowl. Web Intell.*, vol. 1, no. 1/2, pp. 48–80, 2009.
3. P. Fournier-Viger, R. Nkambou, and V. S.-M. Tseng, "RuleGrowth: mining sequential rules common to several sequences by pattern-growth," in *Proc. 2011 ACM Symp. Applied Computing*, 2011, pp. 956–961: ACM.
4. P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo, "CMRules: Mining sequential rules common to several sequences," *Knowl.-Based Syst.*, vol. 25, no. 1, pp. 63–76, 2012.

5. P. Fournier-Viger, T. Gueniche, S. Zida, and V. S. Tseng, "ERMiner: sequential rule mining using equivalence classes," in *Int. Symp. Intelligent Data Analysis*, Springer, 2014, pp. 108–119.
6. M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, no. 1–2, pp. 31–60, 2001.
7. T. Gueniche, P. Fournier-Viger, R. Raman, and V. S. Tseng, "CPT+: Decreasing the time/space complexity of the Compact Prediction Tree," in *Pacific-Asia Conf. Knowledge Discovery and Data Mining*, Springer, 2015, pp. 625–636.
8. N. T. Da, T. Hanh, and P. H. Duy, "A survey of webpage access prediction," in *2018 Int. Conf. Adv. Technol. Commun. (ATC)*, IEEE, 2018, pp. 315–320.
9. N. T. Da, T. Hanh, and P. H. Duy, "Improving webpage access predictions based on sequence prediction and PageRank algorithm," *Interdiscip. J. Inf., Knowl. Manag.*, vol. 14, 2019.
10. N. Thon Da and T. Hanh, "A novel approach based on sequence prediction for webpage access," 2018, CPT; CPT+; Sequence Prediction; Web Mining. vol. 7, no. 4, p. 4, Sept. 2018.
11. T. Gueniche, P. Fournier-Viger, and V. S. Tseng, "Compact prediction tree: A lossless model for accurate sequence prediction," in *Int. Conf. Advanced Data Mining and Applications*, Springer, 2013, pp. 177–188.
12. P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "SPMF: a Java open-source pattern mining library," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3389–3393, Nov. 2014.

AUTHOR PROFILES

Nguyen Thon Da, received his Master's degree in Computer Science from the University of Technology, VNU-HCM in 2013. In November 2016, he was accepted as a PhD Student in Information Systems at the Posts and Telecommunications Institute of Technology, Vietnam. He is now working as IT employee and an assistant teacher at the Faculty of Information Systems, University of Economics and Law, VNU-HCM. His research interests include data mining and pattern and sequence prediction.

Tan Hanh, received his PhD in Informatics from Grenoble INP, France in 2009. Currently, he is Vice President of the Posts and Telecommunications Institute of Technology. His research interests are distributed systems, machine learning, information retrieval, and data mining.

Pham Hoang Duy, strongly connects with IT-related research and education. From 2005 to 2009, he was working on his PhD research project tackling the problem of knowledge representation and defeasible reasoning in multi-agent systems at the University of Queensland in Australia. His research and teaching interests are data-mining techniques and their applications, especially in the domain of computer systems' security.