Interdisciplinary Journal of Information, Knowledge, and Management

*An Official Publication of the Informing Science Institute*
*InformingScience.org*

*IJIKM.org*

**Volume 14, 2019**

# IMPROVING WEBPAGE ACCESS PREDICTIONS BASED ON SEQUENCE PREDICTION AND PAGERANK ALGORITHM

| | | |
|---|---|---|
| Nguyen Thon Da * | University of Economics and Law, VNU-HCM, Ho Chi Minh, Vietnam | dant@uel.edu.vn |
| Tan Hanh | Posts and Telecommunications Institute of Technology, Hanoi, Vietnam | tanhanh@ptit.edu.vn |
| Pham Hoang Duy | Posts and Telecommunications Institute of Technology, Hanoi, Vietnam | duyph@uel.edu.vn |

* Corresponding author

## ABSTRACT

| | |
|---|---|
| Aim/Purpose | In this article, we provide a better solution to Webpage access prediction. In particularly, our core proposed approach is to increase accuracy and efficiency by reducing the sequence space with integration of PageRank into CPT+. |
| Background | The problem of predicting the next page on a web site has become significant because of the non-stop growth of Internet in terms of the volume of contents and the mass of users. The webpage prediction is complex because we should consider multiple kinds of information such as the webpage name, the contents of the webpage, the user profile, the time between webpage visits, differences among users, and the time spent on a page or on each part of the page. Therefore, webpage access prediction draws substantial effort of the web mining research community in order to obtain valuable information and improve user experience as well. |
| Methodology | CPT+ is a complex prediction algorithm that dramatically offers more accurate predictions than other state-of-the-art models. The integration of the importance of every particular page on a website (i.e., the PageRank) regarding to its associations with other pages into CPT+ model can improve the performance of the existing model. |
| Contribution | In this paper, we propose an approach to reduce prediction space while improving accuracy through combining CPT+ and PageRank algorithms. Experimental results on several real datasets indicate the space reduced by up to between 15% |

Accepting Editor Duane Eleuterio Ramirez | Received: November 6, 2018 | Revised: January 14, 2019 | Accepted: September 19, 2018.
Cite as: Da, N. T., Hanh, T., & Duy, P. H. (2019). Improving webpage access predictions based on sequence prediction and pagerank algorithm. *Interdisciplinary Journal of Information, Knowledge, and Management*, *14,* 27-44. https://doi.org/10.28945/4176

| | |
|---|---|
| | and 30%. As a result, the run-time is quicker. Furthermore, the prediction accuracy is improved. It is convenient that researchers go on using CPT+ to predict Webpage access. |
| Findings | Our experimental results indicate that PageRank algorithm is a good solution to improve CPT+ prediction. An amount of though approximately 15 % to 30% of redundant data is removed from datasets while improving the accuracy. |
| Recommendations for Practitioners | The result of the article could be used in developing relevant applications such as Webpage and product recommendation systems. |
| Recommendations for Researchers | The paper provides a prediction model that integrates CPT+ and PageRank algorithms to tackle the problem of complexity and accuracy. The model has been experimented against several real datasets in order to show its performance. |
| Impact on Society | Given an improving model to predict Webpage access using in several fields such as e-learning, product recommendation, link prediction, and user behavior prediction, the society can enjoy a better experience and more efficient environment while surfing the Web. |
| Future Research | We intend to further improve the accuracy of webpage access prediction by using the combination of CPT+ and other algorithms. |
| Keywords | Webpage access prediction, sequence prediction, compact prediction tree, PageRank algorithm |

## INTRODUCTION

Currently, the problem of modelling and predicting a user's browsing behaviour on a web site has attracted a lot of research interest as it can be applied in improving web cache performance, Webpages recommendation, search engines enhancements, understanding and influencing buying patterns, and personalizing the browsing experience (Deshpande & Karypis, 2004). In e-commerce, the next pages prediction is very necessary and critical. The prediction supports companies to deal with issues relating to customers such as their trends in buying and their interests in particular products. In this paper, we propose an effective solution to enhance the performance of the Webpage access prediction. In particularly, our proposed approach is to increase accuracy and efficiency by reducing the sequence space with integration of PageRank into CPT+ model.

In the task of modelling and predicting webpage access, it is desirable to find strategies to analyse and retrieve the interestingness of pages in order to yield meaningful predictions of possible webpage access. Thus, this problem is the main motivation for this research study. In fact, the webpage access prediction on a web site is a major and challenging problem. This problem has become significant because of the non-stop growth of Internet in terms of huge volumes of contents and increasingly quantity of users. Furthermore, while calculating the next page accesses, we may want to consider several kinds of information, such as the webpage name, the contents of the webpage, the profile of the user, the time between webpage visits, differences among users, and the time spent on a page or on each part of the page. Thus, many different kinds of data exist. Therefore, webpage access prediction is a significant topic of research. Among the proposed methods, sequence prediction models using CPT+ (Gueniche, Fournier-Viger, Raman, & Tseng, 2015) are the most popular method due to its performance. In this article, we present an approach that reduces the prediction space while preserving accuracy by integrating an important factor to web pages titled PageRank.

The remainder of the paper is organized as follows. In Section 1, we present a definition of the problem of predicting webpage access, CPT+ and PageRank algorithm. Subsequently Section 2 pro-

poses a novel approach to webpage access prediction. In the following two sections, we present an experimental evaluation and an experimental study. Finally, we conclude with our solution.

# BACKGROUND

## DEFINITION OF SEQUENCE DATABASE

A sequence database used in webpage prediction, *SDP*, is a set of sequences. Let there be a set $P = \{p_1, p_2, \ldots, p_n\}$ of webpages. A sequence database contains sequences $S = \{s_1, s_2 \ldots s_n\}$, where $s_i$ is an ordered list of webpages. Table 1 shows a sequence database containing five sequences. The first sequence, named s1, contains 6 pages. This sequence means that a user has visited pages $p_1, p_2, p_4, p_6, p_3$, and $p_5$, in that order.

**Table 1. An example of a sequence database**

| ID | SEQUENCE |
|---|---|
| s1 | p1, p2, p4, p6, p3, p5 |
| s2 | p4, p3, p4, p6, p2 |
| s3 | p1, p2, p4, p9, p3, p7, p3, p10 |
| s4 | p6, p1, p4, p8, p3, p5 p3, p3 |

The problem of webpage prediction consists of predicting the next web pages in a sequence given the information contained in a set of training sequences.

For example, consider that a user visits the web pages $p_1$, $p_2$, $p_3$, and $p_4$, in that order. The sequence prediction can then be used to preload a webpage or recommend that webpage to the user. Figure 1 illustrates the problem of webpage prediction.

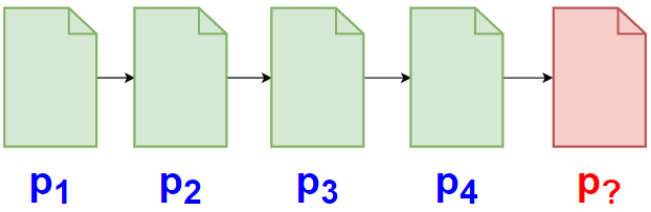


**Figure 1. An illustration of webpage prediction**

In addition, to create sequence databases, especially from weblog data, see (Da, Hanh, & Duy, 2018).

## THE PROBLEM OF WEBPAGE PREDICTION

Webpage prediction is the problem of calculating the next web pages that a user may visit based on the web pages previously visited by that user or by a group of users (Narvekar & Banu, 2015). In general, webpage prediction involves two steps. First, a sequence prediction model is trained to predict webpage access using training sequences of pages visited by one or more users. The second step involves applying the trained model to some new sequence to predict the next webpage in that sequence. Figure 2 illustrates this general process of webpage prediction.

The input is a set of training sequences from one or more users. The output of a prediction is a set of one or more web pages that are deemed to be the most likely to be visited by the user next. Some prediction models may be designed to output a single webpage, while others may be designed to output several pages.

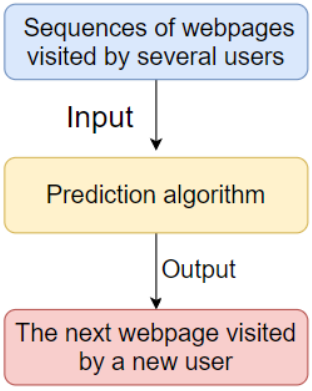


**Figure 2. General workflow of Webpage prediction**

The process of webpage prediction is similar to the problem of classification in data mining. However, key differences relative to the traditional problem of classification are that the time dimension is considered in webpage prediction, and other information, such as the importance of every page and the user's profile may also be considered.

To predict webpage access, many different approaches have been proposed, such as machine learning, association rules (Geetharamani, Revathy, & Jacob, 2015; Yang, Li, & Wang, 2004), sequential patterns (Fournier-Viger, Gomariz, Campos, & Thomas, 2014; Fournier-Viger, Lin, Kiran, Koh, & Thomas, 2017; Mabroukeh & Ezeife, 2010; Mobasher, Dai, Luo, & Nakagawa, 2002), and sequential rules (Fournier-Viger, Wu, Tseng, Cao, & Nkambou, 2015). However, one of the significant limitations of the aforementioned approaches is that they are outmoded. Moreover, they also have several limitations, as follows:

Machine learning-based approaches build lossy models that may ignore relevant information from the training sequences when making predictions (Gueniche, Fournier-Viger, & Tseng, 2013).

According to (Gueniche et al., 2015), the abovementioned models suffer from some major drawbacks. (1) Most of them assume the Markovian hypothesis that each event solely depends on the previous events. The prediction accuracy using these models can thus dramatically decrease. (2) Only part of the information contained in the training sequences is used. Therefore, these models ignore some of the information contained in the training sequences when making predictions, and this feature can severely reduce the accuracy of these models. For instance, Markov models typically consider only the last k items in the training sequences in performing a prediction, where k is the order of the model. A solution to this problem is to increase the order of the Markov models. Nevertheless, increasing the order of Markov models often leads a very high state of complexity, thus making them impractical for many real-life applications.

Currently, regarding to sequence prediction, CPT+ (Gueniche et al., 2015) is up to 98 times more compact and 4.5 times faster than CPT (Gueniche et al., 2013), and CPT+ has the best overall accuracy when compared to six state-of-the-art models from the literature (Gueniche et al., 2015).

## *CPT+*

(Gueniche et al., 2015) proposed solutions to address the drawbacks of CPT (Gueniche et al., 2013); specifically, this earlier study suggests three strategies named frequent subsequence compression (FSC), simple branches compression (SBC) and prediction with improved noise reduction (PNR).

### Frequent subsequence compression (FSC)

According to (Gueniche et al., 2015), frequent subsequence compression (FSC) includes three steps: (1) the identification of frequent subsequences in the training sequences, (2) the generation of a nov-

el item in the alphabet Z of items for every frequent subsequence, and (3) the replacement of every frequent subsequence by the corresponding novel item when inserting training sequences into the prediction tree. Further, a novel data structure is introduced to store the frequent sub-sequences, and it offers a fast means of translating each subsequence into its respective item and vice versa. To reduce the number of nodes in the prediction tree, each frequent subsequence is replaced by a novel symbol (Gueniche et al., 2015).

### Simple branches compression (SBC)

To reduce the size of the prediction tree, an intuitive compression strategy, namely *simple branches compression (SBC)*, has been proposed. In SBC, each single node representing the whole branch takes the place of each simple branch – a branch leads to a single leaf. Traversing the prediction tree from the leaves using the inverted index supports the identification and replacement of the simple branches.

### Prediction with improved noise reduction

The prediction with improved noise reduction (PNR) strategy relies on the observation that noise in training sequences consists of items having a low frequency, where an item's frequency is defined as the number of training sequences containing the item (Gueniche et al., 2015). In applying this observation, PNR removes only items having a low frequency during the prediction process.

According to (Gueniche et al., 2015), the main properties of *prediction with improved noise reduction* (PNR) are that it requires a minimum number of updates on the CT to perform a prediction and noise is defined based on the frequency of items and proportionally to the sequence length.

## *PageRank Algorithm*

PageRank calculation relies on the idea of counting backlinks (citations) to a certain page. It was proposed by Sergey Brin and Lawrence Page, and it provides a way of measuring the importance of website pages.

PageRank algorithm was used to build the very successful search engine Google a huge success (Wu et al., 2008). According to (Page, Brin, Motwani, & Winograd, 1999), PageRank can be calculated using a simple iterative algorithm, and it corresponds to the principal eigenvector of the normalized link matrix of the web. The developers of PageRank give a formula for calculating PageRank towards page A as follows:

$$PR(page\ A) = (1-df) + df(PR(T_1)/C(T_1) + PR(T_2)/C(T_2) + ... + PR(T_n)/C(T_n)) \quad (1)$$

Where:

*PR(page A):* PageRank of page A

$T_i$*:* a page that links to page A

$PR(T_i)$*:* PageRank of page $T_i$

$C(T_i)$*:* the number of pages that page $T_i$ links to

*df:* a damping factor (d = 0.85 is used by many researchers).

# Problem Definition

Currently, to predict Webpage access, a state-of-the-art solution is utilizing sequence prediction. In particularly, CPT and an improved version called CPT+ have been proposed. However, reducing Webpage access prediction time complexity and how to increase speed but to avoid decreasing accuracy is very necessary. Although improved, this is still time consuming. To resolve this problem, we introduce a solution to increase computational efficiency while to remain the accuracy for Webpage access prediction. Our proposal is that, before utilize CPT+ to predict on a sequence database, we

should shorten the size of the original sequence database but avoid the prediction accuracy. In fact, the decline the prediction space is very meaningful because with regard to very large datasets, the prediction would encounter the challenge in terms of time.

## PROPOSED ALGORITHMIC APPROACH

In this section, we propose a method of reducing the space of webpage access prediction based on CPT+ (Gueniche et al., 2015) while preserving or increasing the accuracy of sequence prediction.

Suppose that a sequence database *SD* contains *N* sequences.

*Step 1: Convert the sequence database into a graph database.*

Every pair of contiguous pages $\{p_i, p_j\}$ in a sequence can be considered as a relationship between two vertices or two nodes. Where an arrow $(p_i, p_j)$ goes from $p_i$ to $p_j$, $p_j$ is called the head of the arrow, and $p_i$ is called the tail of the arrow. Moreover, $p_j$ is said to be a direct successor of $p_i$, and $p_i$ is said to be a direct predecessor of $p_j$.

For example, suppose that there are two access sequences $S_1 = \{pA, pD, pZ, pK, pN\}$ and $S_2 = \{pD, pN, pT\}$. Following the discussion above, their sub-graph can be presented as shown in Figure 3.

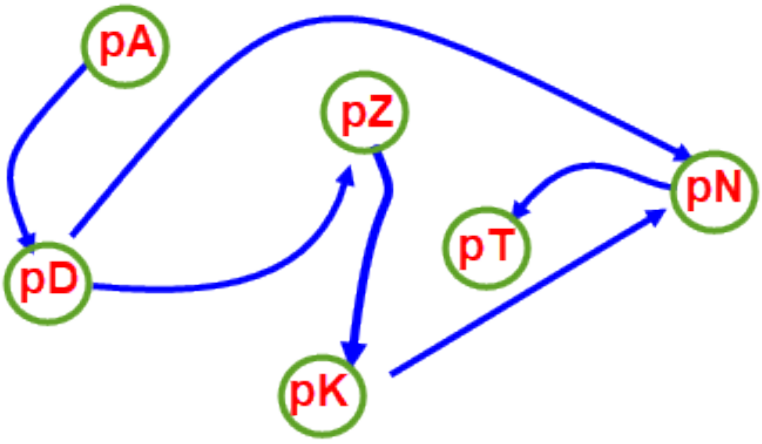


**Figure 3. A sample subgraph is created from a sequence**

Step 2: Determine PageRank of every page

Based on PageRank calculation method, PageRank of every page in the sequence database is calculated.

Step 3: Determine the average PageRank of every sequence

Suppose that the sequence database SD contains N sequences, and $S_j$ is the sequence at position j in the SD.

In the sequence database SD, for each sequence, one page in the sequence has one proper PageRank value. Let M be the number of pages in sequence S, and $p_i$ is the page at position i in the sequence S. The average PageRank of the sequence S can then be calculated as shown below:

$$AVG\_PR\ (S_j) = \frac{\sum_{M}^{i=1} PR(p_i)}{M} \qquad (2)$$

Where: $AVG\_PR(S_j)$ is the average PageRank of all pages in the sequence $S_j$

Step 4: Sort all of the sequences in the sequence database SD by the average PageRank of all of the sequences from high to low without loss of accuracy.

The main purpose of this step is to remove redundant and noisy sequences from the sequence database and to retain useful sequences for prediction.

Let k ∈ (0,100) be the percentage of the size of sequence database. For example, with k = 75 (%) and N = 100000, the new size of the sequence database would be 75000.

In reducing the size of the sequence database, we could choose k randomly. However, to preserve the accuracy of sequence prediction, appropriate values of k should be chosen. In particular, let acc1 be the accuracy of sequence prediction for the original sequence database. Similarly, let acc2 be the accuracy of sequence prediction for the reduced-size sequence database (the new size of the original sequence database). If acc2 ≥ acc1, the chosen value of k is useful. Thus, k (%) of the number of sequences in the original sequence database are retained.

Step 5: Using CPT+ model for sequence prediction

With the reduced-size sequence database obtained from Step 4, the next pages are predicted following CPT+ model.

Figure 4 describes the pseudo code for the above steps.

```
Procedure Build_GraphDatabase
Begin
1. String sfile ← null;
2. Sort(n1);
3.    For k ← 0 to Len(n1) - 1 do
4.    Begin
5.          sfile ← sfile + n1[k] +" ";
6.          For i ← 0  to  Len(arr) -1 do
7.              Begin
8.                  For j←0 to j < Len (arr[i]) -1 do
9.                  If (arr[i][j] = n1[k]) Then
10.                     sfile ←sfile+arr[i][j+1]+ " ";
11.             End
12.           sfile  ←  sfile +"\n";
13.   End
14. WriteFile sfile Adjacency_Matrix
End
```

**Figure 4. Pseudo code for building Graph Database**

The procedure *Average_by_sequences* is used to determine the average value of PageRank by every sequence in sequence database. It is shown in the Figure 5.

Let *arr_avg* be an array containing Average PR values of every sequence.

Let *arr_temp* be an array containing PageRank values of every sequence.

***Pseudo code for calculating Average PR by sequences:***

```
Procedure Average_by_sequences
Begin
1. For i ← 0 to Len(arr_temp) - 1 do
2. arr_avg[i] ← Average_Rows (arr_temp, Len(arr_temp))
End
```

***Pseudo code for Average_Rows:***

```
Function Double Average_Rows(Double arr[ ][ ],int n, int k)
Begin
1. Double S ← 0.0;
2. Double average ← 0.0;
3. For j ← 0 to Len(arr[k]) - 1 do
4.     Begin
5.         S ← S +  arr[k][j];
6.         average ← S / Len(arr[k]);
7.     End
 Return average;
 End
```

**Figure 5. Pseudo code for calculating Average PageRank by sequences and Average Rows**

*Average_Rows* works as follows:
*Line 1, Line 2:* Initialize S (sum of items by rows).
*Line 3:* For loop to visit sequences in sequence database.
*Line 5:* Determine sum of values by sequences.
*Line 6:* Specify average PR's values by every sequence in sequence database.

According to Figure 6, a step-by-step example of proposed algorithmic approach is illustrated.

There are five steps are shown as follows.

- ✓ Input a sequence database
- ✓ Convert links into nodes for a graph database
- ✓ Display nodes and their PageRank
- ✓ Calculate average of PageRank for each sequence
- ✓ Show sequence database (sort decreasingly by average of PageRank for each sequence)

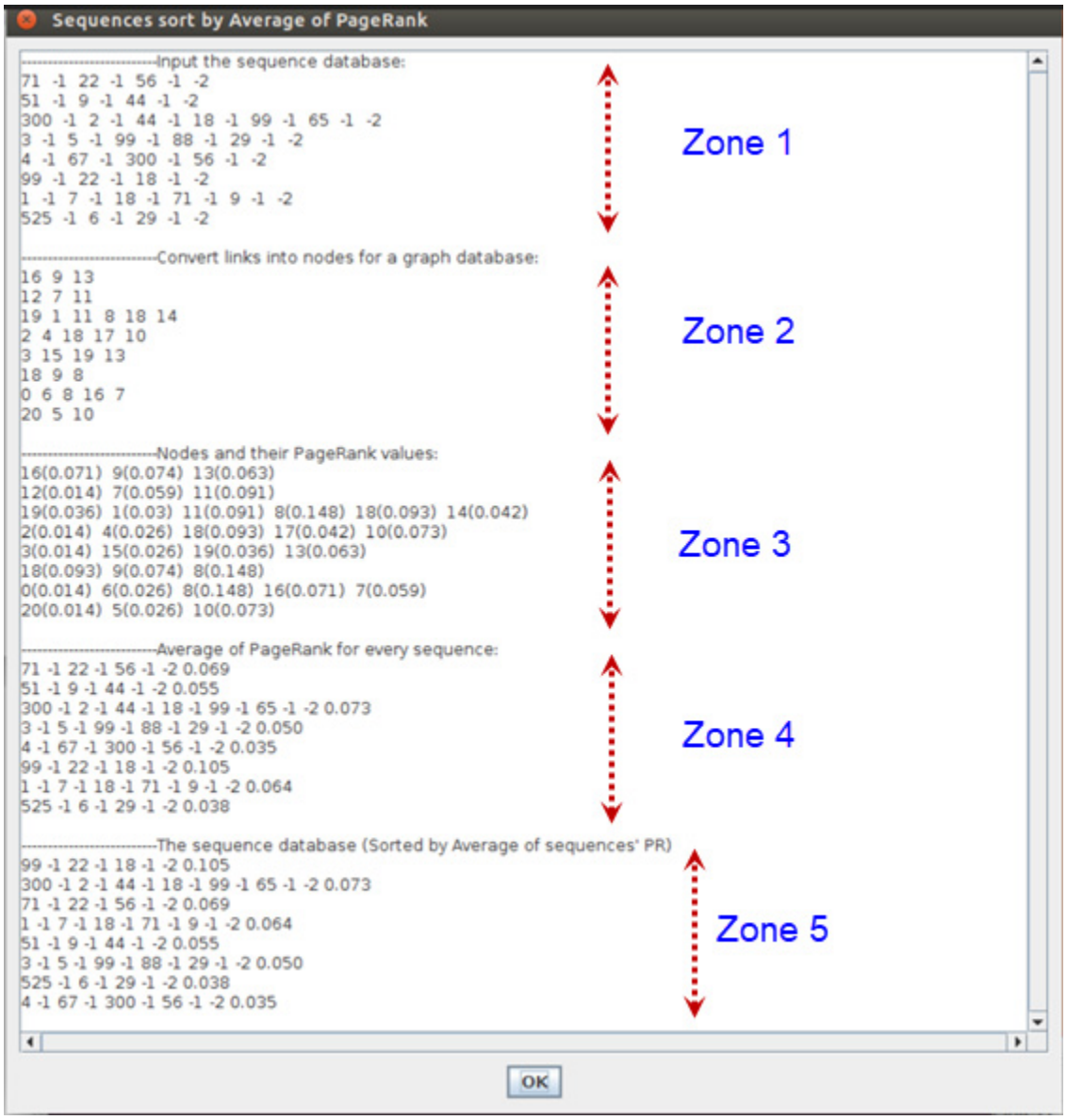


**Figure 6. An example of proposed algorithmic approach**

The zone 1 in the Figure 6 displays 8 sequences in a sequence database. Every sequence contains web pages. For example, the first sequence includes pages such as 71, 22, 56. Each page from this sequence is separated by single space and a -1. The value "-2" indicates the end of the sequence. All pages in this sequence database were encoded by numbers from lowest one to highest one in which the lowest number is 0, the higher number is H, where H is the quantity of all distinct pages. In this case, the page 71 corresponds with node 16, the page 300 corresponds with node 19 and so on. A graph built from the sequence database (see Zone 1 and Zone 2 in Figure 6) is illustrated in Figure 7. The result that gained in Zone 3 was calculated with d = 0.85, the number of nodes is 20, the number of loops is 1000. According to Zone 4, shown in Figure 6, the average of values of Page Rank for every sequence were calculated. For example, the first sequence has the average of values of Page Rank 0.063, the second sequence has the average of values of Page Rank 0.055, and so on. The last zone (Zone 5) shows that sequences in this sequence database were sorted by average of values of Page Rank (from highest value down to lowest one).

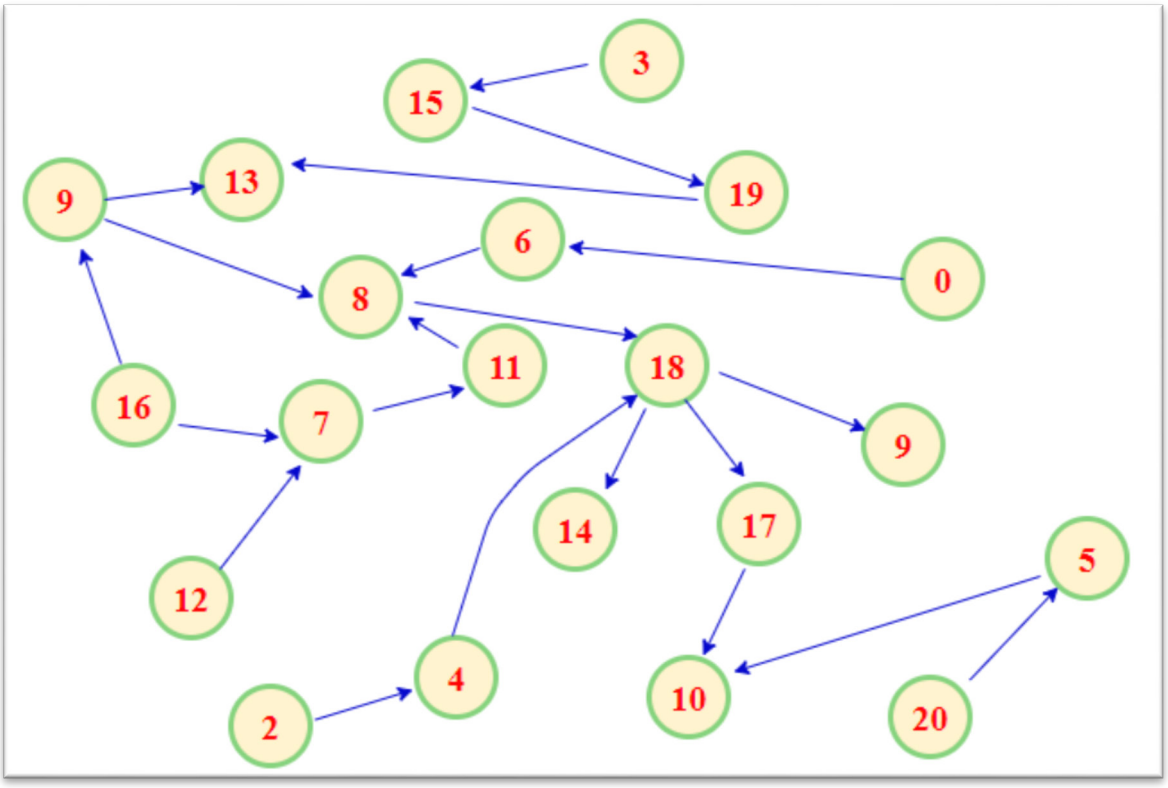


**Figure 7. A graph was created from sequence database**

## EXPERIMENTAL EVALUATION

To evaluate the accuracy of our proposed approach, a set of experiments is 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 Java 8.1 environment with Eclipse Neon.3.

### *DATASETS*

We use three real datasets that have been collected from philippe-fournier-viger.com/spmf/index.php?link=datasets.php They contain sequences of web pages visited by users on websites.

**MSNBC** is a dataset of clickstream data. The original dataset contains up to 989,818 sequences collected from the UCI repository (http://archive.ics.uci.edu/ml/index.php). In this research, we mine on a sub-dataset of MSNBC containing 31,790 sequences and 18 distinct pages.

**FIFA** is a dataset created by processing a part of the web logs from the World Cup. In this paper, we choose a dataset consisting of 20,450 sequences of clickstream data from the website of FIFA World Cup 98 (http://hita.ee.lbl.gov/html/contrib/WorldCup.html). There are distinct 2991 pages in this dataset.

**KOSARAK** is a clickstream dataset collected from a Hungarian news portal (http://fimi.ua.ac.be/data) This dataset contains 69999 sequences and 19986 distinct pages. This dataset is the largest dataset used in our experimental evaluation.

### *EVALUATION FRAMEWORK*

We developed part of the source code related to Page Rank calculation at the link https://cocosci.github.io/MPIPagerank/ (visited on 14-Jan-2019) to complete our proposed approach.

Besides, we utilized the SPMF framework (Fournier-Viger, Gomariz, Gueniche, et al., 2014) for comparing our approach with state-of-the-art approaches applied to the three datasets above. The framework was developed in Java. According to (Gueniche et al., 2013), a prediction has three possible outcomes: (i) The prediction is a success if the generated candidate appears in the suffix of the test sequence; (ii) The prediction is not a match if the predictor is unable to perform a prediction; (iii) Otherwise, the prediction is a failure. We use the accuracy measure to assess the overall performance of each predictor.

$$Accuracy = \frac{|successes|}{|sequences|} \qquad (2)$$

Where: Accuracy (eq. 2) is our major measure to evaluate the accuracy of a given predictor (Gueniche et al., 2013). It is the number of successful predictions compared to the total number of test sequences.

## RESULTS

In this experiment, we use PageRank algorithm to shorten the size of three real datasets such as MSNBC, FIFA and KOSARAK. Then, we check the accuracy on new datasets (after shortening the size of original datasets) by using SPMF framework (Fournier-Viger, Gomariz, Gueniche, et al., 2014). Particularly, we chose 21 different values of *K* at points of 2% interval (through 100% down to 60%) for reducing the size of the sequence databases using our proposed approach presented in Section 3. All mentioned shorten datasets could be downloaded at http://bit.ly/2AniqEm. The results are shown in three figures 8, 9 and 10, respectively.

As illustrated in Figure 8, the original sequence database MSNBC has a prediction accuracy of 46.389% (correspond with the line $x = 46.389$). This is a relatively low. As shown in this figure, the prediction accuracy steadily increases as the dataset of sequences is reduced down to 90%, and after that it continues to improve slightly. This shows the original dataset MSNBC still contains much redundant and meaningless data. It means that using PageRank algorithm to reduce useless data is a appropriate solution.

Figure 9 shows the original dataset FIFA has an accuracy of 99.888% (correspond with the line $x = 99.888$), a rather high accuracy. At the points $K \in [82\%, 100\%)$, the accuracies of datasets with shorten size of FIFA dataset are higher than that of original FIFA dataset.

Also, in Figure 10 the original dataset KOSARAK achieves an accuracy of 99.947% (correspond with the line $x = 99.94$). Although with some fluctuation, the prediction accuracies with shortened versions of KOSARAK dataset are always higher than that of that original KOSARAK dataset, reaching a peak at the point K = 78%. It drops below the original level at K = 66%.

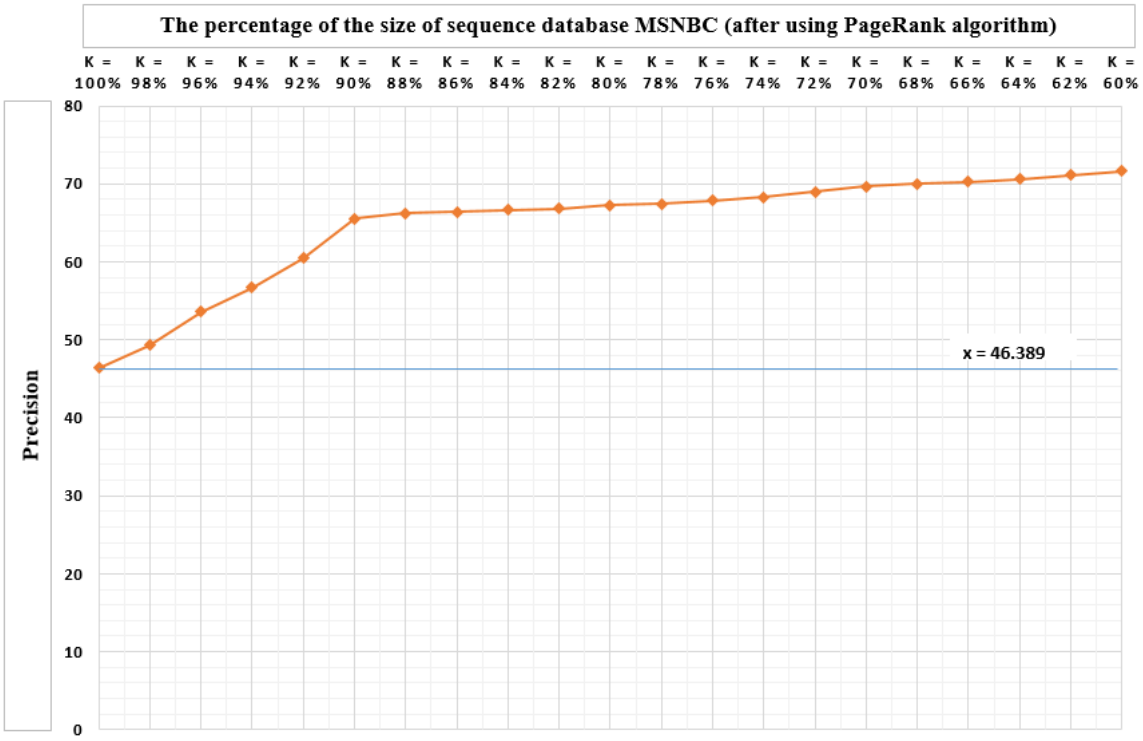


**Figure 8. Reducing the size of MSNBC dataset without the accuracy (up to 40%)**

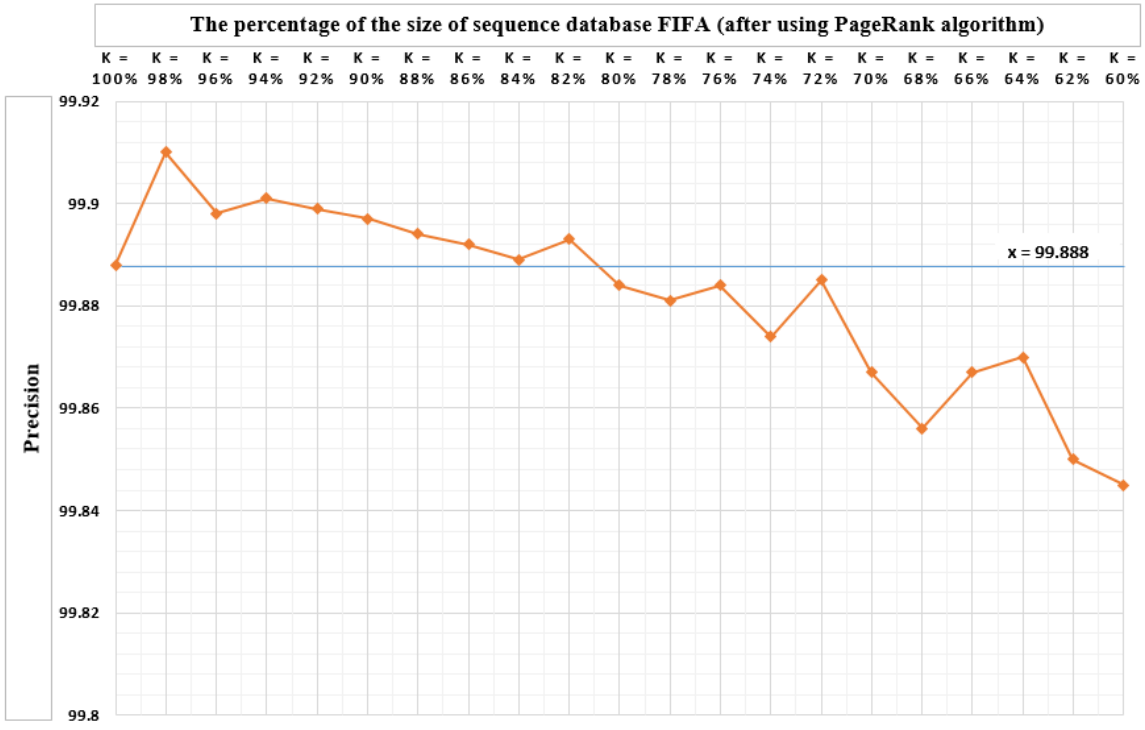


**Figure 9. Reducing the size of FIFA dataset without the accuracy (up to 15%)**

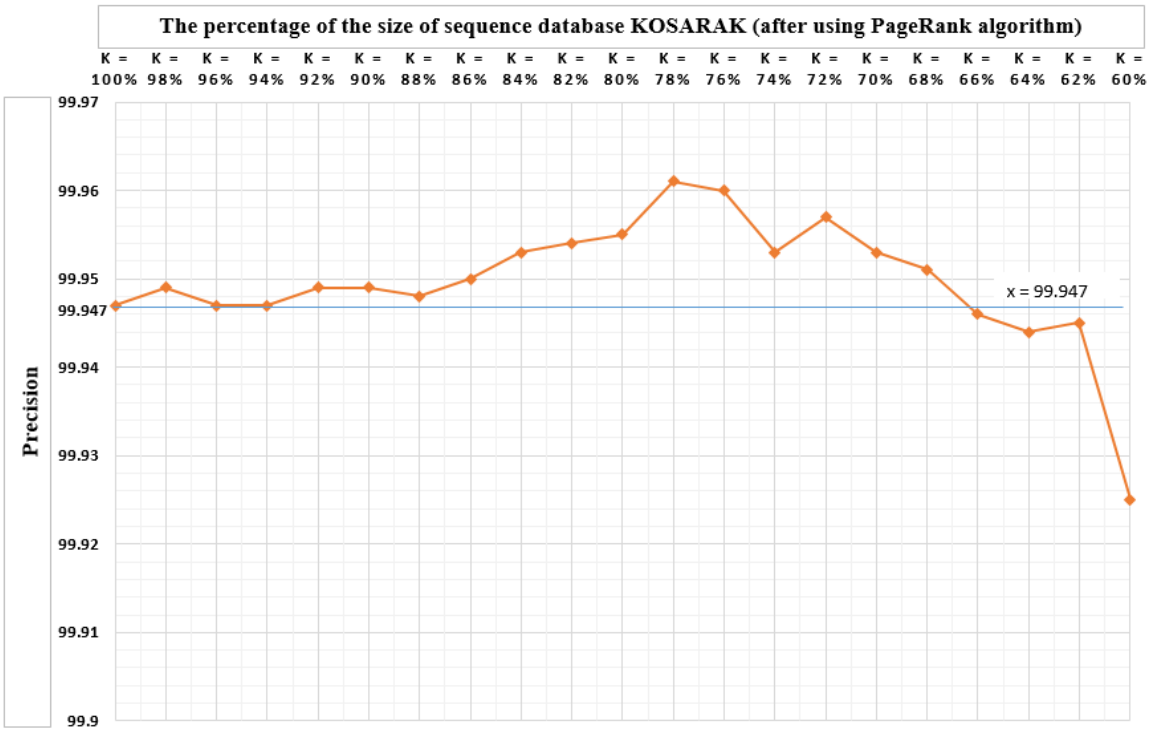


**Figure 10. Reducing the size of KOSARAK dataset without the accuracy (up to 30%)**

To further evaluate our proposed approach, we measure the execution time in milliseconds that the system takes to produce top five best predictions (pages to access next) with and without PageRank algorithm. This evaluation is done on the two datasets FIFA and KOSARAK, with 30 examples of sequence prediction from each dataset. For the dataset FIFA, we chose the *K = 100%* and *K =85%,* and then use CPT+ to predict webpage access to produce the comparison results. Similarly, for the dataset KOSARAK, we chose the *K = 100%* and *K =68%* for comparison. The top five best predictions along with measurements of execution time are listed in Appendices A and B respectively, and the related summary statistics are given in Table 2.

**Table 2. Statistical Results of Execution Time Comparison (N=30)**

| Datasets | Reduction Scales (*k*) | Mean (milliseconds) | Std. Deviation | Difference | *t-statistic (T-test statistic)* | *p-value (Probability value)* |
|---|---|---|---|---|---|---|
| FIFA | 100% | 13308.37 | 1052.72 | 3450.67 | 9.336 | <0.001 |
| | 85% | 9857.70 | 1766.10 | | | |
| KOSARAK | 100% | 14572.00 | 1352.78 | 4110.17 | 16.083 | <0.001 |
| | 68% | 10461.83 | 581.83 | | | |

The experimental results show that using PageRank algorithm to reduce the size of datasets resulted in significant shortening of execution time and thus improved the system's efficiency. As shown in Table 2, the computation of prediction with the reduced dataset of sequences is approximately 1.35 times faster than that of the original dataset in the case of FIFA, and approximately 1.4 times faster in the case of KOSARAK. In both cases, a paired sample t-test found the difference in execution time to be statistically significant.

## FINDINGS AND DISCUSSION

The experimental results show that shortening datasets for webpage prediction is very significant and useful. As illustrated, in datasets having very low accuracies of prediction, redundant and meaningless data are required to be eliminated. In the case of MSNBC, the original prediction accuracy is just less than 50%. This poor performance accounts for the large amount of redundancy in the dataset. As shown by the experiment, the accuracy of MSNBC is improved more than 20% thanks to PageRank algorithm removing useless data. The low accuracy before using PageRank is due to the number of distinct item in this dataset is quite small (only 17 items and the average number of itemsets per sequence is 13.33). Besides, there are a lot of items appear repetitively in the same sequences. For example, the sequence *1 -1 2 -1 3 -1 1 -1 2 -1 1 -1 2 -1 1 -1 2 -1 1 -1 -2* has too many item 2 and they are occur many times.

Although the prediction accuracies of FIFA dataset and KOSARAK dataset are relatively high, they can be improved by applying PageRank algorithm. Furthermore, the prediction runtime is also decreased.

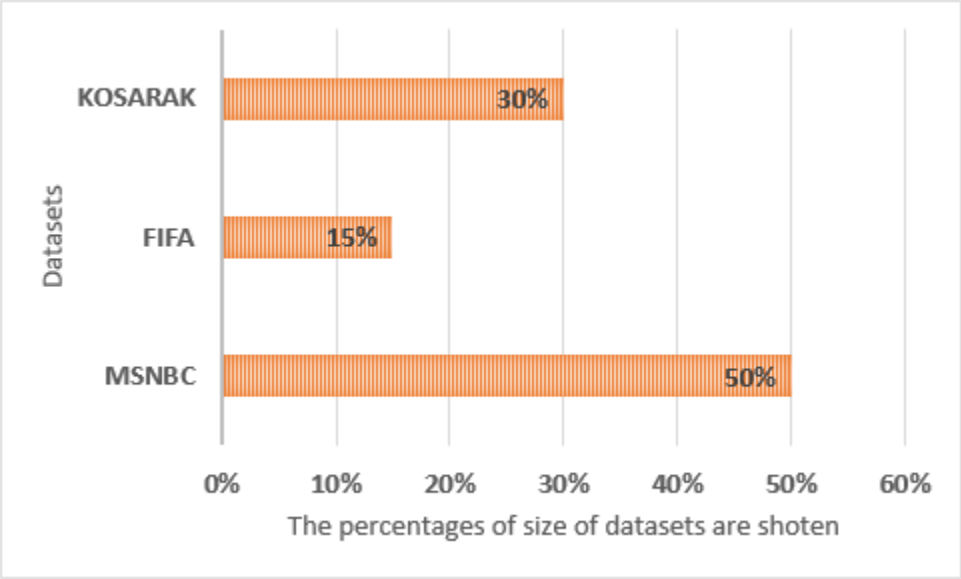


**Figure 11. Shortening the size of datasets KOSARAK, FIFA, MSNBC without accuracy**

According to the results, if the size of dataset is reduced 15 percent to 30 percent (without dropping the prediction accuracy ), the execution time for webpage access prediction also decreases approximately 1.3 to 1.4 times, therefore, increasing computational efficiency. Figure 11 shows that we could shorten the size of sequence database up to 50% with integration of PageRank into CPT without losing accuracy.

## CONCLUSION

Predicting the next item in a sequence over a finite alphabet is essential in a wide range of applications in many domains, especially webpage access prediction. In this paper, we introduce a novel approach to reduce the computational space of webpage access prediction by using PageRank algorithm. Our experimental results on three real datasets show that our proposed approach can remove redundant data from sequence databases but improving the accuracy. The original datasets can be reduced down to 70-85% without significantly compromising the prediction accuracy.

In the future, we aim to further improve the accuracy of webpage access prediction and the execution time for predicting webpage access by using the combination among CPT+ and novel algorithms.

## ACKNOWLEDGEMENT

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM). Besides, we are also grateful to Full Professor Philippe Fournier Viger (philippe-fournier-viger.com/) for his helpful comments and suggestions in order that we could finish this paper.

## REFERENCES

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1-7), 107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

Da, N. T., Hanh, T., & Duy, P. H. (2018). An approach to build sequence database from web log data for webpage access prediction. *International Journal of Computer Science and Network Security, 18*(3), 138-143.

Deshpande, M., & Karypis, G. (2004). Selective Markov models for predicting Webpage accesses. *ACM Transactions on Internet Technology (TOIT), 4*(2), 163-184. https://doi.org/10.1145/990301.990304

Fournier-Viger, P., Gomariz, A., Campos, M., & Thomas, R. (2014). Fast vertical mining of sequential patterns using co-occurrence information. Paper presented at the *Pacific-Asia Conf. on Knowledge Discovery and Data Mining (May 13-16).*

Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., & Tseng, V. S. (2014). SPMF: A Java open-source pattern mining library. *Journal of Machine Learning Research, 15*(1), 3389-3393.

Fournier-Viger, P., Lin, J. C.-W., Kiran, R. U., Koh, Y. S., & Thomas, R. (2017). A survey of sequential pattern mining. *Data Science and Pattern Recognition, 1*(1), 54-77.

Fournier-Viger, P., Wu, C.-W., Tseng, V. S., Cao, L., & Nkambou, R. (2015). Mining partially-ordered sequential rules common to multiple sequences. *IEEE Transactions on Knowledge and Data Engineering, 27*(8), 2203-2216. https://doi.org/10.1109/TKDE.2015.2405509

Geetharamani, R., Revathy, P., & Jacob, S. G. (2015). Prediction of users' web page access behaviour using association rule mining. *Sadhana, 40*(8), 2353-2365. https://doi.org/10.1007/s12046-015-0424-0

Gueniche, T., Fournier-Viger, P., Raman, R., & Tseng, V. S. (2015). CPT+: Decreasing the time/space complexity of the compact prediction tree. Paper presented at the *Pacific-Asia Conf. on Knowledge Discovery and Data Mining (May 19-22).*

Gueniche, T., Fournier-Viger, P., & Tseng, V. S. (2013). Compact prediction tree: a lossless model for accurate sequence prediction. Paper presented at the *ADMA 2013: Advanced Data Mining and Applications (December 14-16).* https://doi.org/10.1007/978-3-642-53917-6_16

Mabroukeh, N. R., & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR), 43*(1), 3. https://doi.org/10.1145/1824795.1824798

Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Using sequential and non-sequential patterns in predictive web usage mining tasks. Paper presented at the *2002 IEEE International Conf. on Data Mining, 2002 (December 9-12). ICDM 2003 Proceedings.* https://doi.org/10.1109/ICDM.2002.1184025

Narvekar, M., & Banu, S. S. (2015). Predicting user's web navigation behavior using hybrid approach. *Procedia Computer Science, 45*, 3-12. https://doi.org/10.1016/j.procs.2015.03.073

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H. . . . Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1-37. https://doi.org/10.1007/s10115-007-0114-2

Yang, Q., Li, T., & Wang, K. (2004). Building association-rule based sequential classifiers for web-document prediction. *Data Mining and Knowledge Discovery, 8*(3), 253-273. https://doi.org/10.1023/B:DAMI.0000023675.04946.f1

# APPENDICES

## APPENDIX A: EXECUTION TIME OF COMPUTING SAMPLE WEBPAGE ACCESS PREDICTION ON FIFA DATASET

| NO. | SEQUENCE OF PAGES (Input) | TOP 5 BEST PREDICTED PAGES (Output) | EXECUTION TIME (Milliseconds) | |
|---|---|---|---|---|
| | | | *k = 100%* (without PageRank algorithm) | *k = 85%* (with PageRank algorithm) |
| 1. | 400, 129, 33 | 101, 376, 164, 216, 202 | 14104 | 10415 |
| 2. | 262, 493, 248 | 28, 3, 80, 109, 266 | 14075 | 9903 |
| 3. | 293, 28, 39 | 214, 92, 112, 80, 5 | 13609 | 11058 |
| 4. | 1119, 675, 564 | 672, 384, 341, 383, 308 | 12040 | 9923 |
| 5. | 124, 27, 222 | 161, 193, 225, 185, 254 | 13330 | 10741 |
| 6. | 1435, 1418, 498 | 577, 289, 472, 679, 674 | 12240 | 9750 |
| 7. | 155, 131, 1 | 30, 98, 13, 18, 36 | 11951 | 9466 |
| 8. | 657, 653, 297 | 760, 710, 714, 685, 102 | 12215 | 9182 |
| 9. | 23, 511, 1006 | 271, 266, 248, 80, 262 | 12564 | 10293 |
| 10. | 1074, 216, 275 | 164, 376, 111, 323, 136 | 12175 | 9428 |
| 11. | 1004, 620, 1034, 1000 | 85, 753, 514, 746, 69 | 12293 | 10070 |
| 12. | 1039, 1053, 1034, 977 | 85, 620, 520, 753, 514 | 13343 | 9767 |
| 13. | 170, 178, 200, 737 | 28, 182, 109, 80, 3 | 12854 | 9640 |
| 14. | 1134, 947, 1138, 1162 | 1150, 1180, 1142, 1149, 1131 | 13999 | 10408 |
| 15. | 1233, 287, 663, 1001 | 85, 514, 341, 383, 384 | 12692 | 10315 |
| 16. | 262 3 109 80 | 28, 182, 72, 63, 51 | 13053 | 9744 |
| 17. | 1060, 899, 905, 988 | 1003, 1023, 2251, 2250, 2007 | 15986 | 9556 |
| 18. | 878, 876, 2172, 2169 | 2175, 2444, 2174, 2435, 2167 | 14753 | 10545 |
| 19. | 612, 377, 1669, 482 | 646, 668, 483, 593, 1733 | 15880 | 10364 |
| 20. | 527, 356, 506, 145, | 28 80, 182, 109, 4, 248 | 13837 | 11210 |
| 21. | 1552, 1558, 1566, 1685, 1688 | 1681, 1680, 1694, 1690, 1571 | 12643 | 10182 |
| 22. | 204, 208, 207, 335, 166 | 225, 185, 254, 168, 2503 | 12600 | 10023 |
| 23. | 264, 164, 376, 136, 111 | 43, 323, 165, 296, 116 | 12939 | 10657 |
| 24. | 1124, 1381, 1383, 1485, 1131 | 564, 1408, 1400, 1133, 118 | 12322 | 10190 |
| 25. | 88, 448, 345, 342, 591 | 383, 480, 384, 621, 341 | 12951 | 9992 |
| 26. | 123, 544, 909, 973, 684 | 1351, 1291, 735, 948, 734 | 13055 | 9580 |
| 27. | 420, 447, 345, 777, 735 | 843, 369, 824, 853, 850 | 13429 | 12007 |
| 28. | 102, 138, 37, 131, 146 | 313, 224, 184, 179, 194 | 14003 | 10557 |

| NO. | SEQUENCE OF PAGES (Input) | TOP 5 BEST PREDICTED PAGES (Output) | EXECUTION TIME (Milliseconds) | |
|---|---|---|---|---|
| | | | *k = 100%* (without Pag-eRank algorithm) | *k = 85%* (with Pag-eRank algorithm) |
| 29. | 87, 835, 805, 845, 1639 | 322, 180, 376, 116, 1074 | 13565 | 1052 |
| 30. | 527, 356, 506, 145, 28 | 80, 182, 109, 4, 248 | 14751 | 9713 |

## APPENDIX B: EXECUTION TIME OF COMPUTING SAMPLE WEBPAGE ACCESS PREDICTION ON KOSARAK DATASET

| NO. | SEQUENCE OF PAGES (Input) | TOP 5 BEST PREDICTED PAGES (Output) | EXECUTION TIME (Milliseconds) | |
|---|---|---|---|---|
| | | | *k = 100%* (without Pag-eRank algorithm) | *k = 68%* (with Pag-eRank algorithm) |
| 1. | 6119, 2876, 101 | 64, 135, 1559, 1037, 148 | 16241 | 10475 |
| 2. | 20004, 6, 19 | 3, 1018, 361, 294, 4961 | 16156 | 9944 |
| 3. | 8009 8119 1980 | 148, 321, 2243, 4238, 942 | 13731 | 10112 |
| 4. | 1830 306 379 | 148, 7, 3, 135, 438 | 14673 | 10192 |
| 5. | 3235 4631 3810 | 64, 3788, 584, 131, 148 | 15205 | 10351 |
| 6. | 5407 13235 83 | 3, 7, 77, 87, 85 | 16719 | 11029 |
| 7. | 2059 2386 2060 | 64, 4068, 670, 1037, 4069 | 15400 | 10463 |
| 8. | 233 1463 1864 | 1865, 235, 148, 90, 1464 | 17200 | 9726 |
| 9. | 5110 305 11295 | 148, 3, 3720, 262, 3520 | 15333 | 10373 |
| 10. | 117 2904 6741 | 120, 118, 126, 2906, 2908 | 13490 | 9941 |
| 11. | 12 1830 1531 | 2943 148, 1044, 473, 242, 7396 | 15470 | 10970 |
| 12. | 1030 1035 1810 9929 | 90, 3, 1019, 777, 520 | 14030 | 11439 |
| 13. | 5101 325 330 332 | 6, 11, 338, 339, 218 | 14562 | 11323 |
| 14. | 25 259 2952 515 | 90, 772, 1132, 2124, 827 | 13956 | 10360 |
| 15. | 86 1068 12618 1978 | 87, 211, 214, 212, 1868 | 13215 | 10147 |
| 16. | 1687 6563 5369 4979 | 2035, 930, 2872, 3992, 155 | 13448 | 11298 |
| 17. | 821 3492 1910 4200 | 3, 7, 666, 215, 317 | 14860 | 9826 |
| 18. | 51 1155 2864 2866 | 148, 2231, 321, 135, 77 | 18221 | 11157 |
| 19. | 4354 423 1031 738 | 136, 633, 467, 148, 2143 | 14960 | 11368 |
| 20. | 1993 1994 7781 7783 | 6, 1997, 218, 1996, 7788 | 13158 | 10878 |
| 21. | 382 3322 808 3 2356 | 148, 423, 135, 64, 473 | 13221 | 10956 |
| 22. | 512 236 4732 2953 515 | 90, 2124, 827, 371, 650 | 15186 | 10174 |
| 23. | 1101 1141 587 442 1734 | 465, 464, 455, 467, 744 | 14170 | 11550 |
| 24. | 4643 1520 695 3141 784 | 148, 3, 672, 1071, 266 | 14282 | 10101 |
| 25. | 220 229 232 6882 2214 | 148, 397, 3322, 694, 442 | 13166 | 9747 |
| 26. | 14 106 7121 108 112 | 114, 64, 131, 130, 852 | 14231 | 9958 |

| NO. | SEQUENCE OF PAGES (Input) | TOP 5 BEST PREDICTED PAGES (Output) | EXECUTION TIME (Milliseconds) | |
|---|---|---|---|---|
| | | | *k = 100%* (without PageRank algorithm) | *k = 68%* (with PageRank algorithm) |
| 27. | 66 4212 4103 944 349 | 7, 3, 666, 215, 46 | 13057 | 9863 |
| 28. | 9631 3 628 64 278 | 7, 148, 135, 205, 49 | 12863 | 9944 |
| 29. | 8963 3825 5716 9700 1923 | 316, 3972, 874, 790, 4509 | 13018 | 10450 |
| 30. | 4442 1837 686 354 1645 | 25, 357, 845, 356, 747 | 13938 | 9740 |

## BIOGRAPHIES

**Nguyen Thon Da** received Master 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 Posts and Telecommunications Institute of Technology, Vietnam. He is now working as IT employee and an assistant teacher at Faculty of Information Systems, University of Economics and Law, VNUHCM. His research interests include data mining, pattern mining, sequence analysis and prediction.

**Tan Hanh** received the PhD degree in Informatics from Grenoble INP, France in 2009. Currently, he is vice president of 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.