Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFICIENT TECHNIQUES FOR PREDICTING SUPPLIERS CHURN TENDENCY IN E-COMMERCE BASED ON WEBSITE ACCESS DATA

[1]VERONICA S. MOERTINI, [2]NIKO IBRAHIM, [3]LIONOV

[1,3]Informatics Department, Parahyangan Catholic University
[2]Information System Department, Maranatha Christian University
Bandung, Indonesia
E-mail: moertini@unpar.ac.id, niko.ibrahim@it.maranatha.edu, lionov@unpar.ac.id

ABSTRACT

Electronic supplier relationship management (e-SRM) is important in order to maintain strong, long lasting and beneficial relationship between e-commerce firms and their suppliers. One important function of e-SRM is to predict suppliers who tend to churn such that early "treatment" can be given. In the e-commerce systems that involve suppliers as the websites users, predicting suppliers' churn tendency can be based on analyzing their frequencies in accessing the e-commerce websites.
Our proposed techniques include data warehouse design (supporting the data collection and preprocessing) and unsupervised algorithms that analyze the preprocessed bitmaps of time series data representing suppliers website access from time to time. Having bitmaps as inputs, our proposed algorithms are efficient (the time complexity is *O(n)*) as proven with our experiments. In experimenting with real world data of an e-commerce system selling hotel rooms, our techniques produce output of supplier segments where each segment has certain churn level tendency and need specific treatment.

Keywords: *Churn Prediction In E-Commerce, Supplier Relationship Management, Web Usage Mining*

1. INTRODUCTION

E-commerce systems come in several business models. Few of the models - such as B2C Transaction Broker, B2C Market Creator, B2B E-procurement and Private Industrial Networks - involve customers (buyers) and suppliers as the websites users [1]. For those models, having electronic supplier relationship management (e-SRM) is important in order to maintain strong, long lasting and beneficial relationship between the e-commerce firms and their suppliers. One of the important functions of e-SRM is to predict suppliers who tend to churn such that early "treatment" can be given to prevent them from churning. To reduce cost and maximize effectiveness, churn prediction techniques have to be fast and accurate to ensure that the right limited suppliers are being targeted for retention treatment.

While electronic customer relationship management (e-CRM) concept has been matured and discussed widely in literatures, e-SRM is a relatively new concept [2]. In e-CRM, many methods for predicting customers' behavior of churning have been developed. The methods adopt traditional methods as well as soft computing [3], where many of these are developed based on *recency, frequency* and *monetary* (see Subsection 2.2 for more discussions) ([4], [5]). Among those methods, we have not found any churn prediction method that is based on the recency and frequency of customers website access. This seems to be justified as e-commerce customers may just visit and shop at an e-commerce website intermittently or occasionally, not regularly, just based on their impulsive need or desire. Contrarily to this fact, in e-commerce systems that involve suppliers as the websites users, the suppliers are expected to continuously manage their profile, products, services or campaigns from time to time. Thus, one approach for predicting suppliers' churn behavior is by adopting web usage mining techniques, specifically by analyzing suppliers recency and frequency in accessing the websites.

In mining web usage, raw dataset must be collected and then preprocessed with certain techniques such that the transformed dataset is ready to be fed into data mining algorithms [6]. In the case of mining

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

suppliers web usage, we view that the better approach is to develop data warehouse that is used to store the preprocessed historical data such that the data needed can be added as well as analyzed periodically. Another important concept that we adopt is bitmap index in DBMS technology, which is known for its efficiency in processing queries involving bit operations, such AND, OR or COUNT [7]. In order to gain speedy computations, in our techniques, data representing the recency and frequency of the suppliers are transformed into bitmaps of time series data.

In this research, we intend to contribute in developing novel techniques used to predict suppliers behavior towards churning based on recency and frequency in accessing the websites. The techniques include data warehouse design (specifically to support the data collection and preprocessing) and unsupervised algorithms for predicting the level of suppliers tendency towards churning by taking inputs of bitmaps representing the recency and frequency of website usage. The use of bitmaps is intended to gain time complexity of $O(n)$ of the proposed algorithms.

This paper is organized as follows: Introduction, literature review depicting relevant concepts, proposed systems and techniques, experiments and conclusions.

2. LITERATURE REVIEW

2.1. e-Commerce Models Involving Online Suppliers

E-commerce comes in several business models. There are few models that involving customers (buyers) and suppliers as e-commerce websites users, such as [1]:

(a) *B2C Transaction Broker*: The e-commerce firms act as brokers who sell suppliers' products or services via the websites and obtain transaction fees. Here, the suppliers and customers do not interact directly. Industries using this model includes financial services, travel services, job placement services.

(b) *B2C Market Creator*: The owner firms uses the websites to create markets that bring buyers and sellers together. Hence, the suppliers and customers do not interact directly.

(c) B2B E-procurement, Exchanges and *Industry Consortia*: The websites act as digital marketplace where suppliers and purchasers conduct transactions.

(c) Private Industrial Networks: The websites are designed to coordinate flow of communication among firms engaged in business together.

The users of customer access those websites mainly to search and purchase products, which can be incidentally, impulsively or infrequently. On the other hand, the suppliers use the websites to maintain company profiles, update products/services offered, create campaigns, and so on, and are expected to be frequent at all times in conducting these activities.

2.2. CRM Techniques

Recency, Frequency, Monetary (RFM) analysis is a traditional approach to analyzing customer behavior in the retailing and have been widely adopted. This analysis divides customers into groups, based on how recently they have made a purchase, how frequently they make purchases, and how much money they have spent ([4], [5]). The RFM analysis uses: (a) Recency: The time (in units such as days/months/years) since the most recent purchase transaction or shopping visit; (b) Frequency: The total number of purchase transactions or shopping visits in the period examined; (c) Monetary: The total value of the purchases within the period examined or the average value (e.g., monthly average value) per time unit. This analysis divides each of the three dimensions of RFM into equal sized chunks and places customers in the corresponding chunk along each dimension.

It is reported in [3] that popular methods for churn predictions in CRM are: (a) Traditional methods: Decision Trees, Regression Analysis, Support vector machine (SVM), Naïve Bayes, k-nearest neighbor (KNN); (b) Soft computing: fuzzy logic, neural networks, and genetic algorithms. Hadden [3] develops supplier churn prediction model based on Neural Networks, Regression Trees and Linear Regression. The data used in developing the models are customer profiles and various data related to customer complaints and repair.

2.3. Web Usage Mining

Clickstream is foundational data. The data can be used to measure pages and campaigns and analyze all kinds of behavior, such as visits, visitors, time on site, page view, bounce rate, and so on [8]. Clickstream and associated data, which are generated from user interactions with websites, can also be analyzed to discover patterns [6]. In general, the process consists of two phases:

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(a) Data preparation: Data of web application logs database are selected as needed, cleaned, identified, parsed, integrated, transformed and then stored in another database. The data analyzed can be categorized into four primary groups, which are usage, content, structure and user data.
(b) Pattern discovery phase: The transformed data are fed into pattern discovery algorithms, the output are analyzed for selecting the useful ones. There are several techniques for analyzing the data used in pattern discovery phase, which are clustering, classification, association and correlation analysis, and so on.
The useful patterns discovered can then be used to personalize services (for examples of such services, see [9] and [4]).

In this research, we choose clustering approach. Clustering is a data mining technique that groups together a set of items having similar characteristics. In the web usage domain, there are two kinds of interesting clusters that can be discovered, which are user clusters and page clusters. Clustering of user transactions records is one of the most commonly used tasks in Web usage mining [6].

2.4. Data Warehouse

Data warehouse is a subject-oriented, integrated, time-variant (historical) and non-volatile collection of data supporting fast and systematic data analysis needed in decision making process. It is populated from the enterprise databases as well as other sources by ETCL (*extract, transform, clean* and *load*) functions [10]. The data warehouse system architecture can be single, two or three layer. Three-layer design consist of source, reconciled and data warehouse layer (see Figure 1). The reconciled layer is a temporary database used to store the selected data from sources, which may need to be cleaned and transformed before loaded into the data warehouse schema. To facilitate the storing of historical websites access data, three-layer architecture is needed.

The basic schema employed in data warehouse and data mart is star schema. It consists of fact and dimension tables. The fact table, which is usually the center of a star schema, contains facts that are linked through their dimensions. Hence, it has primary keys of its linked dimensions and measure attribute(s) that represent specific business aspect or activity. The dimensions provide descriptive characteristics about the facts through their attributes.

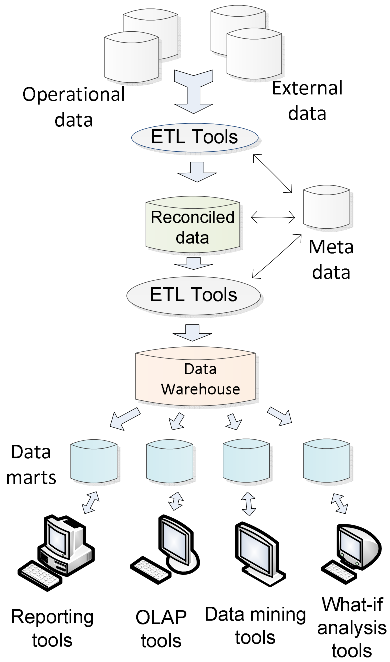


Figure 1: Three layer architecture for a data warehouse system [10].

2.5. Research Results of SRM

[11] reviews supplier selection and order allocation models based on an extensive search in the literature (170 paper during 2000-2010) and show the contributions to supply chain management (SCM). The techniques found in the research results are grouped into Multi Attribute Decision Making (MADM), mathematical programming, Artificial Intelligence methods (case-base reasoning, neural network, decision tree, association rules, cluster analysis), statistical models, fuzzy set theory, and hybrid models. Among the results of selecting (segmenting) suppliers models/techniques, the most common used criteria are material/products quality, lead time or delivery time and price.

3. PROPOSED SYSTEM AND TECHNIQUES

Based on our survey and knowledge we conclude that the frequency and recency of supplier activities (accessing the website for updating their profile, products catalog, creating campaigns, etc.) can be used to predict their tendency of churning. That is, if the frequency of online activities is low or declining during some period of observation, it can be interpreted that the suppliers are less interested (in conducting business activities), which may lead to churn. Hence, the system and techniques that we propose are intended to analyze

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the supplier log activities stored in operational databases.

3.1. Data Warehouse System Architecture

In e-commerce systems, data warehouses can also be designed to support e-CRM and e-SRM. To provide data needed for suppliers' churn prediction based on the website access frequency, the data warehouse should stores the summarized of suppliers websites access data from time to time. Then, time series data representing the recency and frequency of suppliers' activities can be generated from the data mart (created from the data warehouse) and fed into the prediction algorithms. (Note: Time series data is a set of data about individual/agent behavior over time, where the time unit of observation can be day, week, month, or year [12]).

The architecture of the proposed system is shown in Figure 2. It is basically a three-layer data warehouse system (see Subsection 2.4). The supplier web access logs and other data are extracted from data sources (operational databases), which are then cleaned, transform and load onto data warehouse. A data mart containing time series data of frequency and statuses of suppliers' web access is created from the data warehouse, where its content is used to feed the proposed churn prediction algorithms.

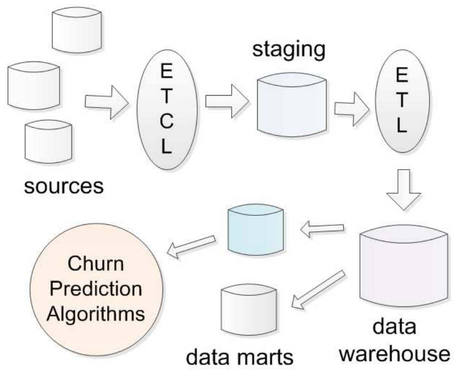


Figure 2: The System Architecture.

3.2. Some Part of Data Warehouse Design

In our previous research results presented in [13], we have developed an incremental method in developing systems supporting clients management for small medium enterprises, where the data warehouse can be built incrementally by developing a data mart at each cycle. Here, we present the design of a data mart needed.

The data mart design is shown in Figure 3. In DimTime table, *UnitTimeOfYear* attribute can be number day of year (having values of 1 to 362), week of year (1 to 52), month of year (1 to 12) or other unit as needed. In FactAccess table, AccessFreq is the frequency of accessing the e-commerce website, while AccessStatus is the status (having value '1' if the frequency is greater than 0 and '0' if otherwise).

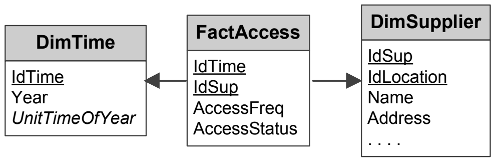


Figure 3: Some Part Of The Data Warehouse Schema.

From the schema, if *UnitTimeOfYear* depicts week, some example of the time series data created in the last 24 weeks that can be queried from the data warehouse is shown in Table 1.

Table 1: Time Series Data Queried From The Data Warehouse.

IdSup	Year	1F	1S	2 F	2S	…	24F	24S
IdS1	2014	2	1	0	0	…	0	0
IdS7	2014	3	1	1	1	…	4	1
IdS23	2014	0	0	1	1	…	2	1
.. . .	. . .							...

Note: *iF* = frequency of week-*i*, *jS* = status of week-*j*

Data Sources and Transform Functions Design:
Among the available data sources in the e-commerce systems, in this web usage mining (see Subsection 2.3), the data needed is usage data. Usage data is the log data collected automatically by the website and application servers representing the behavior of visitors [6]. To populate the designed data mart (see Figure 3), the operational data needed include the logs of suppliers' online transactions. Typical transaction logs tables usually have the following attributes: ID of parties/users (such as suppliers) involved, ID of the related transaction and timestamp.

As known, ETCL functions is run periodically (daily, weekly or others) to add data stored in a data warehouse. The algorithm for transforming data stored in transaction logs into the staging tables is:

Algorithm: Tranform-log
Table accessed:

- Transaction log tables
- StagingTable(IdSup, Year, UnitTimeofYear, AccessStatus, AccessFreq)

Steps:

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

get year // current year
get unit_time // current unit time
compute start_time from unit_time and year
compute end_time from unit_time and year
selected_log_records ← select records having timestamp since start_time to end_time from log tables
create temporary_file where each row contains IdSupplier and array of its transaction timestamps from selected_log_records
for each row in the temporary_file
    freq ← number of timestamp elements in the array
    if freq > 0 then status ←1 else status ← 0
    insert (IdSupplier, year, unit_time, status, freq)
        into StagingTable
end

The content of StagingTable, along with other staging tables, are then read and loaded into data warehouse regularly.

3.4. Input of Bitmap Time Series Data

A bitmap is defined as an array of bits. In DBMS technology, bitmap indexing is known for its efficiency (speed) in querying tables indexed on attributes with small unique values, as querying involve bitmap operations with bits operation (COUNT, AND, OR and NOT) [7]. In creating indices, a bitmap is created for each distinct value of attributes. For attribute value *v*, the bit for a table record is 1 if the record has the value *v*, and is 0 if otherwise. Bitmap indexing is commonly used in data warehouses that manage large volume of data. The proposed algorithms can take the advantages of bitmap index.

The proposed algorithms take bitmap time series data as inputs. Some examples of bitmap time series (selected from Table 1, where attribute of IdSupplier and W*i*_Status are chosen) fed into the proposed algorithms are shown in Table 2 or Table 3. In Table 3, the statuses of each supplier are concatenated into a single bitmap string.

Table 2: Some example of time series dataset.

IdSupplier	1S	2S	…	24S
IdS1	1	0	…	0
IdS7	1	1	…	1
IdS23	0	1	…	1
. . . .				. . .

Table 3: Time series data with 24-bit bitmap string.

IdSupplier	Statuses (24 bits)
IdS1	101011111010 011111111110
IdS7	110101011010 011101110101
IdS23	011100111010 010001110111
. . . .	

3.5. Proposed Algorithms

As discussed in Section 2.3, clustering of user transactions records is one of the most commonly used tasks in web usage mining. We design algorithms to cluster suppliers giving results where each supplier is labeled with its predicted level of churning. In this section, we discuss the data structure (array) that supports efficient computation and the algorithms.

Array of Count (Frequency) of Bit 1: The arrays are designed to store the count of bit 1 in the bitmaps. That is, an array is created for a bitmap with certain length (8, 16, 24 bits, and so on). The bitmap decimal value is used as the index while its bit 1 count is stored as its element value (see Figure 4). The operation of getting an element of an array with its index (look up table), i.e. *arrayName(i)*, is *O(1),* which is very efficient computation.

The following are some examples of bitmaps and their corresponding decimal value (given in '()'):

- 8-bits: 00000000 (0), 00000001 (1), 00000010 (2), . . ., 11111110 (254), 11111111 (255);
- 16-bits: 0000000000000000 (0), 0000000000000001 (1), 0000000000000010 (2), . . ., 1111111111111110 (65534), 1111111111111111 (65535);
- 24-bits: 000000000000000000000000 (0), 000000000000000000000001 (1), . . ., 111111111111111111111110 (16777214), 111111111111111111111111 (16777215).

For bitmap of 32-bits and 64-bits, the maximum value is 4294967295 and 18446744073709551615.

For the arrays depicted in Figure 4, for instance, the operation for obtaining the count of 1 in bitmap "11111110" is simply `countOnes8b[254]`, while "111111111111111111111110" is `countOnes24b[16777214]`.

Index	Value
1	1
2	1
3	2
4	1
. . .	. . .
. . .	. . .
253	7
254	7
255	8

Index	Value
1	1
2	1
3	2
4	1
16777213	23
16777214	23
16777215	24

Figure 4: Array of bit 1 count of 8-bit and 24-bit.

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithms: Our algorithms are designed with the assumption that the patterns of suppliers' frequency in conducting activities (during some period of observation time) used to define predicted churn levels are known in advance. For example:

(1). If in the last period of observation of 24 weeks (6 months), a supplier is less active, i.e. dormant during k% x 24 weeks (for instance, k = 25) or more, then this supplier may churn in the future.

(2). If in the last 6 months, a supplier's frequency in conducting activities decline from month to month, then then this supplier is very likely to churn. If the frequency is declining sometime in the middle of the time only (then up again sometime later), then then this supplier may be churn.

We design 2 algorithms, PredictChurn_1 and PredictChurn_2, with those 2 assumptions and discussed as follows.

PredictChurn-1: The predicted churn level of every supplier is determined by how frequent each supplier conduct online activity (update product catalog, create campaigns, and so on) during the period of observation. Using some defined threshold frequency values of *low, medium* and *high*, the algorithm predicts the churn level of every supplier accordingly.

Algorithm: PredictChurn-1
Input: (a) File containing records of IdSupplier and its bitmap of statuses (see Table 3), fInput; (b) File containing count of 1 (see Figure 4), fCountOne; (c) Value of *threshold1, threshold2, threshold3*
Output: array of IdSupplier who predicted to churn, IdSupChurn[], and its churn level, ChurnLevel[] where the level is 1 = low or 2 = medium or 3 = high
Descriptions: Predicting supplier's churn behavior based on count (frequency) of bit 1: If the count < *threshold1* then cLevel = 3, else if *threshold1* >= count < *threshold2* then cLevel = 2, else if *threshold2* >= count < *threshold3* then cLevel = 1
Steps:

```
Read fInput, store IdSuppliers in IdSup[] and bitmaps in bitStr[]
Read fCountOne and store the array in countOnes[]
Initialize IdSupChurn[]; Initialize ChurnLevel[]
bitmapSize ← length of bitStr[0]
th1 ← threshold1 x bitmapSize;
h2 ← threshold2 x bitmapSize;
h3 ← threshold3 x bitmapSize
j ← 0
for i=0 until size of IdSup[]-1  // evaluate every supplier
   cLevel = 0 //default value: not churn
   decVal ← decimal value of bitStr[i]
   ctOnes ← countOnes[decVal] // look up count
        // of bit 1
   if ctOnes < th1then cLevel = 3  // high
   else if th1>= ctOnes < th2 then cLevel = 2
        // medium
   else if  th2 >= ctOnes <  th3 then cLevel = 1 // low
   if cLevel > 0
      IdSupChurn[j] = IdSup[i]
      ChurnLevel[j] = cLevel
      j← j + 1
return IdSupChurn[],ChurnLevel[]
```

PredictChurn-2: The predicted churn level of every supplier is determined by the frequency changes from "window time" to "window time". Here, the overall bitmap will be divided into windows by masking it with the defined masks. Some example of the bitmap where divided into 4 windows and the corresponding 4 masks are as follows:

1011	0111	1011	1110
window1	*window2*	*window3*	*window4*

```
mask1 = 1111000000000000;
mask2 = 0000111100000000;
mask3 = 0000000011110000;
mask4 = 0000000000001111;
```

The number of windows and masks can be designed as needed (based on the known knowledge). In the detailed algorithm below, we provide some example of known patterns (for predicting churn level) with 4 windows.

Algorithm: PredictChurn-2 //In this algorithm, number of windows used are 4 (four)
Input: (a) File containing records of IdSupplier and its bitmap of statuses (see Table 3), fInput; (b) File containing count of 1 (see Figure 4), fCountOne; (c) Masking bitmaps, mask1, mask2, mask3, mask4
Output: array of IdSupplier who predicted to churn, IdSupChurn[], and its churn level, 1, 2, or 3
Descriptions: Predicting supplier's churn behavior based on the change of count (frequency) of bit 1 in the windows.
Known rules (for examples): If the frequency of web access:
is zero then churn level = 4;
is fully declining then churn level = 3;
is declining at the beginning of 2 windows then churn level = 2;
is declining at the beginning of 1 window then churn level = 1;
is declining at the end of 2 windows then churn level =2;
Steps:
Read fInput store IdSuppliers in IdSup[] and bitmap strings in bitStr[]
Read fCountOne store in count of bit 1 in countOnes[]
Initialize IdSupChurn[]; Initialize ChurnLevel[]

Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
bitmapSize ← length of bitStr[0]
th ← threshold x bitmapSize;
j ← 0
for i=0 until size of IdSup[]-1 //evaluate every supplier in IdSup[]
   cL = 0 //default value: not churn
   //Mask the bitmaps and get the decimal values
   dV1 ← decimal of (binary of bitStr[i] AND mask1);
   dV2 ← decimal of (binary of bitStr[i] AND mask2);
   dV3 ← decimal of (binary of bitStr[i] AND mask3);
   dV4 ← decimal of (binary of bitStr[i] AND mask4);
   //Get the count of bit 1
   ct1 ← countOnes[dV1]; ct2 ← countOnes[dV2];
   ct3 ← countOnes[dV3]; ct4 ← countOnes[dV4];
   ct = ct1 + ct2 + ct3 + ct4
   if ct = 0 then cL = 4
   else if 0 < ct < th
      if ct1>ct2 && ct2>ct3 && ct3>ct4 then
         cL = 3
      else if ct1>ct2 && ct2>ct3 && ct3<ct4 then
         cL = 2
      else if ct1>ct2 && ct2<ct3 && ct3<ct4 then
         cL = 1
      else if ct1<ct2 && ct2>ct3 && ct3>ct4 then
         cL = 2
   if cL > 0
      IdSupChurn[j] = IdSup[i]
      ChurnLevel[j] = cL
      j ← j + 1
return IdSupChurn[],ChurnLevel[]
```

Algorithms Complexity: The complexity of PredictChurn_1 and PredictChurn_2 is *O(n)*, where *n* denotes the number of suppliers. Hence, the algorithms are efficient.

The Use and Implementation of the Algorithms: The algorithms can be implemented and used "standalone" for predicting suppliers based on the frequency of conducting activities as well as combined with other algorithms based on monetary analysis. Those algorithms can be implemented:

- Off database: The algorithms are implemented is a program module that read bitmaps stored in the database or files populated with bitmaps read from the data warehouse.
- In the database: The algorithms are implemented as stored procedures in the DBMS which access tables of bitmaps, which can be indexed with bitmap indexing.

4. EXPERIMENT

The experiments were performed in a personal computer with Intell(R) Pentium 4, 3.0 GHz of CPU, 2 GB of RAM, and 32-bit Windows operating system, where bitmaps data are stored in files.

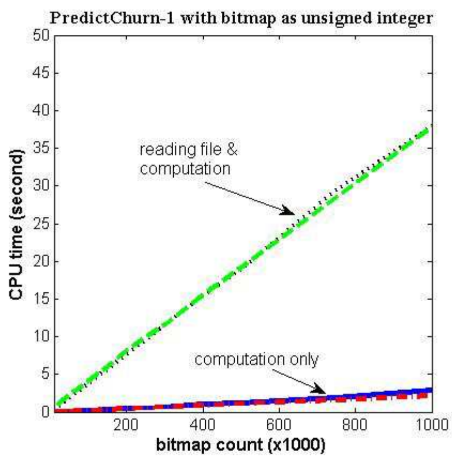
4.1. Time Response Experiment

Experiment Method: In these experiments, we create 2 sets of file containing data simulation of 16-bits and 2 sets of file of 24-bits bitmaps. In the first set of files, the bitmaps are stored as unsigned integers while in the second files the bitmaps are stored as strings. One set include 11 files, where each stores 50K, 100K, 200K, …, 900K or 1M bitmaps. We use the sets of files as the inputs of PredictChurn-1 and PredictChurn-2 algorithm, which are implemented as program modules. Each run of the module is repeated 10 times, where the CPU times for reading input file and computing are recorded in each run. The CPU times are then averaged and the results are discussed below.

In Figure 5, 6 and 7, the CPU time for processing:

- 16-bit bitmaps are presented in blue (for performing computation only) and black (for reading input file and performing computation);
- 24-bit bitmaps are presented in red (for performing computation only) and green (for reading input file and performing computation).

Bitmaps are represented as unsigned integers: As shown in Figure 5, the computation is linear and very efficient. For processing 1,000,000 bitmaps, the response time including for reading input files of PredictChurn-1 is less than 40 seconds. If the time for reading input files is omitted, the response times are less than 5 seconds. (Note: PredictChurn-2 cannot be experimented as the first bits of 0 in bitmaps will be omitted if bitmaps are stored as unsigned integers.)



Journal of Theoretical and Applied Information Technology
30th April 2015. Vol.74 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 5: The time responses of PredictChurn-1 in processing unsigned integer bitmaps.

Bitmaps are represented as strings: As shown in Figure 6 and 7, the computation is linear and also efficient. However, since the computation involve converting strings into unsigned integer values (for obtaining indices used to look up count one arrays), the execution times are approximately 4 times as long compared to the previous experiments. Here, for processing 1,000,000 bitmaps, the response time including for reading input files of both algorithms is less than 200 seconds. If the time for reading input files is omitted, the response times are less than 120 seconds.

Experiment results discussion: The time complexity of *O(n)* of both algorithms have been proven by the linearity of the execution times. If the computation does not involve conversion strings into unsigned integers, the execution times are very efficient. If the computation involves string conversion, three quarter of the times are needed for this operation.

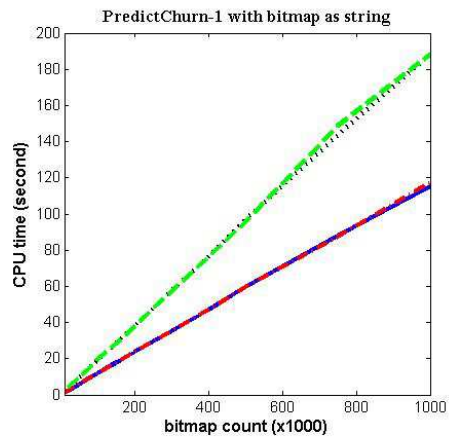


Figure 6: The time responses of PredictChurn-1 in processing string bitmaps.

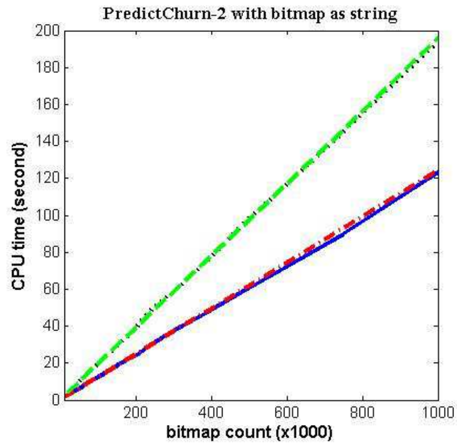


Figure 7: The time responses of PredictChurn-2 in processing string bitmaps.

4.2. Experiment with Case Study Data

The real data used for experiments is obtained from a transaction broker e-commerce selling hotel rooms in Indonesia, which is www.klikhotel.com. Klikhotel has been operating since August 2010 and in 2014 it has more than 1600 hotels (mostly of 3 and 4 star) who act as hotel rooms suppliers. Its customers are individual customers, travel agent and tour operators.

As discussed in [14], in our previous research we designed Supplier Relationship Managements (SRM) for Klikhotel and have implemented many of the SRM features used by the hotels. Few of the features include self-service hotel information management functions used for updating hotel profile, rooms information, and creating/updating campaigns or promotions. (Among those hotels, however, many have been integrated with an international hotel management system, which is also connected with Klikhotel system. Thus they do not use the SRM provide by Klikhotel.) As Klikhotel has been storing hotels transactions in its database, we can create and implement the data mart as depicted in Figure 3.

For this experiment, from the data mart we queried 6 month of weekly time series data (period of July to December 2013) that include of 1229 hotels. Among them, the total hotels being analyzed are 793 hotels as the rests do not use Klikhotel's SRM. Each hotel activity frequency is represented by a bitmap consisting of 24 bits, where each bit represents the activity status of a week. After the bitmaps are processed with the proposed algorithms, the results are shown in Table 4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Clustering Results Of Supplier Hotels.

	Predict-Churn-1 Output	*Predict-Churn-2* Output
Hotels with cL = 1	108	16
Hotels with cL = 2	173	28
Hotels with cL = 3	332	336
Total	613	380
Hotels with fine activity	180	413
TR1	0.74 sec	0.8 sec
TR2	0.24 sec	0.32 sec

cL : churn level
TR1: time response including reading input file
TR2: time response without reading input file

As shown in Table 4, the output of algorithm PredictChurn-1 differ to PredictChurn-2, where the later predicts less hotels needing "treatments" (hotels with cL=2 and cL=3). However, both algorithms predicts that more than 330 hotels have low activity (churn level is 3), which should alert the KlikHotel supplier care division. These hotels can then be fed into an SRM module that generates email alert or other SRM module that predict the behavior of churning based on monetary variable. (The Klikhotel management has acknowledged that the results show consistency with their knowledge of the hotel members' activity.) In the table, it is also shown that it takes less than 1 second to process the 1229 bitmaps data, which is very fast.

5. CONCLUSION

The suppliers' online transaction logs data can be preprocessed into time series data that is then analyzed with efficient clustering algorithms to predict suppliers churn level tendency based on their frequency in accessing website. To perform such tasks, we propose efficient techniques that involve data mart design with its transform function needed to preprocessed raw log data into time series format, and two churn tendency clustering algorithms that process the time series bitmaps. Having time complexity of $O(n)$, the algorithms are proved to be very efficient.

Our proposed techniques are used to predict churn level tendency based on suppliers frequency in accessing a website only. For further works: Techniques processing other data, such as monetary and other necessary data that can be used to predict churn tendency, can be developed. These techniques are then combined with our proposed techniques to obtain better (more accurate) prediction techniques. Other future works: In the case that training and testing data set for classification models (bitmap time series data having a class attribute representing the suppliers churn level) can be obtained or generated, classification models used to predict suppliers churn tendency can be designed and then trained and tested with the data set.

ACKNOWLEDGMENT

We like to thank to the Directorate General of Higher Education, Ministry of Research and Technology and Higher Education, Indonesia who has funded this research in 2015 through Hibah Bersaing scheme.

REFERENCES

[1] K.C. Laudon and C.G. Traver, *E-commerce Business Technology Society 6th ed.*, Pearson Education, Inc., England, 2010.
[2] Mary E. Chenoweth, Nancy Y. Moore, Amy G. Cox, Judith D. Mele, Jerry M. Sollinger, *Best Practices in Supplier Relationship Management and Their Early Implementation in the Air Force Materiel Command*, Prepared for the United States Air Force, Project Air Force, RAND Corporation, 2012.
[3] John Hadden, *A Customer Profiling Methodology for Churn Prediction*, Disertation, School of Applied Sciences, Manufacturing Department, Cranfield University, UK, 2008.
[4] K. Tsiptsis and A. Chorianopoulos, *Data Mining Techniques in CRM: Inside Customer Segmentation*, John Wiley and Sons, Ltd., Pub., United Kingdom, 2009.
[5] Gordon S. Linoff, *Data Analysis Using SQL and Excel*, Wiley Publishing, Inc., Canada, 2008.
[6] Bing Liu, *Web DataMining: Exploring Hyperlinks, Contents, and Usage Data*, Springer-Verlag Berlin Heidelberg 2007.
[7] Silberschatz, Korth & Sudarshan, *Database System Concepts*, 5th Ed, McGraw-Hill, USA, 2011.
[8] A. Kaushik, *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity*, Wiley Publishing, USA, 2010.
[9] R.Kousalya & V. Saravanan, "Personalizing User Directories Through Navigational

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Behavior of Interesting Groups and Achieving Mining Tasks", *Journal of Theoretical and Applied Information Technology*, Vol. 67 No.2, 2014.

[10] M. Golfarelli & S. Rizzi, *Data Warehouse Design: Modern Principles and Methodologies*, McGraw Hill, 2009.

[11] M. Setak, Samaneh S. and A. Alimohammadian, "Supplier Selection and Order Allocation Models in Supply Chain Management: A Review", *World Applied Sciences Journal*, Vol. 18, No. 1, pp. 55-72, 2012.

[12] Kaastra, I. & Boyd, M., *Designing a Neural Network for Forecasting Financial and Economic Time Series*, Neurocomputing, Vol. 10, pp. 215-236, 1996.

[13] Veronica S. Moertini, Niko Ibrahim & Verliyantina, "e-CRM Development Method for e-Commerce System Owned by Small Medium Enterprises", *The 2014 2nd International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E 2014)*, Bandung, Indonesia, 19-21 August 2014.

[14] Niko Ibrahim, Veronica S. Moertini & Verliyantina, "Supplier Relationship Management Model For SME's E-Commerce Transaction Broker Case Study: Hotel Rooms Provider", *Journal of Theoretical and Applied Information Technology (JATIT)*, Vol.71 No.1, pp. 61-70, 2015.