

## Enabling efficient business process mining using flatten sequential structure model

Ang Jin Sheng<sup>1</sup>, Jastini Mohd Jamil<sup>1</sup>, Izwan Nizal Mohd Shaharane<sup>1</sup>, Mohamad Fadli Zolkipli<sup>2</sup>

<sup>1</sup>Department of Decision Science, School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, Malaysia

<sup>2</sup>School of Computing, Universiti Utara Malaysia, Sintok, Malaysia

### Article Info

#### Article history:

Received Sep 27, 2022

Revised Mar 15, 2023

Accepted Mar 24, 2023

#### Keywords:

Business process log data

Frequent subtree structure mining

Semi structured data

Statistical analysis

XML mining

### ABSTRACT

The volume of extensible markup language (XML) format documents is increasing every day due to the development of internet and the use of XML format in business process log file. Storing business process log data in XML format is preferable due to the ability of extensible and storing data irrespective of how it will be represented. However, mining XML format data poses challenges due to its complex data structure and dimensions. This paper proposes a method to convert XML format document into a structured format without ignoring the structural information. Converting semi-structured business process log data into structured format will allow more data mining techniques and statistical test be conducted and extract information from the business process log data. The experiment in this study performs t-test on a set of synthetic data and a set of real-world data to prove that information in business process log can be extracted through normal statistical test. Empirical results show that statistical analysis can be conducted on business process log data especially in XML format after flatten sequential structure model (FSSM) is used.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



### Corresponding Author:

Izwan Nizal Mohd Shaharane

Department of Decision Science, School of Quantitative Sciences, Universiti Utara Malaysia

06010 UUM Sintok, Kedah, Malaysia

Email: nizal@uum.edu.my

## 1. INTRODUCTION

Business process is a set of dependent activities with a structure within a company or business that following some logical order with a target to create desired result [1]. Business process management system (BPMS) arise as a solution to support different aspects of business processes between organization in the 21<sup>st</sup> century [2]. Thus, whenever a business process is executed, BPMS will produce a process log or event log [3]. These event log often also known as “audit trail”, “transaction log” or “history” of business process [4], [5]. Usually, business process event logs usually stored in extensible markup language (XML) format [6]. The first standard for event log is mining extensible markup language (MXML) began in 2003 and updated to extensible event stream (XES) in 2009. XES standard became IEEE standard in 2016.

Aalst *et al.* [6] suggest a different approach besides using interview techniques to obtain insights from business processes, which is analyse the event logs that produced by BPMS. However, it is hard to apply classical data mining techniques or statistical analysis to these event logs due to their semi-structure properties of XML format [7]. Accordingly, different methods and algorithms have been developed to extract information from these XML documents such as frequent subtree mining (FSM) [8]-[10]. FSM mainly looking for the patterns in a tree-structured database by using the support value determine by user. In other words, FSM is an association rule mining in tree structure database.

Nevertheless, there are a few drawbacks using FSM to get insights from business process log data. Firstly, FSM methods usually ignore or do not account the node positional information [11]. This will result in

some information loss during extracting information from business process log data. As a matter of fact, this positional information may be substantial in some application scenarios. Secondly, support value is the only measurement in FSM. It is difficult to look for novel or interesting patterns when the support value is set very low during huge number of rules generated [12]. Furthermore, most of the researchers are focusing on improving the performance of FSM including finding ways to reduce average run time and memory usage [7]. Thus, the results acquired from FSM whether meaningful or interesting is unknown. Statistical analysis is one of many ways to overcome the limitations of FSM. Statistical analysis such as chi square test and regressions can help to filter irrelevant variables to reduce the rules that are meaningless and uninteresting [13].

In this study, a model is proposed to offer statistical approach able to be applied in business process log data specifically in XML format. The performance of business process may differ due to some factor such as customer types, product types, geographically and time although the processes are identical. Usually, process variant analysis [14] is conducted to find out the difference between the business processes. However, statistical analysis such as t-test can be used as alternative to find that whether there is a difference in production from two same business process or two same machines in a production line during performing analysis on business process log data. Therefore, t-test is proposed in this study to determine whether there is a difference between the same process and the factor that influence the outcome of a business process. Two set of business process log data including simulated and real-life data are used in this study.

## 2. RELATED WORKS

In the era of big data, analytics are widely performed to obtain insights and knowledge to improve business process and allow better decision to be made. Therefore, analyzing event logs produce by BPMS is one of the important topics in nowadays. Due to some reasons, observing or analyzing whole population is difficult during practical. The same difficulties sometimes happen on event logs also especially when the population is in large scale. LogRank [15] is invented to sample large scale business process log data become smaller scale or size so that business process discovery can be performed easier. Gaaloul *et al.* [16] proposed applying statistical analysis on workflow log. The model statistical dependency tables (SDT) proposed by them can determine event dependencies by analyzing business process log data statistically. LogLens presented by Debnath *et al.* [17] can detect anomaly from event logs in real-time. On the other hand, log delta analysis proposed by Beest *et al.* [18] can identify behavioral difference between two business process log data.

Usually, business process log data are stored in XML format [3]. One of the reasons these business process log data store in this semi-structured format is the capability of XML format to represent the contextual information among different attribute or metadata in a domain unambiguous method. Nevertheless, it is quite challenging to perform statistical analysis and data mining technique to XML data because of the complex data structure and dimensions (structure dimension and content dimension) [19]. Due to the similar characteristic of XML document and trees structured data, many researchers modelled XML document as an ordered, labelled and rooted trees. Frequent subtree mining is the most generally used for analyzing XML format document. Different algorithms of FSM have been developed by different researchers such as [8]-[10], from the past to improve the performance of FSM and reduce the memory usage and total running time. However, the information extract from XML documents by using FSM is limited. Due to the minimum support set by user is the only measurement used in FSM, the result of FSM only limit to most common tree can be found in the dataset. Thus, common structure of XML can be determined. Besides FSM, there are some other mining algorithm to look for most frequent rule in XML format document such as pre-order linked WAP-tree mining (PLWAP) [20], combination based behavioral pattern mining (COBPAM) [21] and frequent pattern mining [22].

To overcome the downside of FSM [23], proposed database structure model (DSM) to flatten XML format data so that more data mining techniques can be perform on XML documents. Numerous researchers such as [3] and [24] perform clustering or classification on XML format document including business process log data. Their research prove that applying DSM on XML format data have better result compare to FSM. Shaharane *et al.* [13] and Shaharane and Jamil [25] suggest that performing correlation analysis can filter or reduce the unrelated variables in XML document to improve the interestingness of result after performing analytics or data mining. However, the weakness of DSM is assuming all tree structure in XML document are the same. Moreover, attributes are ignored when using DSM although the structural information are preserved.

## 3. PROPOSED FRAMEWORK AND MODEL

### 3.1. Mining business process log framework

To mine business process log data [26], framework is used in this study. Figure 1 illustrates the framework to mine business process log data. Firstly, raw business process log data in XML format is pre-processed. Data that is not related to the transactions and corrupted are filtered. Then, data is extracted and

converted into structured data using flatten sequential structure model (FSSM). Finally, statistical analysis such as t-test is applied to determine the difference between two groups of data in this study.

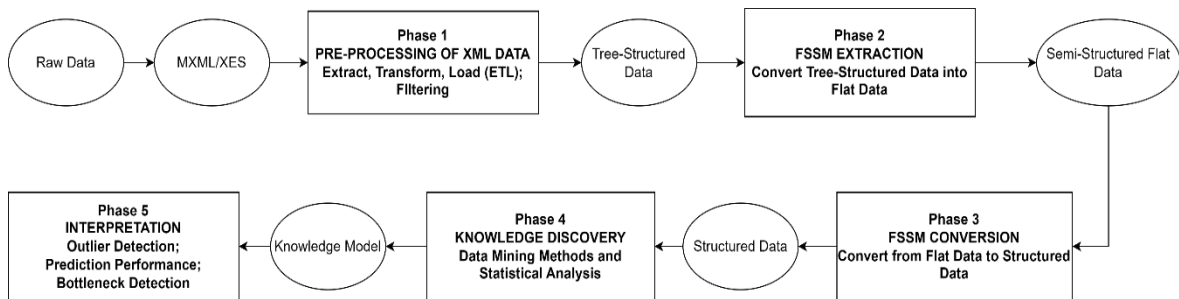


Figure 1. Mining business process log framework

**3.2. Flatten sequential structure model**

FSSM is proposed in this study to convert XML format data into structured data so that more data mining algorithm and statistical approach can be conducted to obtain insights from these semi-structured documents. FSSM is divide into two phases, extraction phase and conversion phase. The data structure and algorithm in each phase of FSSM are explained in detailly in this section. A synthetic tree database  $T_e$  is shown in Figure 2 for explanatory purpose. Two different structures of rooted ordered labelled trees labelled as  $t_0$  and  $t_1$  are illustrated in Figure 2 to show how tree structured format data is converted into structured format.

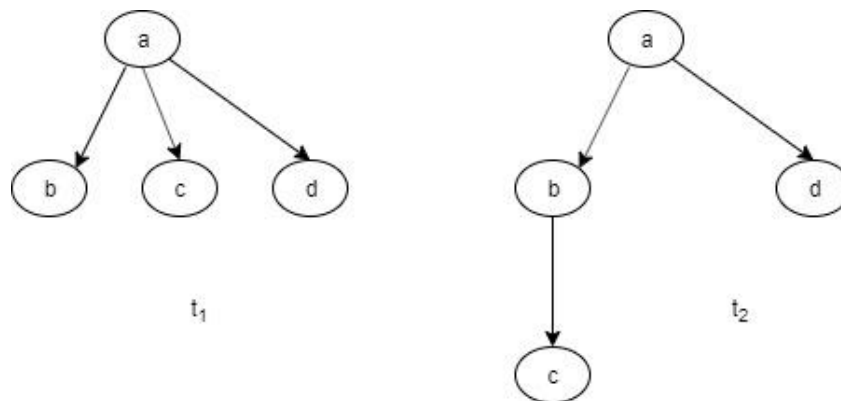


Figure 2. Tree-structured database  $T_e$

**3.3. Data structure in FSSM**

The first phase of FSSM is to record the structural properties of every instance in a tree database. Put differently, the structural information is preserved in FSSM extraction phase. Table 1 shows that the tree-structured database  $T_e$  is flatten and preserving the structural information at the same time. The sequence of the tree structured is viewed from top to bottom, then left to right. Node 'a' is the root of the tree  $t_1$  and  $t_2$ . Before proceeding to the next sibling, a backtrack to it's parent of the node is required. Therefore, '-1' in Table 1 means backtrack. Next, FSSM conversion phase is converting the flatten data from semi-structure format to structured format. Table 2 illustrates the flatten data in FSSM extraction phase is converted into structured format during FSSM conversion phase.

Table 1. FSSM extraction phase flatten data

$T_e$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$t_1$	a	b	-1	c	-1	d	-1
$t_2$	a	b	C	-1	-1	d	0

Table 2. FSSM conversion phase structured data

	a	b	c	d
$t_1$	$t_1a$	$t_1b$	$t_1c$	$t_1d$
$t_2$	$t_2a$	$t_2b$	$t_2c$	$t_2d$

### 3.4. FSSM algorithms

Algorithm 1 shows the pseudocode for finding maximum number of variables among transactions or subtrees. Firstly, the longest chain or the longest tree must be determined before FSSM starts. Thus, the table for first phase can be drafted using the maximum length and variables of the longest tree.

#### Algorithm 1. Finding maximum number of variables

**Input:** XML format dataset

**Output:** Longest Node in dataset

```

1: Define variable:
   Let Maximum = First Node
   Let Maximum Count = 0
   Let Maximum Node = First Transaction
   Let Maximum Variable = empty list
2: for (each transaction in a tree)
3:   let node level = 1 and variable = empty
4:   while (elements not equal 0)
5:     if (node have child)
6:       elements --
7:       node level ++
8:       variable = variable + element
9:     if (node do not have child)
10:      node level --
11:      if (node level equal 0)
12:        exit loop
13:     else
14:       variable = variable + backtrack
15:     count ++
16:   if (count > Maximum Count)
17:     Maximum Count = count
18:     Maximum Variable = variable
19:     Maximum Node = transaction

```

Algorithm 2 illustrates FSSM phase 1, FSSM extraction phase. XML data is extracted and flattened through this phase. The data structure or result of FSSM extraction phase is shown in Table 1. Algorithm 3 shows the pseudocode for FSSM phase 2, FSSM conversion. The flatten data is converted to structured format using FSSM conversion phase. Lastly, the example of outcome of FSSM extraction phase is illustrated in Table 2.

#### Algorithm 2. FSSM extraction (Phase 1)

**Input:** XML format dataset

**Output:** Flatten data

```

1: Define variable:
   Let total column of FSSM table is the length of Maximum Node from algorithm 1
   Let FSSM table = empty
   Let structure table variable = empty
2: for (each transaction in a tree)
3:   let node level = 1, column number = 1, FSSM row = empty
4:   while (elements not equal 0)
5:     if (node have child)
6:       elements --
7:       node level ++
8:       column number of FSSM row ++
9:       if (node contains attribute)
10:        add node name and attribute into FSSM
11:        add attribute name into structure table variable
12:       else
13:        add node name
14:       if (node do not have child)
15:        node level --
16:        if (node level equal 0)
17:          exit loop
18:       else
19:        column number of the FSSM row ++
20:        add '-1' or 'b' into FSSM
21:   while (column number of current transaction < length of total column in FSSM
table)
22:     column number of FSSM row ++
23:     add 0 into the list of FSSM row
24:   add FSSM row into FSSM table

```

**Algorithm 3. FSSM conversion (Phase 2)****Input:** Flatten data from FSSM extraction**Output:** Structured data

```

1: Filter only unique attribute name from structure table variable list from algorithm 2.
2: Put the unique attribute name into columns of FSSM structured table.
3: for (each transaction in FSSM flatten table)
4:     let node level = 0 and structure table = empty
5:     for (each variable in FSSM flatten table)
6:         if (FSSM table variable value == 'b' or FSSM table variable value == '1')
7:             node level --
8:         else
9:             if (FSSM table variable value not equal '0')
10:                node level ++
11:            if (node level > 1)
12:                if (FSSM table variable contain attributes)
13:                    put the attribute value according to the attribute
14:                    / column name in the structure table variable
15:            if (structure table contains any value)
16:                if (node level equal 1)
17:                    put 0 into all attributes / columns which does not
18:                    contain any value
19:            add transaction number into structure table
20:            add structure table into the complete table
21:            empty the structure table

```

**4. PROCEDURE****4.1. Procedure employed**

Brief description procedure in this study to perform t-test on XML format business process event logs are given:

- Firstly, raw data in XML format is cleaned and filtered. Only data with transaction information are remained and used in this study.
- Then, tree structured data is flattened through FSSM extraction phase.
- FSSM conversion phase converts the flatten data into structured data.
- t-test is conducted on the converted structured data. Null hypothesis will be rejected if the p-value is less than 0.05. Else, the null hypothesis failed to be rejected.

**4.2. Hypothesis testing**

In this study, two independent sample t-test is used for hypothesis testing. In the BPIC 2017 dataset, the null hypothesis of t-test is set as there is no differences of the amount requested between the applications is accepted and rejected. On the other hand, null hypothesis of simulated data is set as there is no differences between activity C and activity D.

Steps of using t-test is shown as:

- a) Null hypothesis is set as:
  - $H_0: \mu_1 = \mu_2$
  - $H_1: \mu_1 \neq \mu_2$
- b) Use the (1) to calculate the t-statistic where  $\bar{x}_1$ ,  $\bar{x}_2$  are the sample 1's mean and sample 2's mean of respectively;  $n_1$  and  $n_2$  are the sample size of group 1 and group 2 respectively;  $s_1$  and  $s_2$  are the standard deviation of group 1 and group 2 respectively.
- c) Compute the p-value by comparing the t-statistic with t-distribution.
- d) If the p-value is less than 0.05, reject the null hypothesis.

$$t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (1)$$

**4.3. Summary of the data**

There are two sets of data used in this study. Real life and simulated business process log data. These datasets are described in detail as:

**4.3.1. Simulated data**

The simulated dataset is simulated using processes and logs generator 2 (PLG2) developed by [27]. PLG2 is an application to generate random business processes and its event logs. 200 transactions of data are generated randomly based on the business process illustrated in Figure 3. Activity C generates a value between 500-1,200 whereas activity D generates a value between 800-1,200 randomly.

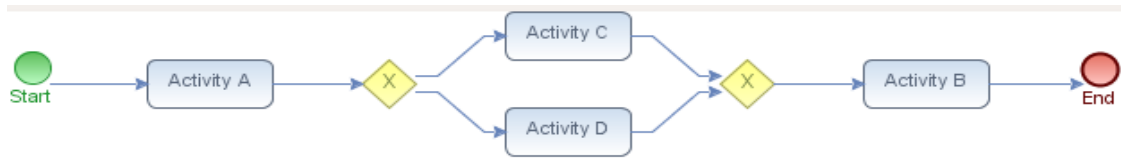


Figure 3. Business process simulated in PLG2

**4.3.2. Real life data**

Real life event log data is provided by the business process intelligence challenge (BPIC) 2017. There are two event logs provided in this challenge including application log and offer log [28]. However, only application event log is used in this study. The event logs contains 26 types of events that can be divided into 3 categories, which are application state changes, offer state changes and workflow events [29]. There are around 31508 of transactions in the document provided by BPIC 2017. However, the first 200 transactions are used in this study. The process flow for BPIC 2017 dataset [30] is shown in Figure 4.

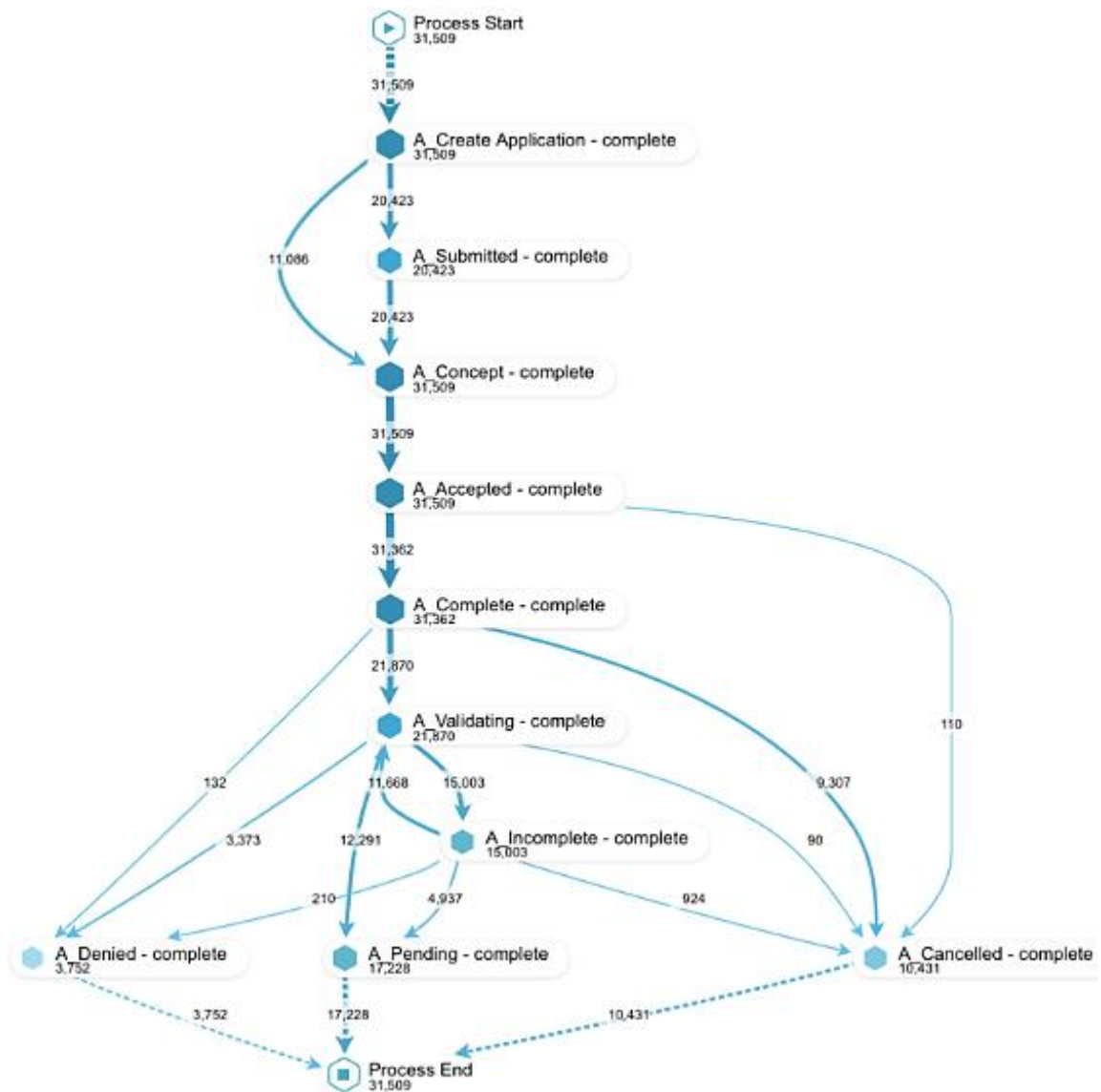


Figure 4. Business process of BPIC 2017 dataset

5. RESULTS AND DISCUSSION

Firstly, the dataset is converted to flat data and then structured data before t-test conducted. The dataset is cleaned and converted using R version 4.05. Then, the data is export to csv format and import into SPSS to conduct t-test. SPSS version 26 is used in this study. Figure 5 shows the screenshot of flatten data of simulation dataset during FSSM extraction.

	A	B	C	D	E	F	G	H	I	J
1	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
2	trace	concept:name-case_67	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
3	trace	concept:name-case_159	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
4	trace	concept:name-case_68	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
5	trace	concept:name-case_181	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
6	trace	concept:name-case_185	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
7	trace	concept:name-case_168	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
8	trace	concept:name-case_11	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
9	trace	concept:name-case_74	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
10	trace	concept:name-case_60	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
11	trace	concept:name-case_147	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
12	trace	concept:name-case_1	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
13	trace	concept:name-case_165	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
14	trace	concept:name-case_71	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
15	trace	concept:name-case_13	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
16	trace	concept:name-case_176	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
17	trace	concept:name-case_78	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
18	trace	concept:name-case_17	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
19	trace	concept:name-case_85	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event
20	trace	concept:name-case_103	b	event	concept:name-Activity A	b	time:timestamp-1970-01-01T07:30:00+07:30	b	b	event

Figure 5. Screenshot of flatten data for simulation dataset

Figure 6 shows the screenshot of structured data for simulation dataset during FSSM conversion phase. Figure 7 shows the screenshot of simulation dataset after filtering activity C and activity D. The name of activity C and activity D change to 1 and 2 respectively.

	A	B	C	D
1	Transaction	concept:name	time:timestamp	number_of_production
2	T7	case_67		0
3	T7	Activity A	1970-01-01T07:30:00+07:30	0
4	T7	Activity C	1970-01-01T08:30:00+07:30	853
5	T7	Activity B	1970-01-01T09:30:00+07:30	0
6	T8	case_159		0
7	T8	Activity A	1970-01-01T07:30:00+07:30	0
8	T8	Activity C	1970-01-01T08:30:00+07:30	601
9	T8	Activity B	1970-01-01T09:30:00+07:30	0
10	T9	case_68		0
11	T9	Activity A	1970-01-01T07:30:00+07:30	0
12	T9	Activity D	1970-01-01T08:30:00+07:30	998
13	T9	Activity B	1970-01-01T09:30:00+07:30	0
14	T10	case_181		0
15	T10	Activity A	1970-01-01T07:30:00+07:30	0
16	T10	Activity D	1970-01-01T08:30:00+07:30	1134
17	T10	Activity B	1970-01-01T09:30:00+07:30	0
18	T11	case_185		0
19	T11	Activity A	1970-01-01T07:30:00+07:30	0
20	T11	Activity D	1970-01-01T08:30:00+07:30	872

Figure 6. Structured data for simulation dataset

	A	B	C	D
1	Transactions	concept:name	time:timestamp	number_of_production
2	1		1 1970-01-01T08:30:00+07:30	853
3	2		1 1970-01-01T08:30:00+07:30	601
4	3		2 1970-01-01T08:30:00+07:30	998
5	4		2 1970-01-01T08:30:00+07:30	1134
6	5		2 1970-01-01T08:30:00+07:30	872
7	6		1 1970-01-01T08:30:00+07:30	627
8	7		1 1970-01-01T08:30:00+07:30	527
9	8		1 1970-01-01T08:30:00+07:30	784
10	9		2 1970-01-01T08:30:00+07:30	1179
11	10		1 1970-01-01T08:30:00+07:30	736
12	11		2 1970-01-01T08:30:00+07:30	1006
13	12		2 1970-01-01T08:30:00+07:30	939
14	13		1 1970-01-01T08:30:00+07:30	554
15	14		2 1970-01-01T08:30:00+07:30	977
16	15		1 1970-01-01T08:30:00+07:30	533
17	16		1 1970-01-01T08:30:00+07:30	730
18	17		1 1970-01-01T08:30:00+07:30	802
19	18		1 1970-01-01T08:30:00+07:30	916
20	19		2 1970-01-01T08:30:00+07:30	976

Figure 7. Filtered data for simulation dataset

Figure 8 illustrates the screenshot of flatten data for BPIC 2017 dataset during FSSM extraction phase. Figure 9 illustrates the screenshot of structured data for BPIC 2017 dataset during FSSM conversion phase. Figure 10 illustrates the screenshot of BPIC 2017 dataset after filtering transaction, loan goal, application type, request amount of the loan and accepted or not for each transaction. The accepted result are true or false is converted to 1 or 0 respectively.

x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
trace	LoanGoal-Existing loan takeover	b	ApplicationType-New credit	b	concept:name-Application_652823628	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Home improvement	b	ApplicationType-New credit	b	concept:name-Application_1691306052	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Home improvement	b	ApplicationType-New credit	b	concept:name-Application_428409768	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Car	b	ApplicationType-New credit	b	concept:name-Application_1746793196	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Home improvement	b	ApplicationType-New credit	b	concept:name-Application_828200680	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Existing loan takeover	b	ApplicationType-New credit	b	concept:name-Application_1085880569	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Existing loan takeover	b	ApplicationType-New credit	b	concept:name-Application_1266995739	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Home improvement	b	ApplicationType-New credit	b	concept:name-Application_1878239836	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Car	b	ApplicationType-New credit	b	concept:name-Application_619403287	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Car	b	ApplicationType-New credit	b	concept:name-Application_1710223761	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Other, see explanation	b	ApplicationType-New credit	b	concept:name-Application_1529124572	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Other, see explanation	b	ApplicationType-New credit	b	concept:name-Application_387012864	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Other	b	ApplicationType-New credit	b	concept:name-Application_1120819670	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Existing loan takeover	b	ApplicationType-New credit	b	concept:name-Application_42838382	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Home improvement	b	ApplicationType-New credit	b	concept:name-Application_180547487	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Remaining debt home	b	ApplicationType-New credit	b	concept:name-Application_1966208034	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Car	b	ApplicationType-New credit	b	concept:name-Application_1806387393	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Not specified	b	ApplicationType-New credit	b	concept:name-Application_1111870538	b	Requeste	b	event	Action-Created	b	org:resource-User_1
trace	LoanGoal-Car	b	ApplicationType-New credit	b	concept:name-Application_1017492916	b	Requeste	b	event	Action-Created	b	org:resource-User_1

Figure 8. Flatten data for BPIC 2017 dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Transaction	LoanGoal	Applicatio	concept:n	Requeste	Action	org:resou	EventOrig	EventID	lifecycle:t	time:time	FirstWith	NumberO	Accepted	MonthlyC	Selected	CreditSco	OfferedAl
2	T2	Existing Ic	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	T2	0 New credi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	T2	0	0 Applicatio	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	T2	0	0	0	20000	0	0	0	0	0	0	0	0	0	0	0	0	0
6	T2	0	0 A_Create	0	Created	User_1	Applicatio	Applicatio	complete	2016-01-0	0	0	0	0	0	0	0	0
7	T2	0	0 A_Submit	0	statechan	User_1	Applicatio	AppIState	complete	2016-01-0	0	0	0	0	0	0	0	0
8	T2	0	0 W_Handle	0	Created	User_1	Workflow	Workitem	schedule	2016-01-0	0	0	0	0	0	0	0	0
9	T2	0	0 W_Handle	0	Deleted	User_1	Workflow	Workitem	withdraw	2016-01-0	0	0	0	0	0	0	0	0
10	T2	0	0 W_Compl	0	Created	User_1	Workflow	Workitem	schedule	2016-01-0	0	0	0	0	0	0	0	0
11	T2	0	0 A_Concep	0	statechan	User_1	Applicatio	AppIState	complete	2016-01-0	0	0	0	0	0	0	0	0
12	T2	0	0 W_Compl	0	Obtained	User_17	Workflow	Workitem	start	2016-01-0	0	0	0	0	0	0	0	0
13	T2	0	0 W_Compl	0	Released	User_17	Workflow	Workitem	suspend	2016-01-0	0	0	0	0	0	0	0	0
14	T2	0	0 A_Accepte	0	statechan	User_52	Applicatio	AppIState	complete	2016-01-0	0	0	0	0	0	0	0	0
15	T2	0	0 O_Create	0	Created	User_52	Offer	Offer_148	complete	2016-01-0	20000	44	TRUE	498.29	TRUE	979	20000	
16	T2	0	0 O_Created	0	statechan	User_52	Offer	OfferState	complete	2016-01-0	0	0	0	0	0	0	0	0
17	T2	0	0 O_Sent (n	0	statechan	User_52	Offer	OfferState	complete	2016-01-0	0	0	0	0	0	0	0	0
18	T2	0	0 W_Compl	0	Deleted	User_52	Workflow	Workitem	ate_abort	2016-01-0	0	0	0	0	0	0	0	0
19	T2	0	0 W_Call aff	0	Created	User_52	Workflow	Workitem	schedule	2016-01-0	0	0	0	0	0	0	0	0
20	T2	0	0 W_Call aff	0	Obtained	User_52	Workflow	Workitem	start	2016-01-0	0	0	0	0	0	0	0	0

Figure 9. Structured data for BPIC 2017 dataset

	A	B	C	D	E
1	Transaction	loan_goal	application_type	request_a	accepted
2		2 Existing loan takeover	New credit	20000	1
3		3 Home improvement	New credit	10000	0
4		4 Home improvement	New credit	15000	1
5		5 Car	New credit	5000	0
6		6 Home improvement	New credit	35000	1
7		7 Existing loan takeover	New credit	13000	1
8		8 Existing loan takeover	New credit	7000	0
9		9 Home improvement	New credit	15000	1
10		10 Car	New credit	15000	1
11		11 Car	New credit	11000	1
12		12 Other, see explanation	New credit	5000	0
13		13 Other, see explanation	New credit	5000	1
14		14 Car	New credit	6850	1
15		15 Existing loan takeover	New credit	29500	1
16		16 Home improvement	New credit	6000	1
17		17 Remaining debt home	New credit	40000	1
18		18 Car	New credit	5000	0
19		19 Not specified	New credit	5000	1
20		20 Car	New credit	15000	1

Figure 10. Filtered data for BPIC 2017 dataset



Tables 3 and 4 summarizes the summary of the simulation and real-life dataset respectively after filtering the data required for t-test. The minimum value for production in simulation data is 501 and maximum value is 1196. On the other hand, the value for request amount for BPIC 2017 dataset is between 5000 and 50000. The mean for simulation dataset and BPIC 2017 dataset are 856.32 and 16359 respectively.

The result of t-test is summarized in Table 5. There is a difference between two production values in simulation data as the p-value is less than 0.05. However, the p-value for BPIC 2017 dataset is higher than 0.05. Thus, there is no difference between whether the application is accepted or rejected for the requested amount in BPIC dataset. By conducting t-test on business process log, difference between two process or outcome can be determined. Therefore, performing statistical test on business process log can extract more information such as finding out the difference between processes, relationships between variables compare to FSM. The limitation of FSM finding frequent pattern of subtree only based on support set by user can be overcome by statistical test to get more knowledge from business process log data.

Table 3. Descriptive statistics for simulation data

Parameter	N	Minimum	Maximum	Mean	Std. Deviation
number_of_production	200	501	1196	856.32	197.708

Table 4. Descriptive statistics for BPIC 2017 data

Parameter	N	Minimum	Maximum	Mean	Std. Deviation
request_amount	200	5000	50000	16359.00	10827.713

Table 5. Results of t-test after using FSSM

Dataset	Null hypothesis	p-value	Decision
Simulation data	H <sub>0</sub> : There is no difference of production value between activity C and activity D. H <sub>1</sub> : There is a difference of production value between activity C and activity D.	0.00	Reject Null hypothesis
BPIC 2017	H <sub>0</sub> : There is no difference of the amount requested between the applications is accepted and rejected. H <sub>1</sub> : There is a difference of the amount requested between the applications is accepted and rejected.	0.057	Failed to reject null hypothesis

## 6. CONCLUSIONS AND FUTURE WORKS

Extracting information from business process log especially in XML format usually done using traditional process mining or frequent structure mining. FSM usually can mine information such as frequent patterns or subtrees in business process log data. However, the interestingness of result and more information such as difference between processes or relationship between variables cannot be determined using FSM. As the business process getting more complex and increasing in numbers, this paper introduces a model that enables wider range of application in data mining or statistical analysis conducted in tree structured business process logs. Two experiments including simulation data and real-life data are done to show the promising capabilities of proposed method. Data mining techniques such as classifications algorithm or more statistical test can be explored using the proposed framework and model in the future research. For example, relationship test such as Pearson correlation test can be done to reduce the variables before doing classification in business process log data.

## ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support provided by the Fundamental Research Grant Scheme with (FR.GS/1/2018/ICT02/UUM/02/5), Universiti Utara Malaysia, Sintok, Kedah, Malaysia, for conducting this research.




## REFERENCES

- [1] J. Kang, Z. Diao, and M. T. Zanini, "Business-to-business marketing responses to COVID-19 crisis: a business process perspective," *Marketing Intelligence and Planning*, vol. 39, no. 3, pp. 454-468, 2021, doi: 10.1108/MIP-05-2020-0217.
- [2] Y. Alotaibi, "Business process modelling challenges and solutions: a literature review," *Journal of Intelligent Manufacturing*, vol. 27, no. 4, pp. 701-723, 2016, doi: 10.1007/s10845-014-0917-4.
- [3] D. B. Bui, F. Hadzic, and V. Potdar, "A framework for application of tree-structured data mining to process log analysis," in *International Conference on Intelligent Data Engineering and Automated Learning*, 2012, pp. 423-434. doi: 10.1007/978-3-642-32639-4-52.




- [4] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining process models from workflow logs," in *International Conference on Extending Database Technology*, 1998, pp. 467-483, doi: 10.1007/BFb0101003.
- [5] M. Sayal, F. Casati, U. Dayal, and M.-C. Shan, "Business process cockpit," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, 2002, pp. 880-883 doi: 10.1016/B978-155860869-6/50086-X.
- [6] W. M. V. Aalst, B. F. V. Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. Weijters, "Workflow mining: a survey of issues and approaches," *Data and Knowledge Engineering*, vol. 47, no. 2, pp. 237-267, 2003, doi: 10.1016/S0169-023X(03)00066-1.
- [7] K. Abe, S. Kawasoe, T. Asai, H. Arimura, and S. Arikawa, "Optimized substructure discovery for semi-structured data," in *European Conference on Principles of Data Mining and Knowledge Discovery*, 2002, pp. 1-14, doi: 10.1007/3-540-45681-3-1.
- [8] M. J. Zaki, "Efficiently mining frequent trees in a forest: algorithms and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1021-1035, 2005, doi: 10.1109/TKDE.2005.125.
- [9] S. Tatikonda, S. Parthasarathy, and T. Kurc, "TRIPS and TIDES: new algorithms for tree mining," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 2006, pp. 455-464, doi: 10.1145/1183614.1183680.
- [10] T. Asai, K. Abe, S. Kawasoe, H. Sakamoto, H. Arimura, and S. Arikawa, "Efficient substructure discovery from large semi-structured data," *IEICE TRANSACTIONS on Information and Systems*, vol. 87, no. 12, pp. 2754-2763, 2004.
- [11] F. Hadzic, M. Hecker, and A. Tagarelli, "Ordered subtree mining via transactional mapping using a structure-preserving tree database schema," *Information Sciences*, vol. 310, pp. 97-117, 2015, doi: 10.1016/j.ins.2015.03.015.
- [12] R. R. Rao and K. Makkithaya, "Identifying risk patterns in public health data through association rules," *Journal of Biomedical Engineering Society of India*, pp. 30-34, 2016.
- [13] I. N. M. Shaharane, F. Hadzic, and T. S. Dillon, "Interestingness of association rules using symmetrical tau and logistic regression," in *Australasian Joint Conference on Artificial Intelligence*, 2009, pp. 422-431, doi: 10.1007/978-3-642-10439-8-43.
- [14] A. Bolt, M. de Leoni, and W. M. V. der Aalst, "Process variant comparison: using event logs to detect differences in behavior and business rules," *Information Systems*, vol. 74, pp. 53-66, 2018, doi: 10.1016/j.is.2017.12.006.
- [15] C. Liu, Y. Pei, Q. Zeng, and H. Duan, "LogRank: an approach to sample business process event log for efficient discovery," in *International Conference on Knowledge Science, Engineering and Management*, 2018, pp. 415-425, doi: 10.1007/978-3-319-99365-2\_36.
- [16] W. Gaaloul, K. Gaaloul, S. Bhiri, A. Haller, and M. Hauswirth, "Log-based transactional workflow mining," *Distributed and Parallel Databases*, vol. 25, no. 3, pp. 193-240, 2009, doi: 10.1007/s10619-009-7040-0.
- [17] B. Debnath et al., "Loglens: a real-time log analysis system," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1052-1062, doi: 10.1109/ICDCS.2018.00105.
- [18] N. R. V. Beest, M. Dumas, L. García-Bañuelos, and M. L. Rosa, "Log delta analysis: Interpretable differencing of business process event logs," in *International Conference on Business Process Management*, 2016, pp. 386-405, doi: 10.1007/978-3-319-23063-4-26.
- [19] L. Candillier, L. Denoyer, P. Gallinari, M. C. Rousset, A. Termier, and A.-M. Vercoustre, "Mining XML documents," in *Data Mining Patterns: New Methods and Applications*, 2008, pp. 198-219, doi: 10.4018/978-1-59904-162-9.ch009.
- [20] C. I. Ezeife and Y. Lu, "Mining web log sequential patterns with position coded pre-order linked wap-tree," *Data Mining and Knowledge Discovery*, vol. 10, no. 1, pp. 5-38, 2005, doi:10.1007/s10618-005-0248-3.
- [21] M. Acheli, D. Grigori, and M. Weidlich, "Efficient discovery of compact maximal behavioral patterns from event logs," in *International Conference on Advanced Information Systems Engineering*, 2019, pp. 579-594, doi: 10.1007/978-3-030-21290-2-36.
- [22] C. C. Aggarwal, "Applications of frequent pattern mining," in *Frequent pattern mining*, 2014, pp. 443-467, doi: 10.1007/978-3-319-07821-2\_18.
- [23] F. Hadzic, "A structure preserving flat data format representation for tree-structured data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2011: Springer, pp. 221-233, doi: 10.1007/978-3-642-28320-8-19.
- [24] N. Ikasari and F. Hadzic, "An assessment on loan performance from combined quantitative and qualitative data in XML," in *International Conference on Discovery Science*, 2012, pp. 268-283, doi: 10.1007/978-3-642-33492-4\_22.
- [25] I. N. M. Shaharane and J. M. Jamil, "Irrelevant feature and rule removal for structural associative classification," *Journal of Information and Communication Technology*, vol. 14, no. 1, pp. 95-110, 2015, doi: 10.32890/jict2015.14.6.
- [26] J. S. Ang, "A framework to analyze business process log in XML format," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 3, pp. 2623-2630, 2021.
- [27] A. Burattin, "PLG2: multiperspective process randomization with online and offline simulations," in *BPM (Demos)*, 2016, pp. 1-6, doi: 10.48550/arXiv.1506.08415.
- [28] L. Blevi, L. Delporte, and J. Robbrecht, "Process mining on the loan application process of a Dutch Financial Institute," *BPI Challenge*, 2017.
- [29] D. Jeong, J. Lim, and Y. Bae, "BPIC 2017: business process mining—A Loan process application," *Seventh International Business Process Intelligence Challenge (BPIC'17)*, 2017.
- [30] F. Berger, "Mining event log data to improve a loan application process," *International Business Process Intelligence Challenge (BPIC'17)*, pp. 1-29, 2017, doi: 10.1007/s12599-020-00649-w.

## BIOGRAPHIES OF AUTHORS






**Ang Jin Sheng**    is a Ph.D. student at School of Quantitative Sciences, University Malaysia. He received his Master Studies in Federation University Australia in 2018. His research interests are data mining, artificial intelligence, and big data analysis. He can be contacted at email: [angjinsheng@gmail.com](mailto:angjinsheng@gmail.com).






**Dr. Jastini Mohd Jamil**    is a senior lecturer and researcher in Data Mining at School of Quantitative Sciences, Universiti Utara Malaysia. She received her Ph.D. in Data Mining from University of Bradford in 2012. She also holds a Master of Computer Sciences with focused on data mining, rough sets and neural networks from Universiti Teknologi Malaysia, and a Bachelor degree in Information Technology (networking) with honours from Universiti Utara Malaysia. Her research interests are solving problems in diverse area using data mining, decision support system and statistical techniques. Her other interests include structural equation modeling, partial least squares, neural networks, rough sets, data pre-processing, handling missing, data and forecasting. She can be contacted at email: jastini@uum.edu.my.



**Associate Prof. Dr. Izwan Nizal Mohd Shaharane**    is a lecturer and researcher at the Department of Decision Science, School of Quantitative Sciences, Universiti Utara Malaysia. He received his Ph.D. from Curtin University, Perth, Australia in 2012. His research areas mainly focus on data mining especially the quality issues, measures of interestingness, evaluation and application of data mining models. He also has a great interest in solving problems in diverse area using data mining, decision support system and statistical techniques. He can be contacted at email: nizal@uum.edu.my.



**Ts. Dr. Mohamad Fadli Zolkipli**    is an Associate Professor at the School of Computing, Universiti Utara Malaysia (UUM). He completed his doctorate degree in Computer Science at Universiti Sains Malaysia (USM) in 2012. His career in academia started when he joined KUKTEM/Universiti Malaysia Pahang (UMP) in July 2002 as academician. His teaching expertise includes data communication and networking, switching and routing and network security. His research interests cover the broad area of digital security. He has published numerous articles in the area of computer systems and networking especially in security domain such as intrusion detection systems, malware analysis and cloud security. As a part of research community, he also involves as a reviewer for conferences and journals. He is currently active in supervising research students of master and doctorate degrees. He can be contacted at email: m.fadli.zolkipli@uum.edu.my.