ISSN 2278-3091
Volume 8, No.3, May - June 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse88832019.pdf
https://doi.org/10.30534/ijatcse/2019/88832019

ABPMDF: Towards a Framework for Automated Model Discovery from Process Event Logs for Business Intelligence

M V Kamal[1], D Vasumathi[2]
[1]Research Scholar, India, kamalmv@gmail.com
[2]Professor, Dept of CSE, JNTUCEH, JNTUH, India, rochan44@gmail.com

ABSTRACT

Process mining has broad spectrum but an important area of interest is known as process discovery which mines process models from event logs of Information Technology (IT) systems. Such process models can provide required business intelligence with analysis and auditing. In the era of big data, process event logs can actually become source of big data. The recorded process logs of an IT system might reflect true dynamics of execution of processes. However, it may have hidden trends and also highly infrequent behaviours which are often known as outliers. The presence of outlier can be interpreted as noise or the trend with least support. When process models are automatically extracted from process event logs, it may result in very rarely travelled paths that pollute or clutter the associated process model. In the literature, many approaches are found on process mining. However, an efficient framework for automated model discovery from process event logs for business intelligence is still desired. In this paper, we proposed a framework known as Automated Business Process Model Discovery Framework (ABPMDF) is proposed with an underlying algorithm named Hybrid Approach for Business Process Log Filtering (HA-BPLF). The algorithm considers entropy with both ratios of directly-follows and directly-precedes and also Laplace smoothing for efficient discovery of process models and removal of highly infrequent behaviours. The framework is evaluated with large real time and synthetic process event logs. The proposed algorithm is found to have better performance over existing algorithms like Filter log using Simple Heuristics (SLF), Filter log using Prefix-Closed Language (PCL) and FilterLog.

Key words: Business process management, business process logs, process mining, model discovery

1. INTRODUCTION

Process mining is the task of discovering process models from business process event logs. The discovered models can help in further research and analysis which will provide valuable insights to enterprises in making strategic decisions. For instance, it is possible to predict the execution time of the processes to assess current performance and improve it further [1]. Broadly process mining is of three kinds known as discovery, conformance and enhancement. Discovering process models from event logs which will be actually a starting point for further analysis. There are many techniques to such as Alpha algorithm to mine process logs. Conformance on the other hand is to compare an existing process with its event logs to know really the recorded process log conforms to the process. In other words, conformance checking measures to know whether the model aligns with the reality. Enhancement is the act of analysing both process and its event logs and improve or extend an existing model [2].

There are many standards or laws that provide strict guidelines on process improvements. Some of them are related to management trends with respect to business intelligence and making well informed decisions. The standards include Corporate Performance Management (CPM), Continuous Process Improvement (CPI), Total Quality Management (TQM) and Six Sigma. There are legislations like Sarbanes-Oxley Act (SOX) 2002 and Basel II Accord 2004 to improve processes and avoid corporate and accounting scandals like WorldCom, Peregrine, Adelphia, Tyco and Enron [2].

Business processes are prone to changes over a period of time. Therefore, analysing processes and finding compliance with its own event logs is a continuous process. Another important research is to find outliers in the process event logs. In the contemporary business era, finding anomaly has its utilities in the real world. The outliers may indicate fraud or inefficiency in the process [11]. Therefore, it is essential to focus on such infrequent behaviours in process event logs. Moreover, there are many business process mining challenges. They include cleaning event data, dealing with complex event logs, generating representative benchmarks, handling of concept drift, dealing with representational bias, balancing quality criteria, mining cross-organizational processes, giving operational support, combining other kinds of analysis with process mining and ensuring usability and understandability for naïve users [21]. Business process simulation (BPS) [23] can be employed to understand the impact of these challenges.

From the literature it is understood that there are many techniques used for process discovery from business process event logs. Obtaining and replaying process models is explored in [2] for conformance checking. More techniques with their review can be found in Section 2. Conditional infrequent behaviours are discovered from event longs in [4]. An automated technique for removal of infrequent

behaviours from process logs is proposed in [26]. In both [4] and [26] that are close to the work in this paper directly-follows relationship is used to discover infrequent behaviours. However, it is not sufficient to produce more accurate results. To overcome this drawback, a comprehensive framework is proposed in this paper which considers both directly-follows and directly-precedes relations in order to improve performance of finding infrequent behaviours. Our contributions in this paper are as follows.

1. A framework named Automated Business Process Model Discovery Framework (ABPMDF) is proposed for more efficient processing of event logs and gain business intelligence.
2. An algorithm known as Hybrid Approach for Business Process Log Filtering (HA-BPLF) is proposed for discovering infrequent behaviours from process logs more efficiently.
3. A prototype is built to evaluate our framework and the underlying algorithm. The empirical results showed better performance over its predecessors.

The remainder of the paper is structured as follows. Section 2 provides review of literature on various aspects of process mining. Section 3 provides preliminaries that help in understanding important terms. Section 4 presents the proposed ABPMDF and describes how the framework performs mining on process event logs. Section 5 presents empirical results and evaluation. Section 6 concludes the paper and gives directions for future work.

2. RELATED WORK

Process mining is widely used to discover process models from event logs. This section reviews literature on prior works of process mining. Aalst, Schonenberg and Song [1] demonstrated that it is possible to discover process models from event logs and then the discovered models are used for prediction and analysis for improvements. The prediction of process times can help to improve services and leverage customer loyalty. They proposed a methodology to predict from process logs and generate business intelligence needed. They used ProM API for achieving this. However, automation of abstractions is not yet done in their method.

While present process models can reflect actual business processes, the historical processes can provide intelligence needed to comprehend on conformance with the current models. Towards this end, Aalst, Adriansyah and Dongen [2] proposed a framework for conformance checking by replaying history on process models. Their exploration of replay is yet incomplete and more comprehensive analysis is needed. Bose and Aalst [3] proposed a comprehensive framework for discovering signature patterns from event logs. Such predictions provide desirable and undesirable behaviours. Their framework is however generic in nature but needs to be improved to support different domains.

Mannhardt *et al*. [4] proposed a process discovery method known as Data-aware Heuristic Miner (DHM). It is data-driven in nature and able to discover infrequent behaviours in process event logs. It analyses control-flow as well. By discovering data and control flow it is possible to know the conditional infrequent behaviours. They explored directly-follows relations which provides insights but the other dependencies other than directly-follows relations. Diamantini *et al*. [5] employed hierarchical graph clustering to discover models from unstructured processes. However, the process discovery techniques employed are generic and needs to be evaluated with different domains.

Aalst *et al*. [6] proposed a novel two-step approach for process mining. It involves both regions and transition systems. The transition system is constructed in such a way that it could avoid over-fitting while the theory of regions is employed to synthesize models. ProM and Petrify are used for implementation. Theory of regions is yet to be tailored towards process mining. Usage or process cubes is another concept in process mining. Towards this end, Bolt and Aalst [7] proposed a formal model for process mining using process cubes. They called it as multi-dimensional process mining. Dimensions like location and organization are used for process cube operations. Besides classical Online Analytical Processing (OLAP) operations like slice and dice are used for further processing [27]. They used process cube view and materialized process cube view in order to analyse process models. They opined that their model does not support concept drift and comparison of cells.

Outliers are the abnormal behaviours in any data. In process mining also it stands true. Sani *et al*. [8] proposed a general purpose filtering method for discovering outliers. For this they employed conditional behavioural probabilities. They implemented Outlier Detection Algorithm for filtering out outliers from process event logs. While exploring processes, it is also important to understand deviations (not outliers but inconsistencies). Leemans *et al*. [9] studied four aspects and found that they are crucial for process mining. They are known as speed, semantics, evaluation and zoomability. Using these aspects, they studied academic workflows and commercial tools and combined to benefit from both the worlds. Provided feature comparison to find gap between them as well. Fitness deviations were found and their work was not integrated with Evolutionary Tree Miner to ascertain intuitive and interactive miner.

Bolt *et al*. [10] explored scientific workflows for the purpose of process mining. RapidMiner with RapidProM extension is used to discover process models from event logs. They also implemented methods for automatic report generation and visualization. They intended to improve the method further using standard workflows. Process mining supports both supervised and unsupervised models. In the former case, it needs training set while the latter does not need explicit training. Nolle *et al*. [11] proposed an unsupervised method for anomaly detection. They employed neural network technology in order to have better analysis of the logs with noise. From this it is understood that quality of process event logs is important for discovering process models and deriving business intelligence.

Nguyen *et al*. [12] explored the concept of auto encoders to improve quality of process logs. The auto encoders are a class of neural networks used to enhance quality of process logs. These neural networks do not need priori knowledge

in order to leverage quality. They opined that generative machine learning techniques cloud be used to improve the performance further. Neural networks have been improved to support deep learning techniques. Mehdiyev *et al*. [13] proposed a multi-stage deep learning method for process predictions. It has unsupervised pre-training component followed by supervised fine-tuning component. However, the understandability of prediction models is to be done further. Schonig *et al*. [14] proposed a process mining framework for overcoming drawbacks of process mining mechanisms. The drawbacks include discovering restricted set of processes, solutions are not efficient and the readability of discovered process models is low. They are yet to work on improving readability with proper visualization techniques.

Under-fitting and over-fitting are two issues related to process mining. Aalst *et al*. [15] proposed a two-step approach for balancing under-fitting and over-fitting. They found that classical process mining methods are not suitable for this kind of research. In their solution, the first step is to construct a transition system using a configurable approach. Afterwards, they used theory of regions in order to synthesize models. New theory of regions is to be investigated to improve the performance. When it comes to process mining, e-Commerce domain is rich in its processes. Poggi *et al*. [16] explored e-Commerce web logs for business process mining. They used Business Process Insight (BPI) platform for collaborative process intelligence. They found important improvements in obtaining BI for high level decision making. They observed that tools related to Business Process Management (BPM) can complement web analytics tools.

Log analysis in real time can improve the utility of it. Debnath *et al*. [17] proposed a real-time log analysis system known as LogLens. It is able to automate anomaly detection from log entries. It supports both stateless and stateful applications for log analysis. LogLens at present supports both visualization and process mining. The accuracy of the tool needs to be improved further when users use it for different domains. When it comes to event streams, detection of drifts is essential in order to identify unpredictable business processes. Ostovar *et al*. [18] proposed a solution to this which overcomes issues like unpredictability and drifts in the intra-traces. Their Drift Detection Algorithm is used to achieve it. However, it is yet desired to have better drift characterization for improving performance.

Evolutionary algorithms are widely used in different mining operations. Vazquez-Barreiros *et al*. [19] proposed a tool named ProDiGen based on Genetic Algorithm (GA). They evaluated it with noisy logs and found it suitable for process mining. They employed hierarchical fitness function for improving performance. Business processes, in the contemporary era, run in distributed environments. By studying them, it is possible to predict event based failures. Borkowski *et al*. [20] proposed an event based failure prediction model for mining distributed business processes. Online machine learning approach is followed to implement it. An algorithm is proposed to have bounded traversal and evaluate datasets. Identification of the modern event sources pertaining to IoT kind of technology is yet to be explored.

While analysing business processes, waste can be eliminated. Verenich *et al*. [22][28] proposed a fine-grained approach for process mining which eliminate wastage pertaining to over-processing. A machine learning approach known as predictive activity ordering is followed. Other types of waste such as defect waste is not yet considered. Apart from avoiding waste, it is essential to have cancellation or error handling features. Leemans and Aalst [24] proposed a process discovery technique along with cancellation features. However, automatic error handling is still desired in their work. Delias and Kazanidis [25] proposed many transformation templates and visualization techniques that can be reused by the people who involve in process mining. Very close to our work in this paper is found in [4] and [26]. However, there is an important drawback when they used only directly-follows relation and not considering directly-precedes relation. This is overcome in this paper besides providing a comprehensive framework for process mining.

3. PRELIMINARIES

In real world distributed applications there exists number of processes. Applications are made up of many processes and each process may have different sequence of services to be invoked. It is a complex phenomenon where the services may be provided by third parties and the runtime scenario is so dynamic. In such complex scenario, it is essential to maintain business process event logs. Business process is the process which is made up of number of services. Log is the set of entries recorded in a file where each entry shows an event and its details. For instance, user initiates authentication. It is an event considered and its details are logged with the date and time values as well. Such logs play vital role in process discovery and process improvement besides process auditing.

Infrequent behaviour in the log indicates that the behaviour is abnormal when compared with other instances. It is also known as noise or outlier. Filtering out such behaviours has utility in process mining. In this paper, directly-follows and directly-precedes are the two important terms used for discovering infrequent behaviours. The former is the “ratio of events of activity a and that are directly followed by an activity b in the given event log”. The latter on the other hand is “the ratio of events that are directly preceded by an activity b in the given event log”.

4. FRAMEWORK FOR DISCOVERING PROCESS MODEL AND FILTERING

We proposed a framework for discovering process model from business process event log (EL) and have an efficient means of filtering infrequent behaviours. Information Technology (IT) wings of various companies maintain business process logs. Such logs are essential for the purposes of auditing and analysis. The logs can have different activities and therefore can be converted later into event logs. The execution of specific process is recorded in the form of a trace reflecting sequence of events. When the

log entries are to be processed or analysed automatically, it is essential to have an automated approach.

4.1 Log and Automaton

First of all, an automaton is generated which is nothing but a direct graph where nodes are denoted as states that occurred in the log. There are two dependencies that can exist among the states. They are known as direct-follows and direct-precedes. An example log and its corresponding automaton adapted from [26] is as in Figure 1.

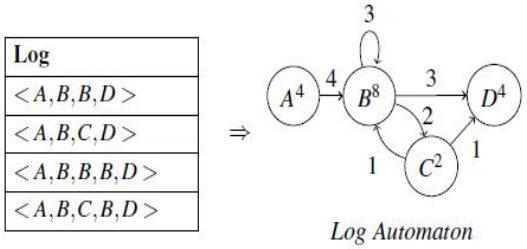

Figure 1: Sample log and its automaton

As presented in Figure 1, the log has entries. Each entry is a set of activities or events. For instance, the first log entry has events such as <A, B, B, D>. The log automaton shows a numeric value inside every state. It reflects the number of occurrences of the event in the log. On the arrow, a value is provided. It indicates number of times the two events occurred in that sequence. Any arc is considered to be infrequent when the frequency is less than a given threshold. Having understood the log and its automaton, now it is the time to understand the proposed framework for discovering processes and events besides pruning infrequent behaviours.

4.2 Proposed Framework

Extracting actionable knowledge, process knowledge in other words, from process event logs has significant utility in IT systems. In the contemporary era, in the wake of distributed computing technologies and realization of complex business processes in the real world, it is essential to have an automated process discovery approach that provides actionable knowledge or business intelligence. Existing algorithms captured behaviour found in the process event logs. However, there was trade-off between accuracy and the complexity of the derived process models [26]. In the wake of potential drawbacks in terms of completeness and the utility of BI from the process event logs, we proposed a framework that has mechanisms to capture business process models by mining process event logs. The proposed framework shown in Figure 2 exploits the usefulness of graphical notations like PetriNets to have process automaton that could be used for discovering infrequent behaviours that limit the advantages of discovered process models. The framework has an underlying algorithm. Unlike its predecessor [26], it makes use of both ratios of directly-follows and directly-precedes as part of the hybrid approach that makes use of entropy and Laplace smoothing. The proposed framework is named as Automated Business Process Model Discovery Framework (ABPMDF).

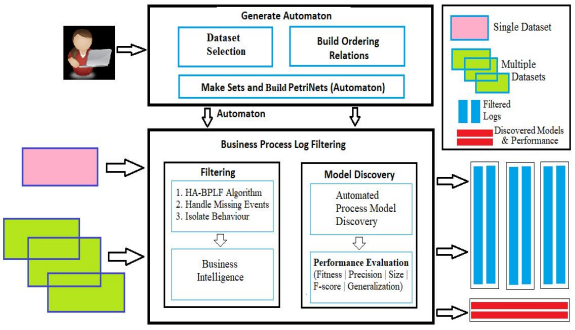

Figure 2: Overview of the proposed framework (ABPMDF)

The framework is realized with the implementation made using Python language and its process mining packages. It takes inputs in the form of raw datasets and also in the form of more refined forms such as automaton files that are generated using Graphviz tool (or package available in Python). If the input is the raw data set (real time or synthetic), there is a phase known as Generating Automaton. It takes selected dataset, build various ordering relations, make sets needed and finally generate PetriNets (visual representation of process log events). In the process, based on process event logs, Graphviz code is generated before producing PetriNets. Sample process event log and corresponding automaton are as follows.

Process Log
<A, C, E, G>
<A, E, C, G>
<B, D, F, G>
<B, F, D, G>

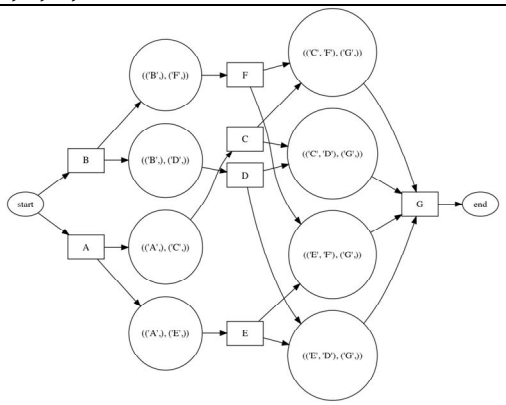

Figure 3: Automaton corresponding to the above process event log.

As shown in Figure 3, it is evident that there are different processes found in the process event log. Accordingly, the flow is automatically generated with an algorithm. The flow in the automaton can help in discovering processes after removing infrequent behaviours. After generation of automaton, business process log filtering takes place. It has both filtering which generates BI and then model discovery is carried out with the BI. Once models are realized, it is possible to subject the models to evaluation. The evaluation metrics used in this paper include fitness, precision, size, generalization and F-score. The result of the proposed framework is discovered business process models and

performance details. A hybrid approach of entropy and Laplace smoothing (see Section 4.1) are used to achieve this.

4.1 Combining Entropy and Laplace Smoothing

Entropy is a measure widely used in data mining and statistics. In data mining algorithms, entropy reflects homogeneity of a given sample. In this paper, entropy is used to identify infrequency of events in event logs. The entropy of event log EL is computed based on the two ratios of activities found in EL. They are known as ratio of directly-follows (rdf) and ratio of directly-precedes (rdp). Thus the categorical distributions of activities in EL can be denoted as rdf(a, EL) and rdp(a, EL) respectively. However, these become not reliable when the #(a, EL) is very small. In some hypothetical case, when #(a, EL) = 1, rdf(a, EL) assigns 1 to an activity reflecting the fact that a single activity in EL to have preceded by and contains probability of 0 with respect to all other activities. In the same fashion, when #(a, EL) = 1, the rdp(a, EL) assigns 1 to a specific activity and then assigns 0 to all other activities. Thus the #(a, EL) = 1 leads to a relation shown below.

F(rdf(a, EL)) = 0 and F(rdp(a, EL)) = 0

It shows that the usage of defined entropy is not reliable in case of #(a, EL) value is very small. In order to overcome this problem, we combined the entropy with a smoothing function known as Laplace Smoothing which is meant for empirical estimation of distribution of activities in terms of succeeding and preceding. For this reason, a smoothed version of the two ratios aforementioned are required. The rdf^s is defined as in Eq. 1.

$$rdf^s(a, b, L) = \frac{\alpha + \#((a,b),EL)}{\alpha(Activities(EL)+1) + \#(a,EL)} \quad (1)$$

Where the parameter for smoothing is denoted as $\alpha \in R \geq 0$. Therefore, the smoothed version of rdf (rdf^s) shows a value rdfs(a, b, EL). Its value lies between an empirical estimation of rdf(a, b, EL) and 1/|activities(EL)|+1 which reflects uniform probability based on the smoothing parameter. In the same fashion rdp is considered. In order to ensure filtering out activities from log EL in such a way that other activities become less chaotic. Towards this end, the concept of total entropy is considered. The entropy value of an event log EL is computed as the sum of entropies of all the activities found in the EL. This is provided in Eq. 2.

$$F(EL) = \sum_{a \in Activities(EL)} F(a, EL) \quad (2)$$

As the above entropy based filter is sensitive to infrequent activities, it is altered to have smoothing version of the same.

$$F^s(EL) = \sum_{a \in Activities(EL)} F^s(a, EL) \quad (3)$$

As shown in Eq. 3, the smoothed version of the filtering function is used for refined performance. This will lead to improvement in filtering process and it will be more useful in filtering out infrequent behaviours in business process event logs.

4.2 Hybrid Business Process Log Filtering

A hybrid approach is followed to filter business process logs in order to get rid of infrequent behaviours. This approach makes use of entropy and Laplace Smoothing in order to filter out log entries. An algorithm named Hybrid Approach for Business Process Log Filtering (HA-BPLF) is proposed.

Algorithm 1: Hybrid Approach for Business Process Log Filtering

Input: Business process event log *EL*
Output: Filtered list of event logs *FEL*

Initialization

1. ***EL'*** **= EL**
2. ***FEL*** **= (*EL'*)**
3. ***activities*** **= null**
4. ***length*** **= Length(getActivities(EL'))**

Filtering

5. **while*length*> 2 do**
6. ***activities*** **= getActivities(*EL'*)**
7. ***a'*** **= arg min$_{a \text{ belongs activities}}$ F^s(*EL'* |*activities*\{a})**
8. ***EL'*** **= *EL'*** **| *activities* \{a'}**
9. ***FEL*** **= *FEL*.(*EL'*)**
10. **endwhile**
11. **return*FEL***

Algorithm 1: Hybrid Approach for Business Process Log Filtering.

As shown in Algorithm 1, there is an iterative process which greedily filters out infrequent behaviours from event log. The algorithm takes EL (event log) as input and produces a list of event logs (FEL). Each element in the list is a list itself. Stated differently, each element in the FEL is a version of FEL with an activity has been filtered out. Thus each subsequent element has one more activity filtered out when the element is compared with that of preceding element. The algorithm continues working until there are only two activities in the event log EL. When two activities are found, it does mean that no relations can be discovered between two activities. This is the reason for the given stop criteria for the algorithm.

5. EXPERIMENTAL RESULTS

Experiments are made on real world and synthetic datasets [26] related to process logs. Python programming language and its related packages including ProM (process mining platform) is used to implement the proposed framework. The results are observed and evaluated in terms of fitness, precision, size, F-score and generalization. The performance of the proposed algorithm is compared with two baseline algorithms like Filter Log using Simple Heuristics (SLF) and Filter Log using Prefix-Closed Language (PCL) besides the filtering algorithm proposed by Conforti *et al*. [26]. The experimental results on both real life and artificial process event logs are as follows.

5.1 Fitness Comparison on Artificial Logs

Conforti *et al.*	Proposed	SLF	PCL
1	1.2	0.8	0.9
1	1.2	0.83	0.79
1	1.2	0.8	0.95
1	1.2	0.85	0.956
1	1.2	0.83	0.9
1	1.2	0.82	0.81
1	1.2	0.81	0.85
1	1.2	0.82	0.6

Table 1: Fitness comparison on artificial logs

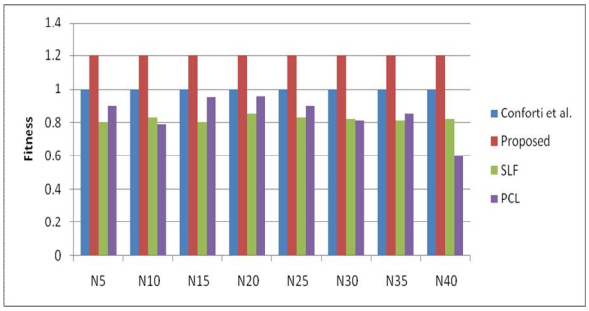

Figure 4: Fitness evaluation on artificial logs

5.2 Precision Comparison on Artificial Logs

Conforti *et al.*	Proposed	SLF	PCL
0.7	0.9	0.5	0.6
0.7	0.9	0.5	0.59
0.7	0.9	0.52	0.65
0.7	0.9	0.49	0.62
0.7	0.9	0.48	0.5
0.7	0.9	0.4	0.55
0.7	0.9	0.39	0.57
0.7	0.9	0.4	0.54

Table 2: Precision comparison on artificial logs

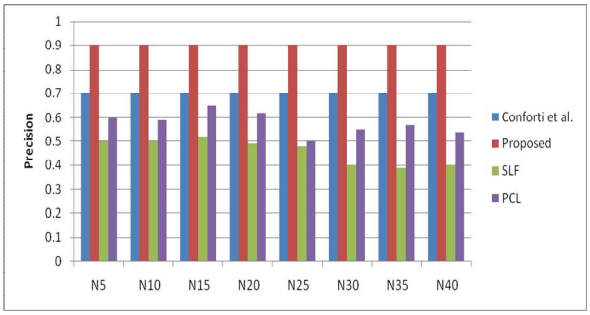

Figure 5: Precision comparison on artificial logs

5.3 F-Score Comparison on Artificial Logs

Conforti *et al.*	Proposed	SLF	PCL
0.8	0.9	0.58	0.7
0.8	0.9	0.58	0.65
0.8	0.9	0.58	0.75
0.8	0.9	0.58	0.74
0.8	0.9	0.57	0.58
0.8	0.9	0.56	0.65
0.8	0.9	0.55	0.7
0.8	0.9	0.54	0.56

Table 3: F-Score comparison on artificial logs

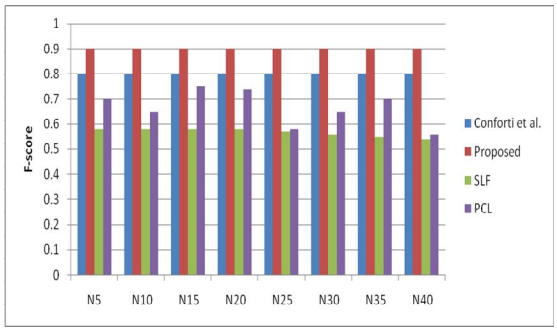

Figure 6: F-Score comparison on artificial logs

5.4 Size Comparison on Artificial Logs

Conforti *et al.*	Proposed	SLF	PCL
50	55	30	50
50	55	35	55
50	55	35	60
50	55	32	55
50	55	40	65
50	55	45	60
50	55	50	55
50	55	55	50

Table 4: Size comparison on artificial logs

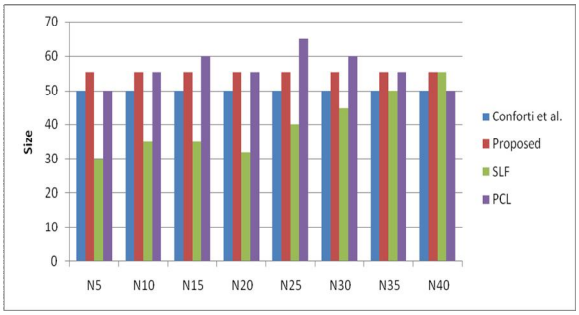

Figure 7: Size comparison on artificial logs

5.5 Fitness Comparison on Real-Life Logs

Conforti *et al*.	Proposed	Original
0.82	0.84	0.9
0.9	0.95	0.92
0.92	0.94	0.81
0.85	0.87	0.86
0.8	0.82	0.85
0.78	0.8	0.95
0.82	0.8	0.78
0.83	0.85	0.81
0.8	0.82	1
0.8	0.82	0.9
0.8	0.82	0.85
0.82	0.84	0.81
0.81	0.83	0.75
0.85	0.87	1
0.8	0.82	0.9
0.82	0.84	0.95

Table 5: Fitness comparison on real time process logs

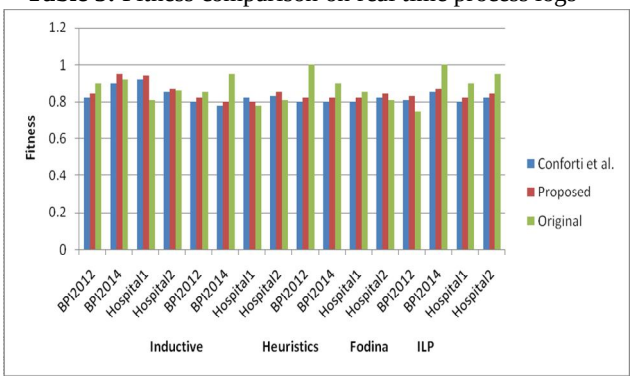

Figure 8: Fitness comparison on real world process logs

5.6 Precision Comparison on Real-Life Logs

Conforti *et al*.	Proposed	Original
0.99	1	0.82
0.55	0.59	0.4
0.65	0.67	0.5
0.65	0.67	0.35
1	1.2	0.7
0.6	0.69	0.5
0.5	0.55	0.3
0.4	0.45	0.33
1	1.2	0.65
0.2	0.25	0.39
0.5	0.55	0.4
0.4	0.5	0.38
0.6	0.65	0.36
0.55	0.57	0.6
0.3	0.35	0.3
0.25	0.3	0.25

Table 6: Precision comparison on real world process logs

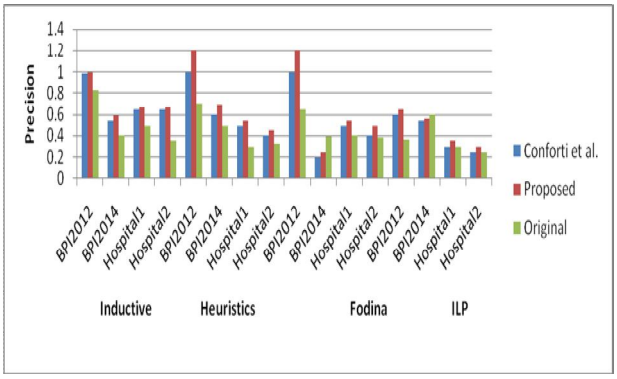

Figure 9: Precision comparison on real world process logs

5.7 F-Score Comparison on Real-Life Logs

Conforti *et al*.	Proposed	Original
0.9	1	0.85
0.65	0.67	0.58
0.78	0.8	0.59
0.78	0.8	0.5
0.9	1	0.8
0.7	0.69	0.7
0.65	0.67	0.42
0.6	0.62	0.44
0.9	1	0.75
0.3	0.35	0.5
0.6	0.65	0.5
0.55	0.57	0.4
0.7	0.72	0.78
0.68	0.7	0.67
0.5	0.55	0.4
0.45	0.47	0.35

Table 7: F-Score comparison on real world process logs

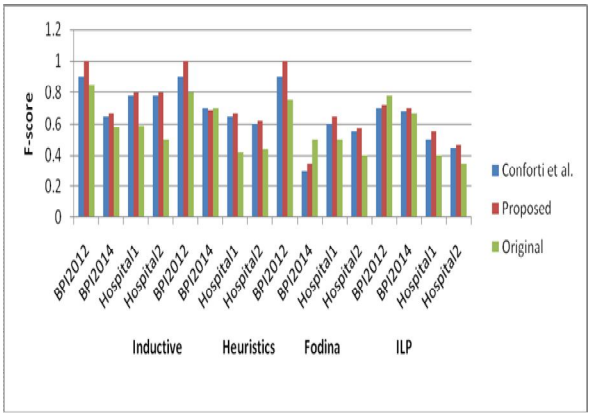

Figure 10: F-Score comparison on real time process logs

5.8 Size Comparison On Real-Life Logs

Conforti *et al*.	Proposed	Original
50	60	70
60	70	60
60	70	80
70	80	85
65	75	90
70	80	190
120	130	180
130	140	195
50	60	100
70	80	110
90	100	120
100	110	150
40	50	50
20	30	40
60	70	60
50	60	80

Table 8: Size comparison on real life logs

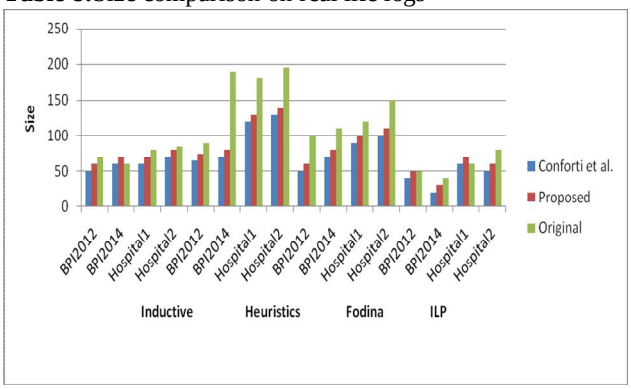

Figure 11: Size comparison on real time process logs

5.9 Generalization Comparison On Real-Life Logs

Conforti *et al*.	Proposed	Original
0.9	1	0.9
0.9	1	0.95
0.95	0.97	0.85
0.9	1	0.9
0.95	0.97	0.9
0.8	0.85	0.95
0.82	0.84	0.78
0.7	0.75	0.8
0.95	0.97	0.99
0.94	0.96	0.95
0.92	0.94	0.94
0.92	0.95	0.93
0.7	0.75	0.55
1	1.1	1
0.9	1	0.8
0.92	0.94	0.9

Table 9: Generalization comparison on real time process logs

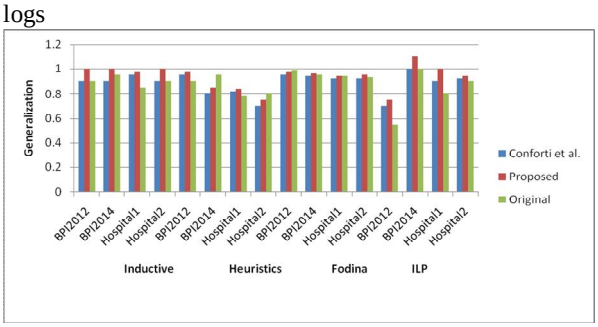

Figure 12: Generalization comparison on real time process logs

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework known as Automated Business Process Model Discovery Framework (ABPMDF) for automatic detection and removal of infrequent behaviours from process event logs. Towards this end an algorithm named Hybrid Approach for Business Process Log Filtering (HA-BPLF)is proposed and implemented. The algorithm makes use of entropy considering both directly-follows and directly-precedes relations besides Laplacian smoothing for better performance. The algorithm takes business process event log as input and produces a filtered list of event logs. ProM platform in Python and other Python packages like Graphviz are used to implement the framework. Empirical study is made with a prototype application. The results are analysed by comparing the proposed algorithm with baselines and the filtering algorithm in [26] in terms of fitness, precision, F-Measure, size and generalization. Experimental evaluation with a prototype application revealed that the proposed method is performing better than state of the art. However, the framework proposed in this paper is not fully realized in this paper. Therefore, here are directions for future work. First, it is important to identify and handle missing events in the event logs for improving quality of discovered process models. Second, isolating behaviours and comparison can help improve the utility of the framework. For instance, infrequent behaviours can be isolated from frequent behaviours so as to enable comparison of logs with similar behaviour.

REFERENCES

[1] W.M.P. van der Aalsta, M.H. Schonenberga and M. Song. (2010). Time Prediction Based on Process Mining. Information Systems, p1-33.

[2] Wil van der Aalst, Arya Adriansyah, and Boudewijn van Dongen. (2014). Replaying History on Process Models for Conformance Checking and Performance Analysis. IEEE, p1-22.

[3] R.P. Jagadeesh Chandra Bose and Wil M.P. van der Aalst. (2013). Discovering Signature Patterns from Event Logs. IEEE, p1-9.

[4] Mannhardt F, de Leoni M, Reijers H.A and van der Aalst W.M.P (2017). Data-driven process discovery: revealing conditional infrequent

behaviour from event logs. Advanced Information Systems Engineering, p1-16. https://doi.org/10.1007/978-3-319-59536-8_34

[5] Claudia Diamantini, Laura Genga and Domenico Potena. (2016). Behavioural process mining for unstructured processes. Journal of Intelligent Information Systems, p1-30. https://doi.org/10.1007/s10844-016-0394-7

[6] Wil M.P. van der Aalst, V. Rubin, B.F. van Dongen, E. Kindler, and C.W. G¨unther. (2012). Process Mining: A Two-Step Approach using Transition Systems and Regions. Springer, p1-36.

[7] Alfredo Bolt and Wil M.P. van der Aalst. (2014). Multidimensional Process Mining using Process Cubes. IEEE, p1-15.

[8] Mohammadreza Fani Sani, Sebastiaan J. van Zelst, Wil M.P. van der Aalst. (2015). Improving Process Discovery Results by Filtering Outliers using Conditional Behavioural Probabilities. IEEE, p1-12.

[9] Sander J.J. Leemans(B), Dirk Fahland, and Wil M.P. van der Aalst. (2015). Exploring Processes and Deviations. Springer International Publishing Switzerland, p1-13. https://doi.org/10.1007/978-3-319-15895-2_26

[10] Alfredo Bolt1 · Massimiliano de Leoni1 · Wil M. P. van der Aalst1. (2015). Scientific workflows for process mining: building blocks, scenarios, and implementation. Int J Softw Tools Technol Transfer, p1-22.

[11] Timo Nolle(B), Alexander Seeliger, and Max Muhlhauser. (2016). Unsupervised Anomaly Detection in Noisy Business Process Event Logs Using Denoising Autoencoders. Springer International Publishing Switzerland, p1-22. https://doi.org/10.1007/978-3-319-46307-0_28

[12] Hoang Thi Cam Nguyen, Suhwan Lee, Jongchan Kim, Jonghyeon Ko, Marco Comuzzi. (2019). Autoencoders for Improving Quality of Process Event Logs. Expert Systems, p1-27.

[13] Nijat Mehdiyev, Joerg Evermann and Peter Fettke. (2018). A Novel Business Process Prediction Model Using a Deep Learning Method. IEEE, p1-27. https://doi.org/10.1007/s12599-018-0551-3

[14] Stefan Schoniga, Cristina Cabanillasb, Stefan Jablonskia, Jan Mendling. (2016). A Framework for Efficiently Mining the Organisational Perspective of Business Processes. Journal of Decision Support Systems, p1-34. https://doi.org/10.1016/j.dss.2016.06.012

[15] W. M. P, van der Aalst, V. Rubin, H. M.W. Verbeek, B. F. van Dongen, E. Kindler and C. W. Günther. (2010). Process mining: a two-step approach to balance between under fitting and overfitting. Softw Syst Model, p1-25.

[16] Nicolas Poggi, Vinod Muthusamy, David Carrera1 and Rania Khalaf. (2013). Business Process Mining from E-commerce Web Logs. IEEE, p1-16. https://doi.org/10.1007/978-3-642-40176-3_7

[17] Biplob Debnath, Mohiuddin Solaimani, Muhammad Ali Gulzar, Nipun Arora, Cristian Lumezanu, Jianwu Xu, Bo Zong, Hui Zhang, Guofei Jiang and Latifur Khan. (2018). LogLens: A Real-time Log Analysis System. IEEE, p1-11. https://doi.org/10.1109/ICDCS.2018.00105

[18] Alireza Ostovar, Abderrahmane Maaradji, Marcello La Rosa, Arthur H.M. ter Hofstede and Boudewijn F.V. van Dongen. (2016). Detecting Drift from Event Streams of Unpredictable Business Processes. IEEE, p1-15. https://doi.org/10.1007/978-3-319-46397-1_26

[19] Borja Vazquez-Barreirosa, Manuel Mucientesa, Manuel Lamaa. (2014). ProDiGen: mining complete, precise and minimal structure process models with a genetic algorithm. Information Sciences, p1-26.

[20] Michael Borkowskia, Walid Fdhilac, Matteo Nardellib, Stefanie Rinderle-Mac and Stefan Schultea. (2018). Event-based Failure Prediction in Distributed Business Processes. Information Systems, p1-20.

[21] Hind Rbigui and Chiwoon Cho. (2017). The state-of-the-art of business process mining challenges. International Journal of Business Process Integration and Management, p1-31. https://doi.org/10.1504/IJBPIM.2017.10009731

[22] Ilya Verenich, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi and Chiara Di Francescomarino. (2016). Minimizing over processing waste in business processes via predictive activity ordering. Springer International Publishing Switzerland, p1-16. https://doi.org/10.1007/978-3-319-39696-5_12

[23] Niels Martin, Benoît Depaire and An Caris. (2015). The use of process mining in business process simulation model construction: structuring the field. Business & Information Systems Engineering. 51 (1), p1-23. https://doi.org/10.1007/s12599-015-0410-4

[24] M. Leemans and W.M.P. van der Aalst. (2017). Modelling and Discovering Cancelation Behaviour. IEEE, p1-19.

[25] Pavlos Delias and Ioannis Kazanidis. (2017). Process Analytics Through Event Databases: Potentials for Visualizations and Process Mining. Springer, p1-13. https://doi.org/10.1007/978-3-319-57487-5_7

[26] Raffaele Conforti, Marcello La Rosa and Arthur H. M. ter Hofstede (2016). Filtering out Infrequent Behaviour from Business Process Event Logs. IEEE Transactions on Knowledge and Knowledge Engineering, IEEE, p1-14.

[27] Danish Ahamad, MD Mobin Akhtar, Shabi Alam Hameed.- A Review and Analysis of Big Data and MapReduce (2019). IJATCSE Feb 2019.

[28] Danish Ahamad, MD Mobin Akhtar, Shabi Alam Hameed.- Analysis of Data Science with the use of Big Data (2018). IJATCSE December 2018.