Annals of Emerging Technologies in Computing (AETiC)
Vol. 5, No. 4, 2021

Research Article

Process Discovery Enhancement with Trace Clustering and Profiling

Muhammad Faizan[1], Megat F. Zuhairi[2,*] and Shahrinaz Ismail[3]

[1,2]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Malaysia
Muhammad.faizan@s.unikl.edu.my; megatfarez@unikl.edu.my
[3]Albukhary International University, Alor Setar, Kedah, Malaysia
sha905@gmail.com
*Correspondence: megatfarez@unikl.edu.my

Received: 12th June 2021; Accepted: 21st September 2021; Published: 1st October 2021

Abstract: The potential in process mining is progressively growing due to the increasing amount of event-data. Process mining strategies use event-logs to automatically classify process models, recommend improvements, predict processing times, check conformance, and recognize anomalies/deviations and bottlenecks. However, proper handling of event-logs while evaluating and using them as input is crucial to any process mining technique. When process mining techniques are applied to flexible systems with a large number of decisions to take at runtime, the outcome is often unstructured or semi-structured process models that are hard to comprehend. Existing approaches are good at discovering and visualizing structured processes but often struggle with less structured ones. Surprisingly, process mining is most useful in domains where flexibility is desired. A good illustration is the "patient treatment" process in a hospital, where the ability to deviate from dealing with changing conditions is crucial. It is useful to have insights into actual operations. However, there is a significant amount of diversity, which contributes to complicated, difficult-to-understand models. Trace clustering is a method for decreasing the complexity of process models in this context while also increasing their comprehensibility and accuracy. This paper discusses process mining, event-logs, and presenting a clustering approach to pre-process event-logs, i.e., a homogeneous subset of the event-log is created. A process model is generated for each subset. These homogeneous subsets are then evaluated independently from each other, which significantly improving the quality of mining results in flexible environments. The presented approach improves the fitness and precision of a discovered model while reducing its complexity, resulting in well-structured and easily understandable process discovery results.

Keywords: ***Incremental trace clustering; Process mining; Pre-processing; Process discovery; Trace profiling***

1. Introduction

Information systems nowadays can reliably document the implementation of business processes [1]. Such a system includes procure-to-pay and order-to-cash processes, monitored by Enterprise Resource Planning (ERP) systems. Process mining [2–4] focuses on turning these event data into actionable and valuable information to detect and rectify process efficiency or enforcement problems. Various process mining techniques [5–8] exist. Some of the methods include automated process discovery, process variant analysis, performance mining, and conformance checking. In process model discovery, the goal is to recreate the overall business process structure in a process model. Conformance checking assesses the degree to which the reported data aligns with the organisation's normative process model. Process variant analysis focuses on the variations of real scenarios, whereas performance mining focuses on evaluating the efficiency of processes.

Process mining is an effective method for evaluating the executions of operational processes based on event data. Current process mining techniques perform well on structured processes but exploring and visualising less-organised processes may pose problems [9]. Unfortunately, in areas where flexibility is

Muhammad Faizan, Megat F. Zuhairi and Shahrinaz Ismail, "Process Discovery Enhancement with Trace Clustering and Profiling", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 1-13, Vol. 5, No. 4, 1st October 2021, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2021.04.001, Available: http://aetic.theiaer.org/archive/v5/v5n4/p1.html.

required, process mining is of utmost concern [10]. However, there is a significant amount of diversity in a flexible environment, which contributes to complicated, difficult-to-understand models.

Nonetheless, inherent issues exist about applying process mining within changing situations. Such conditions typically require a wide variety of potential behaviour, where the analytical findings are similarly unstructured. Numerous studies show that process discovery methods struggle to discover precise and interpretable process models from event-logs generated in highly flexible environments [11].

This paper combines two distinct methods that acknowledge the process mining perspective and increase the quality of a discovered model. When dealing with unstructured processes and reducing process models' complexity, trace clustering is a great option to improve the quality of a discovered model by increasing its fitness, precision and reducing its complexity. The trace clustering method involves a systematic divide-and-conquer approach—however, the trace profiling approach is based on specific features derived from the corresponding trace. Each trace is assigned a numeric value by each object, which is referred to as a metric. Clustering techniques may use a variety of distance measures [12] to quantify the similarity between cases. Several distance metrics can measure the relative-distance between the log's two cases using these feature metrics. Finally, clustering algorithms can be used to group cases into homogenous subsets, depending on how closely they are related. Fig. 1 depicts the framework to improve the quality of the process model used in this article.

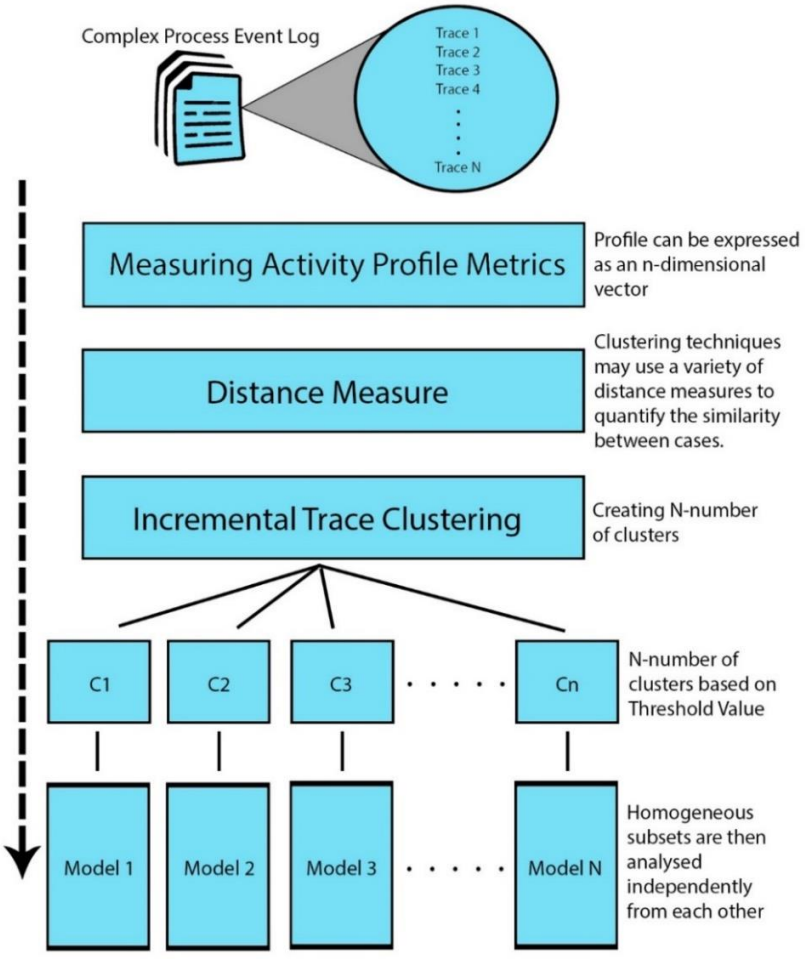

Figure 1: Framework to improve the quality of a process model

2. Process Mining Background Knowledge

Process mining is a collection of tools for analysing and exploiting data obtained by IT systems that support business processes. Process mining's primary objective is to automatically extract information from event-logs generated by IT systems [8], [13]. Information systems keep track of events in various formats, from plain text files to data embedded in massive database structures. Machines and users carry out processes with the assistance of software systems, on which process mining provides three types of activities

after the event-log has been extracted: discovery, conformance testing, and enhancement. Fig. 2 illustrates an overview of the process mining operations.

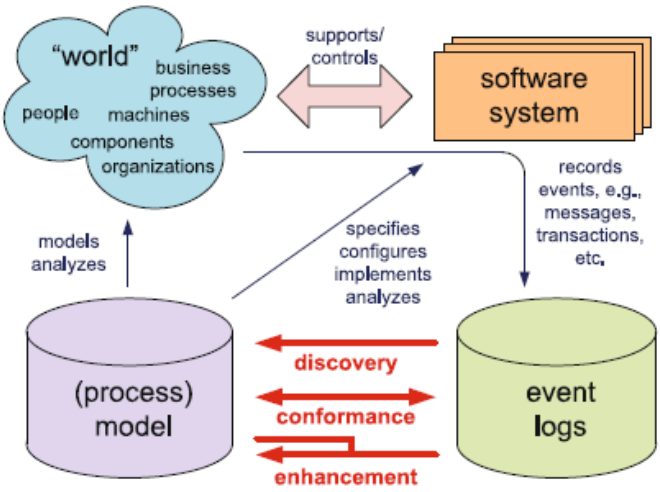

Figure 2: An overview of the process mining operations [2]

Numerous discovery algorithms differentiate between several different perspectives, including the organisation, the process, or the case [14]. The discovery algorithms are designed to concentrate on the process perspective to produce a process model covering the event-log's behaviour. A variety of discovery algorithms can be used, which include α–algorithm and its extensions [15], heuristic mining [7], genetic mining [16], fuzzy mining [6], inductive mining [17] and region-based mining [3]. The organisational perspective is concerned with the relationships between persons and roles due to the handover of work or the execution of related activities. Different characterisations are focused on the case perspective of a process instance. It analyses the data components, as well as the roles and persons associated with the process. However, the values of the associated data items can also be used to classify cases. For example, if a case is a replenishment order, knowing the number of products purchased or the supplier might be valuable information.

Conformance checking is the second process mining method, which involves comparing the process as reported in an event-logs to a specified process classification, e.g., a process model. As a conformance checking method, a variety of techniques have been proposed. In ProM 5, Rozinat et al. [18] introduce a conformance checker plugin that uses a log replay technique to determine which degree an event-log matches the behaviour defined in a Petri net. Another method introduced by Aalst et al. [19] describes the process model's alignments and the event-log to identify deviations. Most of the stated approaches emphasise the conformance of event-logs to process models, i.e., the analysis of deviations between actual and theoretic processes, by measuring the alignment between their event data and existing processes

Enrichment or enhancement is the third technique in process mining, and it focuses on improving process models using knowledge from event-logs. For instance, performance data such as activity waiting times or durations can be extracted and plotted onto process models. The event-log is analysed in a process model to derive rules that explain why certain exclusive paths are chosen.

2.1. Event-Log

Process mining deals with the records of events extracted from information systems. Events can be characterised by several attributes [20]. An event can be set with a timestamp related to an activity conducted by a given person with the associated costs/variants. The specific information provided in an event-log used for process mining is illustrated in Table. 1. Event-logs are the departure point for process mining [21]. The most common assumption about event-logs is that they provide information about every distinct process. Some of the basic terminologies for processing mining event-logs are subsequently discussed.

Case: A case is a specific instance of any process, and a process may contain several cases. Table 1 shows four cases, and each case has a unique identifier referred to as case-id. Case 1 comprises six associated events.

Activity: Each event in a log refers to an activity. An activity forms one step in the process. Table 1 shows invite reviewers, get review 2, get review 1, collect reviews, decide and reject as activities on the first case. A process is composed of activities and the relationship between activities. The following types of relations are the components:

- Loop- Activity X repeats a certain number of times
- Concurrency- Activities X and Y happens mainly at the same time
- Sequence- Activity X follows activity Y
- Decision point- From Activity X, either activity Y or activity Z can be reached further

Event id: Every event can have a specific identifier, known as the event id. However, for identification or mining purposes, this is not widely used.

Data Attribute: Log files contain a single data attribute or even more to provide additional event information. Table. 1 presents the variant as an attribute for the results. Log files can have several other attributes associated with the data.

Resources: Every event in Table. 1 is linked to a resource. However, not all log files contain details about the resources.

Traces: Events records are associated with a specific trace. An event-log is a sequence of traces, and the events are organised sequentially within each trace. Each log event is unique and can be connected to a single trace.

Timestamp: A series of characters or encoded information indicates the date and time of day that a specific activity occurs, which can be precise to a fraction of seconds. Besides, Table. 1 displays a column with a human-readable timestamp. Depending on server configurations where log files are stored, the timestamp could be in different formats.

According to the Disco User Guide[1], event-logs should include at least the following four essentials to apply process mining tools.

i. Timestamp
ii. Case Id
iii. Activity
iv. Resource

Table 1: An example of an event-log file

Case Id	Event Id	Activity	Resource	Variant	Timestamp
1	654321	invite reviewers	Anne	11	01/01/2020 07:00
1	654322	get review 2	Mary	11	05/01/2020 08:00
1	654323	get review 1	John	11	14/01/2020 18:00
1	654324	collect reviews	Mike	11	15/01/2020 10:00
1	654325	decide	Wil	11	18/01/2020 06:00
1	654326	reject	Anne	11	28/01/2020 16:00
2	654327	invite reviewers	Anne	6	14/02/2020 02:00
2	654328	get review 1	John	6	18/02/2020 03:00
2	654329	collect reviews	Mike	6	19/02/2020 05:00
2	654330	invite additional reviewer	Mike	6	19/02/2020 09:00
2	654331	get review X	Carol	6	03/03/2020 16:00
2	654332	decide	Wil	6	08/03/2020 05:00
2	654333	accept	Anne	6	09/03/2020 01:00
3	654334	invite reviewers	Anne	15	25/03/2020 15:00
3	654335	get review 2	Mary	15	02/04/2020 06:00
3	654336	collect reviews	Mike	15	03/04/2020 02:00
3	654337	invite additional reviewer	Mike	15	07/04/2020 05:00
3	654338	get review X	Joe	15	10/04/2020 18:00
3	654339	decide	Wil	15	11/04/2020 11:00
3	654340	accept	Anne	15	11/04/2020 17:00
4	654341	invite reviewers	Anne	21	15/04/2020 20:00
4	654342	get review 1	John	21	19/04/2020 03:00
…	…	…	…	…	…

[1] "Disco User´s Guide." https://fluxicon.com/disco/, https://fluxicon.com/book/read/reference/#disco-user-guide.

3. Problem Formulation

Data is being recorded at a growing pace due to the increased use of information systems [22]. A lot of this data is unstructured and difficult to understand. Process mining is a field that aims to extract knowledge about an organisation's processes, mainly from documented events and turn them into meaningful process models. The common issues found in the resulting discovered process models are a disproportionate amount of task nodes and the formation of a large number of connections leading to the traditional "spaghetti models" [10], [23]. The diversity of an event-log is one significant cause of unstructured mining performance [24], i.e., individual cases are significantly different from each other. In these instances, it is reasonable to assume that one event-log contains multiple implicit process variants, each of which is substantially more structured than the overall process.

The problem of analysing large-scale event-logs can be confronted using the data mining "clustering" technique on event data for pre-processing, which improves process discovery in process mining [25]. Pre-processing is the process of converting data sources' information about content, use, and structure into the data-abstractions required for pattern discovery. This pre-processing stage makes it easier to translate event data into a format that can be used for process mining. However, using the data without any pre-processing will create an immature model that cannot handle most cases. Without pre-processing, the outcomes are often unstructured or semi-structured process models that are hard to comprehend. A typical spaghetti process discovered using conventional process mining techniques is shown in Fig. 3.

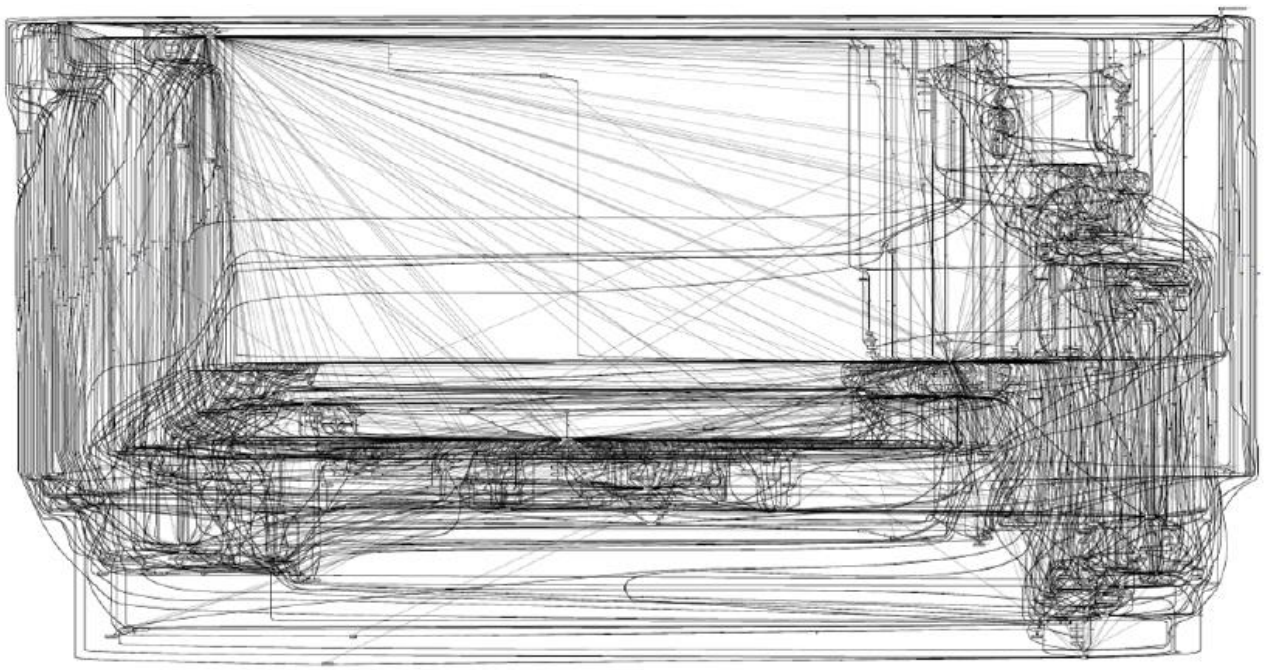

Figure 3: Spaghetti process describing the complexity of a process-model

In the spaghetti model, there is no problem with the map itself; it is a process if you look at every little detail, every minor exception, all in one picture, but the problem with that it is not particularly useful. It cannot derive any valuable analysis insights from such a detailed process map.

4. Trace Clustering

Clustering is an unsupervised data mining technique [26–28] focusing on grouping together homogenous objects. The purpose of clustering is to extract groups/clusters and comparable data objects [29], [37]. In contrast, the objects belonging to two different clusters are dissimilar, as illustrated in Fig. 4. In process mining, the objects are the process instances, i.e., cases in an event-log. To enable cluster creation, it is essential to define a notion of dissimilarity between process instances. Fig. 4. (f) depicts a mixture of traces (distinguished by colour). The objective of trace clustering is to partition the traces into clusters such that traces within a cluster are similar to each other and traces that belong to different clusters are dissimilar, i.e., “grouping traces of the same colour” as illustrated in Fig. 4. (a-e).

The trace clustering technique deals with the heterogeneity in the event-log [30]. Process Mining methods have difficulties dealing with variability caused by heterogeneity, i.e., different usage scenarios are merged into a single spaghetti-like process [31]. Multiple comprehensible models capturing different behaviour classes are preferred over a single spaghetti-like model in such scenarios. A typical method to achieve this is to pre-process an event-log to segregate/cluster homogenous sets of cases and analysing each set of homogenous cases separately. The importance of trace clustering in process mining is illustrated in Fig. 5. A process model derived from a complete event-log is shown on the left-side of the Figure. Understanding this model is very difficult because it is pretty complex to comprehend. The process models mined from the clustered traces are shown on the Figure's right-side. Trace clustering allows the comprehension of process-models by reducing the "spaghettiness." Clustering enables an easier grasp of process models discovered by reducing "spaghettiness" [32]. Such segregation also assumes significance when dealing with vast volumes of data [27], based on the technique "divide and conquer" [33], [34].

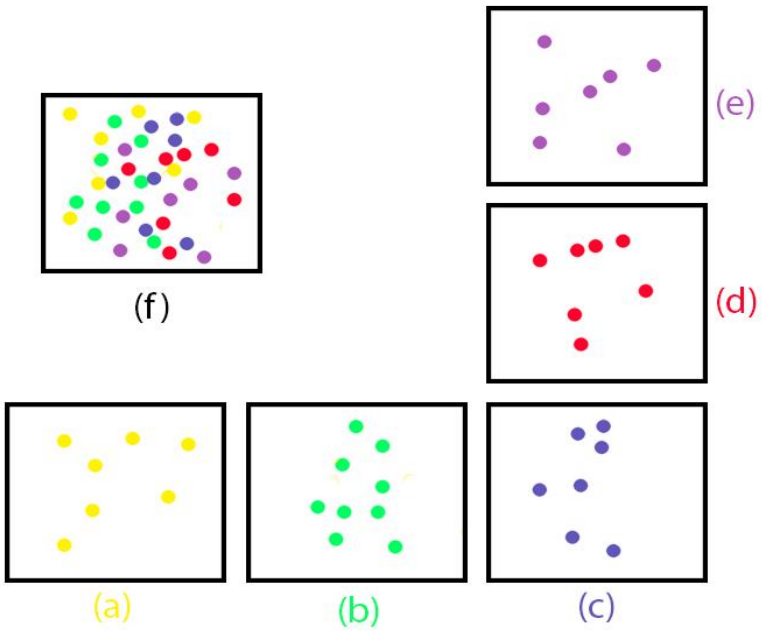

Figure 4: The objective of trace clustering: An event-log containing a heterogeneous mix of traces. Traces represented by the same colour are similar to each other, and traces of different colour are dissimilar

Furthermore, event-logs may contain anomalous, outlier, or noisy traces. The presence of such impurity in logs can also impact the goodness of the mined results. However, the primary constraint of trace clustering on data is that it cannot be used for process mining independently. The data utilised in process mining is not in a format that can be directly used for trace clustering. It required the use of additional techniques in order to group data into clusters. Moreover, trace clustering and profiling give astonishing outcomes on large datasets.

Most of the arguments on 'improved results' through trace clustering are subjective. Three factors influence the partitioning of an event-log into clusters of homogenous cases:

- Features used to characterize a case
- Distance or similarity metrics used
- Choice of clustering algorithm

In the literature, several methods to trace clustering have been suggested. Several techniques apply a form of translation to the learning problem to use current distance-based clustering algorithms. A distance metric between each pair of traces can be calculated by converting an event-log to a vector-space model. On the other hand, the distance between two traces is reflected using techniques that do not convert the distance into an attribute value context. Moreover, model-based clustering methods have also been shown to be applicable to trace clustering.

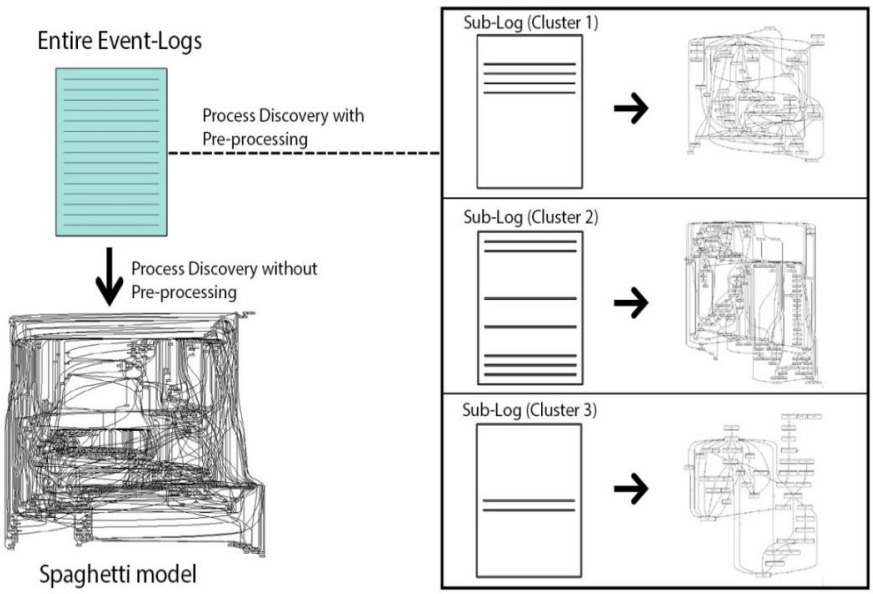

Figure 5: Illustration of the spaghetti model and essential method of trace clustering in process mining

The research in [30] introduces a trace clustering method built on the "MCL" Markov cluster algorithm to discover process deviations and process variants. A stochastic matrix in this methodology represents the transition probability among trace activities. It may independently find an (unspecified) number of clusters of different densities and sizes.

Greco et al. [35] were pioneers in studying process mining for clustering log traces. They proposed the concept of disjunctive workflow schemas "DWS" for divisive trace clustering. They also construct vector space trace clusters over the activities and their transitions to discover descriptive process models on the initial attempt. As the feature vector, Song et al. [10] elaborated on creating so-called trace profiles composed of various trace-perspectives, i.e., to measure the similarity between the traces, a similarity matrix was created. This paper uses the same approach prosed in [10] to create a profiling vector as an input for the clustering algorithm. However, the main idea is to focus on the fitness, precision, and simplicity of a model generated by clustered data using Incremental Trace Clustering.

The assumption behind incremental clustering is that patterns can be considered one at a time and allocated to clusters that already exist. A new data object is allocated to a cluster without significantly affecting the current clusters. Below is a high-level explanation of a standard incremental clustering algorithm.

a. Create a cluster for the first data point.
b. Consider the following data point. Either place this point in an existing cluster or create a new one with it. This task is accomplished based on a set of criteria, such as the distance between the new data point and cluster centroids that already exist.
c. Proceed with step b until all of the data items have been clustered.

The incremental clustering algorithm's main benefit is that the entire pattern matrix does not need to be stored in memory. Typically, it is non-iterative, and as a result, its time requirements are minimal and have minimal space requirements.

5. Trace Profiling (TP)

Evaluation of the similarity of clustered points is crucial for the clustering application [10]. Cases are the instances of processes that left a trace in the log, and points are associated with them. A case is characterised by a well-defined compilation of items in the trace profiling approach, i.e., specific characteristics obtained from the related trace. Each trace is assigned a numeric value by every item referred to as a metric. As a result, we can consider a profile with n items to be a function that assigns a vector (x_1, $x_2, x_3, \ldots\ldots\ldots, x_n$) to a trace. Profiling a log is identified as measuring a collection of traces using several profiles to produce an aggregate vector, which contains the values for each measured item in a predetermined order.

Several distance metrics can calculate the relative-distance between the log's two cases using these feature metrics. Finally, clustering algorithms can be used to group cases into homogenous subsets, depending on how closely they are related. These homogenous subsets can then be analysed independently of one another, significantly improving the quality of mining results in flexible environments.

Table. 2 demonstrates the outcome of expanding activity profiles on the example log from Table. 1. The activity profile contains one item for each form of activity, e.g., event-name, in the log. Counting an activity object is as simple as counting all of the events in a trace with the same activity's name. Each row of the table represents the profile vector of one trace in the log.

Table 2: Activity profiles from Table. 1 example logs

Case Id	Activity Profiles								
	invite reviewers	**get review 1**	**get review 2**	**collect reviews**	**decide**	**invite additional reviewer**	**get review X**	**accept**	**reject**
1	1	1	1	1	1	0	0	0	1
2	1	1	0	1	1	1	1	1	0
3	1	0	1	1	1	1	1	1	0
4	1	1	0	1	1	1	2	0	1
…	…	…	…	…	…	…	…	…	…

Table. 2 shows a profile that captures the details typically found in event-logs. It is simple to expand this approach with custom profiles when extra knowledge is available in an application area, e.g., "performance profile, assets profile, transition profile, and originator profile." However, designing different profiles can increase the quality of trace clustering.

6. Experiment

As shown in Table. 1, the illustration processes are the procedure of reviewing articles for publication. The procedure begins with the process of invite reviewers by an editor. The author submitted the article to the journal and requested publication. A preliminary process starts by inviting reviewers to conduct a review process when the request has been received. After inviting the reviewers, the editor starts getting review responses. Then, the process to collect reviews begins, and based on that, a decision has been made to start the acceptance or rejection process directly. If the decision is not clear to the editor, then the inviting additional reviewer's process begins, and a cycle of decision making started until an acceptance or rejection decision has been made, and the case is ended.

Table. 1 schematically displays event-logs that follow the processes identified earlier. Every row is a series of events and corresponds to a single case. The term "case" refers to a specific row in the event log, while "trace" refers to a sequence of events within that case. As shown in row 1, events are classified by case-identifier (1), activity-identifier (invite reviewers), and originator (Anne). The α-algorithm will automatically discover the Petri net model shown in Fig. 6 based on that log.

To determine the accuracy of the discovered model, a conformance checking can be performed [10], [18], [36]. Fitness and precision are two consistency metrics used in process mining to evaluate process models' behavioural quality. The term fitness refers to how much of the observed behaviour can be enlightened by the model that was discovered, i.e., "Does the observed process comply with the process model's control flow?". According to the model, an average fitness value for the log ranges from 0 to 1, indicating how well the model can accurately capture the behaviour shown in the traces—the percentage of traces in the log that fit the model perfectly. Even if a model has the potential to "replay" traces in logs, it can still be ineffective due to its complexity or the amount of behaviour it requires, i.e., behaviour that is not confirmed by log interpretations. Precision is a comparison of the behaviour activated in the model and the behaviour active in the log at a given condition. The amount of behaviour distinct by the discovered model presented in the event-log is measured by precision, i.e., if a model does not allow for "too much" behaviour, it is said to be precise. A metric for behavioural appropriateness is discussed in [5], i.e., Does the model accurately reflect the process observed?

Conformance checking plug-in “Replay a log on petri net for conformance analysis” and “Measure Precision/Generalization” plug-in in ProM 6.9 is used to measure the fitness and precision of a derived model. Fig. 6 shows the model with a fitness level of 0.73 and a precision of 0.86, i.e., the outcome is not satisfactory because of insufficient patterns. However, a more sophisticated process-mining algorithm

could create a better model. Many of the existing modern techniques generate perfect fit models. The experiment aims that by clustering cases can obtain an optimal and more accurate model while still using the α-algorithm.

More accurate models can be extracted from it by splitting the event-logs into multiple subgroups and discover multiple process models. Clustering techniques may use a variety of distance-measures to quantify the similarity among two cases. The distance measures help in the calculation of inter-pattern similarity and can influence clustering outcomes. Euclidean distance is used in this experiment to measure the similarity between cases.

$$\text{Euclidean distance } (e_a, e_b) := \sqrt{\sum_{x=1}^{n} |c_{ax} - c_{bx}|^2}$$

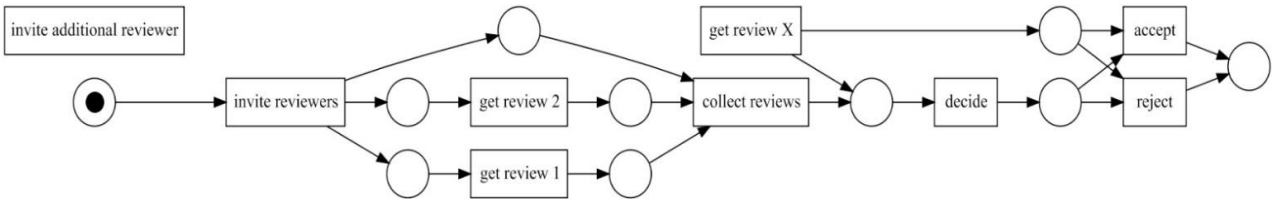

Figure 6: Process Model with the complete event-log

The profile is an n-dimensional vector, with n representing the number of items derived from the process event-log. Thus, the vector $(c_{a1}, c_{a2}, c_{a3}, \ldots\ldots, c_{an})$ corresponds to the case e_a, where each c_{ab} represents the number of times item b has appeared in case a.

Incremental Trace Clustering [29] is the algorithm used in this study to cluster data objects, which means that we can look through patterns one by one and allocate them to existing clusters. Incremental trace clustering focuses to create groups of event-log traces based on the resemblance of process instances. The centroid does not need to be recalculated because it assigns all new data to a cluster without influencing the existing clusters. To determine when a new cluster is formed, the basic concept is to compare the similarity of various processes in each trace. A threshold value specifies whether the trace can add to an existing cluster or build a new cluster. When a new trace is added to an existing cluster, the centroid does not recalculate. Fig. 7 illustrates the working of Incremental Trace Clustering on the activity profile vector.

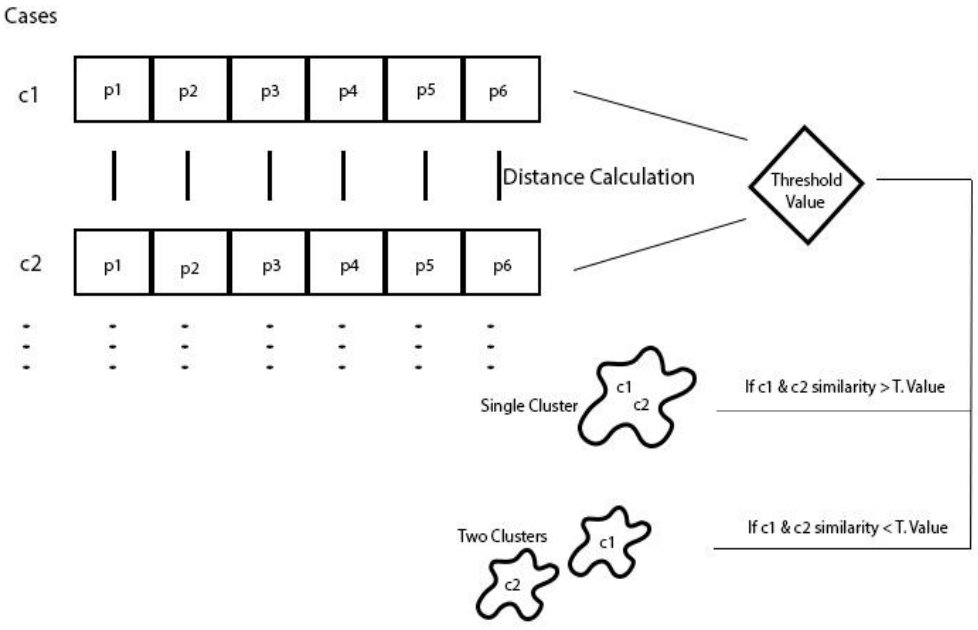

Figure 7: Illustration of Incremental Trace Clustering on Trace Profile Vector

Subsequently, by applying this approach to the example log presented in table. 1, three groups have been classified, i.e., depending on the tasks. The first group **a** is where all primary reviewers respond to the editor's request, i.e., cases where the task of "invite addition reviewer" is missed, along with the task of "get review X." The second group **b** refers to cases where no primary reviewer responded to the editor. These cases are not concerned with reviewer 1 & 2. The third subgroup **c** relates to cases where an editor required an additional reviewer to decide—fig. 8 displays process models built up again from each group using the same α-algorithm. The three models' precision and fitness are 1.000, which indicates that trace clustering

can help classify process variants that match homogeneous sub-sets of cases. Also, the generated models are of much higher quality and easy to understand. This demonstrates that trace clustering allows the separation of different processes from a process-log by splitting them into subsets with similar properties.

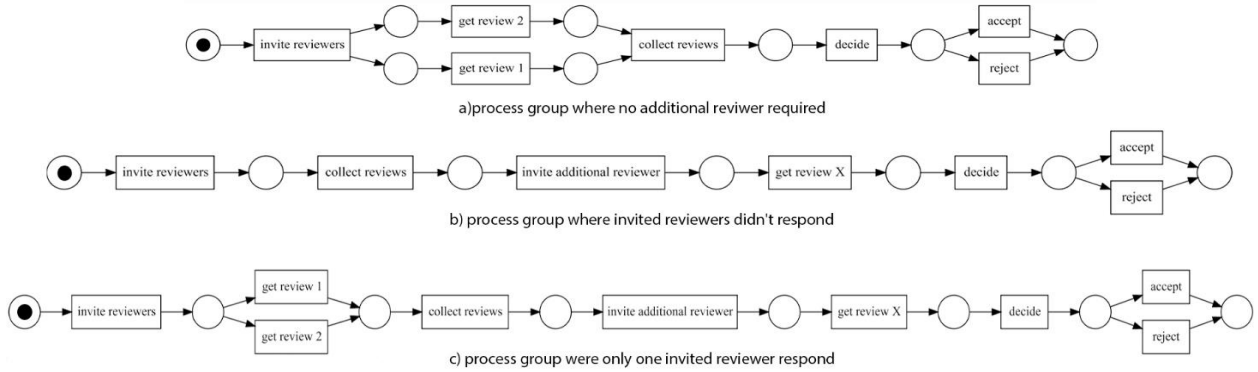

Figure 8: Derived Process Model from three groups

7. Conclusion and Future Directions

Process mining techniques may provide accurate, valuable insights into how real-life processes are being conducted. Process mining is crucial for understanding flexible environments in which participants, including top management, are often unaware of the process's structure. However, in such a flexible environment, process mining algorithms continue to produce unstructured and complex process-models that are difficult to understand and often provide unusable information. Diversity is one explanation for these issues, i.e., processes which can produce a collection of very different organised cases. Pre-processing of event-logs is used to alleviate the problem. Trace Clustering is one of the best solutions to that problem, which helps generate more precise and more accurate process models.

This paper highlighted some significant challenges in process discovery and presented a generic approach that divides the event-log into smaller, more homogeneous subsets of traces to solve diversity-related problems efficiently. The activity profiling concept is used with Incremental Trace Clustering to provide a suitable solution for characterising, comparing, and grouping homogenous objects of traces. It showed that process mining results could be improved by analysing these subsets independently, compared to analysing the whole event-log. This approach significantly increases the effectiveness of mined results, i.e., by improving the quality of a discovered model with an "increase in fitness and precision and reduce its complexity".

Relevant process information is retained as much as possible in this way, resulting in more precise subsets. Moreover, to cluster these traces, we combine two distinct methods that agree with the process mining perspectives. Trace clustering can improve any process mining algorithm results since it operates at the event-log level. Many interesting issues, such as introducing domain-specific profiles, different clustering algorithms, or more refined distance/similarity steps, remain open for future study.

Acknowledgement

The authors should like to acknowledge the support and facilities provided by the Universiti Kuala Lumpur, Malaysia.

References

[1] Jinlin Wang, Xing Wang, Yuchen Yang, Hongli Zhang and Binxing Fang, "A review of data cleaning methods for web information system", *Computers, Materials & Continua*, Print ISSN: 1546-2218, Online ISSN: 1546-2226, vol. 62, no. 3, pp. 1053–1075, 2020, Published by Tech Science Press, DOI: 10.32604/cmc.2020.08675, Available: https://www.techscience.com/cmc/v62n3/38341.

[2] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier *et al.*, "Process mining manifesto", in *Lecture Notes in Business Information Processing (LNBIP)*, Business Process Management Workshops, BPM 2011, vol. 99, Online ISBN: 978-3-642-28108-2, Print ISBN: 978-3-642-28107-5, DOI: 10.1007/978-3-642-28108-2_19, pp. 169–194, Available: https://link.springer.com/chapter/10.1007/978-3-642-28108-2_19.

[3] Mahdi Ghasemi and Daniel Amyot, "Process mining in healthcare: A systematised literature review", *International Journal of Electronic Healthcare*, Print ISSN: 1741-8453, Online ISSN: 1741-8461, pp. 60–88, vol. 9, no. 1, 2016, DOI: 10.1504/IJEH.2016.078745, Available: https://www.inderscience.com/info/inarticle.php?artid=78745.

[4] Edgar Batista and Agusti Solanas, "Process mining in healthcare: A systematic review", in 2018 *9th International Conference on Information, Intelligence, Systems and Applications, IISA 2018*, vol. 1, pp. 1–6, 2018, Published by IEEE, DOI: 10.1109/IISA.2018.8633608, Available: https://ieeexplore.ieee.org/document/8633608.

[5] Sungbum Park and Young Sik Kang, "A Study of Process Mining-based Business Process Innovation", *Procedia Computer Science*, ISSN: 1877-0509, vol. 91, pp. 734–743, 2016, DOI: 10.1016/j.procs.2016.07.066, Available: https://www.sciencedirect.com/science/article/pii/S1877050916312492.

[6] Camilo Alvarez, Eric Rojas, Michael Arias, Jorge Munoz-Gama, Marcos Sepúlveda *et al.*, "Discovering role interaction models in the Emergency Room using Process Mining", *Journal of Biomedical Informatics*, ISSN: 1532-0464, vol. 78, February 2018, pp. 60–77, 2018, Published by Elsevier, DOI: 10.1016/j.jbi.2017.12.015, Available: https://www.sciencedirect.com/science/article/pii/S153204641730285X.

[7] Wil M. P. van der Aalst, "Process discovery from event data: Relating models and logs through abstractions", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 3, Online ISSN: 1942-4795, pp. 1–21, 2018, DOI: 10.1002/widm.1244, Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1244.

[8] Angelina Prima Kurniati, Owen Johnson, David Hogg and Geoff Hall, "Process mining in oncology: A literature review", in *Proceedings of the 2016 6th International Conference on Information Communication and Management (ICICM)*, 2016, Online ISBN: 978-1-5090-3494-9, no. 1, pp. 291–297, 2016, DOI: 10.1109/INFOCOMAN.2016.7784260, Available: https://ieeexplore.ieee.org/document/7784260.

[9] R. P. Jagadeesh Chandra Bose and Wil van der Aalst, "Trace alignment in process mining: Opportunities for process diagnostics", in *Lecture Notes in Computer Science (LNCS)*, vol. 6336, Online ISBN: 978-3-642-15618-2, Print ISBN: 978-3-642-15617-5, pp. 227–242, 2010, Published by Springer-Berlin,Heidelberg, DOI: 10.1007/978-3-642-15618-2_17, Available: https://link.springer.com/chapter/10.1007/978-3-642-15618-2_17.

[10] Minseok Song, Christian W. Günther and Wil M. P. van der Aalst, "Trace clustering in process mining", in *Lecture Notes in Business Information Processing (LNBIP)*, vol. 17, Online ISBN: 978-3-642-00328-8, Print ISBN: 978-3-642-00327-1, pp. 109–120, 2009, Published by Springer-Berlin,Heidelberg, DOI: 10.1007/978-3-642-00328-8_11, Available: https://link.springer.com/chapter/10.1007/978-3-642-00328-8_11.

[11] Pieter De Koninck and Jochen De Weerdt, "Scalable mixed-paradigm trace clustering using super-instances", *Proceedings of the 2019 International Conference on Process Mining (ICPM)*, pp. 17–24, 2019, Published by IEEE, DOI: 10.1109/ICPM.2019.00014, Available: https://ieeexplore.ieee.org/document/8786061/.

[12] Yu Jiang, Dengwen Yu, Mingzhao Zhao, Hongtao Bai, Chong Wang *et al.*, "Analysis of semi-supervised text clustering algorithm on marine data", *Computers Materials & Continua*, Print ISSN: 1546-2218, Online ISSN: 1546-2226, vol. 64, no. 1, pp. 207–216, 2020, Published by Tech Science Press, DOI: 10.32604/CMC.2020.09861, Available: https://www.techscience.com/cmc/v64n1/39138.

[13] Krisztina Tóth, Károly Machalik, György Fogarassy and Ágnes Vathy-Fogarassy, "Applicability of Process Mining in the Exploration of Healthcare Sequences", in *IEEE 30th Neumann Colloquium (NC)*, pp. 000151-000156, 2017, Published by IEEE, DOI: 10.1109/NC.2017.8263273, Available: https://ieeexplore.ieee.org/document/8263273.

[14] Alfredo Bolt, Massimiliano de Leoni and Wil M. P. van der Aalst, "Scientific workflows for process mining: building blocks, scenarios, and implementation", *International Journal on Software Tools for Technology Transfer*, Print ISSN: 1433-2779, Online ISSN: 1433-2787, vol. 18, pp. 607-628, 2016, Published by Springer-Berlin, Heidelberg, DOI: 10.1007/s10009-015-0399-5, Available: https://link.springer.com/article/10.1007/s10009-015-0399-5.

[15] Wil Van Der Aalst, Ton Weijters and Laura Maruster, "Workflow mining: Discovering process models from event logs", *IEEE Transactions on Knowledge and Data Engineering*, ISSN: 1041-4347, vol.16, no.9, pp. 1128-1142, 2004, Published by IEEE, DOI: 10.1109/TKDE.2004.47, Available: https://ieeexplore.ieee.org/document/1316839.

[16] Wahiba Ben Abdessalem Karaa, Amira S. Ashour, Dhekra Ben Sassi, Payel Roy, Noreen Kausar *et al.*, "Medline text mining: An enhancement genetic algorithm based approach for document clustering", in Intelligent Systems Reference Library: Applications of Intelligent Optimization in Biology and Medicine, Switzerland: Springer, vol 96. Online ISBN: 978-3-319-21212-8, Print ISBN: 978-3-319-21211-1, vol. 96, pp. 267–287, 2016, DOI: 10.1007/978-3-319-21212-8_12, Available: https://link.springer.com/chapter/10.1007/978-3-319-21212-8_12.

[17] Illhoe Hwang and Young Jae Jang, "Process Mining to Discover Shoppers' Pathways at a Fashion Retail Store Using a WiFi-Base Indoor Positioning System", *IEEE Transactions on Automation Science and Engineering*, Print ISSN: 1545-5955, Online ISSN: 1558-3783, vol. 14, no. 4, pp. 1786–1792, 2017, Published by IEEE, DOI: 10.1109/TASE.2017.2692961, Available: https://ieeexplore.ieee.org/document/7926395.

[18] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior", *Information Systems*, ISSN: 0306-4379, vol.33, no.1, pp. 64-95, 2008, Published by Elsevier, DOI: 10.1016/j.is.2007.07.001, Available: https://www.sciencedirect.com/science/article/abs/pii/S030643790700049X.

[19] Wil van der Aalst, Arya Adriansyah and Boudewijn van Dongen, "Replaying history on process models for conformance checking and performance analysis", *Wiley Interdisciplinary Reviews Data Mining and Knowledge*

Discovery, Online ISSN: 1942-4795, vol. 2, no.2, pp. 182-192, 2012, Published by Wiley, DOI: 10.1002/widm.1045, Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1045.

[20] Roberto Gatta, Mauro Vallati, Jacopo Lenkowicz, Calogero Casà, Francesco Cellini *et al.*, "A framework for event log generation and knowledge representation for process mining in healthcare", in *Proceedings of the IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI*) 2018, Online ISSN: 2375-0197, Print ISSN: 1082-3409, November 2018, pp. 647–654, Published by IEEE, DOI: 10.1109/ICTAI.2018.00103, Available: https://ieeexplore.ieee.org/document/8576101.

[21] S. Suriadi, R. Andrews, A. H. M. ter Hofstede and M. T. Wynn, "Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs", *Information Systems*, ISSN: 0306-4379, vol. 64, pp. 132–150, 2017, Published by Elsevier, DOI: 10.1016/j.is.2016.07.011, Available: https://doi.org/10.1016/j.is.2016.07.011.

[22] Jinlin Wang, Xing Wang, Yuchen Yang, Hongli Zhang and Binxing Fang, "A review of data cleaning methods for web information system", *Computers, Materials & Continua*, Print ISSN: 1546-2218, Online ISSN: 1546-2226, vol. 62, no. 3, pp. 1053–1075, 2020, Published by Tech Science Press, DOI: 10.32604/cmc.2020.08675, Available: https://www.techscience.com/cmc/v62n3/38341.

[23] Razi Ahmed, Muhammad Faizan and Anwer Irshad Burney, "Process Mining in Data Science: A Literature Review", in Proceedings of the 2019 *13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, 2019, ISBN: 978-1-7281-4956-1, pp. 1–9, Published by IEEE, DOI: 10.1109/MACS48846.2019.9024806, Available: https://ieeexplore.ieee.org/document/9024806.

[24] Alex Meincheim, Cleiton dos Santos Garcia, Julio Cesar Nievola and Edson Emfilio Scalabrin, "Combining process mining with trace clustering: Manufacturing shop floor process-an applied case", *Proceedings of 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Online ISSN: 2375-0197, November 2017, pp. 498–505, Published by IEEE, DOI: 10.1109/ICTAI.2017.00082, Available: https://ieeexplore.ieee.org/document/8371985.

[25] Pan Wang, Wen'an Tan, Anqiong Tang and Kai Hu, "A novel trace clustering technique based on constrained trace alignment", *Lecture Notes in Computer Science (LNCS)*, Online ISBN: 978-3-319-74521-3, Print ISBN: 978-3-319-74520-6, vol. 10745 pp. 53–63, 2018, Published by Springer, Cham, DOI: 10.1007/978-3-319-74521-3_7, Available: https://link.springer.com/chapter/10.1007/978-3-319-74521-3_7.

[26] Nurshazwani Muhamad Mahfuz, Marina Yusoff and Zakiah Ahmad, "Review of single clustering methods", *International Journal of Artificial Intelligence*, ISSN: 2252-8938, vol. 8, no. 3, pp. 221–227, 2019, DOI: 10.11591/ijai.v8.i3.pp221-227, Available: https://ijai.iaescore.com/index.php/IJAI/article/view/20265.

[27] Norsyela Muhammad Noor Mathivanan, Nor Azura Md.Ghani and Roziah Mohd Janor, "A comparative study on dimensionality reduction between principal component analysis and k-means clustering", *Indonesian Journal of Electrical Engineering and Computer Science*, ISSN: 2502-4752, vol. 16, no. 2, pp. 752–758, 2019, Published by Institute of Advanced Engineering and Science (IAES), DOI: 10.11591/ijeecs.v16.i2.pp752-758, Available: http://ijeecs.iaescore.com/index.php/IJEECS/article/view/19983.

[28] Zhuo Zhou, Jiaohua Qin, Xuyu Xiang, Yun Tan, Qiang Liu and Neal N. Xiong, "News text topic clustering optimized method based on TF-iDF algorithm on spark", *Computers, Materials & Continua*, Print ISSN: 1546-2218, Online ISSN: 1546-2226, vol. 62, no. 1, pp. 217–231, 2020, Published by Tech Science Press, DOI: 10.32604/cmc.2020.06431, Available: https://www.techscience.com/cmc/v62n1/38108.

[29] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, Print ISSN: 0360-0300, Online ISSN: 1557-7341, vol. 31, no.3, pp. 264–323, 1999, Published by Association for Computing Machinery, DOI:10.1145/331499.331504, Available: https://dl.acm.org/doi/10.1145/331499.331504.

[30] B. F. A. Hompes, J. C. A. M. Buijs, W. M. P. van der Aalst, P. M. Dixit, and J. Buurman, - "Discovering deviating cases and process variants using trace clustering", in *Proceedings of the 27th Benelux Conference on Artificial Intelligence*, 5-6 November 2015, Hasselt, Belgium, Published by Springer, Cham, Available: https://pure.tue.nl/ws/portalfiles/portal/54308814/Discovering_Deviating_Cases_and_Process_Variants_Using_Trace_Clustering.pdf.

[31] Claudia Diamantini, Laura Genga and Domenico Potena, "Behavioral process mining for unstructured processes", *Journal of Intelligent Information Systems*, Print ISSN: 0925-9902, Online ISSN: 1573-7675, vol. 47, no. 1, pp. 5–32, 2016, DOI: 10.1007/s10844-016-0394-7, Available: https://link.springer.com/article/10.1007/s10844-016-0394-7.

[32] Krisztina Tóth, Károly Machalik, György Fogarassy and Ágnes Vathy-Fogarassy, "Applicability of process mining in the exploration of healthcare sequences", in *IEEE 30th Jubilee Neumann Colloquium, NC 2017*, ISBN: 9781538646373, Published by IEEE, vol. 1, pp. 151–156, Published by IEEE, DOI: 10.1109/NC.2017.8263273, Available: http://ieeexplore.ieee.org/document/8263273/.

[33] H. M. W. Verbeek and Wil M. P. van der Aalst and J. Munoz-Gama, "Divide and Conquer: A Tool Framework for Supporting Decomposed Discovery in Process Mining", *The Computer Journal*, Print ISSN: 0010-4620, Online ISSN: 1460-2067, vol. 60, no. 11, pp. 1649–1674, 2017, Published by Oxford University Press, DOI: 10.1093/comjnl/bxx040, Available: https://academic.oup.com/comjnl/article-abstract/60/11/1649/3804254.

[34] Shazlyn Milleana Shaharudin, Shuhaida Ismail, Siti Mariana Che Mat Nor and Norhaiza Ahmad, "An efficient method to improve the clustering performance using hybrid robust principal component analysis-spectral

biclustering in rainfall patterns identification", *International Journal of Artificial Intelligence*, ISSN: 2252-8938, vol. 8, no. 3, pp. 237–243, 2019, Published by World Scientific, DOI: 10.11591/ijai.v8.i3.pp237-243, Available: https://ijai.iaescore.com/index.php/IJAI/article/view/20269.

[35] Gianluigi Greco, Antonella Guzzo, Luigi Pontieri and Domenico Sacca, "Discovering expressive process models by clustering log traces", *IEEE Transactions on Knowledge and Data Engineering*, ISSN: 1041-4347, vol. 18, no. 8, pp. 1010–1027, 2006, DOI: 10.1109/TKDE.2006.123, Available: https://ieeexplore.ieee.org/document/1644726.

[36] Mohammadreza Fani Sani, Sebastiaan J. van Zelst and Wil M. P. van der Aalst, "Improving process discovery results by filtering outliers using conditional behavioural probabilities", in *Lecture Notes in Business Information Processing*, 2018, Online ISBN: 978-3-319-74030-0, Print ISBN: 978-3-319-74029-4, vol. 308, pp. 216–229, 2018, DOI: 10.1007/978-3-319-74030-0_16, Available: https://link.springer.com/chapter/10.1007/978-3-319-74030-0_16.

[37] Nicolas Pasquier and Sujoy Chatterjee, "Customer choice modelling: A multilevel consensus clustering approach," *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, Published by International Association of Educators and Researchers (IAER), vol. 5, no. 2, pp. 103–120, 2021, DOI: 10.33166/AETiC.2021.02.009, Available: http://aetic.theiaer.org/archive/v5/v5n2/p9.pdf.