A Generic Process Mining Framework for Uncovering Hierarchical Process Model

Junya Tang*
College of Computer Science and Technology
Tongji University
Shanghai, China
junyatang@tongji.edu.cn

Li Li
College of Electronics and Information Engineering
Tongji University
Shanghai, China
lili@tongji.edu.cn

Qingyun Yu
College of Electronics and Information Engineering
Tongji University
Shanghai, China
qingyunyu@tongji.edu.cn

Ying Liu
Institute of Mechanical and Manufacturing Engineering School of Engineering
Cardiff University
Cardiff, UK
liuy81@cardiff.ac.uk

Joseph Paul Zammit
Department of Industrial&Manufacturing Engineering
name of organization
Univeristy of Malta
Msida, Malta
joseph.zammit@um.edu.mt

Emmanuel Francalanza
Department of Industrial&Manufacturing Engineering
Univeristy of Malta
Msida, Malta
emmanuel.francalanza@um.edu.mt

***Abstract*—Taking advantage of the strengths of knowledge engineering and data science, process mining has recently become a popular approach to process management research. Process mining research has focused on creating process models, checking conformance and analysing bottlenecks. Previous studies have helped organisers understand and improve processes in some fields, and some specific questions (for example, discovering a finished process model from structured data) have been studied. However, for a more general study, it is essential to correlate process mining under different conditions and form a generic process mining framework. This paper proposed a generic process mining framework for uncovering hierarchical process models under different conditions. Firstly, the proposed framework unifies process model discovery approaches for structured and unstructured data, providing a general solution that can perform those. Secondly, the framework proposed an incremental solution for ongoing processes based on the approaches for completed processes. Finally, taking unstructured data as a case, a knowledge extraction-based process discovery approach is proposed to build a hierarchical process model by document clustering and sub-process modelling. Experimental studies using real-world data collected from a design project revealed the merits of the proposed approach. The proposed approach can discover more understandable, adaptive process models.**

Keywords—process mining; process management; top-down clustering; incremental process mining; knowledge e xtraction

I. INTRODUCTION

The science of process management entails observing how work is executed in a process to ensure consistency and benefit from improvements made in previous processes [1]. Process mining is a crucial methodology used in process management, which provides fact-based insights to support process improvements by discovering and analysing models from historical process data. Process mining approaches have been applied successfully in many cases, including the manufacturing industry [2-4], financial services [5], and healthcare processes [6]. The developed approaches differ considerably in algorithmic performance, data features and computational complexity. Some process mining framework under specific conditions has been proposed to structure these approaches. However, unstructured data and ongoing processes limit these frameworks significantly.

For example, 80% of knowledge-intensive process data is semi-structured or unstructured documents, including emails, meeting minutes, and conversation records. However, current process frameworks under specific conditions can not be used directly for unstructured data due to knowledge type and process model complexity. In the event log, process information is explicit; for example, activities, operators, and execution times are all structured. In unstructured data such as text data, the process information is implicit and additional information extraction is necessary. Furthermore, traditional process modelling approaches prefer using a flat and linear model such as Petri Net to show the process behaviour [7]. The flat and linear model is intuitive and easy to understand for a simple process. However, for complex knowledge-intensive processes, the complexity of flat models will increase significantly, resulting in high computational costs and complex understanding. Although many efficient process mining approaches can not be applied directly, current process mining frameworks can be used for reference in knowledge-intensive processes due to the similarity that they all consist of a series of activities.

The ongoing process is another limitation of current process mining frameworks because most previous studies are conducted on a completed process [8]. Most process discovery techniques are fully automated, which means it is impossible to interact with the algorithm or repair the model during the discovery process. As a result of these techniques, event log data is required, and a process model is returned describing an observed behaviour. Other than reapplying algorithms, there is no direct extension of existing process models, including the entire extended event data. Although some incremental process mining approaches have been developed for model repair, a structured framework for process model discovery is still missing.

In light of the problem that developed process mining frameworks can not be used under different conditions, this study proposed a generic process mining framework. In detail, the framework contains four modules: data module, model discovery module, incremental module and process analysis module. Firstly, the proposed framework unifies processes for discovering model data for unstructured and structured data, proposing a general solution that can be applied to both data types. Furthermore, the proposed framework extends completed process mining to ongoing process mining and

provides a structured incremental process mining solution. Based on the framework, this paper focuses on the process model discovery of unstructured data and conducts an in-depth case study. To overcome the shortcomings of the flat and linear process model, a knowledge extraction-based process model discovery approach is studied to build a hierarchical process model from unstructured process data.

The rest of this paper is structured as follows. Section 2 reviews relevant studies of the current process mining framework and process model discovery approaches. Section 3 outlines the proposed process mining framework and gives a hierarchical process model discovery approach for unstructured data. Section 4 reports an experimental study using real-world business process data to demonstrate the hierarchical process model. Section 5 gives the results and analysis of the experimental study. Section 6 gives the conclusion of this paper.

II. Literature Review

A. Process Mining Frameworks

The process is the heart of modern organisations, which continuously develop to satisfy changing process requirements. In the current competitive and challenging business world, it is essential to improve business processes consistently through process management [9]. Process mining provides innovation and automation support in multi-stages of process management with data science approaches and has been the trendy research approach [3] Numerous new technologies promote process mining research and increase the difficulty of technology selection in applications. Many process mining frameworks that refine these technologies have been proposed for the two main problems in process management, process model discovery and process analysis.

Some initial process mining frameworks were proposed in some specific application scenarios. For example, Rubin V [10]focused on the software process and proposed a process mining framework including data collection, model discovery, model analysis and feedback. Markovic and Pereira [11] developed a framework for reusing business process models, which introduced the concept of ontology. However, these initial frameworks are designed for specific processes, making extending them to other domains difficult. Therefore, some more universal frameworks were proposed. For example, De Leoni M [12] proposed a general framework for process mining which relates, predicts and clusters dynamic behaviour from event logs to discover the process. This framework conducted a case analysis and divided the raw event logs into various sub-logs before process discovery, improving the framework's generality. To obtain a better model, Okoye K [13] proposed a semantic-based framework that introduces extra implicit process knowledge. However, universal process frameworks have some hypothetical conditions and can not work directly for knowledge-intensive processes containing unstructured data.

Recently, more attention has been put on unstructured data (such as emails, meeting minutes, and conversation transcripts) that provide valuable information [14-17] also focused on email data. He achieved frequent activity discovery via a pattern discovery-based approach with less human intervention. Lijun Lan [18] focused on design process knowledge extraction and design process design discovery from email data collected during a transportation design project. However, these mining schemes from unstructured data have not been integrated into universal process mining frameworks, limiting the generality. Another limitation is the process type. The universal process mining frameworks always focus on completed processes. Traditionally, process mining always aims to improve current or future processes through learning from previous processes, leading to the study objects usually being completed processes. However, some new tasks, such as prediction and repair, focus on ongoing processes [19-22]. Extending current process mining frameworks to ongoing processes is necessary.

B. Process Model Discovery Approach

Process model discovery generally refers to building process models from process data. Process model discovery is the primary task in process mining and is the focus of current process management. Based on a workflow graph, Agrawal [23] presented the first concrete process model discovery approach. After that, various process model discovery approaches have been proposed to address noise, loop and invisible tasks [24, 25]. Nevertheless, the above methods of identifying process models are unsuitable for processes characterised by flexible workflows, such as industrial and product development [26]. These models suffer a weakness: The evaluation viewpoint is the paths in a flat graph mode, commonly WorkflowNets (WFN). Consequently, the discovered models are typically complex networks that are difficult to comprehend. For this reason, the concept of roadmap abstraction was used to simplify the discovered model [27]. Hierarchical graph clustering was also utilised to identify the most effective methods for collaboration [28, 29]. An approach that can discover an easy understanding of the hierarchical process model from unstructured process data is necessary.

C. A Brief Summary

According to the literature review, existing research into process mining has drawn much attention and applied to some areas. Unfortunately, current process mining frameworks have significant limitations facing different hypothetical conditions. They always focus on specific data and process types, such as event logs and completed processes. Although studies on unstructured data have been conducted, a structured scheme has not been formed. For ongoing processes, most studies still stop on model repair, which requires extra data. This paper proposed a generic process mining framework to integrate process mining approaches under different hypothetical conditions.

III. Methodology

Since no specific algorithm can be applied to all processes with diverse data and processes, researchers have to reinvent or fine-tune the scheme according to the practical application. With in-depth research, different process mining approaches share some similarities, even though they differ in data sets, specific techniques, hypothetical conditions and other symptoms. Therefore, we take process model discovery as a centre and propose a generic process mining framework, which can provide a scheme for different process mining questions according to datasets and hypothetical conditions.

A. A Generic Process Mining Framework

As shown in Fig 1, the generic process mining framework consists of four parts: data module, model discovery module, incremental module and process analysis module. The data

module aims to preprocess input data and extract features or knowledge according to needs. The processed data will be inputted into the model discovery module, including trace (i.e., a sequence of events) and extracted process information. Given the inputs, process model discovery advances to construct a hierarchical model by top-down clustering of traces and unstructured documents. The incremental module aims to discover models from ongoing processes, and the inputs are structured and unstructured data streams. It incrementally constructs and updates the hierarchical model by bottom-up abstracting of traces and unstructured data. Based on discovered process models, multi-dimensional process information such as workflow, social network, and task decomposition will be analysed in the process analysis module.

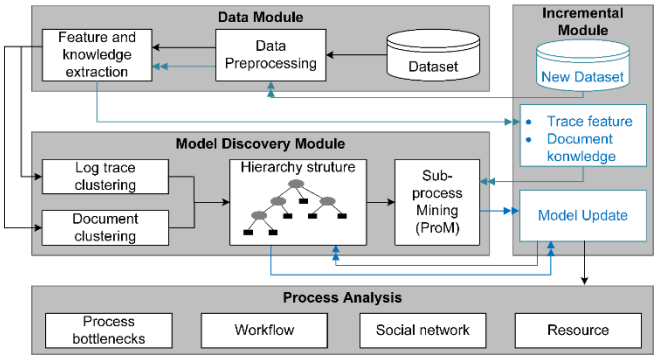


Fig. 1. A generic process mining framework

(1) Dataset Module

In this module, input data is preprocessed to extract relevant features or knowledge. For structured process data, after preprocessing (i.e. data cleaning), conduct feature extraction to determine whether individual traces are similar (usually including feature generation, transformation and selection). For unstructured process data, knowledge extraction is essential due to the implicit information. For different stages of process model discovery, the process knowledge extraction part can be divided into coarse-grained and fine-grained knowledge extraction according to the granularity of knowledge. The coarse-grained knowledge extraction focuses on extracting main topics from process documents, usually using topic modelling approaches. The topic extracted from the process documents summarises the main contents. More fine-grained process knowledge is necessary to reflect how a process was executed. To minimise human intervention, advanced techniques are applied to recognise physical objects involved in the process and their relations, such as natural language processing and knowledge graph-based approaches.

(2) Model discovery module

The model discovery module aims to discover process models from completed processes. Based on extracted knowledge, a process model discovery approach automatically models the underlying processes from the workflow viewpoint to reflect the execution of the activities based on reality. To simplify and improve the understandability of the process model, this module focuses on constructing the hierarchical process model instead of the traditional flat process model. The process model discovery scheme has two steps: model hierarchy construction and sub-process mining. According to data type, there are some differences in details.

As shown in Fig 2, the data module outputs the coarse-grain and fine-grained knowledge for unstructured data. The coarse-grained knowledge supports the model hierarchy construction through document vector generation and clustering. The fine-grained knowledge of each document cluster is then input into ProM for sub-process mining. For structured data, log trace is what we focus on. Through feature generation, feature transformation and feature selection, the data module outputs the trace features, which are input for trace clustering. ProM is then used for discovering sub-processes from each log trace cluster.

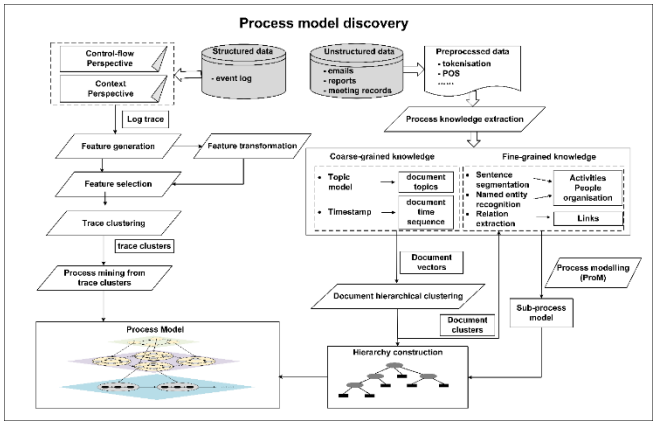


Fig. 2. Details of the model discovery module

(3) Incremental module

The model discovery module indicates the scheme of discovering a hierarchical process model from complete previous process data. It has a hypothetical condition: the process is completed, and all data can be obtained simultaneously. The incremental module gives an incremental process model discovery scheme for ongoing processes in which the hypothetical condition does not hold. This scheme also has two steps: sub-process mining and model structure construction. The incremental model discovery differs from the batch model discovery in structure construction, using a bottom-up abstraction strategy.

As shown in Fig 3, for the first dataset, we first mine its sub-process model according to its fine-grained knowledge or trace feature and then construct the hierarchical structure by merging correlated events layer by layer to obtain the initial current model. As the new dataset arrives, conduct two things: first, obtain its sub-process and update the current sub-process in the bottom layer, then the task abstraction will be conducted and update the sub-process in higher layers. In the hierarchical structure, a higher layer has a higher abstract level.

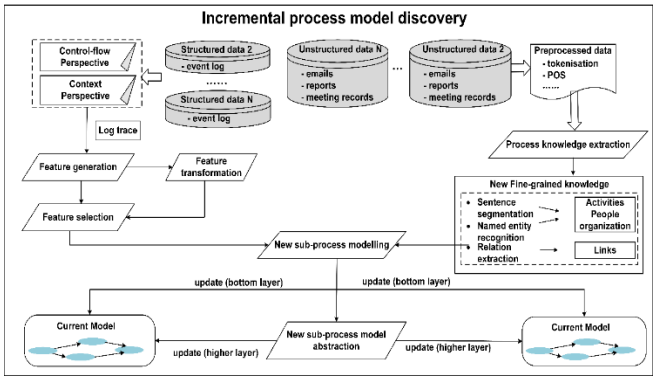


Fig. 3. Details of incremental model discovery module

(4) Process analysis module

The process analysis module aims to draw on the experience of the previous processes and supports the improvement of the current or subsequent processes. Unlike workflow analysis, the process analysis module focuses on multi-dimensional knowledge patterns, including bottlenecks and extra process information analysis.

Bottleneck analysis is crucial for improving previous processes, which is the main content in most process analysis studies. This module analyses process bottlenecks from a global perspective rather than the usual local perspective, with the help of advanced techniques, such as knowledge graph embedding. Besides the universal bottlenecks analysis, some extra process information is integrated from three dimensions, task, personnel and time. As shown in Fig 4, the discovered process model is treated as the central component, and other types of process knowledge, such as temporal process behaviours, social networks, and organisational structure, are linked.

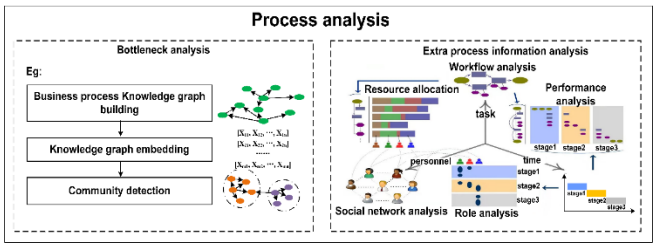


Fig. 4. Details of the process analysis module

As shown in Fig 4, the task dimension focuses on workflow and supports process improvement by analysing the rationality of task settings and assignments and finding deficiencies in task execution. The personnel dimension focuses on the participants and provides extra process information by analysing social networks, organisational role assignments, and human resource allocation [35]. The temporal dimension divides the process into several stages, supporting bottlenecks and task and personnel analysis. Studying the dynamic behaviour of processes requires the analysis of the temporal behaviour of the tasks and participants, including the duration of tasks, the duration of waiting, and the termination time, as well as the temporal and overall frequencies.

B. Hierarchical process model discovery approach for unstructured data

(1) Process knowledge extraction in the data module

For process knowledge extraction, two different granularity approaches are proposed. They are a BTM-based topic modelling approach for extracting coarse-grained knowledge and a natural language processing and knowledge graph-based approach for fine-grained knowledge [36].

Considering the different lengths of process documents, an improved BTM topic model is proposed. BTM was proposed for short texts to solve the world sparsity problem. However, with the text length increasing, redundancy in calculations will occur. A dynamic sliding window is introduced to select biterms from documents of various lengths to solve this problem. The document length is taken into consideration when adjusting the sliding window size. The selection of biterms using a sliding window is shown in Fig 5. The Glove algorithm trains word vectors to remove worthless biterms worth low correlation. Cosine similarity between word vectors is the measured metric for selecting biterms.

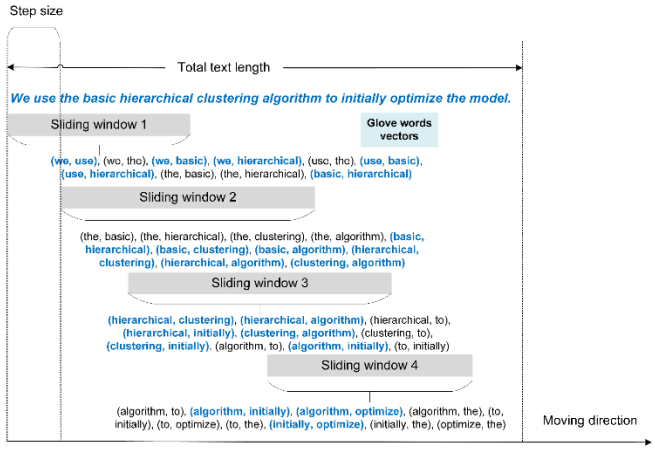


Fig. 5. Coarse-grained knowledge extraction

For fine-grained knowledge extraction, first, some natural language processing approaches formulate raw texts into a knowledge graph, including sentence segmentation, entity extraction, and relation extraction. In process documents, the sentence is more complicated because an entity can span multiple words, and some entities are composed of compound words. To address these problems, rule matching-based dependency analysis of the sentence can be used to help entity extraction. For example, *XX* such as *YY* is a pattern where the type of *YY* is found out from *XX*. Then, the relation between entities can be extracted unsupervised using the sentences' grammar. Nodes and edges are the two main elements in a knowledge graph. The entities and relations extracted are nodes and edges to build a process knowledge graph.

(2) Top-down hierarchical process modelling

According to Fig 2, a top-down hierarchical process model discovery approach was proposed. The approach involves two steps, hierarchy model structure construction and sub-processing model discovery. The process is viewed as a black box in the first step. Then, divide the process into smaller parts to construct a tree structure via document clustering based on each document topic distribution. Each small part can be decomposed until it achieves the desired homogeneity. The second step is mining subprocesses from the document cluster associated with each part. Within each part, the fine-grained knowledge extraction approach is used to extract process knowledge, and a flat process model discovery approach is

used to model the sub-process. The top-down hierarchical process model discovery approach is as follows:

TABLE I. ALGORITHM 1

Algorithm 1: Hierarchical process model discovery through top-down process mining

Input: $D = \{d_1, d_2, \dots, d_n\}$ is the document set, $C = \{c_1, c_2, \dots, c_m\}$ is the document cluster, $D_{c_i}$is the document set in the cluster c_i, $Z = \{z_1, z_2, \dots, z_k\}$ is the topic set, P is the process, SP is the sub-process, d_intra_c is the average intra-cluster distance, and γ is the threshold.

Procedure:

For each d_i in D, do:

Document vector:

$d_i = \{p(z_1|d_i), p(z_2|d_i), \dots, p(z_k|d_i)\}$

Initialisation: $C = D, SP = \emptyset$

While $|C| \neq 0$ do:

If $d\_intra_C \geq \gamma$, do:

$C_{new} = document\ clustering\ (C, Z_C)$,

$C = C \cup C_{new}$

For each c in C, do:

$SP = sub - process\ modelling(c)$

(3) Incremental bottom-up hierarchical process modelling

The incremental bottom-up process model discovery approach is shown in Fig 6. When a new dataset arrives, it will be input into the data module to extract knowledge. According to fine-grained knowledge, discover the sub-process model at the bottom layer and add it to the current sub-process model at the bottom. Based on the updated model at the bottom layer, model abstraction will be conducted layer by layer to obtain the final hierarchical process model until the highest layer is reached (the highest layer number is L).

- Sub-process update：

As shown in Fig 6, new sub-processes will be generated at some layers after the new dataset arrives, which leads to the sub-process updating at these layers. Before updating the sub-process, it is essential to develop appropriate mechanisms to measure how to connect previous and new sub-processes. Two points need to be considered: connect events and their priority relationship.

Using a four-tuple $SP = (A, E, A^s, A^e)$ to represent a flat process model. A is a finite set of events, A^s is the set of starting events, A^e is the set of ending events, $E \subseteq (A - A^s) \times (A - A^e)$ is the relations between events.

① Donate the previous sub-process as $SP_{previous} = (A_{previous}, E_{previous}, A^s_{previous}, A^e_{previous})$, and donate the new sub-process as $SP_{new} = (A_{new}, E_{new}, A^s_{new}, A^e_{new})$. $|A^e_{previous}|$ is the number of events in $A^e_{previous}$ and $|A^s_{new}|$ is the number of events in A^s_{new}. According to different $|A^e_{previous}|$ and $|A^s_{new}|$, set different connect mechanisms. $|A^e_{previous}| = 1$ and $|A^s_{new}| = 1$: the event $a_{new} \in A^s_{new}$ is executed following the event $a_{previous} \in A^e_{previous}$.

② $|A^e_{previous}| > 1$ or $|A^s_{new}| > 1$: known the execution time of two events, $a_{previous\_i} \in A^e_{previous}$, $a_{new\_j} \in A^s_{new}$, measuring the possibility of the relation existing between $a_{previous\_i}$ and a_{new_j} as following :

The execution time interval between $a_{previous\_i}$ and a_{new_j} is donated as Δt, the time window between two connected events is T. Set $p\left(a_{previous\_i}, a_{new\_j}\right) = 1 - \Delta t/T$, if $p\left(a_{previous\_i}, a_{new\_j}\right) \in (0,1)$, consider a_{new_j} is executed following $a_{previous\_i}$, otherwise, consider no relation existing between a_{new_j} and $a_{previous\_i}$.

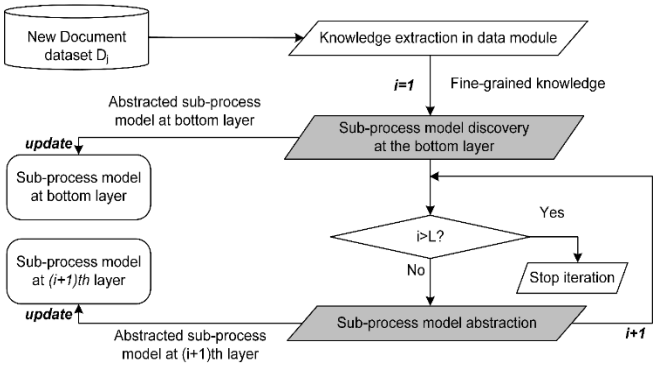


Fig. 6. The incremental bottom-up process modelling scheme

- Sub-process abstraction：

As shown in Fig 3, the incremental module uses a bottom-up mechanism to obtain hierarchical process models. The bottom-up mechanism obtains abstracted sub-processes with different levels at different layers by merging highly correlated small events. Therefore, the event in a higher layer is the abstraction of a set of correlated events in a lower layer, and events in a lower layer are detailed executions of the event in a higher layer. Two fundamental metrics are selected to measure the correlation between small events: neighbourhood ship [33]and context similarity.

The neighbourhoodship measures the time interval of executing two events because events that execute close have a higher probability of being correlated. In detail, for sub-process SP, for each event $a \in$ A, its correlated candidate event is defined as :

$$C(a) = \{\forall a_i \in A_{-a} | (a, a_i) \cup (a_i, a) \in E \wedge |t(a_i) - t(a)| < \tau\} \quad (1)$$

The similarity in context measures the degree of overlap between two events regarding their attributes [18]. In detail, for events a_i and a_j in sub-process SP, two vectors v_i and v_j can be generated by word2vector. The cosine distance $dis(v_i, v_j)$ between v_i and v_j is:

$$\mathrm{dis}(v_i, v_j) = \frac{v_i \cdot v_j}{|v_i| \times |v_j|} \quad (2)$$

The context similarity Sim is :

$$Sim = \frac{1}{dis(v_i, v_j) + 1} \quad (3)$$

Based on the above two metrics, correlated candidates with similar contexts can be merged into an abstracted event in a higher layer. Fig 7 shows an example.

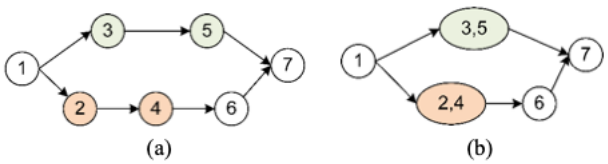


Fig. 7. Example of sub-process abstraction

IV. Experimental Study

The experimental study was mainly conducted using the historical data collected from a project aiming to design an Ants transportation system to track the traffic wave problem in the highway system [37]. The study object is the design process hosted by a university under this project, which is a knowledge-intensive process with unstructured process data. The complex environment of product design and the inherently uncertain nature of innovative design processes lead to an industrial reality that the traditional flat process models can not be understood well. Knowledge extraction-based hierarchical process model discovery can support designers in understanding previous models well and reduce design risks.

First, some public text collections were also utilised to demonstrate the effectiveness of the knowledge extraction approach proposed in the data module. Then an initial experimental study will be conducted to construct a hierarchical process model from the real process data.

A. Data Acquisition and Preparation

The data in this study contains real data and some public data. The real data regarding the design process were emails extracted from Outlook as an XML file. During the design process, the participants used emails to communicate with each other, including exchanging their ideas, discussing works and notifying project activities. During the two years of the project, all 569 emails were required to be sent to a shared address and stored as an XLM file [37]. This XLM file contained all information during the design process, such as activities, resources and personnel interactions. The raw data is unstructured text data, and some basic information, such as design events, participants, resources, and time can be extracted from these texts through coarse-grained and fine-grained knowledge extraction approaches. After acquiring the raw data, filtering deletes blank emails, useless information such as links and marker symbols, and some process-independent notifications. After that, 357 emails were kept for subsequent analysis. The public data is the Google News dataset that contains eight news topic categories: business, computers, culture, science, engineering, health, politics and sports. This public dataset and the real data both contain text data of varying lengths.

B. Experiment Setup

The first experimental study aims to construct a hierarchical process model from the completed process documents of the transportation design project, including hierarchical structure construction and sub-process modelling. The hierarchical structure construction has two steps, filled nodes extraction and filled nodes content extraction. Firstly, determine the document clusters of filled nodes via document clustering. Then, determine the content of each filled node via the proposed coarse-grained knowledge extraction approach (CGKE). The CGKE is a probabilistic topic model considering the computational cost and the amount of data. Because the performance of a probabilistic topic model is easily affected by document length, CGKE uses a sliding window to reduce the impact. An experiment comparing the adaptability of CGKE to different text lengths is also conducted. The CGKE has two outputs. One is the probability distribution of topics over documents, which can be utilised for the vector representation of documents. Another is the probability distribution of words over topics, which determines the content of topics. Two metrics to measure the adaptability of the CGKE to different document lengths are the document vector's quality and the topic content's accuracy.

The document vector is used to determine the number of filled nodes via clustering. Therefore, the quality of the document vector can be indirectly measured by the effect of document clustering. The coherence score can quantify the accuracy of topic content. Since the real design process data is unlabeled, a public dataset is used for the comparative experiments. The public dataset is divided into three subsets according to document length: short document, long document, and long-short document. Two classic topic models, LDA and BTM, performed well in long and short texts and were selected as the baseline. The sub-process can be discovered after obtaining filled nodes and their document clusters. In sub-process mining, fine-grained knowledge is essential. Many natural language processing technologies extract each activity in the sub-process. Input activities and time to process mining tool ProM, the sub-process can be visualised as a control-flow graph.

The second experimental study aims to conduct a hierarchical process model from the incremental process documents of the transportation design project. First, divide the design document according to time. The process lasts 23 months, with 15 -day intervals, and the involved documents are divided into 46 datasets. Then, set the indicator to stop merging. This study sets up the similarity between document clusters as the evaluation index, and layers with different levels of abstraction have different similarity thresholds. Last, set the model update mechanism. In this study, we directly connect the sub-workflow of the merged datasets.

V. Results and Discussion

A. Top-down hierarchical process model discovery

(1) Top-down hierarchical process model discovery

- Extraction of the hierarchical structure：

Document clusters were divided into smaller groups for hierarchical structure extraction via decomposition iterations. Therefore, the hierarchical structure is a tree, including the filled nodes (corresponding to document clusters) and the leaf nodes (corresponding to sub-process models). Each document cluster has a topic that is the total task of the sub-process under that filled node. Take the top layer as an example.

① Extraction filled nodes at the top layer

For the top layer, the big document cluster is the whole process document. To extract the filled nodes in the top layer, the entire process documents are decomposed into K small clusters. In this study, a fusion clustering algorithm was utilised. The algorithm has two main parameters, cluster number K and fusion coefficient λ. According to expert experience, K=7, $\lambda \in [0.2,0.9]$ is set from 0.2 to 0.9, and three common measure metrics, S score (Silhouette-score), CH

score (Calinski-Harabaz score) and DBI score (Davies-Bouldin score) are used to select the best value of λ [37].

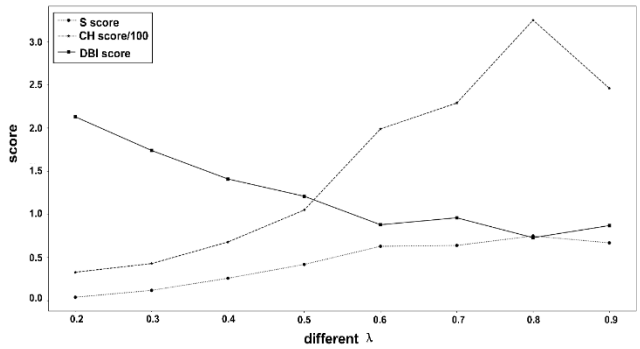


Fig. 8. Three measure metrics values under different λ

The larger S, CH, and smaller DBI scores mean better clustering performance. It can be seen from Fig 8, setting λ from 0.2 to 0.9, when λ is 0.8, the S score and CH score get the largest value while the DBI score gets the smallest value. That means for the fusion clustering algorithm, when the λ is 0.8, the statistical and semantic features of the document can be best utilised. In the top layer, the whole documents were clustered as seven small clusters, and seven filled nodes can be extracted.

② Content extraction of filled nodes

In the top layer, there are seven document clusters. So, seven topics of these clusters correspond to the content of seven filled nodes. Select the top four words to represent each topic, and the extracted topics are "concept paper and student group", "project proposal", "transportation system design", "software application and system simulation", "research paper submission and presentation", "vehicle certain and video presentation", and "entire program optimise".

LDA and BTM are two classic topic models selected as baseline methods to demonstrate the effectiveness of our proposed Coarse-grained Knowledge Extraction (CGKE) approach. The measure metrics are the quality of topics and the quality of representation documents.

The coherence score [38] was used for quality evaluation to perform a more comprehensive analysis. According to the coherence score, words belonging to the same concept will appear together in documents. It must be noted that the coherence score is only used to evaluate top words. The number of top words T is 5. The greater the value of the coherence score, the better the coherence of the topic. The average coherence score was calculated for the whole topic set, and the result is listed in Fig 9.

Fig 9shows the average score of three sub-datasets. It can be seen that CGKE receives the highest coherence score in all the settings. CGKE can extract more accurate and coherent related words from documents of different lengths, whether from visualisation or quantitative analysis.

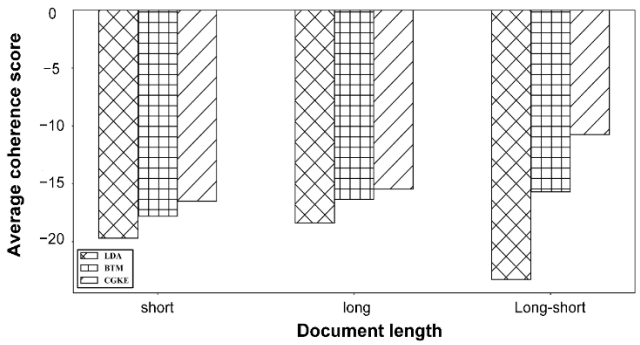


Fig. 9. Average coherence score of the 5 top words (ordered by p(w|d)) discovered by LDA, BTM and CGKE

In the approach proposed, in addition to extracting topic words, the CGKE has another important function as a dimension reduction method for document representation. Topical posterior distributions p(z_i |d) can be represented as vectors for each document. The Google News data has the topic label, and documents with the same label were organised into a cluster. The H score was used to evaluate the quality of the topical representation of documents. The criteria for assessing clusters are that they should have a low intra-cluster distance and a high inter-cluster distance. H score is the ratio of intra-cluster distance to inter-cluster distance. When the topical representation of documents aligns with labelled clusters, the average intra-cluster distance will be small compared to the average inter-cluster distance. That means the best topical representation of a document will have the smallest H score.

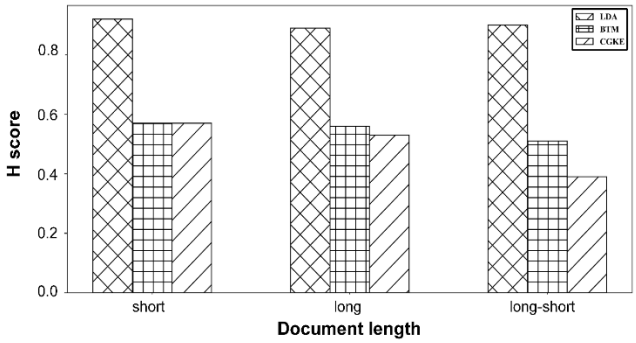


Fig. 10. H score for different methods on the Google_News collection, the smaller value is better

Fig 10 shows the H score of three algorithms on three sub-datasets. BTM and CGKE perform significantly better than LDA on all data sets. On Short News Set, BTM and CGKE get the same H score. On Long News Set and Long-Short News Set, CGKE performs better than BTM, especially on Long-Short News Set. It can be seen that CGKE maintains the advantages of BTM in short texts and significantly improves the representation of documents of different lengths.

- Sub-process modelling:

There are many leaf nodes for each filled node, and each leaf node denotes a sub-process model. As a means of conveying a more straightforward message, Fig 11 illustrates an example of the sub-process model represented by the leaf node.

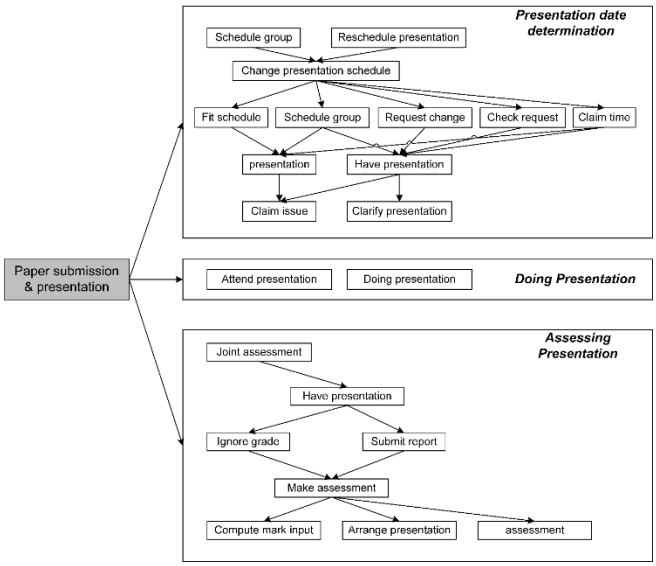


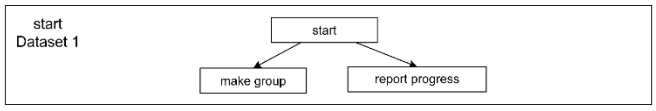
Fig. 11. Example of the sub-process model relating to "paper submission & presentation"

From Fig 11, the "paper submission & presentation" task has three sub-processes connected hierarchically. Among the three sub-processes, the first one illustrates how the presentation data is determined. The scheduling and rescheduling of events are displayed clearly. The second describes how the presentation is prepared, and the third illustrates how the presentation is assessed. In the hierarchical process model, each leaf node corresponds to a sub-process.

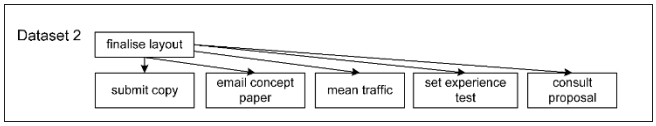
B. Incremental bottom-up hierarchical process model discovery

(1) Incremental sub-process model discovery in the bottom layer

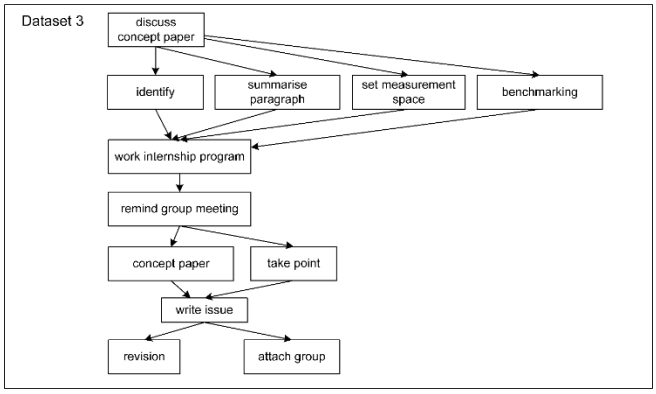
Fig 13 describes several sub-process models discovered in the bottom layer. When a new dataset arrives, the detailed sub-process will first be discovered at the bottom layer, such as Fig 13(a) and Fig 13(b). The end set of the first sub-process model includes two events, and the start set of the second sub-process model has one event. According to the updating mechanism, the correlations between the end events and the start event need to be measured. For example, in Fig 13(a), the end entities are "make group" and "report progress", and in Fig 13(b), the start entities are "finalise layout". The two end entities are all connected to the start entity because their correlation values are over the threshold.



(a)



(b)

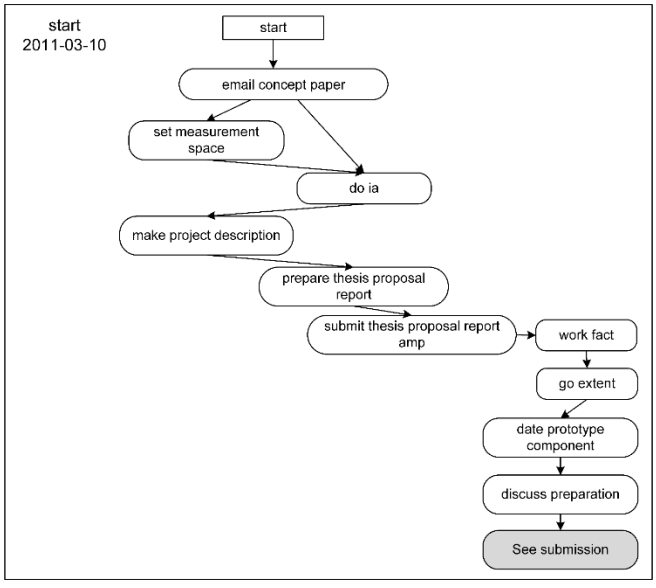


(c)

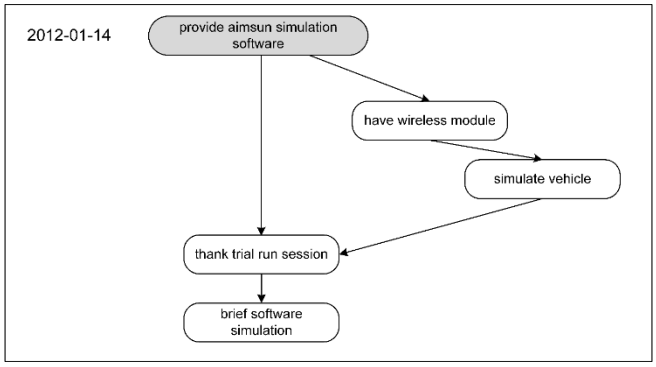
Fig. 12. Examples of detailed sub-process models at the bottom layer ((a) is the sub-process model of the first dataset, (b) is the sub-process model of the second dataset, and (c) is the sub-process model of the third dataset.)

(2) Incremental sub-process model discovery in the abstract layer

The sub-flat model captured the detailed execution of every event. However, such a detailed model can not provide a quick and clear view of the underlying process, leading to model application inefficiency. Therefore, it is essential to simplify the detailed model in the bottom layer. Fig 14 shows the sub-process model in a higher layer, which is more abstract. In the abstracted sub-process model, each task comprises several small tasks in the lower layer. For example, the sub-process model in the bottom layer shown in Fig 13 is abstracted as a task "email concept paper". As a new dataset arrives, we will first obtain its detailed sub-process model in the bottom layer. The detailed model will generate a new hierarchical model with different layer abstract levels, such as Fig 14(b). The new hierarchical model is then used to update the current hierarchical mode by connecting the flat models in corresponding layers.



(a)



(b)

Fig. 13. Abstracted sub-process models in a higher layer ((a) is the abstracted model of the previous dataset, and (b) is the abstracted model of the new dataset. Two models are at the same layer.)

Fig 13 (b) describes the sub-process model of the second dataset, and five events were found. The end set of the first sub-process model includes one event, and the start set of the second sub-process model also has one event. According to the updating mechanism, the start entity will be directly connected to the end entity.

Compared to the detailed flat model, the abstract model shows a clearer workflow of this project, which indicates that the abstracted sub-process model in the top layer can provide a fairly concise and brief description of the entire process.

C. Discussion

Process management is important to the industry. The discovery of process models has long been a concern for various researchers. Process mining provides new methods for process management by linking process science and data science. Some specific questions have been studied during previous research (for example, discovering a finished process model from structured data). However, a generic process mining framework summarising the previous and refining a universal scheme is missing. Due to that, this study gives a generic process mining framework that illustrates different process mining paths according to data and process characteristics. The specific approaches of each part in the framework are not fixed and can be selected according to the data. Unlike current process mining frameworks, the proposed framework considers structured and unstructured data in one scheme. Furthermore, an incremental scheme is proposed for the first time to address the ongoing process mining problem, which can inspire more online process mining approaches and support model repair.

Under this framework, we give a case study focusing on discovering process models from unstructured data, including completed and ongoing processes. We divided the case study into knowledge extraction, completed process model discovery and incremental process model discovery. The first experiment indicates that introducing knowledge extraction methods to process mining can extend current approaches to various process data. Coarse-grained knowledge can provide overall information about the process, such as the main tasks. Fine-grained knowledge can provide detailed information to support process modelling. This study uses an improved topic model to extract coarse-grained knowledge. If the process data is big, deep learning approaches are also suitable. Knowledge extraction is the foundation of the hierarchical process model discovery on unstructured data.

For completed process discovery, this study uses a top-down modelling approach that extracts the hierarchical structure of the model and mines its specific sub-processes. For incremental process discovery, a bottom-up strategy is conducted. Although this study reveals essential findings, there are also limitations. The proposed top-down process model discovery approach is highly dependent on knowledge extraction in the data module and has high requirements for the quality of knowledge extraction. When replying to a specific domain, researchers must adjust the knowledge extraction method according to the characteristics of the data. The proposed bottom-up incremental process model discovery approach may generate extra loops when merging models from a low level to a higher level of abstraction.

It should also be noted that process analysis is introduced for the integrity of the framework. Still, this part is independent content, and this study focuses on discovering the process model, so this part will not be studied in depth in the case study.

VI. Conclusions

A well-understood process model can significantly benefit process analysis and improvement and be the foundation for many other process management tasks. Process mining is the main approach for discovering process models. In this study, we have studied the possibility of achieving so by proposing a generic process mining framework to reduce the limitation that hypothetical conditions bring to process mining approaches. Novelties of the proposed framework include (1) unifying a scheme from process model discovery approaches focusing on structured and unstructured data, (2) extending the scheme for completed processes to ongoing processes, and (3) giving a multi-perspective process analysis direction. Furthermore, a case study that discovers a hierarchical process model from completed and incremental unstructured design process data was conducted based on the process mining framework, and a knowledge extraction-based process discovery approach was proposed. The proposed approach has been tested, and the results provided evidence that the proposed approach can reveal the actual executions of past design processes, both completed and ongoing.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 72171172 and 92367101; the Aeronautical Science Foundation of China under Grant 2023Z066038001; the National Natural Science Foundation of China Basic Science Research Center Program under Grant 62088101; Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100; Chinese Academy of Engineering, Strategic Research and Consulting Program, under Grant 2023-XZ-65.

References

[1] E. Lamine, R. Thabet, A. Sienou, D. Bork, F. Fontanili, and H. Pingaud, "BPRIM: An integrated framework for business process management and risk management," *Computers in Industry,* vol. 117, pp. 103199, 2020.

[2] M. Vještica, V. Dimitrieski, M. M. Pisarić, S. Kordić, S. Ristić, and I. Luković, "Production processes modelling within digital product manufacturing in the context of Industry 4.0," *International Journal of Production Research,* vol. 61, no. 19, pp. 6271-6290, 2023.

[3] R. Lorenz, J. Senoner, W. Sihn, and T. Netland, “Using process mining to improve productivity in make-to-stock manufacturing,” *International Journal of Production Research,* vol. 59, no. 16, pp. 4869-4880, 2021.

[4] Y. Otsubo, N. Otani, M. Chikasue, M. Nishino, and M. Sugiyama, “Root cause estimation of faults in production processes: A novel approach inspired by approximate Bayesian computation,” *International Journal of Production Research,* vol. 61, no. 5, pp. 1556-1574, 2023.

[5] F. Aydemir, Y. U. Pabuccu, and F. Basciftci, “A hybrid process mining approach for business processes in financial organizations,” *Procedia Computer Science,* vol. 158, pp. 244-253, 2019.

[6] D. Schuster, S. J. van Zelst, and W. M. van der Aalst, "Incremental discovery of hierarchical process models." pp. 417-433.

[7] H. Sun, W. Liu, L. Qi, X. Ren, and Y. Du, “An algorithm for mining indirect dependencies from loop-choice-driven loop structure via petri nets,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 52, no. 9, pp. 5411-5423, 2021.

[8] G. Park, D. Schuster, and W. M. van der Aalst, “Pattern-based action engine: Generating process management actions using temporal patterns of process-centric problems,” *Computers in Industry,* vol. 153, pp. 104020, 2023.

[9] H. A. Reijers, “Business Process Management: The evolution of a discipline,” *Computers in Industry,* vol. 126, pp. 103404, 2021.

[10] V. Rubin, C. W. Günther, W. M. Van Der Aalst, E. Kindler, B. F. Van Dongen, and W. Schäfer, "Process mining framework for software processes." pp. 169-181.

[11] I. Markovic, and A. C. Pereira, "Towards a formal framework for reuse in business process modeling." pp. 484-495.

[12] M. De Leoni, W. M. Van Der Aalst, and M. Dees, “A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs,” *Information Systems,* vol. 56, pp. 235-257, 2016.

[13] K. Okoye, S. Islam, U. Naeem, M. S. Sharif, M. A. Azam, and A. Karami, "The application of a semantic-based process mining framework on a learning process domain." pp. 1381-1403.

[14] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, and J. Cheng, “Towards comprehensive support for privacy preservation cross-organization business process mining,” *IEEE Transactions on Services Computing,* vol. 12, no. 4, pp. 639-653, 2016.

[15] M. Mesabbah, W. Abo-Hamad, and S. McKeever, "A hybrid process mining framework for automated simulation modelling for healthcare." pp. 1094-1102.

[16] K. Nadim, A. Ragab, and M.-S. Ouali, “Data-driven dynamic causality analysis of industrial systems using interpretable machine learning and process mining,” *Journal of Intelligent Manufacturing,* vol. 34, no. 1, pp. 57-83, 2023.

[17] M. Elleuch, O. A. Ismaili, N. Laga, W. Gaaloul, and B. Benatallah, "Discovering activities from emails based on pattern discovery approach." pp. 88-104.

[18] L. Lan, Y. Liu, and W. Feng Lu, “Automatic discovery of design task structure using deep belief nets,” *Journal of Computing and Information Science in Engineering,* vol. 17, no. 4, pp. 041001, 2017.

[19] A. Burattin, M. Cimitile, F. M. Maggi, and A. Sperduti, “Online discovery of declarative process models from event streams,” *IEEE Transactions on services computing,* vol. 8, no. 6, pp. 833-846, 2015.

[20] A. Burattin, S. J. van Zelst, A. Armas-Cervantes, B. F. van Dongen, and J. Carmona, "Online conformance checking using behavioural patterns." pp. 250-267.

[21] S. J. van Zelst, A. Bolt, M. Hassani, B. F. van Dongen, and W. M. van der Aalst, “Online conformance checking: relating event streams to process models using prefix-alignments,” *International Journal of Data Science and Analytics,* vol. 8, pp. 269-284, 2019.

[22] N. Navarin, M. Cambiaso, A. Burattin, F. M. Maggi, L. Oneto, and A. Sperduti, "Towards online discovery of data-aware declarative process models from event streams." pp. 1-8.

[23] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining process models from workflow logs." pp. 467-483.

[24] C. Ou-Yang, H.-J. Cheng, and Y.-C. Juan, “An Integrated mining approach to discover business process models with parallel structures: towards fitness improvement,” *International Journal of Production Research,* vol. 53, no. 13, pp. 3888-3916, 2015.

[25] L. Wen, J. Wang, W. M. van der Aalst, B. Huang, and J. Sun, “Mining process models with prime invisible tasks,” *Data & Knowledge Engineering,* vol. 69, no. 10, pp. 999-1021, 2010.

[26] D. Repta, M. A. Moisescu, I. S. Sacala, I. Dumitrache, and A. M. Stanescu, “Towards the development of semantically enabled flexible process monitoring systems,” *International Journal of Computer Integrated Manufacturing,* vol. 30, no. 1, pp. 96-108, 2017.

[27] C. W. Günther, and W. M. Van Der Aalst, "Fuzzy mining–adaptive process simplification based on multi-perspective metrics." pp. 328-343.

[28] C. Diamantini, L. Genga, and D. Potena, “Behavioral process mining for unstructured processes,” *Journal of Intelligent Information Systems,* vol. 47, pp. 5-32, 2016.

[29] J. Liu, J. Wang, X. Liu, T. Ma, and Z. Tang, “MWRSPCA: Online fault monitoring based on moving window recursive sparse principal component analysis,” *Journal of Intelligent Manufacturing*, pp. 1-17, 2022.

[30] S. Nannapaneni, S. Mahadevan, A. Dubey, and Y.-T. T. Lee, “Online monitoring and control of a cyber-physical manufacturing process under uncertainty,” *Journal of Intelligent Manufacturing,* vol. 32, no. 5, pp. 1289-1304, 2021.

[31] R. G. de Sousa, S. M. Peres, M. Fantinato, and H. A. Reijers, "Concept drift detection and localization in process mining: An integrated and efficient approach enabled by trace clustering." pp. 364-373.

[32] R. Zaman, A. Cuzzocrea, and M. Hassani, "An innovative online process mining framework for supporting incremental GDPR compliance of business processes." pp. 2982-2991.

[33] S. Ferilli, "Incremental declarative process mining with woman." pp. 1-8.

[34] A. Armas Cervantes, N. R. van Beest, M. La Rosa, M. Dumas, and L. García-Bañuelos, "Interactive and incremental business process model repair." pp. 53-74.

[35] Y. Zhang, S. Zhang, R. Huang, B. Huang, J. Liang, H. Zhang, and Z. Wang, “Combining deep learning with knowledge graph for macro process planning,” *Computers in Industry,* vol. 140, pp. 103668, 2022.

[36] X. Cheng, X. Yan, Y. Lan, and J. Guo, “Btm: Topic modeling over short texts,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 12, pp. 2928-2941, 2014.

[37] J. Tang, L. Li, Y. Liu, and K.-y. Lin, “Automatic identification of bottleneck tasks for business process management using fusion-based text clustering,” *IFAC-PapersOnLine,* vol. 54, no. 1, pp. 1200-1205, 2021.

[38] R. Zhou, A. Awasthi, and J. Stal-Le Cardinal, “The main trends for multi-tier supply chain in Industry 4.0 based on Natural Language Processing,” *Computers in Industry,* vol. 125, pp. 103369, 2021.