Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 16 / ISSUE 2 / MAY 2021 / 34-62
© 2021 Universidad de Talca - Chile

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762021000200104

A Novel Method for Discovering Process Based on the Network Analysis Approach in the Context of Social Commerce Systems

Leila Esmaeili[1] and Alireza Hashemi Golpayegani[2*]

[1] Amirkabir University of Technology, Department of Computer Engineering and Information Technology, Tehran, Iran, Leila.Esmaeili@aut.ac.ir
[2] Amirkabir University of Technology, Department of Computer Engineering and Information Technology, Tehran, Iran, Sa.Hashemi@aut.ac.ir

Received 22 June 2019; received in revised form 16 April 2020; accepted 12 May 2020

Abstract

Process mining in the context of information systems, which consists of information flows, has been one of the major research areas in the past decade. One of the most common objectives of process mining is the automated business process discovery. There are many challenges in the business process discovery, such as spaghetti models, same-name activities, and discovering loop structures. The researchers have presented a variety of methods that focus on one or more challenges. Due to the importance of commercial systems and the diversity of flows in them, in this research, the process mining problem in the context of social commerce systems is studied. Moreover, the research objective is to present a new method for commercial process discovery that has not been considered before. The proposed method is based on network analysis methods, multi-layered networks (networks with heterogeneous relations), and attributed networks. The results obtained from the proposed method are more precise and more comfortable to understand than the previous ones.

Keywords: Process mining, Process discovery, Network analysis, Community detection, Motifs analysis, Motif discovery, Multi-layered network, Attributed network, Commercial process

1 Introduction

Process mining is a relatively novel research area that, on the one hand, resides between intelligent computations and data mining, and, on the other hand, it resides between process modeling and analysis [81]. The idea of process mining is discovering, monitoring, and improving real processes by extracting knowledge from events log, which is available in today's systems. Moreover, process mining includes three main subjects: process, organization, and case. In the process approach, process mining includes three primary goals. Automated process discovery (i.e., extracting process models from an events log), conformance checking (i.e., monitoring differences by comparing the model with the events log), and improvement (i.e., extending or improving an existing process model employing information about the actual process recorded in some events log) [80], [81].

The automated (business) process discovery has been taken into consideration by practitioners and academics more than other objectives of process mining. The main driver for the growing interest in this area is the need to improve and support business processes in a competitive and swiftly changing environment [81]. The discovered process can provide answers to questions from process analysts and process managers and launch redesign or set up actions [81]. Process discovery empowers organizations to compare the conduct of the process in the events log with the business behavior it would expect from its employees and other stakeholders. Such delta analysis [79] can be beneficial in the context of guaranteeing compliance with new regulations or in the context of business process redesign and optimization. The mismatch of the discovered process and the actual process (designed process) behavior can have multiple causes. Fraud, model incompatibility, process misunderstanding, or development of a model for ideal and unrealistic conditions might be some of the reasons of the unconformity [1].

Despite various studies to discover the business process, some of the most important concerns and challenges associated with the process discovery include the issues of invisible tasks, duplicate tasks, loops, noises, non-free choices, and spaghetti models. Studies that attempt to propose effective methods to solve these issues continue. Furthermore, the use of new approaches and the combination of process mining with other types of analysis techniques to handle problems are other research goals. In addition, studies have so far been conducted in the context of information systems that only include information flows [1], [4], [6], [20], [50], [81], [92]. In this research, we intend to answer this question: Are the previous methods of process model discovery, applicable in other different domain which contains various flows?

Social commerce points out to the usage of the Web 2.0 and its applications in electronic commerce. A significant number of academics, businesses, researchers, and practitioners have been attracted to the social commerce area. An Electronic Social Commerce System (ESCS) includes commercial activities along with social activities. All stakeholders (such as business personnel, customers, partners, and services) can interact with electronic social commerce systems and with each other via provided interfaces [24]. Eelectronic social commerce system includes social relations [9], [31], [24], also information, goods/services, and financial flows [8], [63]. Furthermore, in this context, some commercial procedures are infrequent, so the discovery of them is an issue. Thus, our main research objective is to discover a more precise and straightforward commercial process in the context of electronic social commerce systems from a business process management viewpoint. The features of our proposed method include overcoming the issues of spaghetti models, identifying loop structures, noise handling and increasing the accuracy of the discovered model.

In this research, our main contribution is discovering the process through social commerce systems using network analysis methods and graph theory. In our proposed method, not only the basic concepts of network analysis are employed, but also modeling multi-layered networks, community detection, attributed networks, and motifs discovery and analysis are applied. Such a combination was not presented before. The metrics and methods of network analysis (or social network analysis) have been used to identify the critical and influential resources in organizational mining (one of three process mining approaches) [37], [71], [77], [84]. Furthermore, recently, Appice has applied community detection methods to discover organizational structure [5]. It means that few researchers believe that network analysis is a comprehensive approach in process mining [73].

Our research methodology is based on the waterfall. The waterfall is one of the software development methodologies. The waterfall methodology breaks down the project activities into linear sequential phases. Each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The phases are literature review, problem definition in detail, method design, method implementation, data simulation, data pre-processing, experiments, evaluation of results, and reporting. If it were necessary, some improvements would be added to the proposed method considering the evaluation results.

The paper is organized as follows. Sections 2 and 3 respectively describe the literature review and the theoretical background. The proposed research methodology and the evaluation strategy are raised in section 4. Additionally, the description of the dataset, the experiments' results, and discussions are presented in section 5. Finally, section 6 outlines some of the conclusions, implications, and future works.

2 Theoretical Backgrounds

Theoretical backgrounds are presented in this section. Electronic social commerce system and network analysis are two main backgrounds of this research.

2.1 Electronic Social Commerce System

An electronic social commerce system has two main goals: 1) Achieving commercial goals, 2) Stakeholder engagement, and interaction. Therefore, ESCS provides a platform for doing business activities along with non-business or social activities. It goes beyond a social networking website by addng a business system and a workflow system. All actors, including business employees, customers, and partners, can interact with ESCS and with each other through the intended interfaces [9], [24]. The difference between the main components of social commerce, electronic commerce and social networks are displayed in figure 1.

The implementation of a social commerce system can be based on two types of architecture. The first type is referred to as off-site or (introductory) referential. It includes a social networking system that adds business features, allows advertising and transactions. Examples include Facebook, LinkedIn, and Pinterest. Most retailers use this model. The second architecture is the traditional e-commerce system that utilizes social networking capabilities. In particular, this system allows business-to-consumer (B2C) businesses to gain a better understanding of the customer and serve them better. It is called on-site or direct sales, and examples include Amazon website, Zazzle, Macys, Fab.com, and Etsy [48]. Except for these two types of architecture, there exist auction/tender directors. They are alternative formats for group purchasing operations, such as Groupon and LivingSocial [18].

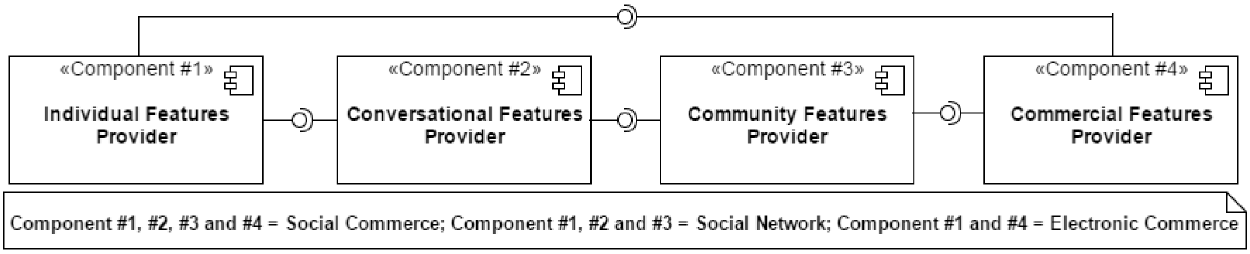

«Component #1»
Individual Features Provider
«Component #2»
Conversational Features Provider
«Component #3»
Community Features Provider
«Component #4»
Commercial Features Provider
Component #1, #2, #3 and #4 = Social Commerce; Component #1, #2 and #3 = Social Network; Component #1 and #4 = Electronic Commerce

Figure 1: Component diagram of s-commerce, e-commerce, and social network systems [24]

2.1.1 Commercial Process

An electronic social commerce system comprises two main components: commercial and social [24]. This paper focuses on the commercial component that supports commercial processes and includes a set of commercial activities. Commercial activities are related to the exchange of finance, goods/services, and information in businesses [48]. Commercial processes are a specific type of business process. The business process is a set of activities that are performed to achieve specific goals. These activities are structured in a relative order. Each activity may have different attributes (for example, people doing the activity). Also, each activity may consist of lower-level sub-activities (a sub-process) or includes decision-making activities and branch nodes [83].

The methods used for business process discovery in the context of information systems are not efficient in the context of social commerce systems. Because they cannot handle or even represent the diversity of working flows. Therefore, we are looking for a new method for discovering process in the context of social commerce systems. Moreover, according to the common components of s-commerce and e-commerce in figure 1, an electronic social commerce system includes all electronic commerce features. Thus, the scope of this research is social commerce, and the focus is just on the commercial process. The s-commerce architecture (off-site or on-site) does not matter.

2.2 Network Analysis

Network analysis involves methods and metrics for extracting knowledge from network structures. Regards to the graph theory, a single-layer social network $SN = (V, E)$ contains a not-null set of vertices (V) (i.e., members or actors) and a set of edges (E) (i.e., homogeneous relations), which indicates an explicit or implicit relationship between two vertices [94]. Different network extraction scenarios lead to the formation of different types of networks [22]. For example, the resources of a process can specify the set of vertices, and the set of edges can define based on the handover of work. In the process mining context, network analysis could be accomplished after extracting the network from the events log.

2.2.1 Multi-Layered Network

There is not always one type of relation (E) among the vertices (V) of a network, or there are not always vertices of the same type in a network. Networks with several relations or vertices are called heterogeneous networks. For example, in a network in which vertices are individuals, there may be three types of relations among vertices: friendship, cooperation, and fellow-citizen. Networks with heterogeneous relations can be modeled in two forms. In the first form, we define a network with weighted edges (the weight of each edge equals the summation of edge types' weight). Obviously, in this situation, the type of relationship cannot be determined using the weight of the related edge. Therefore, the analysis may not be accurate. The second form is to consider each relation type as a separate network. Consequently, the network is represented as a layered network [17], [41]. In this case, each layer might contain separate or shared vertices and edges [17].

2.2.2 Attributed Network

Modeling data based on simple networks or homogenous networks is impossible. Indeed, there is other information in addition to relations among the vertices; for example, in a social network in which vertices are individuals and edges are the friendship among vertices, gender, hometown, and education level can be information that describes each vertex. Thus, when a set of attributes characterises the vertices in the network, we have the attributed network. Attributed networks are widely employed to model real-world systems [43], [49].

2.2.3 Network Communities

Detecting network communities is one of the network analysis challenges. Each community is a subset of the network in which individuals have highly internal relations and fewer connections with individuals outside their community [27]. Actually, in the field of information systems and business process management systems, people inside a community have similar roles and responsibilities and represent organizational units (depending on the scenario of network definition) [95].

Detecting the communities in the field of information systems and business process management systems assists in understanding the organization and improving organizational collaboration. Various traditional clustering methods and algorithms have been proposed by researchers for organizational structure discovery [2], [95]. Only Appice has used community detection methods to discover and analyze organizational units [5]. According to Appice's study, community detection methods provided better results to determine the organizational structure in comparison with the traditional clustering methods.

Clustering algorithms are closely related to community detection algorithms due to their nature and goal of partitioning nodes into groups. Clustering algorithms group the vertices into clusters based on their similarity, while community detection algorithms, which are often modularity-based, analyze the structure of similarity matrix. Modularity [16] is a qualitative function for comparing different divisions of a network that are more cohesive internally than externally. Consequently, community discovery allows us to demonstrate the whole network of an organization at a very comprehensible and dense level, where each community could be a functional group or a real or abstract entity in the system. At this level, community structure contributes significant insights into the network organizational principles [62].

2.2.4 Network Motifs

Many networks contain repetitive patterns that are significantly more frequent than random. Milo et al. (2002) used the term *network motifs* to describe such patterns [39]. Network motifs often have unique functional characteristics and are known as the building blocks of networks and often play essential roles in network performance [40]. Therefore, when a subgraph is repeated with a specific property within the network, it confirms a special behavior or a particular structure in the network. Figure 2 shows the motifs of size three in a simple directed network [54].

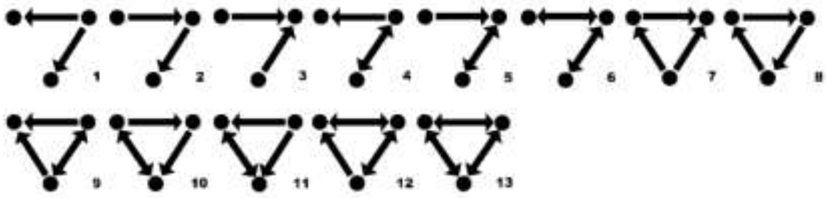


Figure 2: All 13 possible connected directed 3-motifs

Motif discovery algorithms include three main steps: 1) search and determination of the frequency of subgraphs with a given size in the network, 2) identification and categorization of isomorphism subgraphs, and 3) determination of the statistical significance of categories; this is accomplished by comparing the frequency of each class in the desired network with the frequency of that class in random networks. The statistical significance is calculated by the z-score and p-value parameters. The class has a high statistical significance when the z-score is high, and the p-value statistic is low [40]. Hence, motifs are not only repetitive structures in the graph; but also, they have statistical significance concerning the threshold values of z-score and p-value [40], [64]. Also, P-value is employed in other research domains such as image processing [60], [61].

3 Related Works

This section explains three important related concepts to process mining and its' associated studies.

3.1 Business Process Event Logs

From a business process management perspective, event data is related to the execution of one or more business processes and is recorded in a wide variety of data sources such as databases, flat files, message logs, transaction logs, enterprise resource planning (ERP) systems, and document management systems. Each event (*Pid, a, r, t)*, refers to a specific activity (*a*) or a well-defined step in the business process that is executed for a specific item (*Pid*). Each event may be supplemented with additional information, such as the execution resource (*r*), like a person, device, machine or computer server, event timestamp (*t*), event identifier, or other data elements recorded in the events [78]. Each case is an instance of the execution of the business process. A snapshot of an event log based on the process model in figure 3 is shown in table 1. An activity whose execution resource is unspecified is called a task. Each event log contains a set of traces, and each trace contains a sequence of events from the activity. Each trace is associated with a specific case [14].

Activities are not limited to business processes. From the system point of view, web or mobile application systems such as social networks also include a set of activities. Some of the possible activities in social networks include messaging, interlocution, and content sharing [42].

Each resource belongs to at least one role and one group. Role and group determine what each resource should do. Resources that share the same role and group can accomplish the same set of activities. Warehouse keeper and cashier are examples of a role. Transportation and support can be considered as groups. Not all four features (i.e., *Pid, a, r,* and *t*) are needed to discover the process structure.

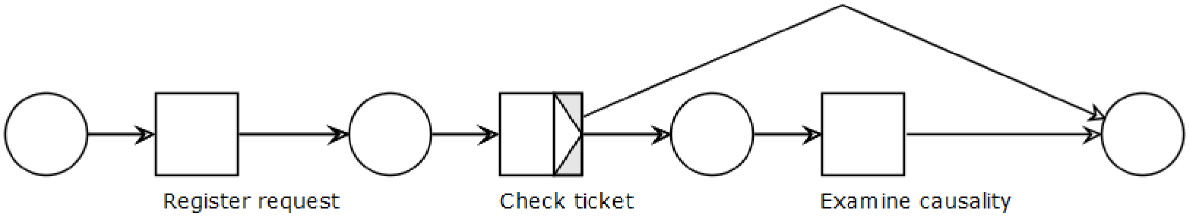


Figure 3: A part of a process based on the Petri net

Table 1: A snapshot of a standard business process event logs base on the process in figure 3

Business Process ID (*Pid*)	Activity Name (*a*)	Timestamp (*t*)	Resource Name (*r*)
1	Register request	2018-05-04: 16:52	Pete
1	Check ticket	2018-05-04: 16:55	Sara
2	Register request	2018-05-04: 16:53	Mike
2	Check ticket	2018-05-04: 16:56	Server 1
2	Examine causality	2018-05-04: 16:58	Ellen
1	Examine causality	2018-05-04: 16:58	Mike
3	Register request	2018-05-04: 17:10	Pete
3	Check ticket	2018-05-04: 17:14	Server 1
…	…	…	…

As shown in Table 1, the event log does not contain the type of workflow in the context of the information systems. For example, it is unclear whether the *Register request* activity makes a flow of information or a financial flow. Whereas, the social commerce system includes different flows.

3.2 Process (Model) Discovery

Process discovery or process model discovery approaches rest on five pillars. In the following, we explain each approach in summary. Some commonly used methods are implemented in the Prom tool (Site 1), the RapidProm plugin (Site 2), or other tools such as Disco (Site 3). Owing to the process discovered is eventually represented and illustrated using the Petri net, process tree, or other methods (Table 2), the notion *Process Model* is used. Thus, in this paper, *Process Model is* a discovered process that can be visualized using one of the visualization methods. In the process model, the order of activities is specified.

Deterministic mining or abstraction methods: Methods developed based on the Alpha algorithm, such as the Beta and Alpha Plus algorithms [90], [91], belong to this category. Aalst introduced the Alpha algorithm [85]. Based on the timestamp of tasks in the event log, this algorithm defines a set of dependency relationships, including causal, parallel, and selection relationships, and maps each relation to a Petri net model. Contrary to their claims regarding the determination of parallel and choice relationships, the implementation of these methods in the Prom tool and the RapidProm plugin lacks the determination of such relations. The most significant problem with such algorithms is to disregard the noise, and the advantage is that they can detect a workflow model for each process and display it as a soundness work flow-net.

Heuristic mining or initiative-based methods: These methods use dependency relationships similar to the deterministic approaches, but they consider the dependencies along with their frequencies. This group of algorithms is based on the fact that with an increase in the frequency of dependency, the probability of the randomness of that relationship decreases and vice versa. The heuristic algorithms consists of three main steps: 1) creating a dependency/frequency table from the event log; 2) creating a dependency/frequency graph from the dependency/frequency table based on a set of heuristic rules; and 3) creating a Petri net using the information in the dependency/frequency graph and its table [68], [89]. The main advantage of these methods is their ability to deal with noise and incomplete information.

Inductive methods or divide and conquer methods: These algorithms operate by the divide and conquer mechanism and include two main steps. In the first step, a stochastic activity (SA) graph is created for process instances. The SA graph is a directed graph, including direct dependencies between activities. In the next step, the resulting SA graph is converted into a workflow model [36]. The mergeSeq, splitSeq, and splitPar algorithms are in this category [34], [35]. Also, the algorithm presented by Schimm [70] is in this group. The important feature of these algorithms is their ability to detect duplicate tasks.

Evolutionary or search-based approaches: These methods are based on the genetic algorithm and consist of three main steps. First, a random initial population of the process model is formed. Then, the fitness index is calculated for each of the process models that form the partition of the primary population. The fitness index determines the degree to which the model justifies the observed behavior in the event log. In the next step, the initial population develops through the cross over and mutation to create the next generation. It thus is evolved from generation to generation, and this evolution continues until a fit model, with a high fitness index, is found. Despite the ability of evolutionary approaches to detect most structures, they impose high computational complexity [1], [19].

Abstraction hierarchy or clustering-based approaches: In these algorithms, for each process instance, a process model is created; these models are clustered at the next step using k-gram, Bag of Activities, or other traditional clustering algorithms. Finally, the primary process model is obtained from the composition of the clusters [12]. In some methods, each process instance needs to be mapped to a vector [72] or a string [11]. Some methods of this category are based on fuzzy logic [28], [32], [86].

One of the significant issues in process model discovery is spaghetti models. The models generated for the processes with a lot of different cases and high diversity of behavior tend to be very confusing and difficult to understand. These complex and large models are usually named spaghetti models [82].

Recently, the spaghetti model issues have been solved by traditional clustering methods such as k-means [4], [7], [87] and hierarchical clustering algorithms [38]. Clustering methods are employed to determine the scope of a process in a way that the discovered model does not contain unnecessary traces and cases [38]. Ariouat et al. also proposed a clustering algorithm (called AXOR) that calculates the similarity based on the distance [6]. However, it is difficult to understand the model and determine the scope of processes if the event log contains several processes.

A summary of the process model discovery approaches compared to the proposed approach is presented in Table 2. It is noteworthy that resistance to disturbance, handling incomplete event logs, and the generation of spaghetti models in the context of information systems have been investigated. Given the differences in social commerce systems, previous approaches in the context of commercial systems may not be as efficient as information systems. Also, since previous methods are specific to information systems and do not consider the diversity of work flow types, they cannot be applied to commercial systems. The last row in Table 2 shows the features of the proposed approach. The present study employs the type of work flow to explore the process in the context of social commerce systems. The proposed approach models the process discovered as an attributed network. This network is convertible to a Petri net.

3.3 Network Analysis Applications in Process Mining

To the best of our knowledge, network analysis (NA) methods have not been applied to process model discovery in process mining. However, it has been used in organizational mining. The purpose of using NA methods in organizational mining is finding relationships among entities (e.g., process resources) and discovering possible bottlenecks to improve performance of the process [95], identifying influential resources in the process [37], [71], [77], [84] and discovering organizational structure [5].

Table 2: Comparison of business process discovery approaches

Features / Approach	**Noise resistant**	**Encounter to incomplete event log**	**Low frequency events/traces detection**	**Generation of simple model (non- spaghetti)**	**Specified Sub-process limit**	**Context of application**	**Identifying complex structures**	**encounter to different working flows**	**Determining decision points (split-join)**	**Determining the type of decision**	**Input**	**Output**
Deterministic mining	×	×	×	×	Slightly	Information systems	×	×	✓	×	Event log (Pid, a, t)	Petri net
Heuristic mining	✓	✓	×	Slightly			✓	×	✓	×		Petri net, Heuristic net, Causal net
Inductive mining	✓	✓	×				✓	×	✓	×		Process tree, Petri net
Evolutionary mining	✓	✓	×		×		×	×	×	×		Petri net, Process tree, Causal net
Clustering-based	✓	✓	×		✓		Slightly	×	✓	×		Petri net, Heuristic net
Proposed approach	✓	✓	✓	✓	✓	Commercial systems	✓	✓	✓	✓	Event log (Pid, a, t) and working flow type	Petri net, Attributed multi-layered network

The NA metrics for analyzing simple networks in organizational mining research are divided into two categories [2], [15], [73]: 1) micro-level metrics: metrics that examine only one specific vertex; such as degree centrality (in-degree and out-degree), betweenness centrality, closeness centrality (in-closeness and out-closeness), eigenvector centrality, and clustering coefficient. 2) macro-level metrics: metrics that analyze the entire network; for example, density, clustering coefficient, and centrality. Each metric has a different interpretation depending on the context of the system and its applications. In order to analyze the extracted network from the event log, research tools such as ProM, NetMiner, and USINET have been employed.

We apply network analysis methods to overcome the problem of spaghetti models and discovering sequencing, selection, and parallelism structures. Also, we cover two common noise and loop structures problems. We also define and use the flow type as a new field in the event log. The primary outcome of our proposed method is an attributed network [43] that we transform it manually into a Petri net, which is the secondary outcome of the method.

4 Problem Definition

One Off-Site social commerce with the B2C business model is desired. The business items are physical goods, and the revenue model is sales. This business is an example of the thousands of similar businesses that are now active all over the world. The process of selling/buying in this business involves various activities. Each activity is executable by a role and a group. Each role and group contain at least one resource (machine or human). Activities can be executed in a predefined structure. However, at runtime, they do not necessarily follow the default structure. For example, deliberate or unintentional sabotage can occur in the system. Some resources might not obey the rules in executing activities and make some exceptions. Execution of each activity makes a type of work flow in the system. These events are recorded in the system and can be retrieved. We call this information a commercial event log. Therefore, we first define the commercial event log.

Definition 1 (Commercial event log): Based on the basic definition of event logs [80] and its extension in this research, $EventLog = (C, A, R, T, L)$, is an event log for a structured process in the context of social commerce systems. The process might contain a loop structure. C represents a set of case identifiers, and A is a set of unique tasks. R is a set of resources (including human and inhuman types). T represents a set of timestamps and L is a set of content labels or flow types. Moreover, L is added to the event logs in this research and was not considered as a feature in previous works. Each resource performs one to n tasks based on its predefined role. Each task has one type of flow. The label L denotes the type of the flow from a_i, done by resource r_i, to the next task a_j, done by resource r_j (i.e., the type of work interaction between the resources r_j and r_i). L includes $\{i, g, f\}$, in which i is the information flow tag, that is, the information exchanged between the resource r_i and the source r_j. Value g is the label of the goods flow, showing that a physical good is transferred between the resources r_j and r_k. Likewise, f is the label of financial flow; it means that some money has been exchanged between the resources r_j and r_k. The process on which $EventLog$ is based has a negligible conversion rate. $EventLog$ has a significant degree of noise due to the lack of completion of the process [55], [57]. In fact, not all traces are completely performed, as if the end of the traces has been deleted.

Different roles have no common tasks and resources. Meanwhile, owing to there is no limitation for the number of human resources, it is variable in various executions of the business process. Therefore, we have:

- There is a resource r_j in R which is an inhuman resource.
- The number of customers is higher than the total number of human resources.
- Also, to simplify the problem, the resources cannot have multiple roles.

We provide the following formal definition for the main problem of the research:

Problem 1: Given the commercial $EventLog$ in the context of social commerce systems, our research problem is finding a function like f which discovers the commercial process that has been based on those events and stored in the event log. The discovered process is an attributed multi-layered network. It can be converted to a Petri network [80], [83].

$$f{:}\,(EventLog) \rightarrow Attributed\ Multi-layered\ Net\ or\ PetriNet \tag{1}$$

According to the differences between social commerce systems and information systems, also the challenges of process discovery, the main research questions are:

- Is it possible to find the commercial process using network analysis metrics and methods (especially motif discovery and analysis)?
- Does the proposed method have better performance in comparison with the previous methods from the business process management viewpoint?

5 Research Methodology

The research methodology is shown in figure 4. The methodology of the research Waterfall. Literature review and problem definition have been presented previously. The most critical step in the methodology is the design of the proposed framework. The proposed framework leads to a new process discovery method. Thus, it will be explained in the following. Data simulations, experiments, and evaluation are also presented in section 6.

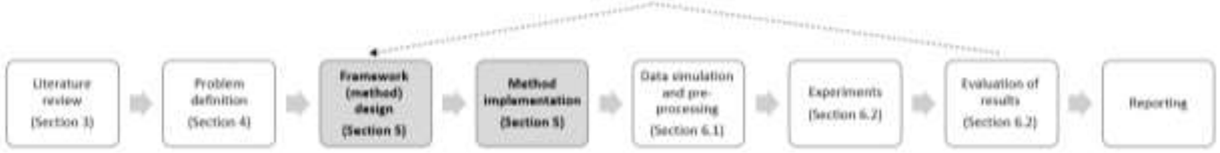

Literature review (Section 3)
Problem definition (Section 4)
Framework (method) design (Section 5)
Method implementation (Section 5)
Data simulation and pre-processing (Section 6.1)
Experiments (Section 6.2)
Evaluation of results (Section 6.2)
Reporting

Figure 4: The proposed research methodology

5.1 Proposed framework

The proposed framework for discovering the commercial process is presented in figure 5. It consists of two main steps: the preprocessing and discovery of the process. Both stages are based on concepts and methods of network analysis. Each section is explained below.

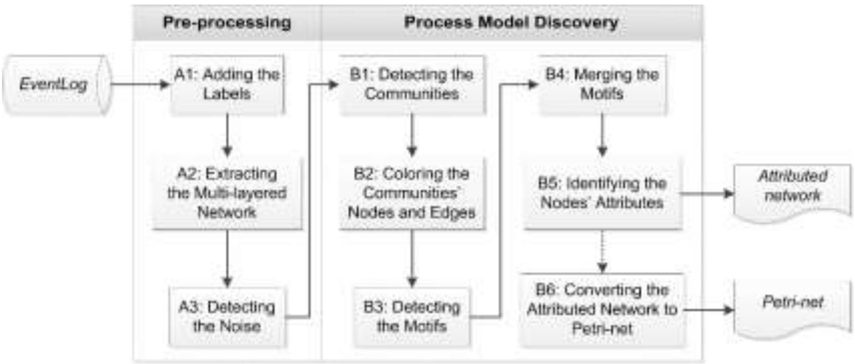

Pre-processing
Process Model Discovery
EventLog
A1: Adding the Labels
A2: Extracting the Multi-layered Network
A3: Detecting the Noise
B1: Detecting the Communities
B2: Coloring the Communities' Nodes and Edges
B3: Detecting the Motifs
B4: Merging the Motifs
B5: Identifying the Nodes' Attributes
B6: Converting the Attributed Network to Petri-net
Attributed network
Petri-net

Figure 5: The proposed framework of process discovery

5.1.1 Pre-Processing Phase

This step consists of three main activities: adding the required labels to the event log (A1), multi-layer network extraction (A2), and initial noise identification (A3). Based on definition 1, each record in $\boldsymbol{EventLog}$ contains five attributes under table 3. Due to the variety of Start and End activities for different items, the two Start and End labels for each CaseID are added before the start of the first activity and after the last activity is completed (Table 4). Then a network structure based on Table 4 is made.

Table 3: A sample of event log ($EventLog$)

CaseID	Flow Type	Resource	Task Name	Time
CASE_1	I	c-346	open DG'home page	t1
CASE_1	I	dgs01	show DG's home page	t2
CASE_1	I	c-346	set search filters	t3
CASE_1	I	dgs01	show result	t4

Table 4: Example of $EventLog$ after step A1

CaseID	Flow Type	Resource	Task Name	Time
CASE_1			Start	t0
CASE_1	I	c-346	open DG'home page	t1
CASE_1	I	dgs01	show DG's home page	t2
CASE_1	I	c-346	set search filters	t3
CASE_1	I	dgs01	show result	t4
CASE_1			End	t5

Definition 2 (Multi-layered activity network): According to the definition of multi-layered networks [17], [41], $ActN_L = (V_{ActN_L}, E_{ActN_L}, L)$ is defined as a multi-layered, static, directed and weighted activity network. $L = \{i, g, f\}$ is the set of flow types in $EventLog$. In other words, it specifies the type of flow transferred from one activity/resource to another. If the value of L is not specified, it is considered as the whole set (i.e., $\{i, g, f\}$). V_{ActN_L} is a set of activities (or tasks) in one of L's layers, and we have $V_{ActN_L} \subseteq A$. $E_{ActN_L} \subseteq V_{ActN_L} \times V_{ActN_L}$ is also a set of directed and weighted relations in one of L's layers. For example, there is an edge from the vertex a_i to the vertex a_j on the network $ActN_i$, if there is at least one activity of vertex a_i before the activity a_j. Therefore, a_i transfers certain information to a_j. The weight of each edge is equal to the absolute frequency of the sequence of two activities a_i and a_j in $EventLog$ based on the type of flow. Consequently, for each sequence of CaseID, there is a corresponding subgraph in network $ActN_L$ (see table 5).

Table 5: An example of $EventLog$ after step A2

CaseID	Flow Type	Resource	Task Name Source	Task Name Target	Time	Weight
CASE_1	I	-	Start	open DG'home page	t0	1
CASE_1	I	c-346	open DG'home page	show DG's home page	t1	1
CASE_1	I	dgs01	show DG's home page	set search filters	t2	1
CASE_1	I	c-346	set search filters	show result	t3	1
CASE_1	I	dgs01	show result	End	t4	1

After extracting the structure of the activity network from the event logs, the initial noisy edges are identified and removed. The identification of noisy edge is based on the knowledge of the research field and the absolute frequency of cases related to that edge. The edges with a small number of case dependencies might be noise. In the context of commercial processes, about 5% of cases are fully accomplished [55], [57]. Therefore, in this study, if the edge is associated with less than ε percentage of cases, it is detected as noise. Increasing this threshold will result in the removal of more edges. If the absolute frequency of edge-related cases follows a normal distribution, we can apply a normal distribution [74] to noise removal.

On the other hand, according to the previous two steps, the edges that connect a_i to End are eliminated, that in the event logs, there is an activity such as a_j after a_i. In fact, the existence of an edge between a_i and End signifies an incomplete end of the process.

5.1.2 Process Model Discovery Phase

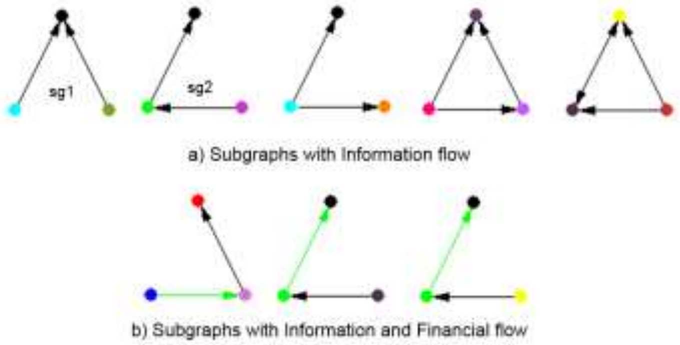
The process model discovery in figure 5 consists of 6 main activities: detection of communities in the integrated multi-layered activity network (B1), coloring vertices and edges in each community of the activity network (B2), identification of the motifs in each community of the colored activity network (B3), merging the motifs to identify the main structure of the process model (B4), determining the features of each vertex (activity), or converting the multi-layered activity network into a multi-layered attributed network (B5) and converting attributed network to Petri net model (B6). The last activity is optional.

In our research, the purpose of community detection (B1) is to solve the problem of spaghetti models. Until now, traditional clustering methods have been used to solve this problem. The difference between traditional clustering techniques and community detection methods, which are based on network structure, is explained in Section 2.2. Different methods and algorithms for detecting communities are suggested by the researchers [33], [56], [76]. Given the fact that at this stage, the goal is to determine the scope of dense subgraphs of the network, we can map $ActN_L$ network layers into a simple network. As a result, community detection methods in simple networks (e.g., [65], [75]) can be applied to our case. In this research, the community detection method introduced by Girvan and Newman [30], [69] is used. An implementation of this method is available in Gephi (Site 4) as a plugin (Girvan-Newman Clustering). Communities identified by this method do not overlap. This means that each vertex will belong only to one community. It identifies communities by removing non-significant edges (removal of the edges having low betweenness centrality).

In the next step, after identifying the communities in the activity network $ActN_L$, the underlying structures should be determined. It is necessary to distinguish between types of relationships, as well as (i.e., tasks) to identify the fundamental structures (i.e., motifs). To do this, we color the vertices and edges (B2). Each vertex (i.e., task) in each community of the network is characterized by a unique color. Moreover, any edges between two activities are characterized by a unique color. Given that the network $ActN_L$ has three layers corresponding to three types of flows, three colors are chosen. At this stage, the edges connecting two communities and the vertices of those edges are considered within the sub-region while identifying the motifs. Therefore, we can consider interconnectivity between the structures.

After coloring the vertices and edges in the activity network, it is time to extract the base subgraphs or network motifs (B3). At this stage, the subgraphs with three vertices are identified. Subgraphs with larger sizes can be formed based on subgraphs with size three [54], [64]. Various methods have been proposed for discovering motifs. In this research, the ESU algorithm is used [40]. The reasons for applying this algorithm are: 1) It is one of the probabilistic algorithms and has low time complexity. 2) This algorithm is a network-centric algorithm and searches for all subgraphs of a given size [93]. ESU algorithm is implemented in FANMOD (Site 5). An advantage of this tool is the ability to distinguish between the edges and vertices of the network based on color. Since all three-sized subgraphs are not significant and meaningful, finally the subgraphs must be filtered based on the Z-score and/or P-value. In this research, motifs are filtered and selected based on P-value $(|p - value| \leq 0.05)$ and/or Z-score $(|z - score| \geq 1.65)$. The reason for choosing these two parameters is that motifs with a P-value of less than 0.05 and a Z-score of higher than 1.65 are more likely to be significant motifs. The probability of being formed randomly is less than 5% for them.

Figure 6 shows the number of 3-vertices subgraphs that can appear in different layers (i.e., *i*, *p*, and *f*). The color of each vertex is specified by B2 based on the task name or task ID. In addition to the importance of p-value and z-score in selecting subgraphs, the number of colors of each subgraph and the structure of the subgraphs are also substantial. For example, subgraphs like those in column A of table 6 are not crucial in discovering the process structure, because these subgraphs are equivalent to the subgraphs in column B of table 6, and they have no further information on the process structure. Therefore, selecting subgraphs of column B is preferable to subgraphs of column A in table 6. Table 6 lists only some trivial subgraphs for one type of flow.



a) Subgraphs with Information flow

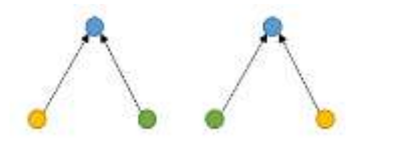
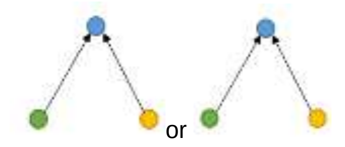
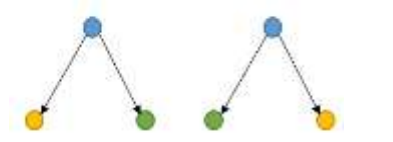
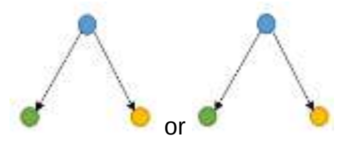
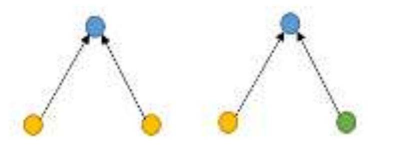
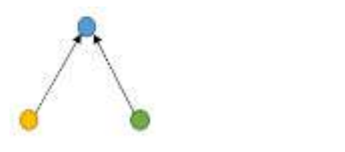
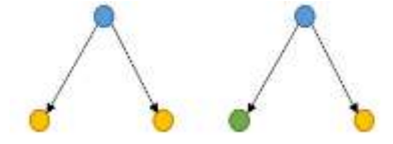
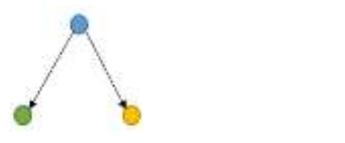
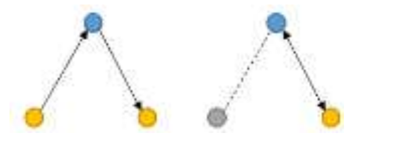
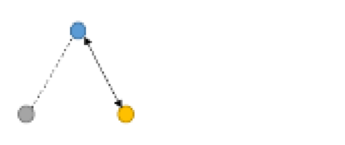
b) Subgraphs with Information and Financial flow

Figure 6: Some of the discovered subgraphes with size three

After the identification of meaningful motifs, we must identify the main structure of the process model (B4). The structure of the process model is constructed by merging essential motifs based on the color of vertices. Since motif detection methods do not consider loops, if the beginning and the end of an edge is a single vertex (i.e., self-loop), that edge is not displayed in the process model structure and thus removed. For example, the subgraph sg1 and sg2 in figure 6 are merged from the black vertex. Then the graph in figure 7 is constructed.

So far, the main structure of the process has been identified. After that, we seek to determine the attributes for the vertices of the network $ActN_L$ (B5) to discover the logical structures (sequence, parallelism, and selection). Two attributes are defined. Possible values for the two attributes are given in table 7. The first attribute accepts three values, including AND, XOR, and Simple. The second attribute is called LevelTwo, whose value can be Split or Join. Therefore, the network $ActN_L$ is converted to an attributed activity network $AttActN_L$ by specifying attributes for the vertices of the activity network.

Table 6: Some of the discovered and preferred subgraphs for discovering process structure

ID	(A) Discovered subgraphs	(B) Preferred subgraph
1		 or
2		 or
3		
4		
5		

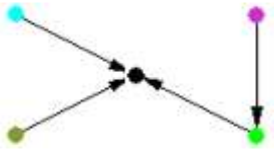


Figure 7: The constructed graph after merging two selected motifs from figure 6

Table 7: Logical structures and equivalent attribute values

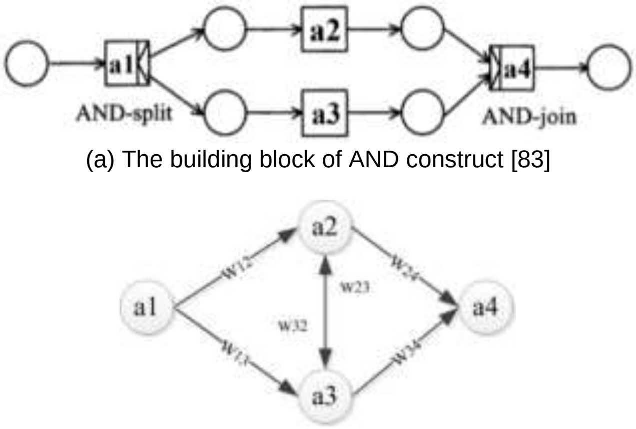
Structure name	Value of attribute 1 - LevelOne	Value of attribute 2 – LevelTwo
Parallelism	AND	Split – Join
Selection	XOR	Split – Join
Sequence	Simple	Null

Definition 3 (Attributed multi-layered network): According to the attributed network definition [49], $AttActN_L = (V_{AttActN_L}, E_{AttActN_L}, L, A)$ is defined as a multi-layered activity network with attributes. In this definition, L, $V_{AttActN_L}$ and $E_{AttActN_L}$ are equivalent to L, V_{ActN_L} and E_{ActN_L} in definition 2 respectively. The attributes associated with each vertex are defined as vector A_i. Also, i is the index of vertices in the set $V_{AttActN_L}$. Each vertex in the network $AttActN_L$ can have more than one pair of attributes (i.e., LevelOne and LevelTwo).

Setting up attributes is determined based on network structure $ActN_L$ and the weight of the relations in the network $ActN_L$.

Rule 1: The Petri network-based process model can be constructed based on building blocks [83]. The parallelism building block is shown in figure 8 (a). In the network $ActN_L$, the equivalent structure of parallelism building block will appear differently (see figure 8 (b)). The structural difference in the network $ActN_L$ is due to the order in which the activities are performed. For example, the order of execution of tasks can differ based on the start time of the task by resources. Figure 8 (b) is derived from merging two motifs 9 and 11 in figure 2. Hence, in figure 8 (b), the edges a_2a_3 and a_3a_2 should be deleted. By removing these two edges, the network structure $ActN_L$can be mapped to the AND structure in the Petri network. By removing the intermediary edges, the weights of the related edges should be updated.

For example, by removing the edge a_2a_3, its weight should be added to edges a_1a_3 and a_2a_4. In this example, the value of the attribute LevelOne and LevelTwo for the task a_1 would be AND and Split, respectively AND and Join are determined for task a_2 the same way (see figure 9). The weight of the output edges of task a_1 and the weight of the input edges to the task a_4 may differ by size ε.



(a) The building block of AND construct [83]

(b) Structure of AND at $ActN_L$

Figure 8: (a) Building block of AND in the Petri net, and (b) The structure of AND in the network $ActN_L$ before removing the intermediary edges

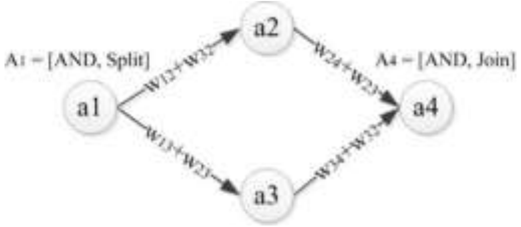


Figure 9: The structure of AND in the network $AttActN_L$

Rule 2: Sequence structures are identified based on the input and output degrees of each vertex in the network $ActN_L$. Accordingly, if we have $InDegree(a_i) = OutDegree(a_i) = 1$, then the LevelOne property value for the vertex a_i is equal to Simple. The LevelTwo feature is not initialized. Figure 10 indicates the task of a_2 with the attribute LevelOne = Simple**.**

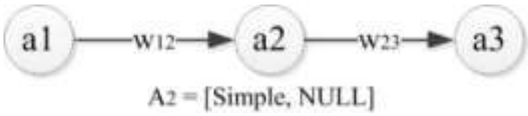


Figure 10: An example of the sequence structure in the network $AttActN_L$

Rule 3: If the in-degree of a task such as a_i in the network $ActN_L$ is greater than 1, the value of LevelTwo property for that task is Join. Similarly, if the out-degree of a task, such as a_j, in network $ActN_L$ is more than 1, the value of LevelTwo property for that task is Split. An example of these two values is given in figure 10 for a_2 and a_4 tasks**.**

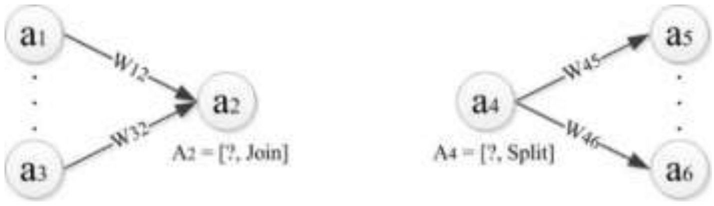
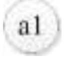
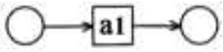
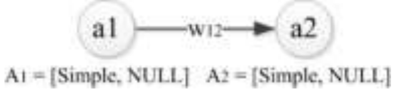
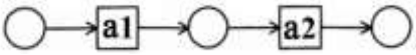
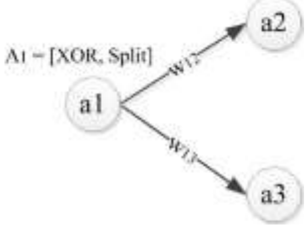
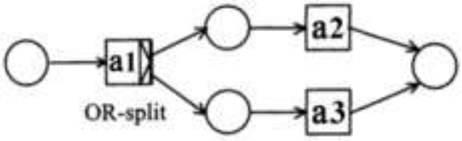
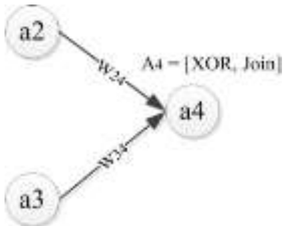
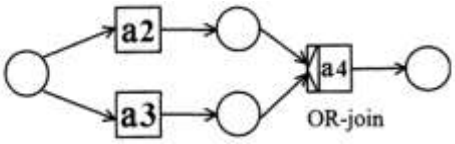
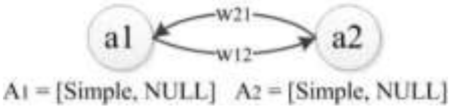
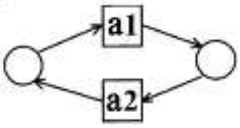
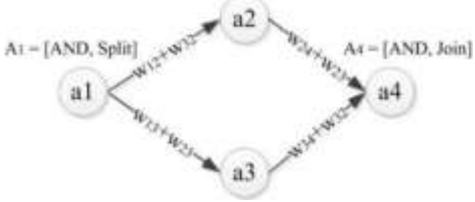
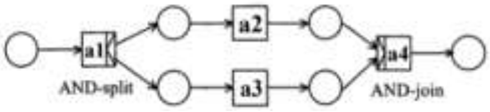


Figure 11: An example of an attribute with the value Join and Split

Table 8: Mapping the network $AttActN_L$ structures to building blocks in Petri net

Structure in $AttActN_L$	**Building block in Petri Net**
	 Basic building block
 a1 W12 a2 A1 = [Simple, NULL] A2 = [Simple, NULL]	 Sequence construct
	 Explicit OR Split construct
	 Explicit OR Join construct
 a1 W21 W12 a2 A1 = [Simple, NULL] A2 = [Simple, NULL]	 Interaction construct
 A1 = [AND, Split] a2 A4 = [AND, Join] a1 W12+W32 W24+W23 a4 W13+W23 W34+W32 a3	 AND construct

Rule 4: Based on figure 11, if the weights of the input edges to a task such as a_i are not equal, the LevelOne property for that task is XOR. If the difference between the weights of the edges is ε, then the LevelOne property for that task is AND. XOR or AND is determined similarly based on the weight of the output edges of task a_j.

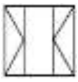
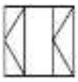
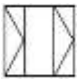
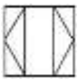
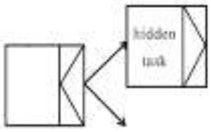
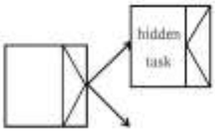
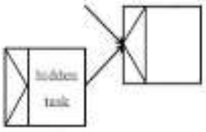
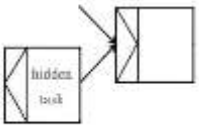
Exception 1: In rule 4, if the difference between weights of the edges is ε and the edges are from two different layers $L = \{g, f\}$, then the LevelOne property for that activity is XOR. In business processes, there can not be both financial and goods flows simultaneously [29], [53]. Therefore, the logical structure is not parallelism**.**

Exception 2: If $OutDegree(a_i) = 0$, then a_i is the final task in the process. Similarly, if $InDegree(a_i) = 0$, a_i is the initial task in the process**.**

Most of the process discovery methods provide the final model based on Petri net. Therefore, the discovered model, which is represented as an attributed network, should be converted to a Petri net based on the basic concepts of Petri networks (B6). This conversion is necessary for evaluation. More information on Petri networks is presented by van der Aalst [83]. Any Petri net-based process model can be constructed based on the building blocks. A model based on building blocks is a safe and soundness model [83]. Table 8 lists the base structures in the network $AttActN_L$ and the equivalent building blocks on the Petri nets**.**

More than a pair of attributes may be assigned to a task in the network $AttActN_L$. According to table 9, for showing it on a Petri net, it is either replaced by one transition or mapped to two transitions. In another case, the additional transition is a hidden task or an invisible task. The hidden tasks are related to the silent steps that are used only for the purpose of routing (available in the process model) and are not present in the event logs [3], [92].

Table 9: Relationship between vertices with several attributes in the network $AttActN_L$ with equivalent transitions in Petri net

ID	LevelOne-LevelTow (1)	LevelOne-LevelTow (2)	The equivalent transition in Petri net
1	AND/XOR-Join	AND-Split	 and-join-and-split  xor-join-and-split
2	XOR/AND -Join	XOR-Split	 and-join-xor-split  xor-join-xor-split
3	XOR-Split	AND-Split	 and-split-xor-split  xor-split-and-split
4	XOR-Join	AND-Join	 and-join-xor-join  xor-join-and-join

5.2 Evaluation Metrics and Strategy

In order to evaluate the proposed method, the performance of this method is compared with some conventional methods that were widely used in previous process discovery researches and are available within the tools. These methods are presented in table 10. To compare the extracted models, those models that are not based on the Petri net are manually converted to Petri net. Besides, to make the proposed method more reliable, activity network construction is also done, regardless of the type of flows. The results which are based on simple network modeling are compared with the results of multi-layered network modeling in which there is a distinction among different flows.

Table 10: Specification of process discovery methods at the evaluation stage

ID	Method name	Tool	Output type	Reference
PMD01	Alpha Miner Classic	RapidProm	Petri net	[85]
PMD02	Alpha Miner +	RapidProm	Petri net	[91]
PMD03	Alpha Miner #	RapidProm	Petri net	[92]
PMD04	Fuzzy Miner	RapidProm	Fuzzy model	[32]
PMD05	Heuristics Miner	RapidProm	Heuristic net	[88], [89]
PMD06	Inductive Miner	RapidProm	Petri net	[44]
PMD07	Inductive Miner - Incompleteness	RapidProm	Petri net	[46]
PMD08	Inductive Miner - Infrequent	RapidProm	Petri net	[45]
PMD09	Disco Miner	Disco	Fuzzy model	[28]
PMD10-ML	Multilayered Attributed Network PMD	No stand-alone tool is available.	Attributed net/ Petri net	Proposed method
PMD10-S	Simple Attributed Network PMD		Attributed net/ Petri net	Proposed method

5.2.1 Adding Noise to Dataset

The performance of methods in facing noise is also examined. There are five types of noise in the event logs [89]: missing head, missing tail, missing body, missing event, and exchanging event. In the context of commercial systems, missing tail happens most frequently (see Section 6.1). Therefore, in this research, the other four types of noise generated, and the rate is changed from 0 to 30%. Thus, the event log always contains noise type 2 at 95%. This means 5% of the users in the social commerce context complete the purchasing process, and 95% of them do not complete the process [53].

5.2.2 Evaluation Metrics

In this research, the performance evaluation of the proposed method is examined from two viewpoints: simplicity of the model, and model accuracy [13]. The first one measures the complexity of the process. Process discovery methods usually lead to spaghetti process models. Studying the spaghetti models is very hard. So simplicity is the model's ability to identify the subprocesses/processes formed in the model. The accuracy of the model is examined at two levels: 1) to what extent the overall structure of the model (regardless of the attributes of the vertices) is determined accurate and in accordance with the initial process model? 2) How exactly are the attributes of the vertices identified?

The False Discovery Rate (FDR), and precision or Positive Predictive Value (PPV) are used to evaluate the performance of the proposed method. These measures are calculated based on the confusion matrix (see equations 1 and 2). The best method has the highest precision and lowest FDR. Jaccard index is applied for similarity computing between two models [26], [59], [67]. It equals to the ratio of intersection of edges to the union of edges.

$$PPV\ (precision) = \frac{tp}{tp + fp} \tag{2}$$

$$FDR = \frac{fp}{fp + tp} \tag{3}$$

$$Jaccard = \frac{tp}{tp + fp + tn} \tag{4}$$

We define the Customized F-Measure (CFM) based on the previous studies [21], [58] to compare discovered models with the primary process model. CFM is employed for the significance test. This measure is derived from the confusion matrix and the number of edges among tasks in the primary process model (see equation 4). CFM is the harmonic mean of m1 and m2. CFM is high for a model that discovers a high number of significant edges and less number of wrong edges.

$$CFM = 2\ \times \frac{m1\ \times\ m2}{m1 + m2} \tag{5}$$

$$m1 = \frac{tp}{tp + tn} \tag{6}$$

$$m2 = \frac{tp}{\#edges\ in\ primary\ process\ model} \tag{7}$$

We employ the percentage error to calculate the error of identified attributes. Percentage error is an expression of the difference between a measured value and the accepted value. The formula for calculating percentage error is in equation 7.

$$percentage\ error = \frac{|experimental\ value -\ accepted\ value|}{accepted\ value} \times 100 \tag{8}$$

6 Results and Discussion

This section includes two subjects, dataset simulation in details and results of experiments.

6.1 Dataset Simulation based on the User Behavior Modelng

The appropriate dataset is not available for social commerce systems. Our research dataset is simulated based on user behavior modeling of a famous Iranian electronic commerce system (i.e., DigiKala) within the period of October 1, 2016, to November 20, 2016. The business process and behavior of users were modeled using WoPeD (Site 6) and simulated based on real data from Digikala's users. Generating a dataset based on behaviors and real patterns has two main advantages in comparison with the real dataset: 1) The real data includes a set of specific behaviors in the data collection period. Therefore, all possible behaviors are not included, or some behaviors are repeated very rarely. However, by changing the simulation settings, we can generate various behaviors and investigate them. 2) If the dataset is limited to reality, the results are limited to the specific case study. Even the size of the dataset might be small. Hence the results cannot be reliable, and the solutions will not be decisive. Nevertheless, if the dataset is simulated based on user behavior modeling, different possible modes might be included and produced sufficiently. Other studies [10], [23], [51], [52], [53], [66] have been used user behavior modeling and simulation techniques to prepare the dataset. Therefore, due to the reasons mentioned above and the lack of flow type in available datasets, we used simulation to build an appropriate dataset.

The $EventLog$ (see definition 1) dataset, includes the implementation of 1000 different instances (or cases) of an electronic purchasing process. The process must be executed multiple times so that all tasks and procedures have been fired at least once. Only 5% of the cases completed the process in the simulation. That is the conversion rate in real e-commerce processes [55], [57]. The event log contains significant amounts of noise (missing tail). There was not this amount of noise in the event log of other research fields. The process involves loop structure and various sub-processes. The number of different major procedures of the process is over 16. The count of $EventLog$ records is 6216. There are a total number of 46 different main tasks in the process. Also, on average, 6.16 activities were performed per case.

6.2 Results of Experiments

The evaluation of model simplicity and model accuracy are presented in this sub-section. Moreover, the discussion of the results is at the end.

6.2.1 Model Simplicity

First, the simplicity of the discovered model is examined. Table 11 shows the simplicity of the model based on the following two questions with different noise levels in the $EventLog$. It should be reminded that all process models were converted to Petri network and then compared.

- S1- How simple and understandable is the model?
- S2- How specific is the scope of sub-processes/processes?

The results of the evaluation of the model's simplicity are presented based on the average responses of 8 experts for different noise levels presented in table 11. The knowledge domains of experts are business process management and business process re-engineering.

The experts are analysts and designers of e-commerce systems who have mastered the concepts and methods of managing business processes and reengineering business processes. We identified 11 people by snowball method and sent models to them, but received only 8 responses. Experts rated the two phrases for each model: S1- the discovered model is understandable and straightforward, and S2- the range of sub-processes/processes in the model are clear. Responses were denoted by ranks in the range [1, 5] (i.e., 5-point Likert scale [47]). A score of 1 is defined as Strongly disagree, and a score of 5 is defined as Strongly agree. Table 11 lists the average scores. In table 11, S1 and S2 are respectively these two sentences: *Model is not complicated* and *the scope of sub-processes is specified.*

Table 11: The evaluation results of the simplicity of the discovered models

Process Model Discovery (PMD) Method	**Noise-0%**		**Noise-10%**		**Noise-20%**		**Noise-30%**	
	S1	**S2**	**S1**	**S2**	**S1**	**S2**	**S1**	**S2**
PMD01	1.25	1.25	1.25	1.25	1	1	1	1
PMD02	1.5	1.25	1.5	1.25	1.25	1	1	1
PMD03	1.5	1.25	1.5	1.25	1.25	1	1	1
PMD04	**5**	3.5	**5**	3.5	**4.75**	3.5	**4.75**	3.25
PMD05	**5**	3.75	**5**	3.75	**4.75**	**3.75**	**4.75**	**3.5**
PMD06	3.25	3.5	3.25	3.5	3	3.25	2.75	2.75
PMD07	3	**4.25**	3	**4.25**	3	**4**	2.75	**3.5**
PMD08	3.25	**4**	3.25	**4**	3.25	**3.75**	2.75	3.25
PMD09	1	1	1	1	1	1	1	1
PMD10-ML (based on Multi-layered network)	4.75	5	4.75	5	4.5	5	4.5	4.75
PMD10-S (based on Simple network)	4.5	5	4.5	5	4.25	5	4.25	4.75

The proposed method and other methods were tested on the process models with 20% noise represented in figure 12 and Appendix A, respectively. Figure 12 shows the model discovered by the proposed method based on the modeling of attributed networks. The vertices with the same color represent a community of tasks (activities) that have strong relationships with each other. In other words, each community shows the scope of tasks associated with a sub-process. In the proposed method, the scope of each sub-process is identified clearly and can be represented at a level of abstraction. An example is shown in figure 13, which provides an overall overview of the discovered process model. We remove vertices' attributes for model simplicity at this view. In the proposed method, in addition to the model's comprehensiveness, the number of processes/sub-processes is known. If the method were based on simple network modeling (PMD10-S), the simplicity and comprehensiveness of the model would be less than that of multi-layered network modeling.

The model discovered by other methods (except for PMD04 and PMD05) is not simple. Consequently, understanding the resulting models from those methods is difficult. Actually, the models are spaghetti ones, and they have complicated structures. The scope of sub-processes and even the number of sub-processes in the model are not obvious. Of course, the sub-processes scope in the PMD07 and PMD08 method is almost clear, but not as clear as the proposed method. In the proposed method, the scope of sub-processes/processes is well known due to the use of community detection method that is based on the network structure.

Deterministic mining and inductive mining methods (i.e., PMD01, PMD02, PMD03, PMD06, PMD07, PMD08, and PMD09) do not have acceptable performance (in terms of simplicity) while facing noise. The PMD09 method is implemented in Disco. By manually filtering some links, a simpler model can be obtained within this tool. In this case, S1 might increase, which is similar to PMD04 based on the fuzzy theory.

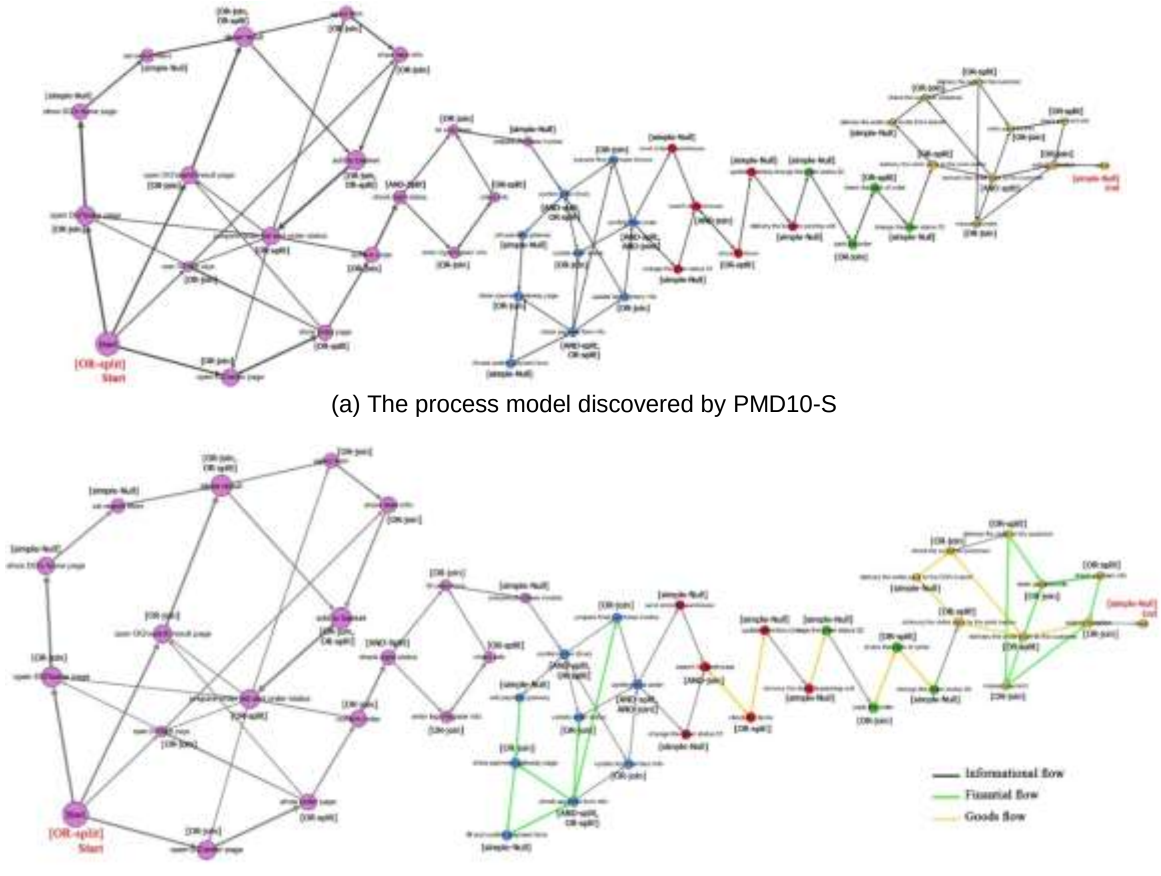
The simplicity S1 of the methods PMD04 and PMD05 are higher than the proposed method. The reason is that there is not an appropriate layout for linear visualizing of the process model in Gephi. Thus, if an algorithm is designed for drawing the network linearly, the simplicity S1 of the proposed method will be improved.

6.2.2 Model Accuracy

As explained, the accuracy of the model is examined using the accuracy of the structure and attributes. The structural accuracy of the identified connected tasks in the discovered model is presented in table 12. Table 12 shows the ratio of the number of connected discovered tasks to the total number of tasks in the original process model. Models discovered by PMD01, PMD02, and PMD03 do not have supreme structural accuracy for identifying connected tasks.

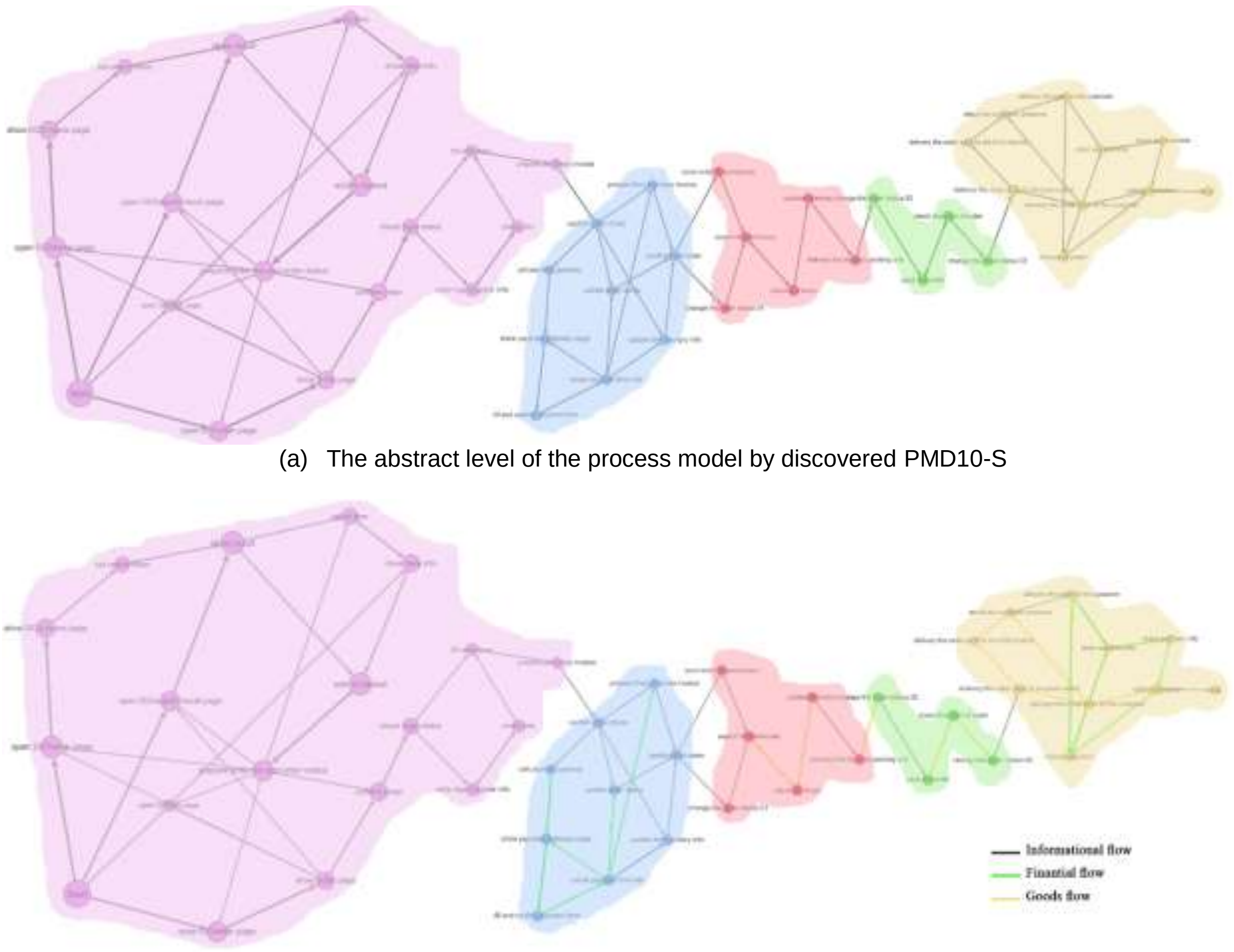
The structural accuracy of the discovered relations among the tasks is also examined. Figures 14, 15, 16, and 17 illustrate the results. Based on our results, our proposed method has the most structural accuracy. According to figure 14, the precision of models PMD05, PMD06, and PMD08 is significant, but PMD06 and PMD08, are not accurate because the number of discovered relations among tasks is too large compared to other methods. This issue is observable in figures A1 and A2 in the Appendix A. There are links among many tasks in the models owing to the black transition, which results in the identification of wrong relations. Consequently, PMD06 and PMD08 have lower Jaccard values (see figure 14). The ratio of undiscovered edges to correctly discovered edges among tasks is high in Alpha models (i.e., PMD01, PMD02, and PMD03). These methods did not discover a significant number of relations among tasks in the initial process model due to the focus on simplicity (see figure 16).

Since it is not possible to identify the most accurate model just based on one of the metrics of Jaccard, PPV, and FDR, the results are also evaluated by CFM. Figure 17 displays the accuracy of models based on CFM. Figure 17 clearly and fairly indicates that PMD10-S and PMD10-L methods are more accurate than others. The proposed method can determine the structure of the process model more accurately by detecting noise and identifying the meaningful network structures of activity $ActN_L$. Also, due to PMD10-S identifies motifs based on the color of the vertices, the structural accuracy has a negligible difference compared to the PMD10-ML method. PMD10-ML performs model discovery and identifies motifs based on the color of the vertices and the color of the edges among them.



(a) The process model discovered by PMD10-S

(b) The process model discovered by PMD10-ML
Figure 12: The discovered process model by our proposed method



(a) The abstract level of the process model by discovered PMD10-S

(b) The abstract level of the process model by discovered PMD10-ML
Figure 13: The abstract level of the process model discovered by the proposed method

Although, in published documents of other methods, it is declared that logical structures are identified, in the last implementation of the methods in the RapidMiner and Disco, none of them determines the logical structures. Therefore, in the case of the accuracy of vertices attributes, only the results for the proposed method are presented. Table 13 demonstrates the accuracy of attributes assigned to the activities. Based on the rules set out in step 5, the attributes of each activity are well-defined. The accuracy of the attributes in the PMD10-ML method is more than when the graph is modeled without distinction of flows. As a result, the structural accuracy and attribute accuracy of the multi-layered network modeling are better than the simple network.

Table 12: The ratio of the number of connected discovered tasks to the total number of tasks in the original process model

Method	Noise-0%	Noise-10%	Noise-20%	Noise-30%
PMD01	0.91	0.91	0.87	0.83
PMD02	0.98	0.98	0.96	0.91
PMD03	0.67	0.67	0.52	0.43
PMD04	1	1	1	1
PMD05	1	1	1	1
PMD06	1	1	1	1
PMD07	1	1	1	1
PMD08	1	1	1	1
PMD09	1	1	1	1
PMD10-ML	1	1	1	1
PMD10-S	1	1	1	1

In the process model, the number of the attribute with value XOR is more than the number of value AND. Due to this fact, wrong identification of one attribute with value AND leads to a large change in percentage error.

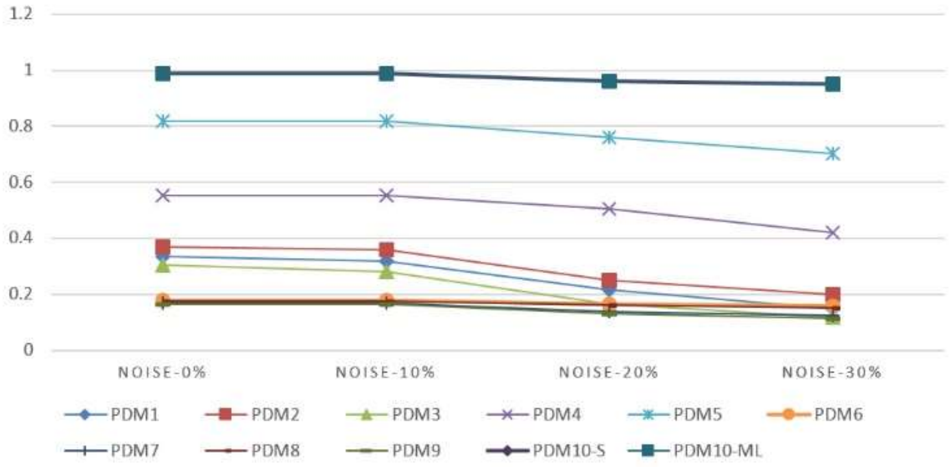


Figure 14: Structural accuracy of models based on the Jaccard

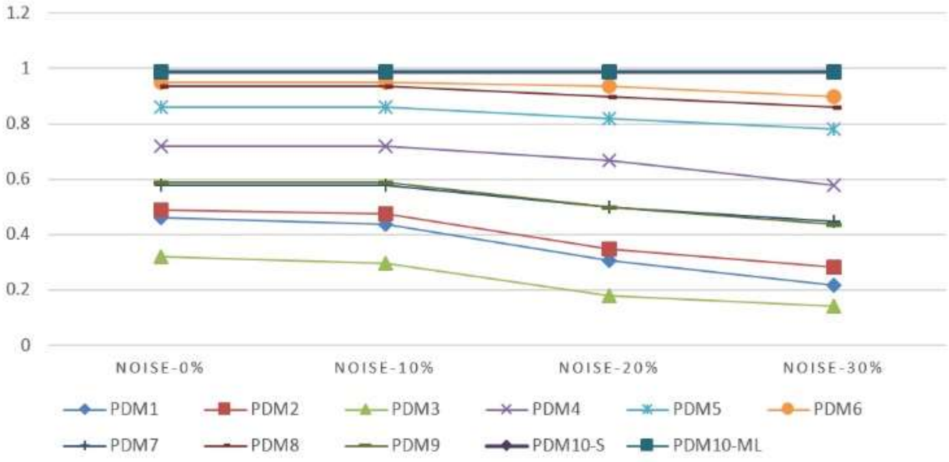


Figure 15: Structural accuracy of models based on the PPV/Precision

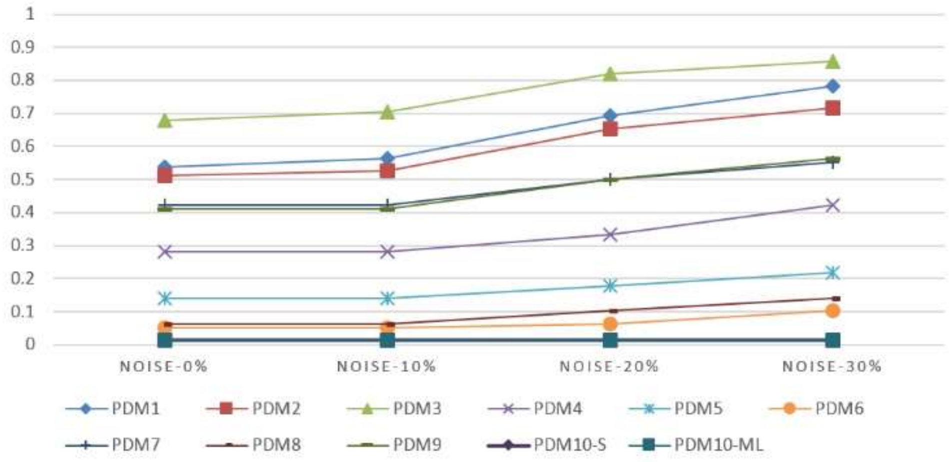


Figure 16: Structural accuracy of models based on the FDR

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762021000200104

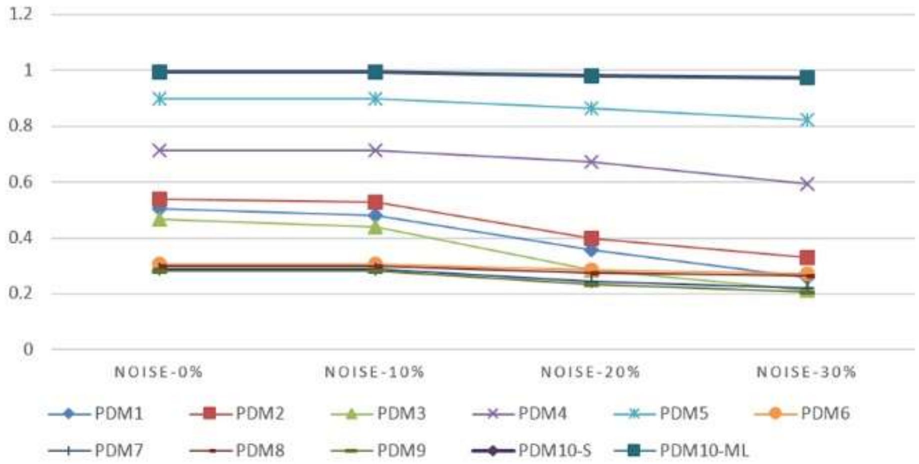


Figure 17: Structural accuracy of models based on the CFM

Table 13: Evaluating the accuracy of the attributes based on the percentage error

Value of attribute	Method	Noise-0%	Noise-10%	Noise-20%	Noise-30%
Simple	PMD10-ML	7.69%	7.69%	0	7.69%
	PMD10-S	7.69%	7.69%	0	7.69%
AND Split	PMD10-ML	33.33%	33.33%	33.33%	33.33%
	PMD10-S	66.67%	66.67%	66.67%	66.67%
XOR Split	PMD10-ML	11.76%	11.76%	17.65%	11.76%
	PMD10-S	17.65%	17.65%	23.53%	17.65%
AND Join	PMD10-ML	0	0	0	0
	PMD10-S	0	0	0	0
XOR Join	PMD10-ML	0	0	5.26%	5.26%
	PMD10-S	0	0	5.26%	5.26%

6.3 Discussion

Table 14 presents a summary of the essential features of the proposed method that has led to meet the process discovery challenges in the context of social commerce systems. Community detection in the network $ActN_L$ based on the Girvan-Newman method [30], [69] made it possible to identify sub-processes in which activities have a dense connection. Each community is a set of activities associated with a sub-process that is strong in terms of internal communication and weak in terms of relations outside the sub-process to which it belongs. Hence, by identifying communities, a macro-vision of the underlying sub-processes/processes in the event log is obtained. It leads to a better perception of the model and simplification of the discovered model.

If the event log is related to more than one organization or business, and we face a cross-organizational mining problem, communities in the first stage can represent the scope of any organization or business. Therefore, it shows inter-organization relationships. Then we can perform community detection inside each community. Detecting communities makes it possible to identify the internal processes of each organization.

The distinction of work flows among activities has made it easier for the human user to understand the discovered process model. As far as we have reviewed, the event log did not have the feature of flow type. We suggest adding flow type to the event log as a valuable data field that helps to identify the process model more accurately. Furthermore, the type of work flow provides valuable insight into the process that can provide process mining goals.

The higher accuracy of the proposed method compared to other methods is due to the discovery of significant structures using motif identification. Because the motifs are meaningful subgraphs, they are not random structures. After identifying the motifs and selecting the important motifs, the edges of the activity network $ActN_L$ that did not appear in any of the motifs are removed. Network analysis methods such as identifying motifs are focused on the structure of relationships, so better results were obtained than other methods. Since motif detection methods are not capable of identifying self-loops, our proposed method is not able to identify this structure. However, other loop structures are well recognized.

The Alpha series algorithms (PMD01, PMD02, and PMD03) tend to focus on simplicity so that the model does not take the form of a spaghetti model. Due to the considerable amount of noise in the event log, the model discovered by these algorithms did not show good results. In all three models, there are some disconnected tasks, which reduces the simplicity and accuracy of the model. The model discovered by PMD01 and PMD03 even have dangling (tasks that are not connected to End).

PMD09 (implemented by Disco) is based on the fuzzy theory. This tool is human-centred and is suitable for someone familiar with process mining. So, for a user who does not have enough knowledge, it is not appropriate. By changing the settings and filtering the tasks and relationships among them, a simpler model is obtained. PMD04 is also based on fuzzy theory, so, a simpler PMD09 model will be a model similar to that of PMD04.

Although other methods claim that they identify logical structures, implementation of these methods in RapidMiner and Disco does not have logical structures. Only a human user can deduce logical structures by observing the model. In the proposed method, we can determine the attribute of each task/activity based on the structure of the network subgraphs as well as the in-degree and out-degree of each activity, the type of relation, and the weight of the communication. Attributes determine the type of activity based on the logical structure and decision making.

It has been shown that in addition to Petri networks, the fuzzy model, and heuristic net, we can also represent the discovered model based on the attributed network. Displaying a model based on the attributed network, in which the subgraphs are determined based on community detection, as well as the distinct work flows, contribute to a greater understanding of the model. So, research contributions are:

- Modeling of the process model discovery problem using multilayered networks was first performed in this study.
- The use of motif discovery methods and the discovery of statistically significant structures confirm that these structures are not formed randomly or by chance. Therefore, there is a meaningful and significant relationship between activities. The composition of meaningful structures leads to discovering the business process structure. The proposed method is a block method. Each subgraph and motif are equivalent to a block of a process that their combination forms a business process model. Social network analysis methods and metrics were used to discover the process model for the first time in this study.
- Awareness of the type of flow generated by different activities can help to discover a more accurate, understandable and transparent model. The business process model discovered by our proposed method is multi-perspective. The model includes both the control flow and the type of workflow generated for each activity in the process.
- The business process was modeled based on attributed and multilayered networks for the first time in this research. The proposed model can be converted to Petri nets.

Consequently, from the business process management viewpoint, the previous methods do not discover the commercial process accurately and straightforward compared to the proposed method.

Table 14: Features of the proposed method

Features	Noise resistant	Encounter to incomplete event log	Low frequency events/traces detection	Generation of simple model (non-spaghetti)	Specified Sub-process limit	Identifying complex structures	Encounter to different working flows	Determining decision points (split-join)	Determining the type of decision
How does the proposed method do it?	Motif discovery and selection of subgraph based on the statistical significance			Separating work flows using multi-layered networks, and community discovery		Discovery and merging of subgraphs (creating the main structure of process from infrastructure and basic small components)	Graph modeling based on the multi-layered networks	Based on the in/out degree of activities	Using the type of work flow, weight of the relationships between activities, business rules, and graph mining

7 Conclusion and Future Works

Network analysis is a powerful and widely used tool in process mining, which has been taken into account so far in organizational mining. In this study, for the first time, a new approach was developed to discover the process model using network analysis. The proposed approach is based on graph theory and utilizes the methods and concepts of network analysis. We also included a new feature in the events log: the flow type. In the proposed method, multi-layered networks (heterogeneous networks) modeling, attributed networks, community detection, and the discovery and analysis of motifs were employed. It was shown that community detection leads to the simplicity of the model and the determination of the sub-processes' scope. A more accurate model is also obtained using multi-layered network modeling. Furthermore, the proposed model was able to determine the logical structures and priorities of the tasks/activities more accurately in comparison to other methods.

Moreover, based on the results, if the event log does not contain the flow type, the simple network-based approach is still usable, and the results are better than other similar methods. The attributed network was also used for the first time as a method for illustrating the process model discovered in this research. The attributed network was converted to the Petri net in order to be compared to other methods. The advantages of the proposed approach are the ability to deal with incomplete information, noise, model simplicity, and high accuracy.

7.1 Limitations

The limitations of the present study are mainly related to the complexities of the problem and the lack of adequate and appropriate data set. In real social commerce systems, additional information can be obtained regarding the execution of business processes, such as the start and end time of each activity, contextual information relevant to each case, and resource information. Another limitation is the inability to examine all mentioned information and their effects on each other and to evaluate their impact on process mining. Therefore, it may not produce very accurate results. Human resource interactions can also be governed by different terms and conditions, and different parameters affect them. It is challenging to consider all the effective parameters and use them in problem modeling. Consequently, it is not possible to investigate all of these complexities in this study.

7.2 Implications

Due to the clarification and mining of new knowledge, process mining affects managers' decision-making and leads to more significant intuition and representation. In the fast-changing, dynamic, and competitive business environment, business processes improving and supporting is a vital need [81]. The process discovered from executed process logs in information systems or commercial systems can answer the questions of managers and process analysts. Also, discovering the real executed process enables businesses to redesign the initial process, optimize it, and perform improvement actions [80]. By comparing the initial process and the actual process behavior in the event log (Delta Analysis), businesses can follow the process in compliance with the regulations. Non-conformity of the discovered process with the initial process (the designed one) can have many different reasons, e.g., fraud, model incompatibility, or developing a model for ideal and unrealistic situations [79].

According to previous work [25], the distinction between different work flows improves identifying and discovering critical points of the process. Non-integrating of different flows leads to a better understanding and insight into the real executed process. As a result, business process analysts can more accurately analyze the discovered process for each work flow. For example, while some parts of the process that contain a flow of goods conform with the initial process, other parts of the process that contain an information flow that differs from the initial process. Process discovery does not determine the reason for these differences. Thus, in the next step, it is necessary to identify them. The proposed network analysis-based approach has the potential to utilize cross-organizational mining and the discovery of complex process models. Because by applying community detection, it is possible to determine the scope of the processes of each organization, and then individually study each identified community. In other words, it is possible to identify and analyze communities hierarchically, and in each community extract the process model.

Process mining of the interactions among sensors in the context of IoT (internet of things) can also be defined. Due to the different nature of flows created among objects, multi-layered network modeling lead to more efficient and more accurate insights in this context. As in this study, the difference between modeling of multi-layered network and the simple network was observed.

Further, due to the simplicity of the discovered model and its high accuracy (even if the type of flow is not identified), this can be added to the Prom tool or the RapidProm plugin in the RapidMiner. Of course, it's also possible to add it to the Gephi network analysis tool or NetworkX network analysis library in Python. It is also feasible to filter the edges by weight. This can be done by a human user. Filtering the edges leads to the simplicity of the discovery model, but some existing relationships among the tasks in the primary model may be eliminated. But it is, however, a useful feature that can be considered. Additionally, many statistical reports can be based on network analysis, for example, the most important tasks based on centrality metrics.

7.3 Future Works

Although the proposed approach of this research has high accuracy and simplicity of the model, some issues need to be addressed in the future. Firstly, in futures works, we want to evaluate the proposed method based on the replay fitness and precision. These two metrics focus on the primary event log and the event log generated by the discovered process model. In this paper, we aim to discover a model that is as consistent as possible with the original process model. Therefore, we did not examine the matching of the primary event log and the discovered process model. Instead, we examined the conforming of the original process model and the discovered process model of the event log. As a second subject, we also intend to apply the proposed method to the non-commercial event log and examine its application to other domains. Additionally, we intend to determine the working group and role for each activity and identify the OR structures in order to improve the business process model discovered. We also want to use social relations among resources to discover the process model in futures, due to social relations influence on working interactions [25]. As a result, a multi-perspective model will be developed that does not include only the control flow. We intend to measure the simplicity of the model based on the Analytic Hierarchy Process (AHP) method. It is also worth trying to develop a tool that can fully visualize the final model based on the proposed method. We hope these are interesting to researchers.

Websites List

Site 1: PRoM
http://www.promtools.org

Site 2: RapidPRoM
http://www.rapidprom.org

Site 3: Disco
http://www.fluxicon.com/disco/

Site 4: Gephi
https://gephi.org/

Site 5: FANMOD
http://www.theinf1.informatik.uni-jena.de/motifs/

Site 6: WoPeD
https://woped.dhbw-karlsruhe.de/?page_id=22

References

[1] S. Alizadeh and A. Norani, ICMA: A new efficient algorithm for process model discovery, Applied Intelligence, vol. 48, pp. 1-18, 2018.
[2] C. C. Alves, Social Network Analysis for Business Process Discovery. Portugal: Technical University of Lisbon, 2010.
[3] A. K. Alves de Medeiros, Genetic process mining, Ph.D. dissertation, Eindhoven University of Technology, 2006.
[4] A. K. Alves de Medeiros, A. Guzzo, G. Greco, W. M. P. van der Aalst, and A. J. M. M. Weijters, Process Mining Based on Clustering: A Quest for Precision, presented at the Business Process Management Workshops, in Proceedings International Conference on Business Process Management, 2007, pp. 17-29.
[5] A. Appice, Towards mining the organizational structure of a dynamic event scenario, Journal of Intelligent Information Systems, vol. 50, no. 1, pp. 165-193, 2018.
[6] H. Ariouat, K. Barkaoui and J. Akoka, Improving process models discovery using AXOR clustering algorithm, presented at the Information Science and Applications, Lecture Notes in Electrical Engineering, Springer, Berlin, pp. 623-629, 2015.
[7] H. Ariouat, A. H. Cairns, K. Barkaoui, J. Akoka, and N. Khelifa, A two-step clustering approach for improving educational process model discovery, presented at the 2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Paris, France, 13-15 June, 2016.
[8] A. B. Badiru, Handbook of Industrial and Systems Engineering, Second Edition. CRC Press, Taylor & Francis Group, 2014.
[9] Y. Baghdadi, From e-commerce to social commerce: A framework to guide enabling cloud computing, Journal of Theoretical and Applied Electronic Commerce Research, vol. 8, no. 3, pp. 12-38, 2013.
[10] F. Bezerra and J. Wainer, Algorithms for anomaly detection of traces in logs of process aware information systems, Information Systems, vol. 38, no. 1, pp. 33-44, 2013.
[11] R. P. J. C. Bose and W. M. P. van der Aalst, Context aware trace clustering: Towards improving process mining results, in Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA, 2009, pp. 401-412.
[12] R. P. J. C. Bose and W. M. P. van der Aalst, Trace clustering based on conserved patterns: Towards achieving better process models, in Business Process Management Workshops (S. Rinderle-Ma, S. Sadiq and F. Leymann, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 170-181.
[13] J. C. A. M. Buijs, B. F. van Dongen and W. M. P. van der Aalst, Quality dimensions in process Discovery: The importance of fitness, precision, generalization and aimplicity, International Journal of Cooperative Information Systems, vol. 23, no. 01, p. 1440001, 2014.
[14] J. J. Buijs, Flexible evolutionary algorithms for mining structured process models, Technische Universiteit Eindhoven, 2014.

[15] L. Chen, M. Mulvenna, D. Quinn, Social Network Analysis: A Survey, International Journal of Ambient Computing and Intelligence, vol. 4, no. 3, pp. 46-58, 2012

[16] A. Clauset, M. E. J. Newman and C. Moore, Finding community structure in very large networks, Physical Review E, vol. 70, no. 6, pp. 1-6, 2004.

[17] J. Cui, F. Wang and J. Zhai, Citation Networks as a Multi-layer Graph: Link Prediction and Importance Ranking, 2010.

[18] R. G. Curty and P. Zhang, Social commerce: Looking back and forward, presented at the ASIST 2011, New Orleans, LA, USA, vol. 48, no. 1, pp. 1-10, 2011.

[19] A. K. A. de Medeiros, A. J. M. M. Weijters and W. M. P. van der Aalst, Genetic process mining: A basic approach and its challenges, in Business Process Management Workshops (C. J. Bussler and A. Haller, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 203-215.

[20] J. de San Pedro and J. Cortadella, Discovering duplicate tasks in transition systems for the simplification of process models, in Proceedings of the 2016 International Conference on Business Process Management, Business Process Management, Seville, Spain, vol. 9850, 2016, pp. 108-124.

[21] M. Dehghan Bahabadi, S. A. Hashemi G. and L. Esmaeili, A novel C2C e-commerce recommender system based on link prediction: Applying social network analysis, International Journal of Advanced Studies in Computer Science & Engineering, vol. 3, no. 7, pp. 1-8, 2014.

[22] Z. Ebadi Abouzar, L. Esmaeili and S. A. Hashemi G., Centrality measures analysis in overlapped communities: An empirical study, presented at the 8th International Symposium on Telecommunications, Tehran, Iran, 27-28 Sept., 2016.

[23] L. Esmaeili and S. A. Hashemi G., Rural ntelligent public transportation system design: Applying the design for re-engineering of transportation ecommerce system in Iran, International Journal of Information Technologies and Systems Approach, vol. 8, no. 1, pp. 1-27, 2015.

[24] L. Esmaeili and S. A. Hashemi G., A systematic review on social commerce, Journal of Strategic Marketing, vol. 27, no. 4, pp. 317-355, 2019.

[25] L. Esmaeili and S. A. Hashemi G., Conformance checking of the activity network with the social relationships structure in the context of social commerce, Journal of Theoretical and Applied Electronic Commerce Research, vol. 15, no. 2, pp. 93-121, 2020.

[26] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters, vol. 27, no. 8, pp. 861-874, 2006.

[27] D. R. Ferreira and C. Alves, Discovering user communities in large event logs, Lecture Notes in Business Information Processing, vol. 99, pp. 123-134, 2012.

[28] C. W. Günther and A. Rozinat, Disco: Discover your processes, BPM 2012 Demonstration Track, vol. 940, pp. 40-44, 2012.

[29] H. Ghavamipoor and S. A. Hashemi Golpayegani, A reinforcement learning based model for adaptive service quality management in e-commerce websites, Business & Information Systems Engineering, vol. 62, pp. 159-177, 2019.

[30] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, in Proceedings of the National Academy of Sciences, USA, 2002, pp. 7821-7826.

[31] J. Gorner, J. Zhang and R. Cohen, Improving trust modeling through the limit of advisor network size and use of referrals, Electronic Commerce Research and Applications, vol. 12, pp. 112-123, 2013.

[32] C. W. Günther and W. M. P. van der Aalst, Fuzzy Mining - adaptive process simplification based on multi-perspective metrics, in Business Process Management (G. Alonso, P. Dadam and M. Rosemann, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 328-343.

[33] S. Harenberg et al., Community detection in large-scale networks: a survey and empirical evaluation, Wiley Interdisciplinary Reviews: Computational Statistics, vol. 6, no. 6, pp. 426-439, 2014.

[34] J. Herbst, A machine learning approach to workflow management, in Machine Learning: ECML 2000 (R. López de Mántaras and E. Plaza, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 183-194.

[35] J. Herbst and D. Karagiannis, Integrating machine learning and workflow management to support acquisition and adaptation of workflow models, International Journal of Intelligent Systems in Accounting, Finance & Management, vol. 9, no. 2, pp. 67-92, Jun. 2000.

[36] J. Herbst and D. Karagiannis, Workflow mining with InWoLvE, Computers in Industry, vol. 53, no. 3, pp. 245-264, Apr. 2004.

[37] H. Jeong, H. Kim and K. P. Kim, Betweenness centralization analysis formalisms on workflow-supported org-social networks, presented at the 16th International Conference on Advanced Communication Technology, Pyeongchang, South Korea, 16-19 Feb., 2014.

[38] J.-Y. Jung and J. Bae, Workflow clustering method based on process similarity, in Proceedings of the ICCSA’06 of the 2006 International Conference on Computational Science and Its Applications, Glasgow, UK, 2006, pp. 379-389.

[39] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs, Bioinformatics, vol. 20, no. 11, pp. 1746-1758, Jul. 2004.

[40] Y. Kavurucu, A comparative study on network motif discovery algorithms, International Journal of Data Mining and Bioinformatics, vol. 11, no. 2, p. 180, 2015.

[41] P. Kazienko, K. Musial, E. Kukla, T. Kajdanowicz, and P. Bródka, Multidimensional social network: Model and analysis, in Proceedings of the 2011 International Conference on Computational Collective Intelligence, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 6922, 2011, pp. 378-387.

[42] E. Khadangi, A. Bagheri and A. Zarean, Empirical analysis of structural properties, macroscopic and microscopic evolution of various Facebook activity networks, Quality & Quantity, vol. 52, no. 1, pp. 249-275, 2018.

[43] C. Largeron, P.-N. Mougel, R. Rabbany, and O. R. Zaïane, Generating Attributed Networks with Communities, PLOS ONE, vol. 10, no. 4, p. e0122777, 2015.

[44] S. J. J. Leemans, D. Fahland and W. M. P. van der Aalst, Discovering block-structured process models from event logs - a constructive approach, in Application and Theory of Petri Nets and Concurrency (J.-M. Colom and J. Desel, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 311-329.

[45] S. J. J. Leemans, D. Fahland and W. M. P. van der Aalst, Discovering block-structured process models from event logs containing infrequent behaviour, in Business Process Management Workshops (N. Lohmann, M. Song and P. Wohed, Eds.). Cham: Springer International Publishing, 2014, pp. 66-78.

[46] S. J. J. Leemans, D. Fahland and W. M. P. van der Aalst, Discovering block-structured process models from incomplete event logs, in Application and Theory of Petri Nets and Concurrency (G. Ciardo and E. Kindler, Eds.). Cham: Springer International Publishing, 2014, pp. 91-110.

[47] Q. Li, A novel Likert scale based on fuzzy sets theory, Expert Systems with Applications, vol. 40, no. 5, pp. 1609-1618, 2013.

[48] T.-P. Liang and E. Turban, Introduction to the special issue - social commerce: A research framework for social commerce, International Journal of Electronic Commerce, vol. 16, no. 2, pp. 5-13, 2012.

[49] L. Liao, X. He, H. Zhang, and T.-S. Chua, Attributed social network embedding, IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 12, pp. 2257-2270, 2018.

[50] X. Lu, D. Fahland, F. J. H. M. van den Biggelaar, and W. M. P. van der Aalst, Handling duplicated tasks in Process discovery by refining event labels, in Proceedings of the 2016 International Conference on Business Process Management, Lecture Notes in Computer Science, Springer, Cham, vol. 9850, 2016, pp. 90-107.

[51] S. Mardani, M. K. Akbari and S. Sharifian, Fraud detection in process aware information systems using mapreduce, in Proceedings of the 2014 6th Conference on Information and Knowledge Technology (IKT), Shahrood, Iran, 2014, pp. 88-91.

[52] S. Mardani and H. R. Shahriari, A new method for occupational fraud detection in process aware information systems, in 2013 10th International ISC Conference on Information Security and Cryptology (ISCISC), Yazd, Iran, 2013, pp. 1-5.

[53] K. Mark and L. Csaba, Analyzing Customer Behavior Model Graph (CBMG) using Markov Chains, in Intelligent Engineering Systems, 2007 International Conference on, Hotel Griff, 2007, pp. 71-76.

[54] M. Märtens, J. Meier, A. Hillebrand, P. Tewarie, and P. Van Mieghem, Brain network clustering with information flow motifs, Applied Network Science, vol. 2, no. 1, Dec. 2017.

[55] W. C. McDowell, R. C. Wilson, and C. O. Kile, An examination of retail website design and conversion rate, Journal of Business Research, vol. 69, no. 11, pp. 4837-4842, Nov. 2016.

[56] N. Meghanathan, Ed., Community Detection in Large-Scale Social Networks: A Survey, in Graph Theoretic Approaches for Analyzing Large-Scale Social Networks:, IGI Global, 2018, pp. 189-206.

[57] D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes, A methodology for workload characterization of E-commerce sites, in Proceedings of the 1st ACM conference on Electronic commerce - EC '99, Denver, Colorado, United States, 1999, pp. 119-128.

[58] B. Minaei-Bidgoli, L. Esmaeili, and M. Nasiri, Comparison of group recommendation techniques in social networks, in 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2011, pp. 236-241.

[59] M. Mojaveriyan, H. Ebrahimpour-komleh, and S. jalaleddin Mousavirad, IGICA: A hybrid feature selection approach in text categorization, International Journal of Intelligent Systems and Applications, vol. 8, no. 3, pp. 42-47, Mar. 2016.

[60] S. J. Mousavirad and H. Ebrahimpour-Komleh, Human mental search-based multilevel thresholding for image segmentation, Applied Soft Computing, Apr. 2019.

[61] S. J. Mousavirad, H. Ebrahimpour-Komleh, and G. Schaefer, Effective image clustering based on human mental search, Applied Soft Computing, vol. 78, pp. 209-220, 2019.

[62] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai, Dynamic social community detection and its applications, PLOS One, vol. 9, no. 4, 2014.

[63] S. M. Rahman and M. S. Raisinghani, Electronic Commerce: Opportunity and Challenges. USA: Idea Group Publishing, 2000.

[64] M. Rocha and P. G. Ferreira, Motif Discovery Algorithms, in Bioinformatics Algorithms, Elsevier, 2018, pp. 221-236.

[65] M. A. Rodriguez and J. Shinavier, Exposing multi-relational networks to single-relational network analysis algorithms, Journal of Informetrics, vol. 4, no. 1, pp. 29-41, Jan. 2010.

[66] G. Ruffo, R. Schifanella, M. Sereno, and R. Politi, WALTy: a user behavior tailored tool for evaluating web application performance, in Proceedings of the 2004 Third IEEE International Symposium on Network Computing and Applications (NCA 2004), Boston, MA, USA, 2004, pp. 77-86.

[67] C. Sammut and G. I. Webb, Eds., Encyclopedia of Machine Learning. New York ; London: Springer, 2010.

[68] R. Sarno, Y. A. Effendi and F. Haryadita, Modified time-based heuristics miner for parallel business processes, International Review on Computers and Software (IRECOS), vol. 11, no. 3, p. 249, 2016.

[69] K. Sathiyakumari and M. S. Vijaya, Community detection based on girvan newman algorithm and link analysis of social media, in Digital Connectivity - Social Impact, vol. 679 (S. Subramanian, R. Nadarajan, S. Rao, and S. Sheen, Eds.). Singapore: Springer Singapore, 2016, pp. 223-234.

[70] G. Schimm, Mining exact models of concurrent workflows, Computers in Industry, vol. 53, no. 3, pp. 265-281, 2004.

[71] J. Song, M. Kim and H. Kim, A Framework: Workflow-Based Social Network Discovery and Analysis, presented at the 13th International Conference on Computational Science and Engineering, 2010, pp. 421-426.

[72] M. Song, C. W. Günther and W. M. P. van der Aalst, Trace clustering in Process Mining, in Business Process Management Workshops, vol. 17 (D. Ardagna, M. Mecella, and J. Yang, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 109-120.

[73] M. Song and W. M. P. van der Aalst, Towards comprehensive support for organizational mining, Decision Support Systems, vol. 46, no. 1, pp. 300-317, 2008.

[74] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Second. Pearson Addison-Wesley, 2005.

[75] L. Tang, X. Wang and H. Liu, Community detection via heterogeneous interaction analysis, Data Mining and Knowledge Discovery, vol. 25, no. 1, pp. 1-33, 2012.

[76] S. S. Thorat and S. M. Shinde, A Survey on community detection, International Journal of Science and Research, vol. 4, no. 1, pp. 670-673, 2013.

[77] S. Uddin and M. J. Jacobson, Dynamics of email communications among university students throughout a semester, Computers & Education, vol. 64, pp. 95-103, 2013.

[78] W. M. P. van der Aalst, Workflow mining: A survey of issues and approaches, Data and Knowledge Engineering, vol. 7, no. 2, pp. 237-267, 2003.

[79] W. M. P. van der Aalst, Business alignment: using process mining as a tool for Delta analysis and conformance testing, Requirements Engineering, vol. 10, no. 3, pp. 198-211, 2005.

[80] W. M. P. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag Berlin Heidelberg, 2011.

[81] W. M. P. van der Aalst et al., Process mining manifesto, in BPM 2011 Workshops, Part I, LNBIPF (Daniel et al. (Eds.). France: Clermont-Ferrand, 2012, pp. 169-194.

[82] W. M. P. van der Aalst and C. W. Gunther, Finding structure in unstructured processes: The case for process mining, presented at the Seventh International Conference on Application of Concurrency to System Design, Bratislava, Slovakia, 2007.

[83] W. M. P. van der Aalst and K. van Hee, Workflow Management Models, Methods, and Systems. Massachusetts London, England: The MIT Press Cambridge, 2002.

[84] W. M. P. van der Aalst, H. A. Reijers, and M. Song, Discovering social networks from event logs, Computer Supported Cooperative Work, vol. 14, no. 6, pp. 549-593, 2005.

[85] W. van der Aalst, T. Weijters, and L. Maruster, Workflow mining: discovering process models from event logs, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128-1142, 2004.

[86] B. F. van Dongen and A. Adriansyah, Process mining: Fuzzy clustering and performance visualization, in Business Process Management Workshops, vol. 43 (S. Rinderle-Ma, S. Sadiq, and F. Leymann, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 158-169.

[87] G. M. Veiga, Developing process mining tools: An implementation of sequence clustering for ProM, Master, Technical University of Lisbon, 2009.

[88] A. J. M. M. Weijters and J. T. S. Ribeiro, Flexible heuristics miner (FHM), in Proceedings 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Paris, France, 2011, pp. 310-317.

[89] A. J. M. M. Weijters and W. M. P. van der Aalst, Rediscovering workflow models from event-based data using little thumb, Integrated Computer-Aided Engineering, vol. 10, no. 2, pp. 151-162, 2003.

[90] L. Wen, J. Wang and J. Sun, Detecting implicit dependencies between tasks from event logs, in Frontiers of WWW Research and Development - APWeb 2006, vol. 3841 (X. Zhou, J. Li, H. T. Shen, M. Kitsuregawa, and Y. Zhang, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 591-603.

[91] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, A novel approach for process mining based on event types, Journal of Intelligent Information Systems, vol. 32, no. 2, pp. 163-190, 2009.

[92] L. Wen, J. Wang, W. M. P. 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.

[93] S. Wernicke, A faster algorithm for detecting network motifs, in Algorithms in Bioinformatics, vol. 3692 (R. Casadio and G. Myers, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 165-177.

[94] M. Zhang, Social network analysis: History, concepts, and research, in Handbook of Social Network Technologies and Applications (B. Furth. Ed.). Boston, MA: Springer 2010, pp. 3-21.

[95] W. Zhao and X. Zhao, Process mining from the organizational perspective, in 17th International Symposium, ISMIS, vol. 5722 (J. Rauch, Z. W. Raś, P. Berka, and T. Elomaa, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 701-708.

Appendix A: The Discovered Process Models Based on the Others Methods are Displayed in this Appendex

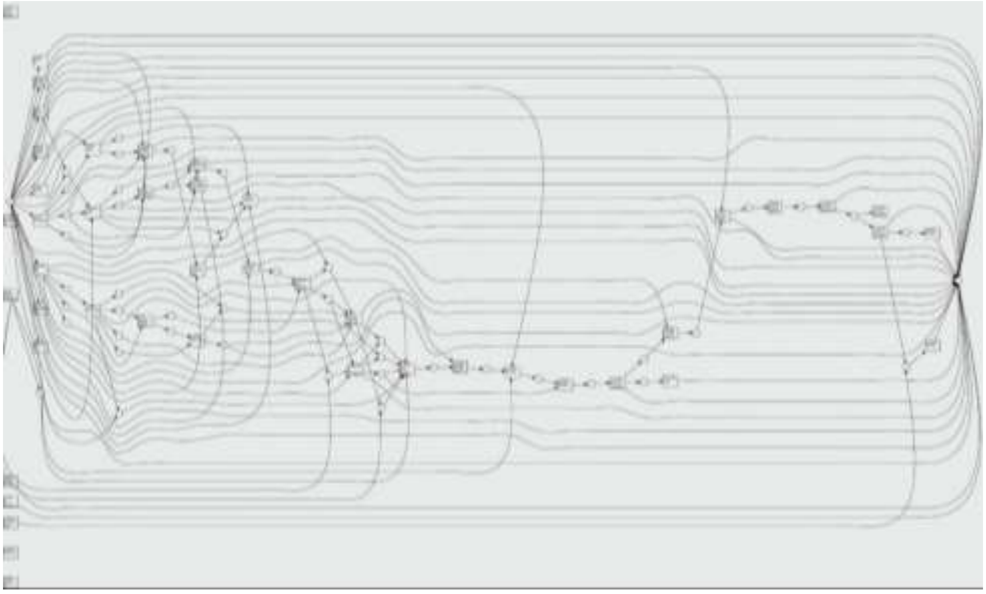


Figure A1: Process model discovered by PMD01

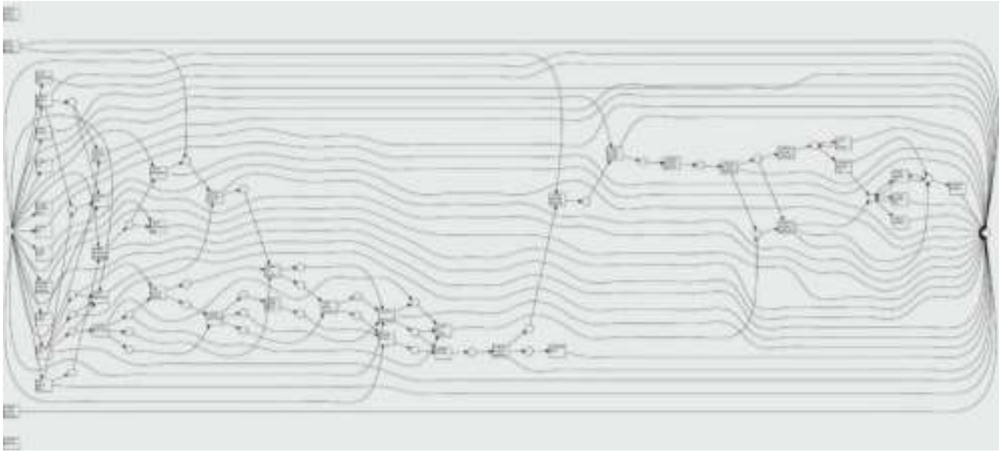


Figure A2: Process model discovered by PMD02

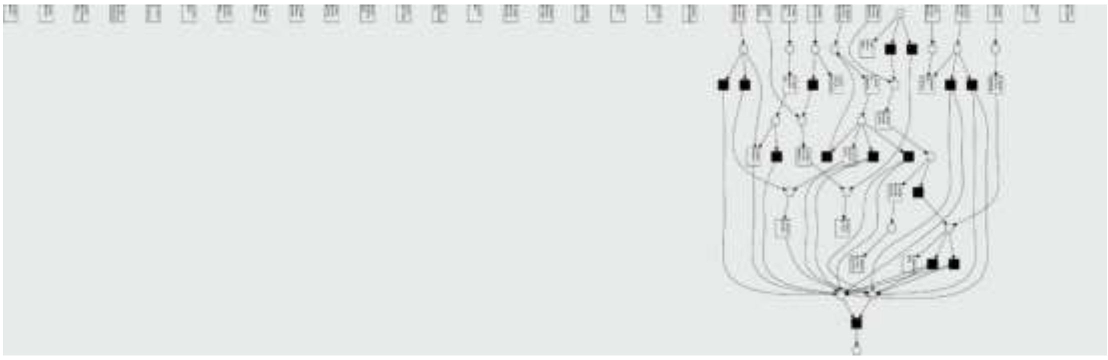


Figure A3: Process model discovered by PMD03

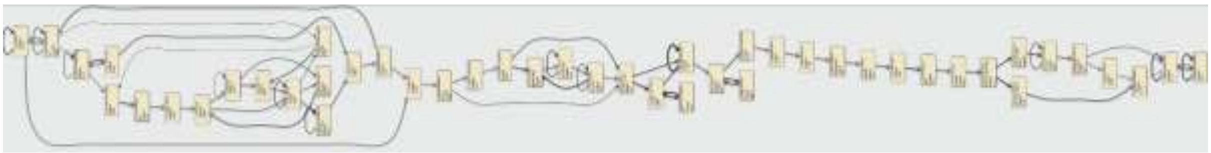


Figure A4: Process model discovered by PMD04

Figure A5: Process model discovered by PMD05

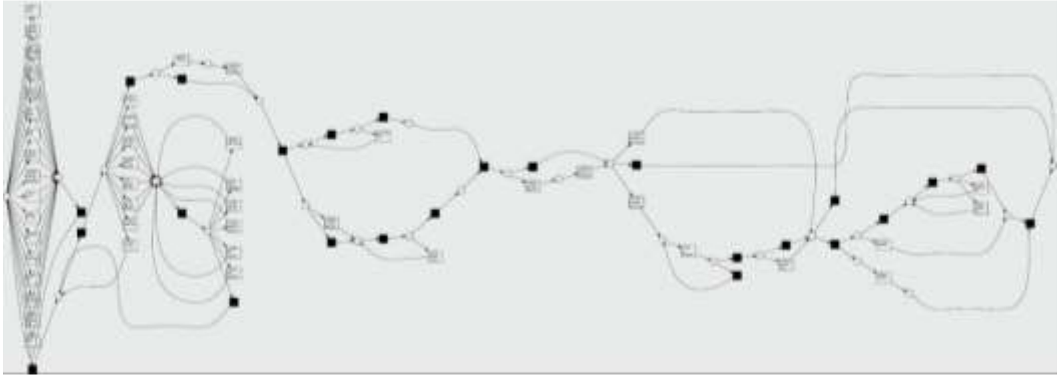


Figure A6: Process model discovered by PMD06

Figure A7: Process model discovered by PMD07

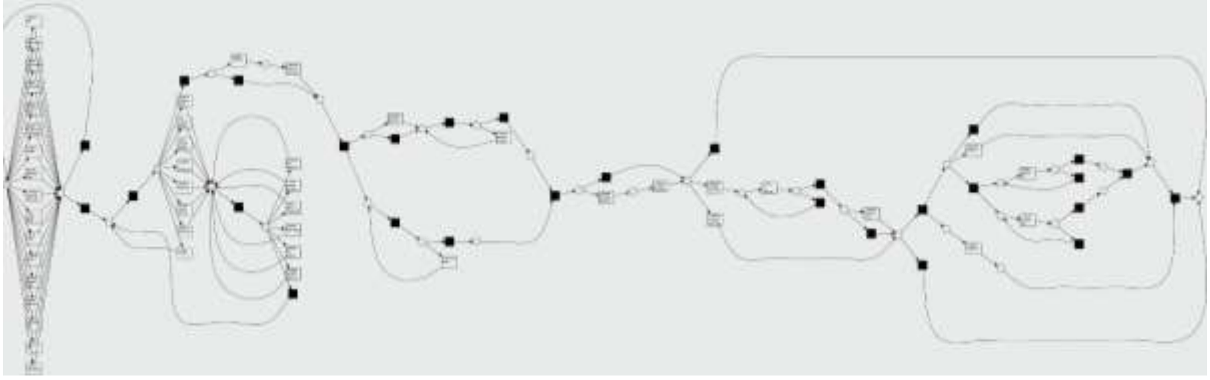


Figure A8: Process model discovered by PMD08

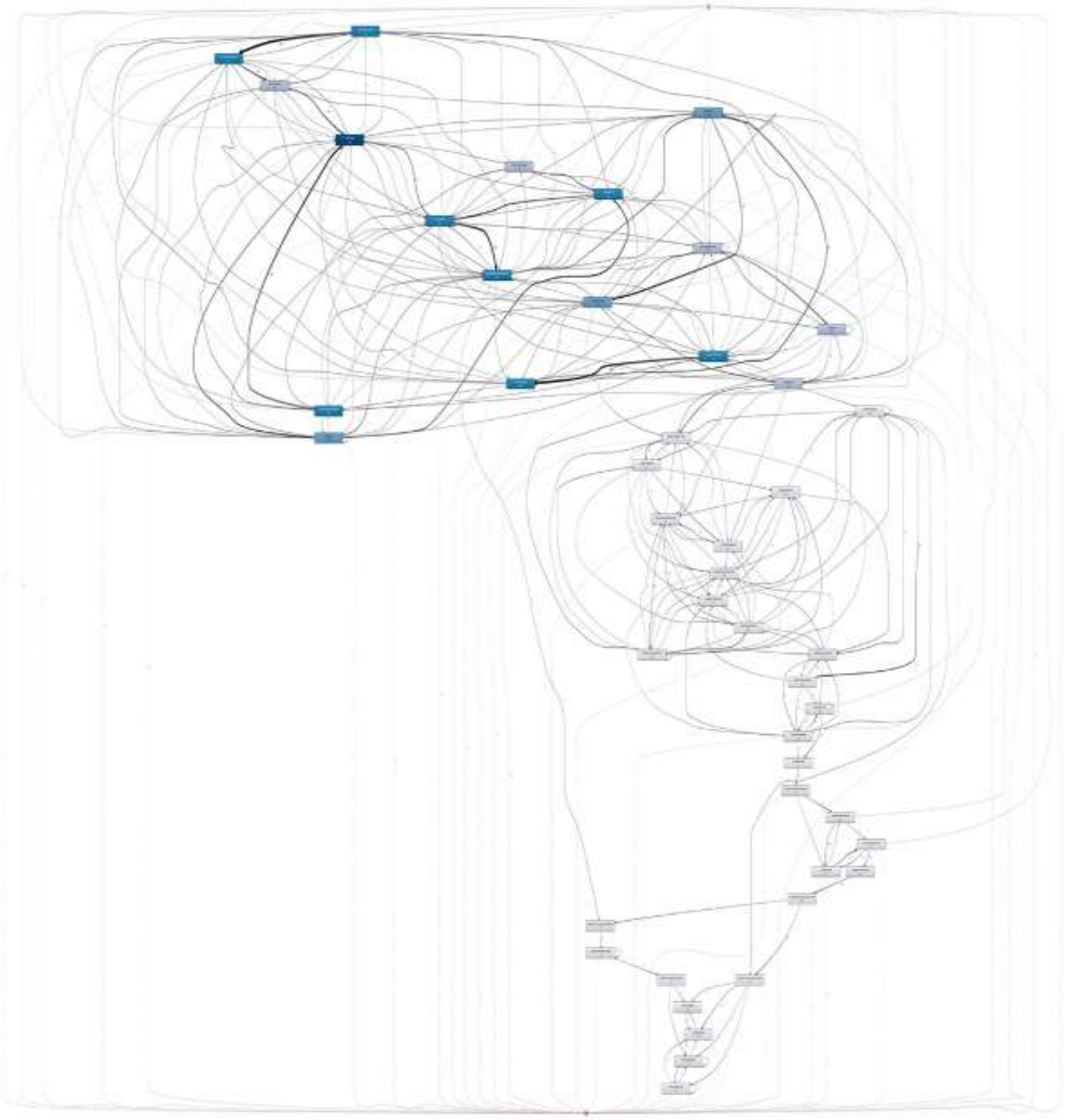


Figure A9: Process model discovered by PMD09