



# A new method for organizational process model discovery through the analysis of workflows and data exchange networks

Roshanak Aghabaghery<sup>1</sup> · Alireza Hashemi Golpayegani<sup>1</sup> · Leila Esmaeili<sup>1</sup>

Received: 18 May 2019 / Revised: 14 November 2019 / Accepted: 13 January 2020 / Published online: 27 January 2020  
© Springer-Verlag GmbH Austria, part of Springer Nature 2020

## Abstract

Nowadays, organizations use process-aware information systems to understand and apply rapid changes to their processes. Process mining techniques automatically extract true dimensions of organizational processes including process models from data sets like event logs stored in these information systems. In most studies performed in the area of process model discovery, only information of the event logs is used. However, in this research, a novel method of process discovery is proposed, which uses event logs as well as the information on the data exchange among organizational roles, which is derived from physical generalized flow diagram model. This information formed the basis of a two-layered network that represents handover flow and data exchange flow among organizational roles. Then, by extracting and analyzing motifs existing in this network, five rules are set that map motifs with certain features to logical structures constructing process models. Finally, by integrating those structures, the process model will be discovered. The advantage of the proposed method over the previous ones is that from the business process management viewpoint, it is more efficient in detecting sophisticated structures in the process model. It is also highly resistant to noise. These benefits are derived from the fact that it exerts data exchange information along with event log information. Doing various experiments and evaluation of their results using the  $F$ -measure confirmed the superiority of this method to previous ones from the viewpoint of the business process management.

**Keywords** Information systems · Process mining · Social network analysis · Motif discovery · Model discovery

## 1 Introduction

Today, organizations need to make rapid changes to their process models to be able to survive in the competitive world of business. The implementation of these changes first calls for an accurate understanding of the current organizational processes. To attain this goal, organizations have resorted to process-aware information systems such as the workflow management systems (WFMSs), enterprise resource planning (ERP), and customer-relationship management (CRM) systems (Van der Aalst 2014). These systems record information on the current actual organizational processes, but do not present an accurate model of these processes. Organization managers generally picture presumptive models of organizational processes, but there are often differences

between the actual process model and these presumptive models.

Consequently, there is a requirement for techniques which are able to automatically extract the actual process models from the data sets stored in the information systems. The process mining concept was framed to achieve this goal (Burattin 2015). One of the data sets stored in the information systems is event log. An event log contains the information on events unfolding during the organizational processes. Every event offers information on an activity (task), timestamp, case, and performer of the activity. Event logs have significant importance because the information contained in these logs is used in all process mining studies. Process mining techniques are generally classified into three categories (Van der Aalst 2014; Burattin 2015; Van Dongen and van der Aalst 2005):

- (a) Process discovery: In these techniques, the process model is extracted without any presumptive model using the data in the information systems.

✉ Alireza Hashemi Golpayegani  
sa.hashemi@aut.ac.ir

<sup>1</sup> ATECS Lab, Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Hafez Ave, Tehran, Iran

- (b) Conformance checking: In these techniques, the actual process model obtained through process discovery methods is compared to a presumptive model.
- (c) Extension: In these techniques, the presumptive model serves as the input and it is improved based on the information extracted from the information systems.

This research revolves around the process discovery techniques. The previous researches on process discovery have been fraught with the following problems (Aleem et al. 2015):

- (a) The inability to identify some complicated structures such as the non-free choice structures, short-length loops, hidden tasks, and duplicate tasks.
- (b) The inability to resist noise.
- (c) The inability to tackle incompleteness.
- (d) The inability to simplify the process model and producing sophisticated models for complicated processes. (These models are called the spaghetti models.)
- (e) The high computational complexity.

All of these problems eventually result in the development of process models, which either lack adequate efficiency or are incomprehensible and unanalyzable. Due to these problems, the present research is conducted to propose a novel process discovery technique that solves some of the problems described above to the possible extent and discovers more accurate process models than the previous methods.

All of the existing process discovery methods use only event logs. However, other types of information such as information on data exchange among processes' roles also exist in the information systems. Since organizational roles perform the process activities, the relations among these roles can disclose information on the process model. Therefore, in the approach proposed in this research, the event logs are used along with the information on data exchange among the roles, which is derived from the physical generalized flow diagrams (PGFDs).

In the proposed solution, network analysis techniques are used to discover the process model. It is also assumed that the information in the information systems is only about one process. To this end, event logs and the PGFD model are used to develop a bilayer network under a suitable scenario. This network represents the handover flow and data exchange flow in the process model. Afterward, by extracting the network motifs and subjecting them to structural analyses, rules are set to identify the logical relationships among the process activities. Finally, the current process model of the organization is discovered following the rules mentioned above.

This article is composed of seven sections: Section one presented an introduction to process mining and its necessities. The research objectives and methodology were also described in this section after explaining the relevant problems. Section two presents a review of the process discovery studies and compares their features in a table. Afterward, based on this table, the differences between the proposed solution and the previous methods are explained. In addition, the previous researches done in the area of process discovery which use social network analysis are introduced in this section as well. In section three, before the description of the proposed approach, the fundamentals concepts used in this approach are introduced. The research problem is mathematically modeled in section four, and section five offers a detailed description of each step of the proposed model. The evaluation method is also introduced in this section. The assessment results are presented and analyzed in section six, and finally, section seven provides the summary of the research steps and suggestions on the future research.

## 2 Related works

Process discovery methods are the pivots of this research. Hence, the previous researches on process discovery are reported in this section. In addition, since this research utilizes social network analysis techniques, the previous researches using these techniques in the area of process discovery are also introduced.

### 2.1 Process model discovery methods

The studies on process mining have mainly attempted to propose a process model discovery method. These studies are classified into five major categories by approach:

- (i) Deterministic mining approaches: The methods developed based on the alpha algorithm, such as the beta and alpha plus algorithms (Wen et al. 2006, 2009), belong to this group. The alpha algorithm introduced by Van der Aalst and Song (2004) was based on the running time of tasks in event logs; this algorithm defines a set of dependencies including causal, parallel, and choice relationships and maps each relation to a Petri net. The most important pitfall of these algorithms is that they overlook noise. Their advantage is that they can discover a workflow per process and display it as an SWF-net.<sup>1</sup> A SWF-net is a workflow for all of whose transitions it is always possible to reach to the end place and

<sup>1</sup> Sounded workflow Network.

for any transition there is a firing sequence enabling it (Van der Aalst et al. 2004).

- (ii) **Heuristic mining methods:** These methods use dependencies similar to the deterministic approaches, but they consider the dependencies along with their frequencies. These algorithms are based on the fact that with an increase in the frequency of a dependency (vice versa), the odds of randomness of that dependency decrease. Heuristic algorithms consist of three steps: (1) creating a dependency/frequency table from the event log; (2) creating a dependency/frequency graph based on a set of heuristic rules; and (3) creating a Petri net using the information in the dependency/frequency graph and dependency/frequency table (Weijters and van der Aalst 2003). The primary advantage of these methods is their ability to resist noise and handle incompleteness.
- (iii) **Inductive approaches:** These algorithms function by the divide-and-conquer mechanism and involve two major steps. In step one, a stochastic activity graph (SAG) is created for the process instances. The SAG graph is a directed graph showing the direct dependencies among the activities. Afterward, the resulting SAG graph is transformed into a workflow model (Herbst and Karagiannis 2003). The mergeSeq, splitSeq, and splitPar algorithms proposed by Herbst (2000, 2002) and the algorithm introduced by Schimm (2003) come into this category. The important advantage of these algorithms is their ability to detect duplicate tasks.
- (iv) **Evolutionary approaches:** consisting of three main steps, these methods are based on the genetic algorithm (GA). First, a random initial population is created from the process model. Next, the fitness index is calculated for each process model forming the initial population. The fitness index determines the degree to which the model explains the behavior contained in the event log. After that, the initial population evolves by dint of the crossover and mutation genetic operators to create the next generation. Therefore, each generation evolves more than the previous generation step by step until a fit model with a large fitness index is found. Despite the ability of evolutionary approaches to detect most structures, they impose high computational complexity (Medeiros et al. 2006).
- (v) **Clustering-based approaches:** In these algorithms, first a process model is developed for each process instance. Afterward, the models are clustered and the juxtaposition of the clusters yields the main process model. In these approaches, process instances are clustered by the k-gram and bag of activities algorithms. To cluster the process instances using the algorithm introduced by Song et al. (2009), each process instance is shown as a vector, but there is no information on the con-

tents of activities per process instance. In the edit distance approach introduced by Bose and van der Aalst (2009), this problem is solved by assuming two process instances as a string and both strings are compared. The difference between the two strings is expressed as a cost. In trace clustering introduced by Bose and van der Aalst (2010), the process instances are compared, and the similarities between the two process instances are identified based on a fixed set of features. Finally, the similar instances are put in one cluster.

Table 1 presents a comparison of the previous researches. It shows the features, advantages, and disadvantages of each method.

All of the previous process discovery methods only use the event log data as seen in Table 1. In this paper, a novel process model discovery approach is proposed which uses the event log data as well as the information on data exchange among the roles. Given that this approach uses the information on the data exchange among the roles along with the event log data, it is expected to generate a more accurate process model. In this approach, social network analysis techniques are employed to discover the process model.

## 2.2 Application of social network analysis in the area of process discovery

Process mining researches can be focused in many perspectives, the most important of which are (Van der Aalst et al. 2004):

- (a) **Process perspective ('How')**: it focuses on the control flow which means the ordering of activities. The goal here is to find a good characterization of all the possible paths, expressed in terms of a Petri net.
- (b) **Organizational perspective ('Who')**: it focuses on the resources that means which resources are involved and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show relations among individual resources.
- (c) **Case perspective ('What')**: it focuses on properties of cases. Cases can be characterized by their paths in the process or by the values of the corresponding data elements.

The focus of all previous studies used social networks in the area of process discovery is on organizational perspective. These researches extract a social network whose nodes are resources. In the social network, edges have been formed based on three scenarios (Song and van der Aalst 2008):

**Table 1** Comparison of the previous process discovery methods

Method	Source of information	Approach	Advantages	Disadvantages
Deterministic mining approaches (Van der Aalst et al. 2004; Wen et al. 2006, 2009)	Event log	Definition of causal relationships among activities and determination of successions, parallel, and choice relations based on causal relationships	The ability to produce a sound workflow model per event log	Overlooking noise
Heuristic mining approaches (Weijters and van der Aalst 2003)	Event log	Determining the succession, parallel, and choice relationships among activities and considering the frequency of these relationships	Noise resistance	The inability to identify sophisticated structures such as non-free choices and short-length loops
Inductive mining approaches (Herbst 2000, 2002, 2003; Schimm 2003)	Event log	Creating the SAG graph for the process and converting it into a workflow model	The ability to identify duplicate tasks	Neglecting noise
Evolutionary approaches (Medeiros et al. 2006)	Event log	Using the genetic algorithm	The ability to identify most complex structures such as non-free choices and nested choice/parallel structures	High computational complexity
Clustering-based mining approach (Song et al. 2009; Bose and Van der Aalst 2009, 2010)	Event log	Generating a process model per process instance, clustering the resulting models, and creating a model for the main process through the juxtaposition of these clusters	The ability to simplify complicated models	The inability to identify complicated structures such as non-free choices and short-length loops

- (a) Handover scenario: In this scenario, an edge is formed between two nodes if, in a process trace, a role performs an activity on a case and then hands over to another role to perform a new activity.
- (b) Joint activity scenario: In this scenario, edges are formed among resources who perform same activities in different traces.
- (c) Joint cases scenario: In this scenario, edges are formed among resources who cooperate with each other on a single case. In other words, edges are formed among resources who perform activities in a single trace. The difference between this scenario and handover scenario is that, here, the order according to which activities are performed in a trace is not important.

In previous researches, based on the extracted social networks and using the social network analysis techniques and metrics such as betweenness, resources are classified in terms of organizational roles (Van der Aalst et al. 2007). In this research, the social network analysis techniques are utilized in the area of process perspective. In other words, they are exercised to discover the process control flow.

### 3 Fundamentals concepts

The approach proposed in this article is centered on a set of fundamentals, which are introduced in this section.

#### 3.1 Social network analysis

A social network is a network of people or groups and the relationships among them. People and groups constitute the nodes, while the edges are formed of the social relationships among people such as friendship, kinship, business, and shared interests. Social network analysis is a subgroup of network analysis techniques and refers to the use of processes, methods, and instruments that lend a better insight into the relationships and structures and help extract significant information (Esmaeili et al. 2011; Dehghan et al. 2014).

Data mining and analysis methods are generally classified into two categories: (1) classic techniques such as clustering and classification and (2) network analysis techniques. In the classic methods, a large volume of data is generated. Besides, these methods are offline. However, network analysis methods convert a large amount of data into a network; thus, they offer higher processing speed. They are also online (Arif 2015). Hence, the network analysis techniques were adopted in this study. Given that the organizational roles constitute network nodes in this study, the network is considered as a social network.

#### 3.1.1 Motifs

In network analysis, subgraphs with special features that are repeated in a network are called motifs. When a subgraph with a particular feature is repeated in a network, a specific behavior or structure is manifested in that network. Techniques known as motif discovery algorithms have been defined to discover the motifs. All motif discovery algorithms involve three steps: (1) searching and determining frequency of subgraphs of a certain size in a network; (2) identifying the isomorphic subgraphs and classifying them into different isomorphic classes; (3) identifying the statistical significance of the classes (by comparing the frequency of each class in the network to the frequency of that class in random networks). This statistical significance is measured by the  $z$ -score and  $p$  value criteria. A class with high  $z$ -core and low  $p$  value is statistically significant (Kavurucu 2015).

The motif discovery algorithms are classified by: (1) the method of subgraph enumeration and (2) the method of subgraph searching. As for subgraph enumeration, motif discovery algorithms are grouped into two classes (Kavurucu 2015; Wong and Baur 2010):

- (a) Tree-based algorithms: These algorithms enumerate all network subgraphs using a tree. The disadvantages of these algorithms are that they are time-consuming and costly because, with an increase in the network size and the subgraph size, the number of subgraphs grows exponentially.
- (b) Probabilistic algorithms: These algorithms sample the subgraphs instead of fully enumerating them and estimate their frequencies. The probabilistic algorithms have shorter run times than the full enumeration algorithms. Therefore, they are more suitable for discovering motifs in large networks and large-sized motifs. However, their sampling procedures are not entirely unbiased.

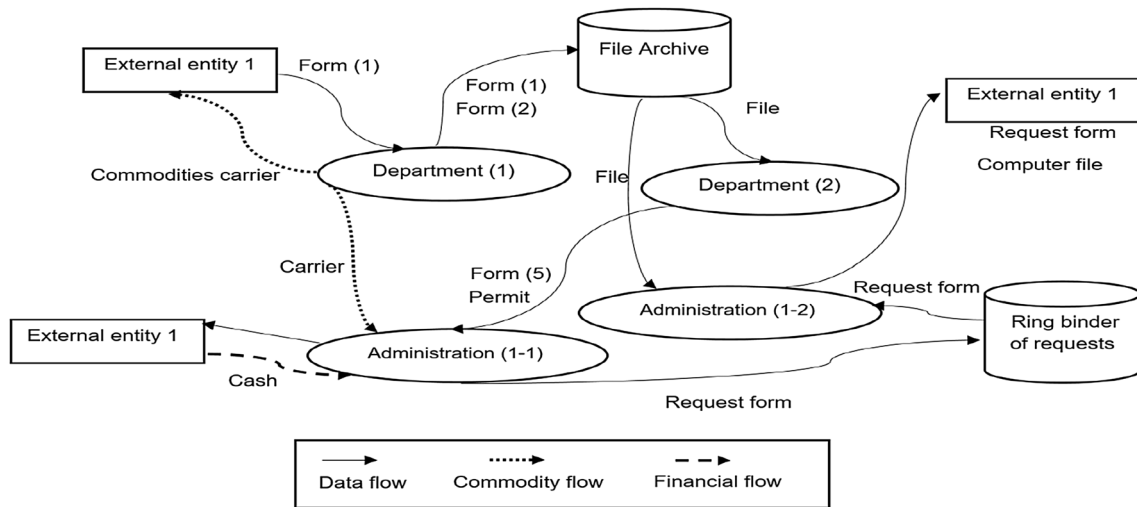
Motif discovery algorithms are also classified into two categories by subgraph searching mechanism (Kavurucu 2015; Wong and Baur 2010):

- (a) Network-based algorithms: These algorithms search all subgraphs of a given size in a network and estimate the total frequency of the subgraphs.
- (b) Motif-based algorithms: These algorithms only search a given subgraph in the network and calculate the frequency of only that subgraph in the entire main network.

Table 2 lists all motif discovery algorithms devised so far. This table categorizes the motif discovery algorithms.

**Table 2** Classification of motif discovery algorithms

	Motif-based algorithms	Network-based algorithms
Tree-based algorithms	Grochow and Kellis (2007)	MA Visto (Schreiber et al. 2005) NeMo Finder (Chen et al. 2006) Kavosh (Kashani et al. 2009) G-tire (Ribeiro and Silva 2010) QuateXelero (Khakabimamaghani et al. 2013)
Probabilistic algorithms	MODA (Omidi et al. 2009)	MFinder (Kashtan et al. 2004) ESU (Wernicke and Rasche 2006) RAND-ESU (Wernicke 2005)



**Fig. 1** Physical generalized flow diagram on the process level

In this research, RAND-ESU algorithm introduced by Wernicke (2005) is used for motif discovery. This algorithm is selected because firstly it is a probabilistic algorithm and has low computational complexity. Secondly, it is a network-based algorithm, which searches all subgraphs of a certain size. It should be noted that the logical structures of the relationships among the activities are manifested as different subgraphs in the network. Hence, motif-based algorithms do not suit this research because these algorithms only search a certain subgraph. Thirdly, the sampling approach used in this algorithm is almost fair as compared to the other probabilistic algorithms (Wernicke 2005). This algorithm has been implemented by FANMOD,<sup>2</sup> which is also used in this research.

### 3.2 Physical generalized flow model

The physical generalized flow model is a diagram that identifies the organizational flows among the organizational

entities (Whitten et al. 2007). Here, organizational entities represent the organizational roles. Moreover, organizational flows are grouped into three categories: (1) data flows; (2) material and service flows, and (3) financial flows (Whitten et al. 2007). These flows are identified by the physical generalized flow diagram (PGFD). The physical generalized flow diagram is defined on three levels: (1) context level, (2) system level, and (3) process level (Whitten et al. 2007). In this paper, the physical generalized flow diagram on the process level is used. Figure 1 depicts a physical generalized flow diagram (PGFD) on the process level. In this figure, the flows among organizational roles are identified.

In this research, only the information on the data flows is used, and the financial flows and commodity flows are neglected. The PGFD model information is recorded as a table in the information system. Table 3 shows a sample PGFD table in the information system. This table shows from which role what kind of flow is transferred to which role. It also identifies the flow content and the flow form.

<sup>2</sup> Fast Network Motif Detection tool.



**Table 3** The PGFD table on the process level in an information system

From role	To role	Flow type	Flow form	Flow content
Department (2)	Administration (1–1)	Data and information	Electronic form	Usage permit
Administration (1–1)	Ring binder of requests	Data and information	Electronic form	Balance check request
File archive	Administration (1–2)	Data and information	File	Customer information

**Table 4** Data set-related hypotheses

Activity-related hypotheses	The processes are structured: the sequence of accruing activities remains invariant with different cases and in different traces The set of activities are variant in different traces The activity range remains unchanged in all traces
Role-related hypotheses	The roles are predefined and predetermined The activities performed by each role are predetermined and predefined and remain unchanged during traces. This information is stored in the role-activity matrix Every role can perform several activities, but each activity must be done only by one role
Event log-related hypotheses	The event log only contains information of one process Every event in an event log consists of the following information: (1) the case fed into the process, (2) the performed activity, (3) the performer of the activity, (4) the activity start time, and (5) the activity complete time
PGFD-related hypotheses	The data flows among different roles in different traces remain unchanged

### 4 Problem statement and modeling

The present research approach is based on a series of fundamental hypotheses about data sets. Table 4 presents a summary of these hypotheses.

In this section, the research problem is modeled based on the above hypotheses. In this research, the problem inputs include the event logs, the process-level PGFD model, and the role-activity matrix. The event log is defined as an ordered tuple of sets as follows:

$$EL = (CN, St, Ct, A, RS). \tag{1}$$

In Eq. (1),  $CN$  denotes the input cases set,  $St$  is the activity start time,  $Ct$  represents the activity completion time,  $A$  shows the process activities set, and  $RS$  is the process resources set. Each of the aforementioned sets is defined as follows:

$$CN = \{c_1, c_2, \dots, c_n\}. \tag{2}$$

$$A = \{a_1, a_2, \dots, a_n\}. \tag{3}$$

$$RS = \{rs_1, rs_2, \dots, rs_n\}. \tag{4}$$

In Eq. (2)  $c_i$  shows the input case number, in Eq. (3)  $a_i$  is the performed activity, and in Eq. (4)  $rs_i$  represents the performer of the activity. The other input is the process-level PGFD model; this input is also defined as an ordered tuple of sets as follows:

$$PGFD = (FR_i, FA_i, TR_i, TA_i, T). \tag{5}$$

In Eq. (5),  $FR_i$  is the set of roles sending the flows,  $FA_i$  is the set of activities performed by the flow-sending roles,  $TR_i$  shows the set of roles receiving the flows,  $TA_i$  denotes the set of activities performed by the flow-receiving roles, and finally,  $T$  denotes the flows type. Each of the above sets and the roles set are also defined as follows:

$$R = \{r_1, r_2, \dots, r_n\}. \tag{6}$$

$$FA_i \subseteq A. \tag{7}$$

$$TA_i \subseteq A. \tag{8}$$

$$TR_i = \{TR_i \subseteq R, a_i \in TA_i\}. \tag{9}$$

$$FR_i = \{FR_i \subseteq R, a_i \in FA_i\}. \tag{10}$$

$$T = \{data, financial, item\}. \tag{11}$$

In Eq. (6),  $r_i$  shows an organization role, and in Eq. (11) ‘data’ refers to the data flow type, ‘financial’ represents the financial flow type, and ‘item’ shows the material and service flow type. Finally, the third input is the role-activity matrix stored in the information system, which is defined as follows:

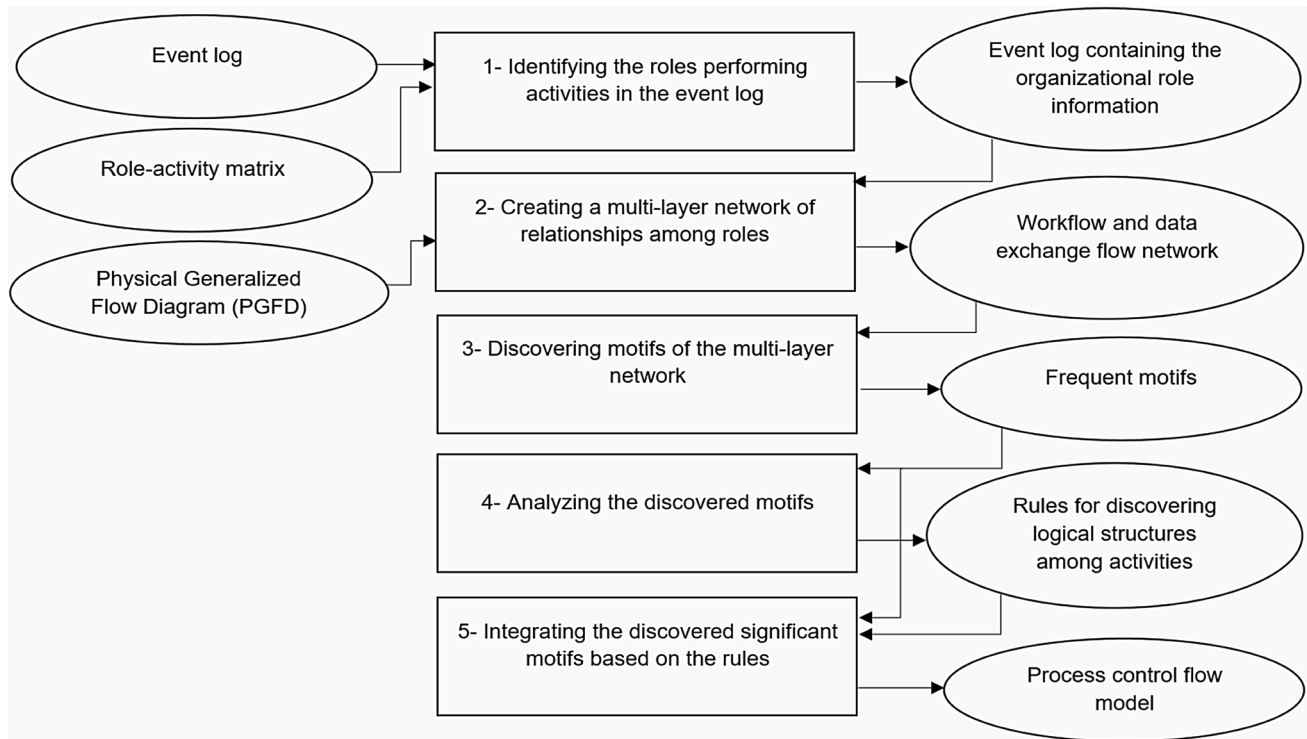


Fig. 2 Five steps of proposed solution to the process discovery problem

$$RA = [x_{a,r_j}], \quad x_{a,r_j} \in \{0, 1\}, \quad a_i \in A, \quad r_j \in R. \quad (12)$$

In Eq. (12),  $RA$  is the role-activity matrix, wherein the columns show the roles and the rows show the activities. The entries in this matrix are either 0 or 1. If entry  $x_{a,r_j}$  equals 1, activity  $a_i$  is performed by role  $r_j$ . As mentioned, in the proposed approach the input information is transformed into a bilayer network under a suitable scenario to represent the workflow and data exchange. This network is defined as the following ternary:

$$N = (V_{i,j}, E, FT). \quad (13)$$

$$V_{i,j} = \{v_{i,j} | i \in R, j \in A\}. \quad (14)$$

$$E \subseteq V_{i,j} \times V_{i,j}. \quad (15)$$

$$FT = \{data, handover\}. \quad (16)$$

In Eq. (14)  $V_{i,j}$  is the network nodes set, in Eq. (15)  $E$  is the set of edges, and in Eq. (16)  $FT$  reflects the edges type. The edge type may be a dataflow which is denoted by ‘data’ or a workflow which is represented by ‘handover.’ After creating the bilayer network of the workflow and data exchange, the network motifs are extracted and motifs

with certain features are mapped to a logical structure of the activity relationships in accordance with certain rules. Finally, the discovered logical relationships are integrated, and the resulting model is mapped to a Petri network. Each step of the solution is described in detail hereunder.

## 5 Research methodology

In this section, first the proposed method of this research will be described, and then, the assessment strategy will be introduced.

### 5.1 Proposed method

The methodology of this research consisted of five steps. These five steps are illustrated in Fig. 2.

#### 5.1.1 Identifying the roles performing activities in the event log

In this research, a standard format is considered for event logs. In this format, the event log initially contains no information on the organizational roles performing the activities. Therefore, this information needs to be added to the event log to determine which role performs each activity. In order to add this information, the role-activity matrix stored in the



**Table 5** Role-activity matrix

	Sale manager	Transportation manager	Financial manager
Customer relationship	✓		
Contract preparation			✓
Shipment		✓	
Order checking	✓		

information system is used. An example of this matrix is presented in Table 5. In this table, the rows show the activities and the columns represent the roles. As seen in Table 5, a role can be responsible for several activities. For instance, the ‘customer relationship’ and ‘order checking’ activities are both performed by ‘sale manager.’ However, the opposite is not true. In other words, each activity is assigned to only one role. The role information can be added to the event log based on these rules and the role-activity matrix.

When roles information is added to the event log, the new event log is defined as an ordered tuple of sets as follows:

$$EL' = (CN, St, Ct, A, RS, R). \tag{17}$$

In Eq. (17) like Eq. (1),  $CN$  denotes the input cases set,  $St$  is the activity start time,  $Ct$  represents the activity completion time,  $A$  shows the process activities set,  $RS$  is the process resources set, and  $R$  represents the role set which is defined in Eq. (6).

### 5.1.2 Creating a multilayer network based on inter-role relationships, from event logs and PGFD

In this step, a bilayer social network is developed to represent the workflow and data exchange flow in the process. In this network, the information in the new event log, which includes information on roles now, and the PGFD-driven information on the exchange of data among the roles, is used. The network modeling scenario is also as follows:

- **Nodes** These are the roles responsible for performing different activities in the network. It is worth mentioning that some roles can handle multiple activities in the process. Hence, at the time of defining the nodes, the activity titles are used along with the role titles to be able to determine the activity performed by a given role in the next steps. According to Eq. (14),  $v_{i,j}$  is a network node that  $i$  refers to a role and  $j$  refers to the activity that is done by role  $i$ .
- **First-layer edges** In the first layer, the edges among nodes are formed based on ‘handover.’ In the handover

scenario, if two following activities exist with a causal relationship for a case and the first activity is performed by role  $i$  and the second activity is performed by role  $j$ , a handover from  $i$  to  $j$  takes place. The formation of the network under the handover scenario is completely described by Van der Aalst and Song (2004). Based on Eqs. (13) and (16), the amount of  $FT$  for the first layer of edges is ‘handover.’ According to Eq. (17), if  $e_i \in EL'$  is an event, two operations are defined on  $EL'$ :

$$\pi_r(e_i) = r_i \in R. \tag{18}$$

$$\pi_a(e_i) = a_i \in A. \tag{19}$$

If  $t_c$  is a process trace belonging to case  $c \in CN$  and  $\rightarrow$  denotes a causal relationship:

$$\begin{aligned} v_{a_i,r_i} \supseteq v_{a_j,r_j} \exists 0 \leq n < |t_c| \pi_r(e_i) &= r_i \wedge \pi_a(e_i) \\ &= a_i \wedge \pi_r(e_{i+n}) = r_j \wedge \pi_a(e_{i+n}) = a_j \wedge e_i e_{i+n} \end{aligned} \tag{20}$$

In Eq. (20),  $v_{a_i,r_i} \supseteq v_{a_j,r_j}$  denotes a handover flow, which means in process trace  $t_c$  belongs to case  $c$ , in event  $e_i$  roles  $r_i$  executes activity  $a_i$  on case  $c$ , and then in event  $e_{i+n}$  role  $r_j$  executes activity  $a_j$  on case  $c$ .

- **Second-layer edges** An edge is formed between two roles responsible for different activities, if, in the PGFD model, a data flow is exchanged between those two roles. Based on Eqs. (13) and (16),  $FT$  for the second layer of edges is ‘data.’ Based on Eq. (5), if  $e_i \in PGFD$ , five operations are defined on  $PGFD$ :

$$\pi_{fa}(e_i) = a \in FA_i. \tag{21}$$

$$\pi_{fr}(e_i) = a \in FR_i. \tag{22}$$

$$\pi_{ta}(e_i) = a' \in TA_i. \tag{23}$$

$$\pi_{tr}(e_i) = r' \in TR_i. \tag{24}$$

$$\pi_t(e_i) = t \in T. \tag{25}$$

Then, a data flow between nodes  $v_{a_i,r_i}$  and  $v_{a'_i,r'_i}$  is defined as:

$$\begin{aligned} v_{a_i,r_i} \triangleright v_{a'_i,r'_i} \exists 0 \leq i \leq |PGFD| \exists e_i \in PGFD \pi_{fa}(e_i) \\ &= a_i \wedge \pi_{fr}(e_i) = r_i \wedge \pi_{ta}(e_i) \\ &= a'_i \wedge \pi_{tr}(e_i) = r'_i \wedge \pi_t(e_i) = \{\text{data}\}. \end{aligned} \tag{26}$$

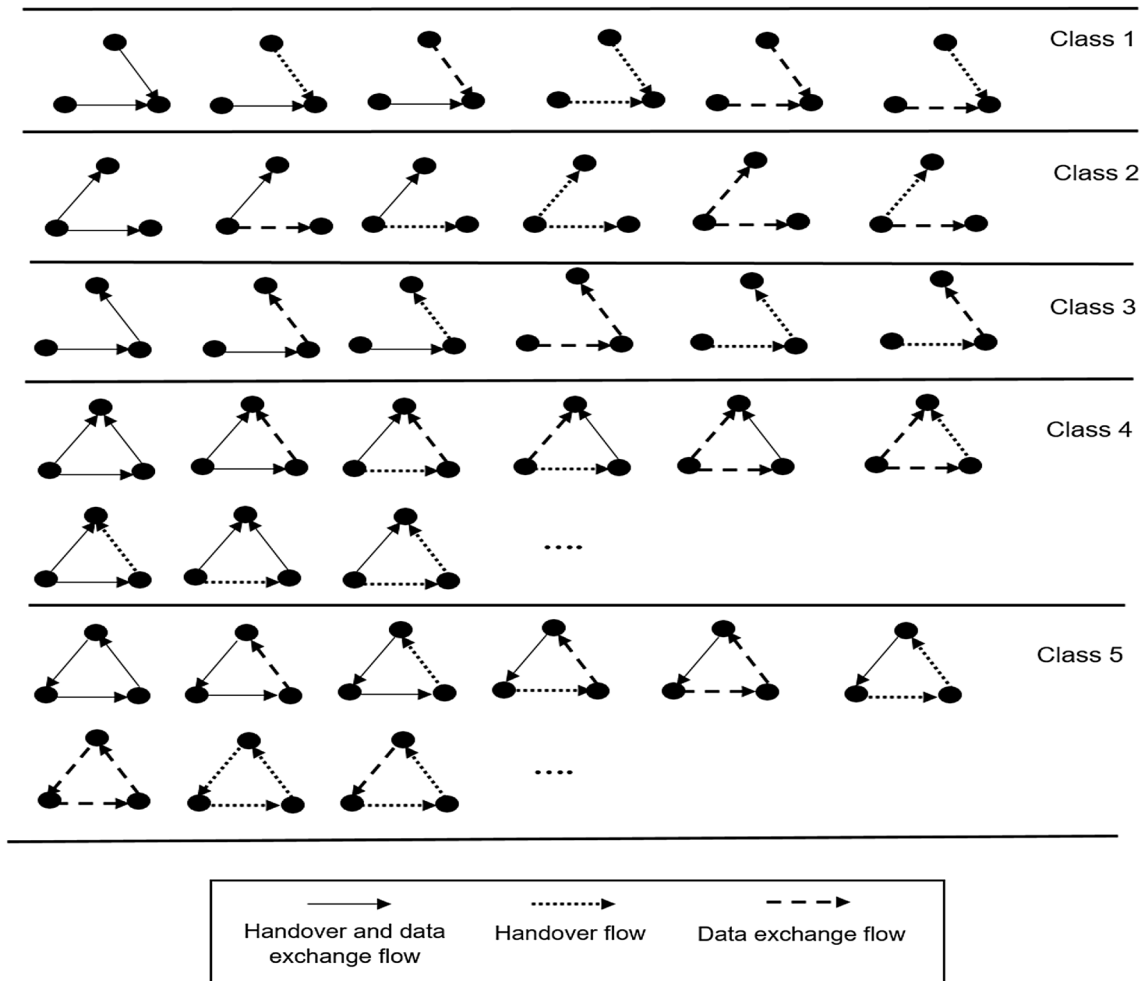


Fig. 3 Classification of size three subgraphs existing in the bilayer network

In Eq. (26),  $v_{a_i, r_i} \triangleright v_{a'_j, r'_j}$  denotes a data flow which means role  $r_i$  who executes activity  $a_i$  sends some data to role  $r'_j$  that is going to execute activity  $a'_j$ .

The network described above is a directed and weighted network.

### 5.1.3 Discovering motifs in the multilayer network

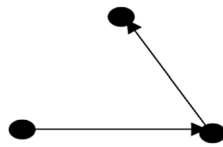
In this phase, ternary motifs in the bilayer workflow and data exchange flow network are extracted. The reason for selecting ternary motifs is that with a decrease in motifs size, their detection is accelerated and size three motifs are the smallest network motifs that can reveal logical structures in process models. These motifs provide adequate information for the identification of logical structures through activity relationships without a need for examining larger motifs. To extract the motifs, the Rand-ESU algorithm implemented in FANMOD is used in this research. The computational complexity

of Rand-ESU is  $O(n^K)$  that  $n$  is the number of nodes, and  $k$  is the size of subgraphs (Wernicke 2005). Thus, for ternary motifs, its computational complexity is equal to  $O(n^3)$ .

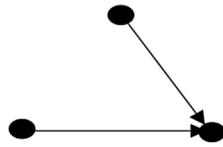
One of the advantages of FANMOD is the possibility of discerning edges and nodes according to their type. According to Eqs. (16) and (13),  $FT$  set shows that there are two types of flow in the bilayer network of  $N$ : ‘data’ and ‘handover.’ Therefore, these flows divide the network edges into three categories: (1) edges only containing the data exchange flow; (2) edges only containing the handover flow, and (3) edges containing both the data exchange flow and the handover flow. These three types are discerned in FANMOD by assigning specified codes to edges. Figure 3 presents some samples of size three subgraphs resulting from FANMOD. In this figure, three groups of edges are observable.

After extracting the subgraphs, the identified subgraphs are filtered by the  $z$ -score and  $p$  value values to identify the significant motifs as follows:

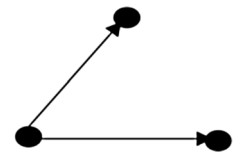
**Fig. 4** The motif representing a sequence among the activities



**Fig. 5** The motif representing a joint node



**Fig. 6** The motif representing a split node



data exchange flow is overlooked, and the edges only contain the handover flow.

$$|z - \text{score}| > 1.65. \tag{27}$$

$$|p - \text{value}| \leq 0.05. \tag{28}$$

The reason for selecting these threshold limit values (TLVs) in Eqs. (27) and (28) is that subgraphs with  $z$ -score values higher than 1.65 and  $p$  values lower than 0.05 are most probably significant motifs and the likelihood of randomness of these motifs is below 5% (Bjorn and Falk 2008).

### 5.1.4 Classification of motifs for identifying logical structures in the process

In this step, considering the resulting structure of the ternary motifs, rules are set to map motifs with specific structural features to legal logical structures in process models. These rules are as follows:

- (i) *Rule no. 1* If the nodes in the ternary motif are connected consecutively by two edges containing handover flows so that the input and output degrees of none of the nodes are higher than one, the activities corresponding to these nodes are also consecutive in the process logical structure. Figure 4 depicts the motif structure that applies to rule no. 1. In this figure, the data exchange flow is neglected, and the edges only contain the handover flow.
- (ii) *Rule no. 2* In a size three motif, if two edges containing handover flows that enter a node, the activity corresponding to that node in the process Petri net, is a joint node. The structure of the motif, to which rule no. 2 applies, is illustrated in Fig. 5. In this figure, the data exchange flow is overlooked, and the edges only contain the handover flow.
- (iii) *Rule no. 3* In a size three motif, if two edges containing handover flows that leave a node, the activity corresponding to that node in the process Petri net, is a split node. The structure of the motif, explained by rule no. 3, is depicted in Fig. 6. In this figure, the

The logical structures of the process model can be mostly figured out based on the above three rules. The question is that since all of the aforesaid rules are based on the handover flows, what information can be derived from the data exchange flows? In response to this question, it should be said that although the second and third rules identify the joint and split structures, these rules do not determine the types of these structures. Here, the data exchange flows are used to determine whether a joint/split structure is an ‘AND’ or ‘XOR’ structure. Hence, the two other rules are set as follows.

- (iv) *Rule no. 4* If data exchange takes place among the activities in a joint/split structure, those activities are parallel and the joint/split nodes are ‘AND’ nodes. The philosophy behind this rule is that activities that are parallel are somehow connected. For instance, all belong to the same organizational unit, and thus, there are most probably data exchange flows among the roles performing these activities.
- (v) *Rule no. 5* If there is no data exchange among the activities in a joint/split structure, those activities can be in the choice state with one another and the joint/split nodes are also ‘XOR’ nodes. The philosophy behind this rule is that the choice structures divide the process model into several branches. As a result, the activities that are in the choice state, in relation to each other, are independent of one another. Hence, there is no relationship among their roles including the data exchange flow.

### 5.1.5 Discovering the process control flow model by integrating the resulting motifs

In this step, which is the last step of the proposed approach, the process control flow structure is obtained based on the rules defined in the previous step. In this step, first the nodes hosting each motif should be identified. In this research, one of the features of FANMOD is used for this purpose. In FANMOD, nodes are discernable as well as edges. This feature allows for the identification of the nodes related to a given motif. Through assignment of a code to each node, FANMOD differentiates among the isomorphic motifs on

different nodes; hence, it will be possible to identify the activities and roles related to each node. Generally speaking, the motif discovery algorithms function based on the following three strategies to calculate the importance of the subgraphs (Wong and Baur 2010):

- (i) F1: This strategy allows for the overlap of nodes and edges.
- (ii) F2: In this strategy, only nodes are allowed to overlap and edges must be entirely separated.
- (iii) F3: In this strategy, neither nodes nor edges are allowed to overlap.

FANMOD uses the F1 strategy. Hence, motifs may share nodes. Next, the separated motifs should be joined according to their shared nodes. Afterward, the logical relationships among the activities corresponding to each node are identified based on rules no. 1 to 3. Finally, the type of joint and split nodes are determined by rule nos. 4 and 5, resulting in the process control flow structure.

## 5.2 Evaluation strategy

To evaluate the proposed approach, first, the effect of factors changing the process behavior on the proposed method will be studied. Next, the efficiency of this method will be compared to the previous methods. To conduct this comparison, the alpha, alpha plus, and heuristic mining algorithms are selected from the group of prior methods. The mechanism of each algorithm is described in the following.

- *Alpha algorithm* This algorithm, which is explained by Van der Aalst and Song (2004), is a deterministic algorithm. In sum, three kinds of dependency relationships, viz. succession, parallel, and choice, are assumed depending on the activity performing time in event logs. Afterward, each relation is mapped to a Petri net.
- *Alpha plus algorithm* This algorithm is the extended version of the alpha algorithm, except that it not only takes the explicit dependencies into account, but also defines a set of implicit relationships among the activities based on several incidence theories (Wen et al. 2009).
- *Heuristic mining algorithm* This algorithm functions similar to the alpha algorithm except that it not only takes the dependencies into account but also values the frequency of relationships to detect noise (Weijters and van der Aalst 2003).

In this research, since the primary process models are simulated and available, the efficiency of the proposed method is measured using the  $F$ -measure, which is the mean weight of Precision and Recall and is obtained via Eq. (29). Precision and Recall are calculated by Eqs. (30) and (31),

respectively. Here, Precision refers to the ratio of the correctly detected relations to all relations found by the solution. On the other hand, Recall refers to the ratio of relationships identified successfully to total actual relationships in the process model. Therefore, the proposed method is evaluated from the viewpoint of business process management. In other words, conformance checking between the discovered process model and the initial process model is the subject.

$$f = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}. \quad (29)$$

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}. \quad (30)$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}. \quad (31)$$

In Eqs. (30) and (31), TP is the true-positive rate and shows the number of relationships among the process activities detected correctly by the solution. FP is the false-positive rate and indicates the relationships that do not actually exist among the process activities but are wrongly identified by the solution. FN is also the false-negative rate which refers to the number of relationships that exist in the primary process model but are not detected by the solution.

## 6 Results and discussion

To assess the proposed method, a set of experiments are designed. In this respect, first the workflow matrices are created for three different processes. A workflow matrix identifies what activities are performed by what roles or resources and it also identifies the order in which activities are performed. Figure 7 depicts part of a workflow matrix belonging to the first process. This process belongs to a company that provides different internet services such as the ADSL service.

In Fig. 7, the rows are activities and the columns are organizational roles. In this matrix, the order of performing activities is also identified. According to a workflow matrix, a process model can be generated. The process models are simulated in WoPeD,<sup>3</sup> which is an open-source software developed to provide modeling, simulating, and analyzing processes described by workflow networks. Event logs can be generated by simulating processes in WoPeD. Figure 8 illustrates the first process model generated based on the workflow matrix depicted in Fig. 7.

<sup>3</sup> Workflow Petri net Designer.

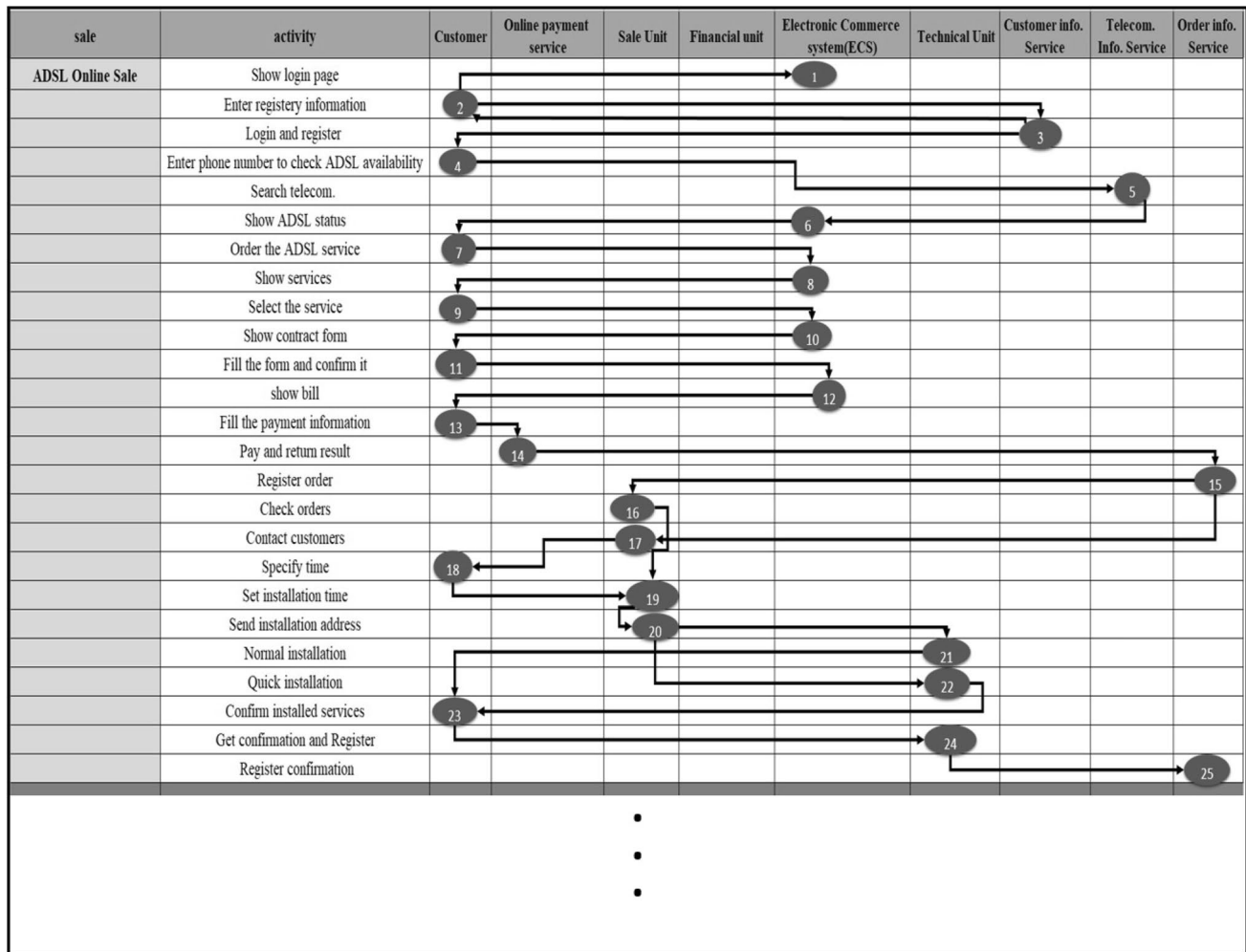


Fig. 7 Part of workflow matrix for the first process

In the next step, the PGFD models of all three processes are made concerning the related workflow matrices. Table 6 illustrates the first process’s PGFD model that is created based on the workflow matrix shown in Fig. 7.

Table 6 depicts different flows exchanging among organizational roles who are responsible for certain activities. In this research, only data flows are used. After that, eight event logs are generated per process by changing the parameters determining the process behavior and simulating the processes in WoPeD. Hence, a total of 24 event logs (8 × 3) are created. According to previous researches, the behavior of a process is affected by three factors (Weijters and van der Aalst 2003):

- The number of input cases within a given period of time: In this research, the number of input cases varies from 100 cases to 400 cases within a 30-day period.
- Noise level: There are five ways of producing noise in an event log: (1) Missing head, (2) Missing body, (3) Miss-

ing tail, (4) Missing event, and (5) Exchange event. In this study, noise was produced using all of the mentioned methods in the range between 0 and 40%.

- The probability of activities being fired in the process: For example, if activities *a* and *b* are in a choice state with each other, probabilities *p* and  $1 - p$  lead to the selection of activities *a* and *b* in each iteration, respectively.

Finally, the proposed method is applied to each experiment set which is the combination of an event log and the PGF model of each process. Figure 9 shows the integrated motifs for one of the experiment sets of the first process. These motifs are extracted from the bilayer network of data flow and handover flow and integrated based on rule nos. 1, 2 and 3 that determine sequential, split, and joint patterns among activities.

In Fig. 9, based on rule no. 2 and rule no. 3 there are split structures among activities { ‘Register order’, ‘Contact



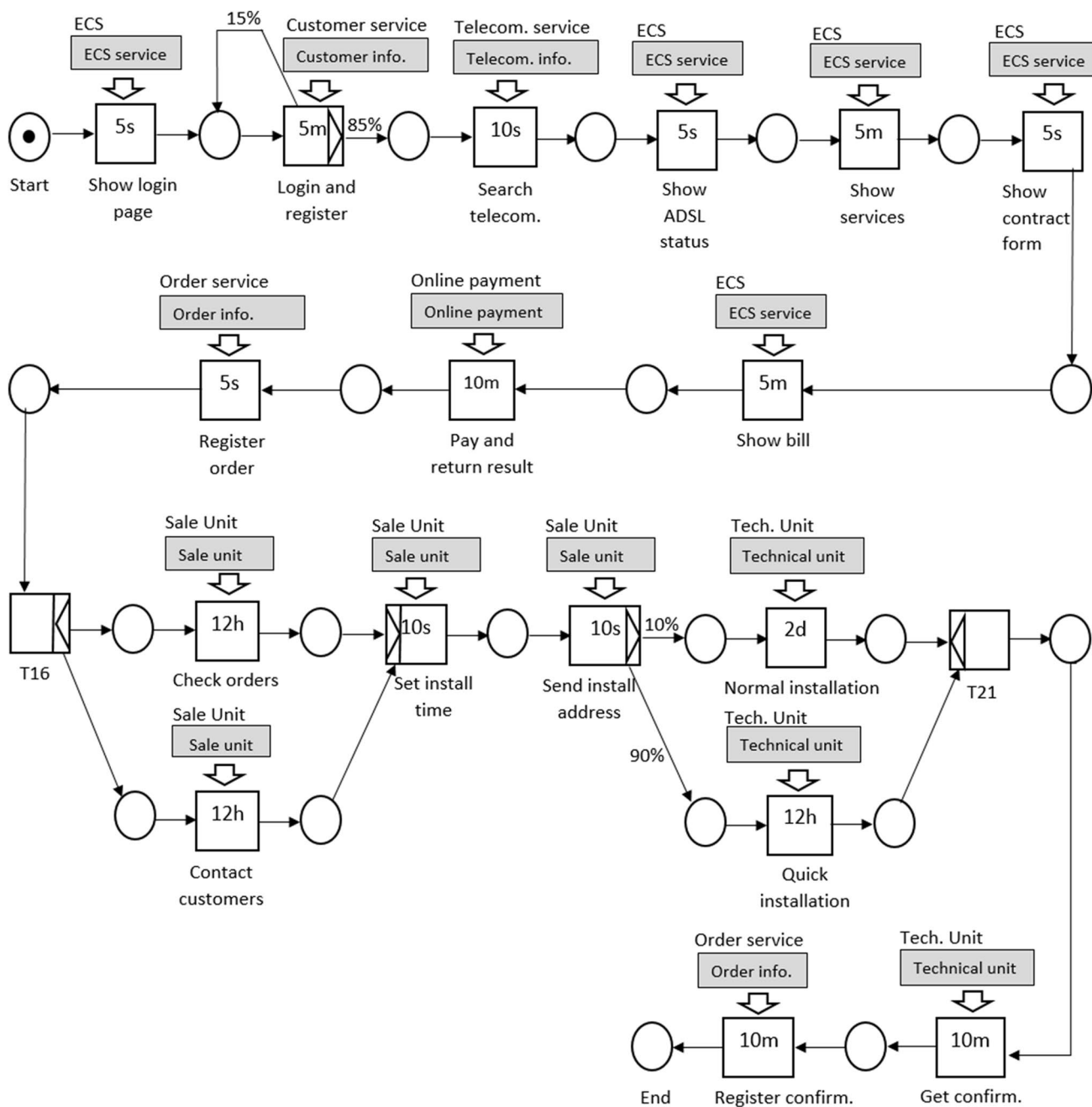


Fig. 8 First process which is modeled and simulated in WoPeD

customers', 'Check orders'} and {'send install address', 'Quick installation', 'Normal installation'}, and there are joint structures among activities {'Register order', 'Contact customers', 'Set install time'} and {'Quick installation', 'Normal installation', 'Get confirm'}. According to rule no. 4 and rule no. 5, the type of split/joint structures among {'Register order', 'Contact customers', 'Check orders'} and {'Register order', 'Contact customers', 'Set install time'} are 'AND,' since there is a data flow between 'Contact

customers' and 'Check orders', and the type of split/joint structures among {'send install address', 'Quick installation', 'Normal installation'} and {'Quick installation', 'Normal installation', 'Get confirm and Register'} are 'XOR,' because there is no data flow between 'Quick installation' and 'Normal installation.' After exerting the mentioned rules and mapping the motifs to Petri net patterns, the control flow of the process is derived. Figure 10 depicts the control flow of the first process created by the proposed method.



**Table 6** PGDF table of the first process

Trail role	From activity	To role	To activity	Flow type	Flow form	Flow content
Customer	Enter registry information	ECS service	Show login page	Data	e_form	Login info.
ECS service	Show login page	Customer info. service	Login and register	Data	e_record	User info.
Customer	Enter phone number to check ADSL availability	Telecom. info. service	Search telecom.	Data	e_form	Phone number info.
Telecom. Info. Service	Search telecom.	ECS service	Show ADSL status	Data	e_record	Telecom. status info.
ECS service	Show ADSL status	Customer	Order ADSL service	Data	e_	Telecom. status info.
Customer	Order ADSL service	ECS service	Show ADSL status	Data	e_form	Request ADSL service
ECS service	Show ADSL status	ECS service	Show services	Data	e_record	Request info.
ECS service	Show services	Customer	Select the service	Data	e_form	Available services info.
Customer	Select the service	ECS service	Show services	Data	e_form	Chosen services info.
ECS service	Show services	ECS service	Show contract form	Data	e_record	Chosen services info.
ECS service	Show contract form	Customer	Fill the form and confirm it	Data	e_form	Contract info.
Customer	Fill the form and confirm it	ECS service	Show contract form	Data	e_form	Complete contract
ECS service	Show contract form	ECS service	Show bill	Data	e_record	Complete contract
ECS service	Show bill	Customer	Fill the payment information	Data	e_form	Bill info.
Customer	Fill the payment information	ECS service	Show bill	Data	e_form	Payment info.
ECS service	Show bill	online payment service	Pay and return result	Data	e_record	Payment info.
Online payment service	Pay and return result	Order info. service	Register order	Financial	e_record	Payment
Online payment service	Pay and return result	Order info. service	Register order	Data	e_record	Payment result
Order info, service	Register order	Sale unit	Check orders	Data	e_record	Order info.
Sale unit	Check orders	Sale unit	Contact customers	Data	e_form	Order info.
Customer info. Service	Login and register	Sale unit	Contact customers	Data	e_record	Contact info.
Sale unit	Contact customer	Customer	Specify time	Data	phone call	Request installation time and address
Customer	Specify time	Sale unit	Contact customers	Data	phone call	Installation time and address
Sale unit	Contact customer	Sale unit	Set installation time	Data	e_form	Installation time and address
Sale unit	Check orders	Sale unit	Set installation time	Data	e_form	order info.
Sale unit	Send installation address	Technical unit	Quick installation	Data	form	Installation time and address
Sale unit	Send installation address	Technical unit	Normal installation	Data	form	Installation time and address
Technical unit	Quick installation	Customer	Confirm installed services	Service	ADSL service	Installed ADSL service
Technical unit	Normal installation	Customer	Confirm installed services	Service	ADSL service	Installed ADSL service
Customer	Confirm installed services	Technical unit	Get confirm.	Data	paper	Confirmation
Technical unit	Get confirm.	Order info. Service	Register confirmation	Data	e_form	Confirmation info.

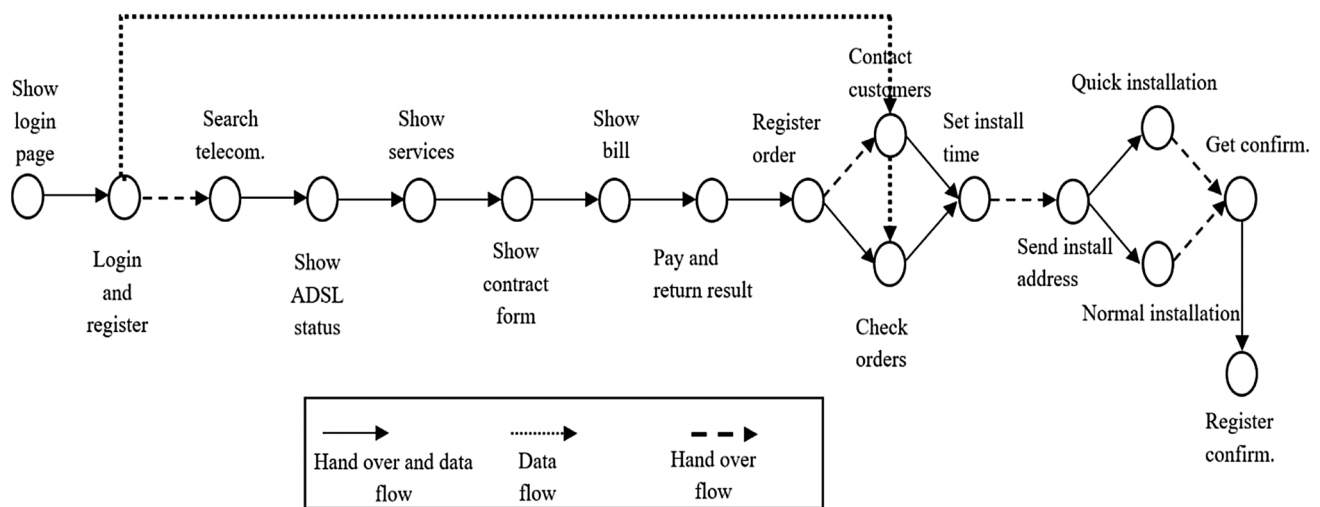


Fig. 9 Integrated motifs for the first process resulted from the proposed method for the

In the following, the characteristics of experiments and the results of implementing the proposed method on the experiment sets are explained.

### 6.1 Results

In Tables 7, 8, and 9, the specifications of the eight experiments carried out on the first, second, and third processes are listed, respectively. These tables show the number of input cases fed into the process in each experiment along with the likelihood of activities being fired and the noise percentage variations. For the probability of activities being fired, each row belongs to a choice structure. For instance, in Table 9, which belongs to the third process, three rows correspond to the probability of activities being fired. These three rows show the existence of three choice structures in the third process model, and each row shows the probability of the activities being fired in a choice structure.

Figures 11, 12, and 13 show the Precision, Recall, and *F*-measure values for each of the eight experiments on the first, second, and third processes, respectively. In these diagrams, the horizontal axis indicates the experiment number and the vertical axis shows the Precision, Recall, and *F*-measure values.

In the first four experiments, the noise for each process is set to zero (Tables 7, 8, and 9). However, the rate of cases fed into the process and probability of activities being fired in these four experiments are variant. In the second four experiments, a noise percentage is applied to the event logs. As seen in Figs. 11, 12, and 13, the results from the first four experiments on each process are the same. However, in the second four experiments, the efficiency of results partly declines with an increase in noise. Therefore, the number of cases fed into the process and the probability of activities

being fired change the trend in the event log but do not affect the results of the proposed method. The only factor influencing the performance of this method is noise.

After studying the determining factors, the number of cases fed into the process and the probability of activities being fired are ruled out in this step and only noise percentage is taken into account. Next, the performance of the proposed approach is compared to the performances of the alpha, heuristic mining, and alpha plus algorithms. These three algorithms are all implemented as plug-ins in ProM. Therefore, all event logs containing the noise generated per process are entered onto ProM software. Afterward, each of the three algorithms is applied to the inputs to measure the Precision, Recall, and *F*-measure of the results. In Figs. 14, 15, and 16, the ratio of the *F*-measure value to the noise percentage is presented per selected algorithm along with the results from the application of the proposed method to all three processes. In these diagrams, the horizontal axis shows the noise percentage and the vertical axis indicates the *F*-measure value.

### 6.2 Discussion

Figures 11, 12, and 13 suggest that in the first four experiments on each process, the results remain unchanged. In these experiments, noise is zero and only the input cases and the probability of activities being fired, change. In the next four experiments, noise is added to the event log, and the increase in noise undermines the results efficiency. Therefore, as seen in Figs. 11, 12, and 13, the number of input cases and the probability of activities being fired do not affect the results from the proposed method, and noise is the only determinant.

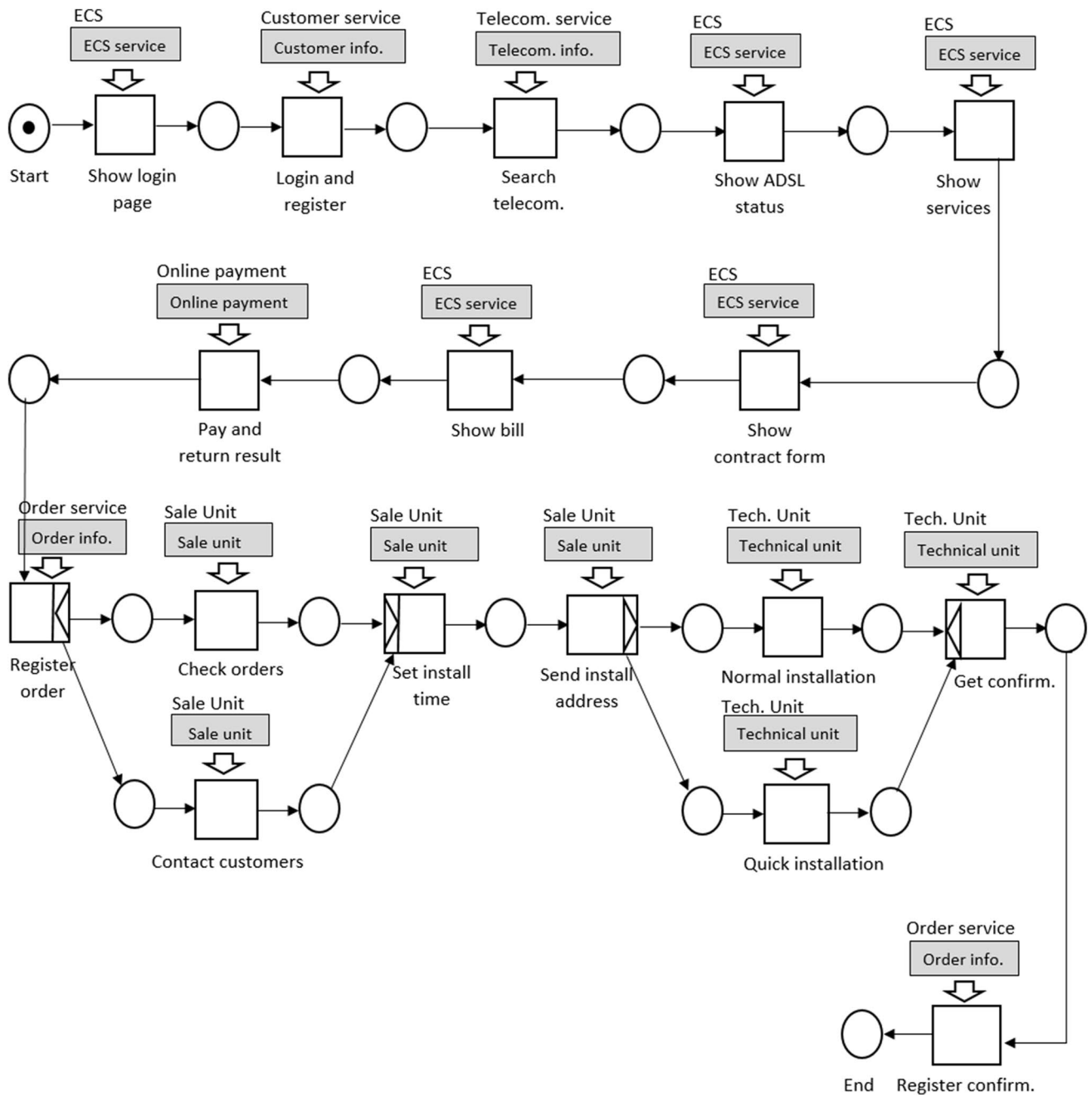


Fig. 10 Control flow of the first process resulted from the proposed method

Table 7 The specifications of the experiments carried out in the first process

	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8
Number of cases	100	200	300	400	100	200	300	400
Probability of the activities being fired	10–90%	15–85%	20–80%	30–70%	10–90%	15–85%	20–80%	30–70%
Noise	0	0	0	0	10%	20%	30%	40%

**Table 8** The specifications of the experiments carried out in the second process

	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8
Number of cases	100	200	300	400	100	200	300	400
Probability of the activities being fired	10–90%	15–85%	2080%	30–70%	10–90%	15–85%	20–80%	30–70%
Noise	0	0	0	0	10%	20%	30%	40%

**Table 9** The specifications of the experiments carried out in the third process

	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8
Number of cases	100	200	300	400	100	200	300	400
Probability of the activities being fired	10–90%	15–85%	20–80%	30–70%	10–90%	15–85%	20–80%	30–70%
	10–90%	15–85%	20–80%	30–70%	10–90%	15–85%	20–80%	30–70%
	5–95%	10–90%	10–90%	15–85%	5–95%	10–90%	10–90%	15–85%
Noise	0	0	0	0	10%	20%	30%	40%

Moreover, Figs. 11, 12, and 13 indicate that although Recall is high in all eight experiments, it is not equal to 1. In other words, the proposed method is not capable of identifying some structures. These structures include the hidden and duplicate tasks. The hidden tasks are those tasks that are only in charge of routing in the process model. No information on these tasks is stored in the information system. Therefore, the proposed method is not capable of identifying these tasks. However, it is possible to enable the proposed approach to determine the hidden tasks in the future using a set of incidence theories.

On the other hand, duplicate tasks are activities that are manifested more than once in the process model, but the proposed method is currently unable to differentiate among them. This pitfall can also be avoided in the future by initially relabeling the activities. To some up, as seen in Figs. 11, 12, and 13, the results of the proposed method are influenced by the logical structures in the primary process model and the noise recorded in the event log.

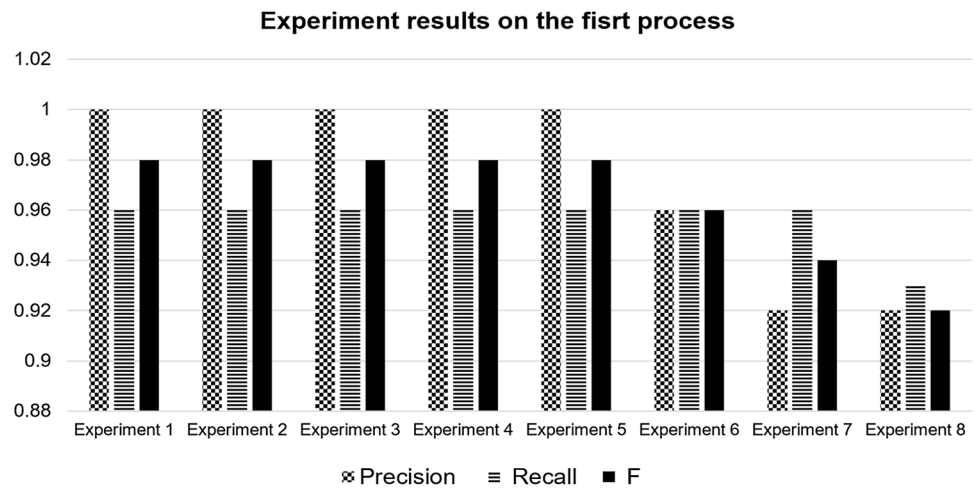
In Figs. 14, 15, and 16, the number of input cases and the probability of activities being fired are ruled out to only consider the effect of noise percentage. These figures also present the comparison of the results of the proposed method to the results of the alpha, alpha plus, and heuristic mining algorithms. As seen in Figs. 14, 15, and 16, despite the effect of noise recorded in the event log on the proposed approach, this method is more noise resistant than the previous methods. Besides, with an increase in noise percentage, the *F*-measure value remains high and does not decrease significantly. As reflected by Figs. 14, 15, and 16, among the three selected algorithms, only the alpha plus algorithm is initially highly accurate. However, with an increase in noise the efficiency of this method is severely decreased.

## 7 Conclusions and future work

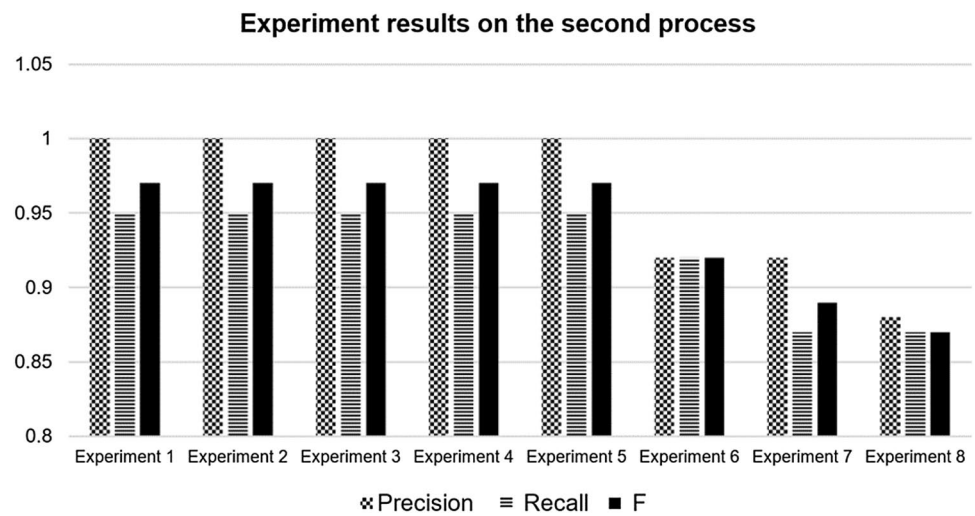
In this research, a new approach for process model discovery is introduced. This new approach identifies the process model based on the information in the event log and the information on the exchange of data among organization roles (which is derived from the PGFD model). In this approach, first a bilayer social network of the handover flow and data exchange flow among the roles in charge of different activities in the process is created using the event log information and PGFD model. Next, the process model consisting of the logical relationships among the process activities and their orders is obtained by discovering the network motifs. To assess the proposed approach, three different processes are designed and simulated and a set of experiments is carried out on the data sets related to each process. Finally, the efficiency of the resulting models is measured using the *F*-measure. The assessment results reflected the higher effectiveness of the proposed method as compared to the previous methods. Some of the advantages of the proposed method are its high resistance to noise and its ability to handle incompleteness. Moreover, since in addition to the information in the event log, the information on data exchange is also used to identify the logical relationships among activities, this approach is capable of detecting many complicated structures (such as the non-free choice structures); hence, it is more precise than the previous methods.

As indicated by the assessment results, the proposed approach is highly precise and highly resistant to noise. Therefore, the implementation of this approach within a standard framework (e.g., as a plug-in in ProM) can help managers to discover accurate organizational process models. This tool enables organizational managers to identify the actual current process models; thus, they will be able to make rapid changes.

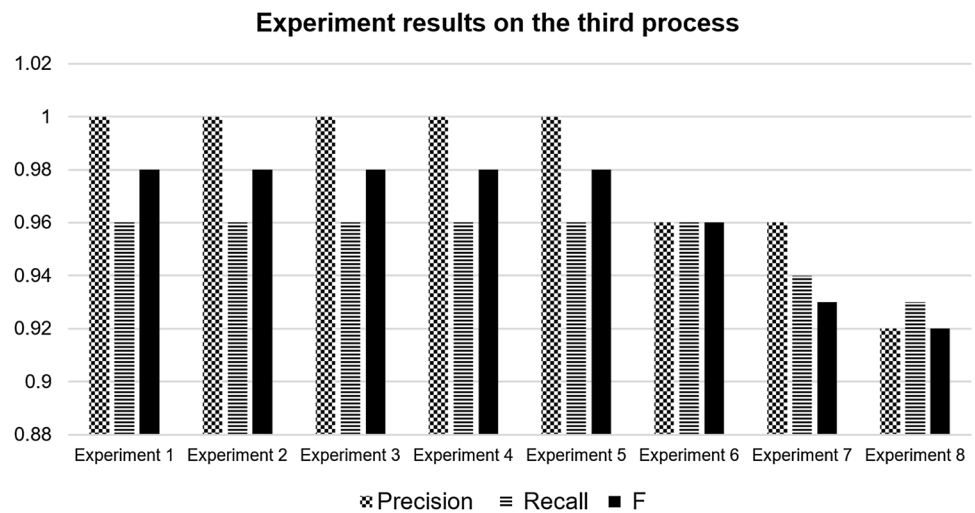
**Fig. 11** Results from the experiments in the first process



**Fig. 12** Results from the experiments in the second process



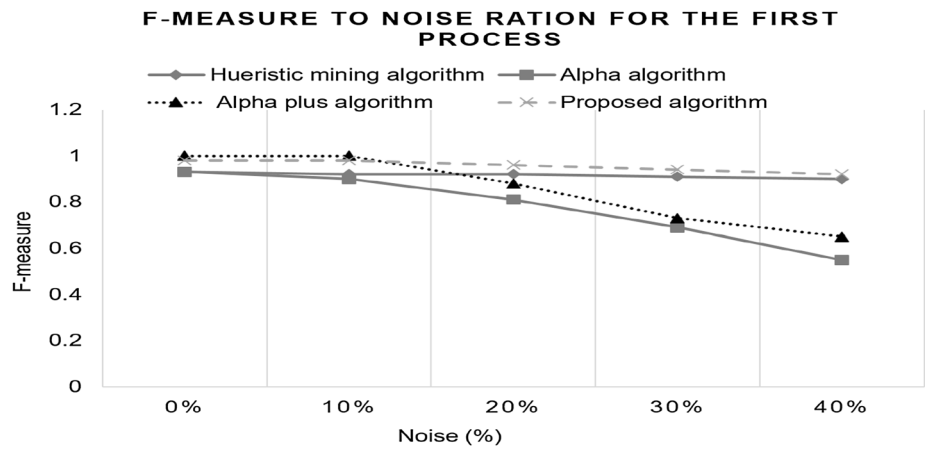
**Fig. 13** Results from the experiments in the third process



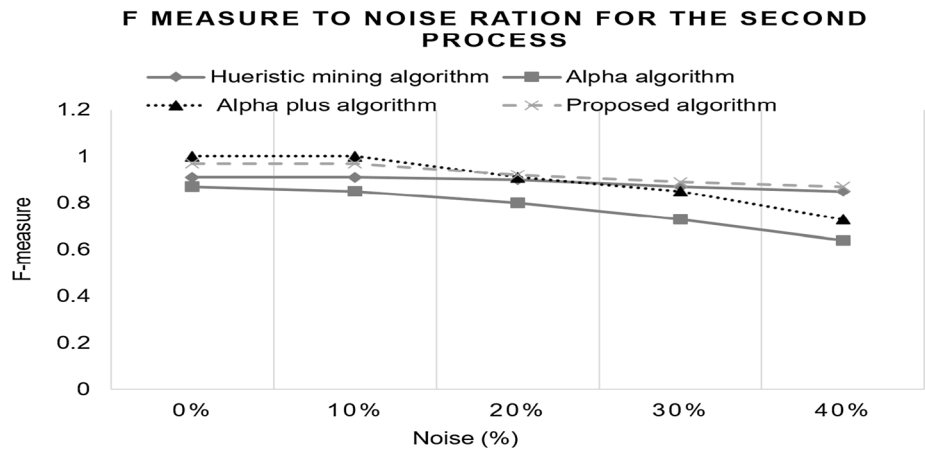
Despite the high efficiency and the resistance of the proposed method to noise, the following requirements must be addressed in the future researches:

- Adding simplifying mechanism to the proposed approach for preventing the development of the spaghetti models. It is possible through the community detection methods.

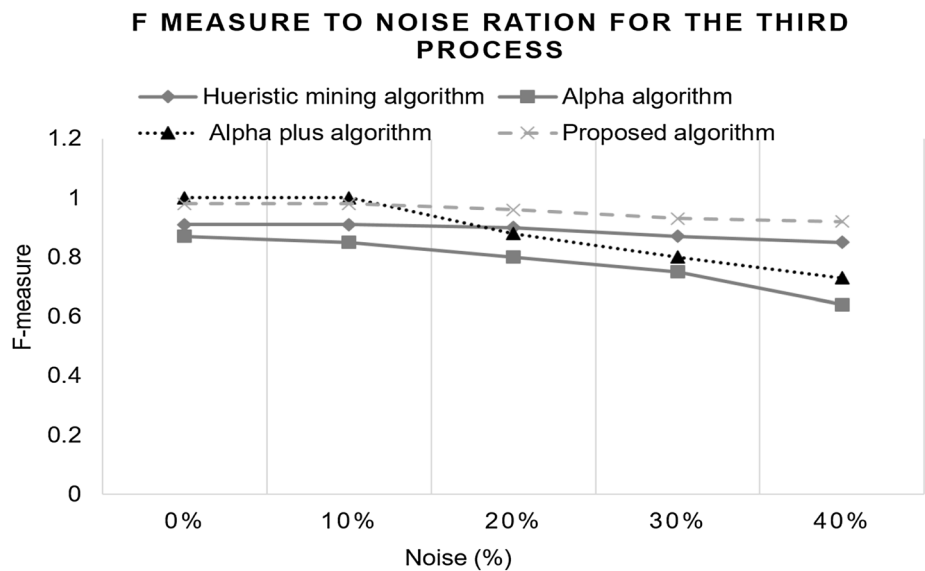
**Fig. 14** *F*-measure-to-noise (%) ratio in the first process



**Fig. 15** *F*-measure-to-noise (%) ratio in the second process



**Fig. 16** *F*-measure-to-noise (%) ratio in the third process



- Improving the recommended approach for enabling it to identify all complicated structures including duplicate and hidden tasks through defining incident theories and relabeling activities.
- Applying the proposed method to actual data sets.
- Evaluating the computational complexity of the proposed method against the other methods.



**Acknowledgements** We would like to thank all those whose assistance proved to be a milestone in the accomplishment of our end goal, especially members of Advanced Technologies in Electronic Commerce and Services Laboratory (ATECS Lab) at Amirkabir University of Technology.

## References

- Aleem S, Capretz LF, Ahmed F (2015) Business process mining approaches: a relative comparison. *Int J Sci Technol Manag* 4:1557–1564
- Arif T (2015) The mathematics of social network analysis: metrics for academic social networks. *Int J Comput Appl Technol Res* 4:889–893. <https://doi.org/10.7753/IJCATR0412.1003>
- Bjorn HJ, Falk S (2008) *Analysis of biological networks*. Wiley, New York
- Bose RPJC, Van der Aalst WMP (2009) Context aware clustering: towards improving process mining results. In: *Proceeding of the ninth SIAM international conference on data mining*. Sparks, Nevada, pp 401–412. <https://doi.org/10.1137/1.9781611972795.35>
- Bose RPJC, van der Aalst WMP (2010) Trace clustering based on conserved patterns: towards achieving better process models. In: Rinderle-Ma S, Sadiq S, Leymann F (eds) *Business process management workshops BPM 2009*, vol 43. *Lecture notes in business information processing*. Springer, Berlin, pp 170–181
- Burattin A (2015) *Process mining techniques in business environments, theoretical aspects, algorithms, techniques and open challenges in process mining*. Springer, Switzerland
- Chen J, Hsu W, Lee ML, Ng SK (2006) NeMoFinder: dissecting genome-wide protein–protein interactions with meso-scale network motifs. In: *Proceeding of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, New York, pp 106–115. <https://doi.org/10.1145/1150402.1150418>
- Dehghan BM, Golpayegani AH, Esmaeili L (2014) A novel C2C e-commerce recommender system based on link prediction: applying social network analysis. *Int J Adv Stud Comput Sci Eng* 3:1–8
- Esmaeili L, Nasiri M, Minaei-Bidgoli B (2011) Analyzing Persian social networks: an empirical study. *Int J Virtual Commun Soc Netw* 3:46–65. <https://doi.org/10.4018/jvcsn.2011070104>
- Grochow JA, Kellis M (2007) Network motif discovery using subgraph enumeration and symmetry-breaking. In: *Speed T, Huang H (eds) Research in computational molecular biology RECOMB 2007*, vol 4453. *Lecture notes in computer science*. Springer, Berlin, pp 99–106
- Herbst J (2000) A machine learning approach to workflow management. In: de Mántaras RL, Plaza E (eds) *Machine learning: ECML 2000*, vol 1810. *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*. Springer, Berlin, pp 183–194
- Herbst J, Karagiannis D (2002) Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In: *Proceedings of ninth international workshop on database and expert systems applications*, IEEE, Vienna, New Jersey, pp 745–793. <https://doi.org/10.1109/DEXA.1998.707491>
- Herbst J, Karagiannis D (2003) Workflow mining with InWoLVE. *J Comput Ind* 53:245–264. <https://doi.org/10.1016/j.compind.2003.10.002>
- Kashani Z, Ahrabian H, Elahi E, Nowzari-Dalini A, Ansari E, Asadi S, Mohammadi S, Schreiber F, Masoudi-Nejad A (2009) Kavosh: a new algorithm for finding network motifs. *BMC Bioinform* 10:1–12. <https://doi.org/10.1186/1471-2105-10-318>
- Kashtan N, Itzkovitz S, Milo R, Alon U (2004) Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics* 20:1746–1758. <https://doi.org/10.1093/bioinformatics/bth163>
- Kavurucu Y (2015) A comparative study on network motif discovery algorithms. *Int J Data Min Bioinform* 11:180–204. <https://doi.org/10.1504/IJDMB.2015.066777>
- Khakabimamaghani S, Sharafuddin I, Dichter N, Koch I, Masoudi-Nejad A (2013) QuateXelero: an accelerated exact network motif detection algorithm. *PLoS ONE* 8:68–73. <https://doi.org/10.1371/journal.pone.0068073>
- Medeiros AKA, Weijters AJMM, van der Aalst WMP (2006) Genetic process mining: a basic approach and its challenges. In: Bussler CJ, Haller A (eds) *Business process management workshops BPM 2005*, vol 3812. *Lecture notes in computer science*. Springer, Berlin, pp 203–215
- Omid S, Schreiber F, Masoudi-Nejad A (2009) MODA: an efficient algorithm for network motif discovery in biological networks. *Genes Genet Syst* 84:385–395. <https://doi.org/10.1266/ggs.84.385>
- Ribeiro P, Silva F (2010) G-Tries: an efficient data structure for discovering network motifs. In: *Proceeding of the 25th ACM symposium on applied computing*. ACM, New York, pp 1559–1566. <https://doi.org/10.1145/1774088.1774422>
- Schimm G (2003) Mining exact models of concurrent workflows. *J Comput Ind* 53:265–281. <https://doi.org/10.1016/j.compind.2003.10.003>
- Schreiber F, Schwobbermeyer H (2005) MAVisto: a tool for the exploration of network motifs. *Bioinformatics* 21:3572–3574. <https://doi.org/10.1093/bioinformatics/bti556>
- Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. *Decis Support Syst* 46(1):300–317. <https://doi.org/10.1016/j.dss.2008.07.002>
- Song M, Günther CW, van der Aalst WMP (2009) Trace clustering in process mining. In: Ardagna D, Mecella M, Yang J (eds) *Business process management workshops BPM 2008*, vol 17. *Lecture notes in business information processing*. Springer, Berlin, pp 109–120
- Van der Aalst WMP (2014) *Process mining discovery, conformance and enhancement of business processes*. Springer, Berlin
- Van der Aalst WMP, Song M (2004) Mining social networks: uncovering interaction patterns in business processes. In: Desel J, Pernici B, Weske M (eds) *Business process management BPM 2004*, vol 3080. *Lecture notes in computer science*. Springer, Berlin, pp 244–260
- Van der Aalst WMP, Weijters AJMM, Maruster L (2004) Workflow mining: discovering process models from event logs. *IEEE Trans Knowl Data Eng* 16:1128–1142. <https://doi.org/10.1109/TKDE.2004.47>
- Van der Aalst WMP, Reijers HA, Weijters AJ, Van Dongen BF, Medeiros AKA, Song M, Verbeek HMW (2007) Business process mining: an industrial application. *Inf Syst* 32(5):713–732
- Van Dongen BF, van der Aalst WMP (2005) A meta model for process mining data. In: *Proceedings of the CAiSE workshops (EMOI-INTEROP workshop)*, Ceur-ws.org, Aachen, pp 309–320
- Weijters AJMM, van der Aalst WMP (2003) Rediscovering workflow models from event-based data using little thumb. *Integr Comput Aided Eng* 10:151–162. <https://doi.org/10.3233/ICA-2003-10205>
- Wen L, Wang J, Sun J (2006) Detecting implicit dependencies between tasks from event logs. In: Zhou X, Li J, Shen HT, Kitsuregawa M, Zhang Y (eds) *Frontiers of WWW research and development: APWeb 2006*, vol 3841. *Lecture notes in computer science*. Springer, Berlin, pp 591–603
- Wen L, Wang J, van der Aalst WMP, Wang Z, Sun J (2009) A novel approach for process mining based on event types. *J Intell Inf Syst* 32:163–190. <https://doi.org/10.1007/s10844-007-0052-1>
- Wernicke S (2005) A faster algorithm for detecting network motifs. In: Casadio R, Myers G (eds) *Algorithms in bioinformatics WABI 2005*, vol 3692. *Lecture notes in computer science*. Springer, Berlin, pp 165–177

Wernicke S, Rasche F (2006) FANMOD: a tool for fast network motif detection. *Bioinformatics* 22:1152–1153. <https://doi.org/10.1093/bioinformatics/bt1038>

Whitten JL, Bentley LD, Lonnie D (2007) *System analysis and design methods*. McGraw-Hill, New York

Wong EA, Baur B (2010) On network tools for network motif findings: a survey study. *Data Min Bioinform* 9:122–134

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.