Journal of Computer Virology and Hacking Techniques
https://doi.org/10.1007/s11416-021-00398-8

ORIGINAL PAPER

Anomaly detection in business processes logs using social network analysis

Mina Ebrahim[1] · Seyed Alireza Hashemi Golpayegani[2]

Received: 5 November 2020 / Accepted: 23 August 2021

Abstract
This paper presents an approach to detect anomalies in process-aware information systems. This approach is based on process mining and uses social network analysis metrics to detect anomalous behavior. The main idea is to prove that applying the organizational perspective using social network analysis metrics can detect anomalies that follow a normal flow but are executed by unauthorized users. The proposed approach has been evaluated using artificial event logs and the cross-validation method. The F-measure evaluation results show that this approach is even effective in the worst case, the highest anomaly rate.

Keywords Social network analysis · Process aware information system · Process mining · Anomaly detection

1 Introduction

During the financial scandals of large companies such as Enron, Parmalat, WorldCom, related to misrepresentations and fraudulent financial statements, in 2002, the US government developed new certifications and manuals to return public confidence to the capital market. As a result, organizations are forced to adopt process-aware information systems(PASs) to automate their business processes [1].

A process-aware information system is a software system that manages and executes operational processes involving people, applications, and information sources based on process models [2]. Since organizations need a rapid response to strategic changes or changes in business models between partners, adopting normative process-aware information systems that suggest the execution of the same set of tasks may impose severe drawbacks on the organizations' competitiveness. Thus, organizations need flexible process-aware information systems, although flexibility may compromise security [3].

Since detecting abnormal events can help adopt process-aware information systems without losing security properties, many studies have been conducted since 2005 on anomaly detection in process-aware information systems to re-balance the trade-off between security and flexibility [3].

Most research on anomaly detection has been conducted using process mining and has emphasized the process perspective [1,3–10]. Therefore, the basis for anomaly detection in these studies has been the order of executing activities. However, in real life, anomalies can be affected by all three data, process, and organizational perspectives or even a combination of them [10].

Also, in most of these studies, it is assumed that rare traces are anomalous. However, it should be noted that although an anomalous trace is not necessarily a rare trace, the opposite of it is not valid; a rare trace is not necessarily an anomalous one [4].

However, despite studies conducted to extract process mining models from the organizational perspective [11,12] no research has been conducted that directly investigates this perspective to detect anomalies in process-aware information systems.

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

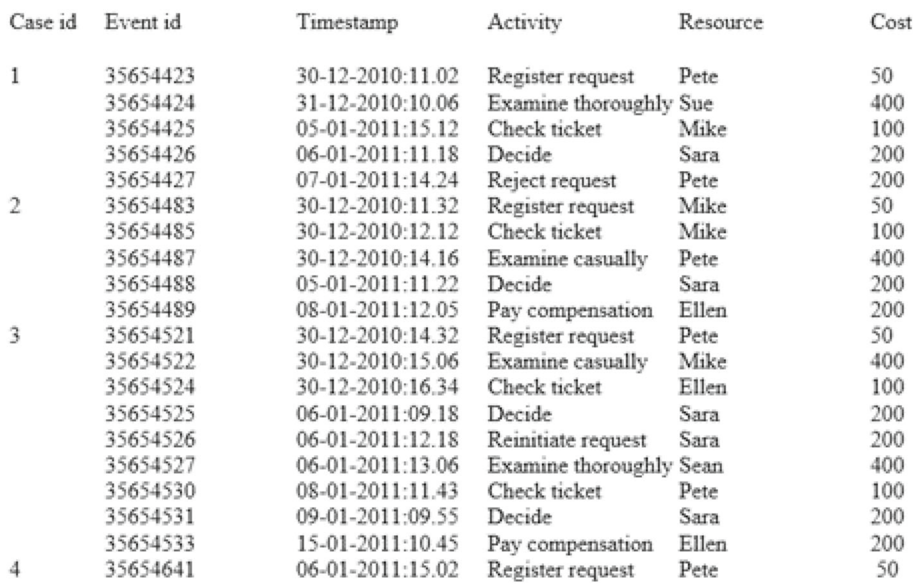
Mina Ebrahim
minaebrahimi@aut.ac.ir

[1] Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

[2] Department of Computer Engineering and IT, Amirkabir University of Technology, Valiasr Street, Tehran 15875-4413, Iran

Published online: 03 September 2021

Fig. 1 Part of an event log: each line corresponds to an event [13]



Case id	Event id	Timestamp	Activity	Resource	Cost
1	35654423	30-12-2010:11.02	Register request	Pete	50
	35654424	31-12-2010:10.06	Examine thoroughly	Sue	400
	35654425	05-01-2011:15.12	Check ticket	Mike	100
	35654426	06-01-2011:11.18	Decide	Sara	200
	35654427	07-01-2011:14.24	Reject request	Pete	200
2	35654483	30-12-2010:11.32	Register request	Mike	50
	35654485	30-12-2010:12.12	Check ticket	Mike	100
	35654487	30-12-2010:14.16	Examine casually	Pete	400
	35654488	05-01-2011:11.22	Decide	Sara	200
	35654489	08-01-2011:12.05	Pay compensation	Ellen	200
3	35654521	30-12-2010:14.32	Register request	Pete	50
	35654522	30-12-2010:15.06	Examine casually	Mike	400
	35654524	30-12-2010:16.34	Check ticket	Ellen	100
	35654525	06-01-2011:09.18	Decide	Sara	200
	35654526	06-01-2011:12.18	Reinitiate request	Sara	200
	35654527	06-01-2011:13.06	Examine thoroughly	Sean	400
	35654530	08-01-2011:11.43	Check ticket	Pete	100
	35654531	09-01-2011:09.55	Decide	Sara	200
	35654533	15-01-2011:10.45	Pay compensation	Ellen	200
4	35654641	06-01-2011:15.02	Register request	Pete	50

1.1 Objectives and outlines

The present study aims to provide an approach for anomaly detection in process-aware information systems using a social network analysis approach focusing on both control flow and organizational perspectives.

Despite other studies, rare traces are not considered anomalies in this study, and traces that are not executed according to the normal model or unauthorized users perform them are assumed to be anomalous.

Additionally, since there is no pre-determined information about normal and anomalous data, the unsupervised approach has been selected among the basic anomaly detection approaches. Simultaneously, this approach tries to keep the false alarm rate or the ratio of the number of cases or users who do not behave abnormally but are placed mistakenly in the class of cases with abnormal behavior low.

The rest of the paper is organized as follows: Sect. 2 includes the theoretical background of the proposed anomaly detection system. Related research conducted on anomaly detection in process-aware information systems is reviewed in Sect. 3. In Sect. 4, the proposed anomaly detection model based on process mining and social network analysis approach is explained along with a description of each step of the problem-solving method and how to implement the proposed strategy. How to perform the evaluation and the results are presented in Sect. 5. Section 6 includes the conclusion of the proposed approach, future research recommendations, and the current study's limitations.

2 Theoretical background

Scientific theories and models that have been chosen as the basic concepts required to conduct this research to diagnose differences between a standard model and observed behaviors in PAISs are studied in this section.

2.1 Process mining

Today's information systems are recording events in so-called event logs. Information systems log enormous amounts of detailed information about the activities that have been executed; such data is known as event logs. An event refers to an activity (i.e., a well-defined step in the process) and is related to a particular case (i.e., a process instance).

Figure 1 illustrates the typical information present in an event log [13].

Process mining aims to discover, monitor, and improve real processes by extracting knowledge from these event logs [13]. Event logs can be used to conduct three types of process mining: conformance checking (comparing the actual process with some prior model), discovery (deriving a model from scratch), and Extension(extend or improve an existing process model) [13].

Process models can be seen as the "maps" describing organizations' operational processes [13]. Various notations exist to model operational business processes. Petri nets are the oldest and best-investigated process modeling language allowing for the modeling of concurrency. A Petri net is a bipartite graph consisting of places and transitions. When modeling business processes in Petri nets, we often consider a subclass of Petri nets known as WorkFlow nets (WF-nets).

Process mining algorithms for process discovery can transform the information shown in figure 1 into process models. For instance, the basic α-algorithm [5]. discovers the Petri net of figure 2 when providing it with the input data in figure 1.

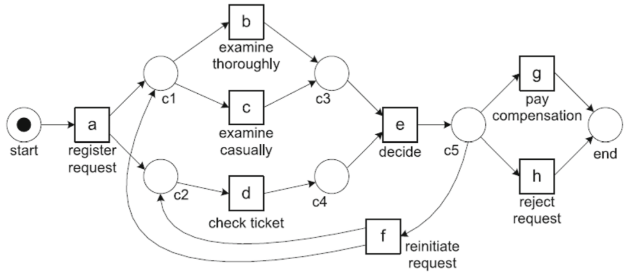


Fig. 2 Example of a discovered process model by α-algorithm [13]

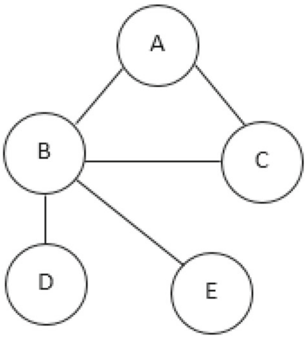


Fig. 3 An example of a friendship network

WoPeD (Workflow Petri net Designer) is an easy-to-use, Java-based open-source software tool, which can edit, simulate and analyze workflow nets. The WoPeD editor offers full support for the class of workflow nets, including operators, triggers, sub-processes, resource assignments, and quantitative parameters like task service times or branching probabilities of XOR splits [14]. The present study focuses on the conformance type of process mining and considers e-commerce systems as an information system to implement and evaluate the proposed methodology.

In this paper, a process instance or a trace is considered a set of activities users can perform on an e-commerce system. Moreover, an appropriate process model is considered a normal and acceptable process to navigate an e-commerce system. This model is used to extract all the normal traces of activities that users can execute. These normal traces and role-activity matrix can provide a role-trace matrix, indicating which trace of activities each user can perform.

2.2 Social network analysis

A social network is a network that shows social relationships among individuals in a community. Graphs usually represent the structure of such networks. Therefore, networks are often regarded as equivalent to graphs.

A graph is composed of two fundamental units: nodes and edges. A pair of nodes define every edge. According to the application field, nodes can represent various individual entities (e.g., people, organizations, and countries).

In turn, an edge is a line that connects two nodes and, analogously, it can represent numerous kinds of relationships between individual entities (e.g., communication, cooperation, and friendship).

Edges may be directed or undirected, depending on if the nature of the relation is asymmetric or symmetric [15].

The underlying structure of such networks is the object of the study of Social Network Analysis (SNA).

SNA methods and techniques were thus designed to discover patterns of interaction between social nodes in social networks. Hence, the focus of SNA is on the relationships established between social entities rather than the social entities themselves.

This technique's primary goal is to examine both the contents and patterns of relationships in social networks to understand the relations among nodes and the implications of these relationships [15].

In this paper, the anomaly concept is drawn in the form of a social network. Closeness centrality and clustering coefficient, two of the metrics for social network analysis, are used to detect abnormal behavior nodes.

Closeness centrality is a rough measure of a node's overall position in the network, giving an idea about how long it will take to reach other nodes from a given starting node. It is the mean length of all shortest paths from one node to all other network nodes [15].

In the social network context, closeness is a measure of reachability that measures how fast a given actor can reach everyone in the network [15]. The normalized closeness centrality metric for a node is obtained from Eq. 1.

$$C_c(i) = \left(\left(\sum_{(j=1)}^{N} d(ij) \right) / (N-1) \right)^{-1} \tag{1}$$

Where i is the node for which we intend to calculate the closeness centrality, *d(ij)* is the shortest distance of node i from each node in the network, and N is the number of nodes in the network.

In the friendship network of figure 3, each node represents a person, and the edges represent a friendship relationship between two people. Here, the *d(ij)* value for node B is 4, and the closeness centrality is 1. It means node B is an influential user that can quickly spread information within the network.

Social networks are naturally transitive, which means that a given node's friends are also likely to be friends. This transitivity property is quantified by a clustering coefficient that can be global, i.e., computed for the whole network, or local, i.e., computed for each node. In the context of the local clustering coefficient, transitivity is a local property of a node's neighborhood that indicates the level of cohesion between a node's neighbors. Therefore, this coefficient is given by the fraction of pairs of nodes, neighbors of a given node connected by edges [15]. The local clustering coefficient for a node can be calculated by equation 2.

$$C_i = (2L_i)/(k_i(k_i - 1)) \tag{2}$$

K_i represents the degree of node i, and L_i indicates the number of edges between neighbors of node i. The value of this measure is between 0 and 1.

In the network of figure 3, the value of L_i for node B is 1, K_i is equal to 4, and its clustering coefficient is 0.16. Moreover, the clustering coefficient values for nodes A and C are one, and nodes D and E are 0.

3 Related works

Conducted research on anomaly detection in business process context can be divided into three main categories:

- Based on process mining approach and focus on flow control perspective.
- Based on neural networks and analyze the temporal behavior of activities.
- Based on machine learning, deep learning, and natural language processing.

None of these works deal explicitly with the organizational perspective; they refer only to the cases' control flow or data values. In the following, these works and research conducted in social network analysis will be reviewed to investigate the efficiency of closeness centrality and clustering coefficient metrics for finding outliers from the organizational perspective.

Aalst and Medeiros first raised anomaly detection using process mining in the process-aware information systems in research [5]. They presented two anomaly detection methods that are supported by the α-algorithm. A drawback of this work is that it demands a known "normal" log, but a known "normal" log may not be available in applications domains that demand flexible support.

In [4] and [6], Bezerra and Wainer presented three approaches to detect anomalous traces: sampling, threshold, and iterative approaches. Nevertheless, as pointed out by the authors, the methods presented in [4,6] have severe practical limitations, directly resulting from the adopted process mining algorithm, which can not deal with larger logs.

In [7] and [3], the rationale for detecting anomalous traces is that given a process model mined with the help of a process mining algorithm, α-algorithm, the compliance level between this model and the whole log is inferior when the log comprises an anomalous trace. In these two studies, the traces whose compliance level variance is more significant than a given threshold value are considered abnormal traces.

The conformance metrics presented in [16], [17] have been used in these studies to quantify a log and model compliance. These metrics are based on two dimensions: fitness and appropriateness. The fitness dimension is measured by a metric with the same name. In comparison, the appropriateness dimension is measured by two metrics, structural and behavioral appropriateness. In 2008 they added the size metric. The anomaly detection algorithm based on the size metric has shown the best accuracy; it has correctly classified nearly 91% of traces from the logs [3]. As a result of their work, the authors cite three extension points that may influence the accuracy of the anomaly detection algorithm: process mining algorithm, a noise metric, and sampling size.

In joint research in 2008, Bezerra, Wainer, and van der Aalst [1] presented an approach to detect anomalous traces using available process mining tools of the ProM framework. As the authors have stated, the model's fitness metric is not very precise. This approach does not automatically find the appropriate model and requires manual inspections by the security responsible.

The authors of [8] presented an approach for the genetic process mining, arguing that the fitness function explained in this study allows for extra behavior that is not seen in the log and needs to be improved.

Therefore, they presented a new approach for process mining using a genetic algorithm [9]. This approach uses an appropriate fitness function that considers the completeness and accuracy of the model. In this approach, if a group of individuals can fit all the traces, an individual with less extra behavior will have more fitness [9]. This genetic algorithm is implemented as a plug-in in the ProM framework.

Bezerra et al. [10] conducted another study in 2013 in which they improved their three algorithms proposed in [4] and [6] and provided a new evaluation. The results obtained from the implementation of these three algorithms show a lot of false positives.

In Nolle et al. [18], proposed a system that relies on neural network technology. In this study, the authors used the autoencoder neural network to detect anomalies related to the flow control perspective during business processes execution. The proposed method can detect anomalous traces without the need for prior knowledge. The authors of this study are skeptical about the effectiveness of their proposed approach in high anomaly rates.

The presented anomaly detection approaches, to now, are limited to the control-flow perspective. For example, an anomaly may follow a normal flow but produce anomalous data or be executed by unauthorized users or executed in an unauthorized run time. Therefore, data, time, and organizational perspectives should also be considered to provide more accuracy.

In [19], the authors analyze the temporal behavior of individual activities to identify anomalies in single process instances. However, malicious users can split an attack on different process executions. Thus an anomaly detection approach that can consider multiple process instances is required. To this end, Böhmer and Rinderle-Ma [20] proposed an unsupervised anomaly detection heuristic that exploits the temporal dependencies between multiple instances. This approach must rely upon a model built from noise-free data, which is unlikely in real cases. To overcome this issue, Nolle et al. [21] proposed a system relying on autoencoder neural network technology to deal with the noise in the event log and learn a representation of the underlying model.

Another group of anomaly detection approaches emerges from Machine Learning methods applied in business process contexts. In [21,22], the authors use an autoencoder to model process behavior. The technique encodes the event log using one-hot encoding and trains the autoencoder using the log as both the input and the output. The mean squared error between the input and output is measured, and given a threshold, anomalous instances are highlighted. The main drawback of the approach is that vector sizes increase linearly with the number of activities, which is costly resource-wise.

Moreover, the one-hot encoding technique produces very sparse vectors, further increasing computational overhead. In [18,23], the authors proposed a deep learning method considering both control and data perspectives to overcome this issue.

The technique uses a deep neural network trained to predict the next event. An activity or attribute with a low execution probability is interpreted as an anomaly given the network probability score. However, the computational cost of deep learning methods is very high, which hinders its application in many scenarios.

Tavares et al. [24] use the Trace2-vector representations to identify anomalous cases. The method's core is based on Natural Language Processing encoding of textual data. For that, they mapped activities and traces as words and sentences, respectively, before applying the word2vec encoding algorithm. Word2vec captures contextual information, i.e., it models activities' surroundings, such that a vector represents each activity. Traces with uncommon encoding are potential anomalies.

The drawback of most previous research is that they focused on business processes from the activity (control-flow) perspective, while less attention has been devoted to the organizational perspective. Process mining from the perspective of an organization is also called organizational mining. Organizational mining can be divided into four categories: organizational structure discovery, SNA, resource allocation, and role mining [11].

SNA aims to find the relationship between entities and explore possible bottlenecks in social networks to improve process efficiency. Community discovery is an essential issue in SNA. Community is a subset of social networks, and people within a community share similar roles and responsibilities. Discovering communities in a social network can help understand the specialization and cooperation in the organization [25].

The authors of [25] proposed the clustering coefficient as a measure to detect communities in graphs. It has been found that the nodes in a more central region of the cluster usually have a higher clustering coefficient value than the others.

Some path-based methods quantify nodes' importance, such as Closeness centrality [26], which detects influential nodes within a network. Influential users are the ablest to spread information within the network.

Three social network analysis techniques to detect fraud have been described in [27]: neighborhood metrics, centrality metrics, and collective inference algorithms. The approaches and the metrics proposed in [27] for the network analysis are only applicable when some network nodes are labeled.

Additionally, in [28], the authors state that three features in a graph can indicate the occurrence of anomalies in the graph: Near-cliques and stars, Heavy vicinities, and Dominant heavy links.

Reviewing previous studies about the use of social network analysis in fraud detection, community detection, and finding influential nodes ensures that social network analysis metrics could help detect anomalies in logs of PAISs.

4 Methodology

Current research assumes a prior appropriate process model, which can be discovered using process discovery algorithms from event logs or has been reverse-engineered from the information system structure. For the implementation and evaluation of the proposed approach in this study, an e-commerce system is considered a process-aware information system due to its navigation structure guiding the flow of work in the system. The website's structure of e-commerce systems represents the normal execution flow within these systems; therefore, it can discover the appropriate process model. Another assumption of this research is an existing role-based access control matrix, determining which roles can navigate each normal procedure.

The problem-solving methodology of this study consists of the following five steps:

- Extract normal traces from the process model.
- Create a role-trace access control matrix.
- Convert the set of traces in the abnormal log to a social network.

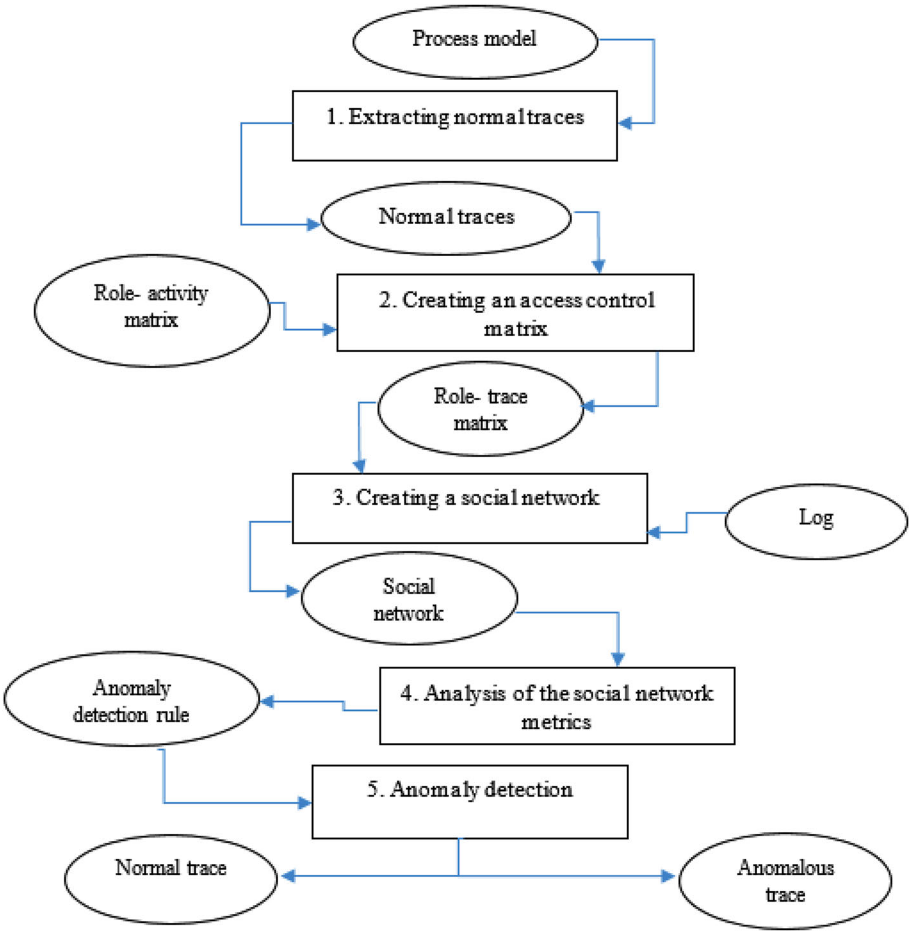


Fig. 4 The problem-solving method proposed for the anomaly detection in event logs

- Analysis of the metrics for the social network
- Classifying traces and detecting anomalous instances

Figure 4 shows a schematic of the methodology proposed for solving anomaly detection problems in event logs. Each of the proposed methodological steps is explained below.

Each of the proposed steps is explained below.

4.1 Extracting normal traces from the process model

In the first step of the problem-solving methodology, normal traces are extracted from the discovered process model using the WoPeD process modeling tool. By running the quantitative simulation properties in WopeD and considering that the process model used is a normal process model with no anomalous traces, cases entering this model navigate normal traces. These traces for each case are displayed in a log.

4.2 Creating a role-trace access control matrix

Based on the assumptions considered in this study, an existing role-activity access control matrix shows authorized roles for executing activities on the e-commerce website. Therefore, based on the role-activity access control matrix and the normal traces extracted in the previous step, a role-trace access control matrix is extracted in the second step that shows each role is authorized to perform which traces.

4.3 Converting the set of traces in the abnormal log to a social network

The logs that include the format defined in this approach must have log label columns consisting of the user's IP address, the trace navigated by each user, and the user's role. In Sect. 5.1, the data format will be explained in more detail. In the third step of the problem-solving methodology, each log, including normal and abnormal traces, is converted to a social network using the role-trace access control matrix. The scenario considered for modeling the network and defining its nodes and edges based on the logs is as follows:

Nodes: Users executing process instances
Edges: An edge is drawn between two users who execute different traces with the same role, while each of the traces is not a normal trace, or they are normal traces, but the role of the user executing the traces is

> not an authorized role for the trace according to the access control matrix. The resulting network will be an undirected, unweighted, and explicit network. It can be deduced from the definition presented for the edges that rare traces are not considered an anomaly. Additionally, the anomaly defined in this study is contextual. A scenario may be regarded as a definition of an anomaly in the context of the process-aware information system workflow. In contrast, the same scenario may not represent an anomaly in other contexts. After defining the scenario proposed for illustrating a network to display and analyze the logs' anomaly, the next step is to discuss how to implement the proposed approach for anomaly detection. The implemented algorithm receives each log's trace to illustrate the network and compares it with the other traces in the log. Suppose users with the same role execute each pair of these traces, but the two traces are different. In that case, at least one of the activities performed in each trace is different from the other. Then, It is investigated in the role-trace access control matrix to see whether the two traces are among the normal traces and can be navigated in the structure of the e-commerce website or not moreover, whether the role of users performing these traces is a part of the authorized roles for the two traces or not. If at least one of the roles or traces in the access control matrix is not a part of the authorized roles or normal traces, an edge is drawn between the two users. Following the implementation of the proposed scenario, how the metrics and techniques of the social network analysis are used to detect the network nodes' anomaly are discussed in the next step.

Considering the definition of the edges, it is obvious that the subject of the anomaly and its definition is modeled as a social network in which the nodes will be users executing the procedures. The resulting network will be an undirected, unweighted, and explicit network.

It can be deduced from the definition presented for the edges that unlike many studies conducted in this field, rare traces are not considered as anomaly. Additionally, the anomaly defined in this study is contextual so that a scenario may be considered as an anomaly only in the context of the processes of execution on e-commerce websites, while the same scenario may not be an anomaly in other contexts.

After defining the scenario proposed for illustrating a network to display and analyze the anomaly in the logs, it is time to discuss how to implement the proposed approach for the anomaly detection. The implemented algorithm, to illustrate the network, receives each trace of the log and compares it with the other traces in the log. If each pair of these traces is executed by users with the same role, but the two traces are different, i.e. at least one of the activities performed in each trace is different from the other, it is investigated in the role-trace access control matrix to see whether the two traces are among the normal traces that can be navigated in the structure of the e-commerce website or not, and whether the role of users performing these procedures is a part of the authorized roles for the two traces or not. If at least one of the roles or traces in the access control matrix is not a part of the authorized roles or normal traces, an edge is drawn between the two users.

Following the implementation of the proposed scenario, in the next step, the way in which the metrics and techniques of the social network analysis are used to detect the anomaly among the network nodes is discussed.

4.4 Analysis of the social network metrics

In the fourth step, using the measures and techniques of network analysis, the metrics required to analyze the social network resulting from the third step to detect anomalies are determined. Based on the values of these metrics, the classifier rule for detecting anomalies is introduced. The social network analysis metrics are divided into node-level and network-level metrics based on evaluating only one node or the entire network [15].

In the problem of detecting anomalies from an organizational perspective, the focus will be on individuals. In other words, when the problem is illustrated as a network, the metrics for the network analysis based on which anomalies can be detected will be the node-level metrics. However, it cannot be said explicitly that only node-level metrics are suitable for detecting abnormalities. Therefore, to determine the appropriate metrics, it is necessary to perform different experiments on the networks obtained from the logs and investigate different problem parameters' effect on the proposed approach.

Based on the approaches proposed in the studies [27], [28], the present research's main idea for anomaly detection is closeness centrality and clustering coefficient metrics. The experiments conducted in this study have shown that the clustering coefficient and closeness centrality metrics are among the suitable metrics for social network analysis to detect anomalies from the organizational perspective. How these metrics can be used to detect frauds and how they can be effective are discussed below.

According to the anomaly definition, there are three types of anomalies in this study: Role anomaly, Activity anomaly, and Role and activity anomaly.

There is a straight edge between users with role anomaly, i.e., users who execute normal traces out of their access authorizations and nodes with the same role due to different navigation. It can be interpreted that in a network including anomaly, users with role anomaly have a short distance from the normal nodes and abnormal nodes with the same role. Thus, it can be said that they will have high centrality.

Also, users with activity anomalies, i.e., users who navigate non-normal traces, have little or no connection to their neighbors. Therefore, it can be said that they have a low clustering coefficient. Accordingly, the following hypothesis can be presented based on the concepts mentioned above:

> Hypothesis (1): In a network in which the relationship between the nodes is based on the presence of at least one of the two nodes in different types of anomaly, anomalous nodes are those that have a high degree of centrality or low clustering coefficient.

The high centrality or low clustering coefficient's value differs in logs with different anomaly rates; the exact value of detecting anomalies is presented below based on the experiments conducted.

According to the proposed scenario, two nodes will be connected if they have different navigation with the same role. If the log's anomaly rate is low, there will be no connection or a slight connection between the node and the one-step distance nodes. It means that other nodes with the same role have the same and authorized navigation, and therefore there is no connection between them. However, if the anomaly rate is high, all the node neighbors may be connected, and this approach may not detect the anomaly. This issue needs further investigation, which is discussed below.

It should be noted that although the closeness centrality and the clustering coefficient are considered as metrics for detecting anomalies, it cannot be said that the clustering coefficient is only used for the detection of anomalies in activities. This metric can also detect anomalies in roles where users have normal but unauthorized navigation depending on how the network is formed and the number of links between the nodes. Therefore, we will use both metrics to detect anomalies.

4.5 Classification of traces and the detection of anomalous instances

Finally, in the fifth step of the problem-solving methodology, the rule for classifying the traces into normal and anomalous traces is created based on the analysis of social network metrics values introduced in the previous step. This rule can be stated as follow:

Classifier Rule :“ If a node has a low clustering coefficient or a high closeness centrality, the node may have activity or role anomaly.”

5 Evaluation

A real or artificial event log of the process-aware information systems is needed to implement the proposed approach. An important point about the log is that we need to know precisely which process instance is normal or anomalous and which user has performed the correct process instance according to the role and authority assigned to him. In the real logs, to ensure the correct number of anomalies and frauds, the logs must be manually checked, which is difficult and time-consuming, and the result of this approach depends on the expert's knowledge. Therefore, to know the logs' contents completely, it is possible to create a log that includes normal and anomalous data.

Additionally, since this study seeks to detect anomalies in e- systems, the artificial logs must have a structure that corresponds to the logs obtained from the users' navigation on e-commerce websites. In this study, the logs created based on the simulation of the execution paths on the e-commerce website are used to evaluate the proposed approach's performance.

Appropriate metrics for determining a binary classifier's quality that classes are usually positive and negative are precision, accuracy, and recall [10].

In this study, positive means that a role or trace is anomalous, and negative means that the role or trace is normal. The two metrics of precision and recall have been used to determine the binary classifier's quality.

5.1 Data collection method

The appropriate process model of a process-aware information system with a known navigation structure is required to implement and evaluate the proposed methodology. A typical kind of this system is an e-commerce system.

Therefore, in this paper, an e-commerce website's workflow or process model is simulated on the client-side and the server-side to create an artificial event log. The event log created is as similar to the actual log as possible.

Using WoPeD as a process modeling tool, the structure of a hypothetical e-commerce website (such as Amazon and eBay) based on business processes supported by the website is modeled and then simulated in the form of a Petri net. Following modeling an e-commerce website's structure, possible and authorized traces based on user roles are simulated and produced. For this purpose, the Quantitative Simulation property in the WoPeD is used. In this study, the parameters considered were the simulation time of 2 months, the case arrival rate of 500, the service method of FIFO, and the service and the arrival distribution function of Poisson.

The point that should be noted is mapping the log obtained from the simulation and the data structure for navigating a website.

The log obtained from the simulation includes the case identifier, activities performed by a case, and each activity's duration. However, in this study, the log file must include each user's IP address, the links they have clicked (URL), and the user's role on the e-commerce website.

In this study, the method of mapping between the concepts available on the e-commerce website and the process model is as follows:

- Each link on the website is equivalent to an activity in the process model or a Petri net transition.
- Each page of the website is equivalent to one place in the Petri net.
- Each case number is equivalent to the user's IP address.
- Each user's navigation on web pages from arrival to leaving the website is equivalent to a trace in the process model.

Since users have no role in the log obtained from the simulation, we assign each procedure's roles before creating anomalies using the role-trace access control matrix. As the execution path on a website is not anomalous so far, there are no anomalies in the log obtained from it. However, to evaluate the proposed approach, it is necessary to create an anomalous log; three types of anomalous traces are created for each normal trace on the website, and the trace turns into a trace out of the role-trace access control matrix:

Anomalous trace (1):	An activity is added to the trace.
Anomalous trace (2):	An activity is removed from the trace.
Anomalous trace (3):	Another activity replaces an activity in the trace.

The normal trace randomly replaces one of the three anomalous traces to create an activity anomaly for 10% of the log traces. Additionally, 2% of users are randomly replaced by an unauthorized role to create a role anomaly, which may also occur for users with activity anomaly, causing role and activity anomalies.

5.2 Evaluation methodology

In this study, F-measure has been used to evaluate the proposed approach according to equation 3. F-measure is the weighted average of precision and recall, which reaches its best value at one and worst at zero. The precision and recall are also calculated according to equations 4 and 5, respectively. The precision indicates the ratio of correctly identified anomalous instances to the number of identified anomalies. In contrast, the recall indicates the ratio of correctly identified anomalous instances to the total number of actually anomalous instances.

$$F - measure = 2/(1/precision + 1/recall) \quad (3)$$

$$precision = TP/(TP + FP) \quad (4)$$

$$recall = TP/(TP + FN) \quad (5)$$

The true positive (*TP*) indicates the number of anomalous instances, and the system has correctly identified them. The false positive (*FP*) indicates the number of normal instances incorrectly identified as anomalous instances. Finally, the false negative (*FN*) indicates the number of anomalous instances that have been incorrectly identified as normal instances.

5.2.1 The cross-validation for the anomaly detection

Since anomaly rate is essential in this research, a cross-validation method with minor changes has been used to evaluate the proposed approach. In this method, one-fifth of the data is randomly selected as a test data set each time the experiment is performed, and the remaining is considered as training data set. Therefore, anomalies with different rates are placed in each data set.

5.2.2 Parameters for the evaluation of the proposed approach

The parameters that have the highest effect on the efficiency of the approach proposed in this study are the rates of role and activity anomalies. The proposed approach's performance has been evaluated in two test data sets for different parameter values.

5.2.3 Experiment 1

In this experiment, the efficiency of the proposed approach was investigated for the low anomaly rates. Table 1 shows the values of the parameters for experiment 1.

Based on the list of users who have role or activity anomalies and the result of calculating clustering coefficient and closeness centrality metrics from the training data sets, the following rules are obtained for the anomaly detection: $0.542 \leq$ Closeness Centrality ≤ 1 , $0 \leq$ Clustering Coefficient < 0.3

These rules mean that users with the clustering coefficient ranging from 0 to 0.3 have role or activity anomalies, or both of them. Moreover, the closeness centrality metric for users with role or activity anomalies or both of them ranges from 0.542 to 1.

The two rules above lead to standard results as both detect the same nodes as anomalous nodes. However, it is possible that some anomalous nodes cannot be detected using the

Table 1 Parameters for experiment 1

Number of process instances	Frequency of the experiment repetition	Number of training instances in each repetition	Number of test instances in each repetition	Percentage of the role anomaly in the test data set	Percentage of the activity anomaly in the test data set
500	20	400	100	Ranging from 0 to 10%	Ranging from 0 to 20%

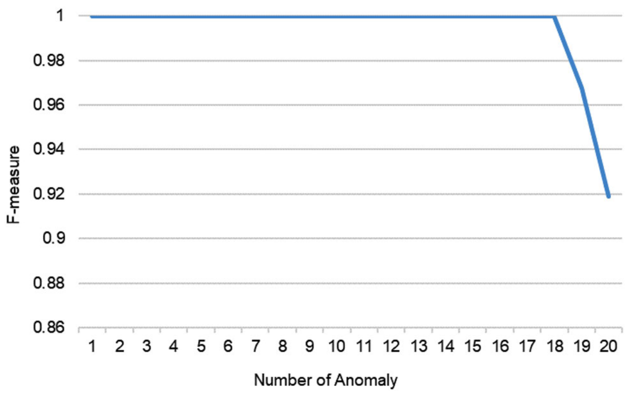


Fig. 5 The diagram of the evaluation of the proposed approach's performance in logs with a low anomaly rate

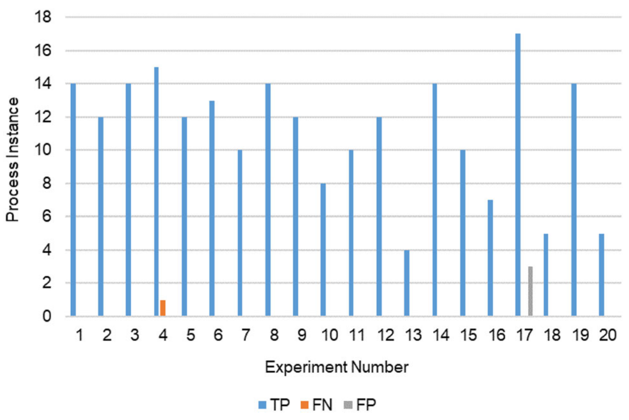


Fig. 6 The diagram of evaluating the proposed approach's performance in logs with a low anomaly rate

clustering coefficient metric, but they can be detected by the closeness centrality metric and vice versa.

Figure 5 shows the F-measure's value versus the number of anomalies in the logs with a low anomaly rate based on the experiments performed. According to figure 5, increasing anomaly rates from 4 to 17 in logs with 100 records causes F-measure's value to decrease from 1 to approximately 0.92.

The anomaly detection approach focuses on decreasing the FN rate (false negative), i.e., the number of anomalous instances is considered normal [10].

According to figure 6, the false-negative rate in the 4^{th} repetition of the experiment is 1, and in the 17^{th} repetition is 13. The rate of anomalies is higher in these two experiments compared to other repetitions. These two repetition causes more experiments to evaluate the proposed approach's performance in case of a high anomaly rate.

Table 2 shows the mean values of F-measure, recall, and precision for this experiment. When evaluating an anomaly detection approach's performance, it should be considered that the recall's value is as close to 1 as possible. It means the rate of the false negative is decreased, and at the same time, the value of precision is high, i.e., less normal instances are considered anomalous [10].

Therefore, according to the values presented in table 2, this approach is highly effective when a low rate of anomalies occurs in the logs.

5.2.4 Experiment 2

This experiment aimed to investigate the effectiveness of the proposed approach in high anomaly rates. Table 3 shows the values of the parameters for experiment 2.

Figure 7 shows the F-measure's value versus the number of anomalies with high anomaly rates in the logs. In these experiments, the F-measure value has been approximately between 0.8 and 0.95, indicating that this approach has an appropriate performance when there is a high anomaly rate. Figure 8 also shows the values of false positive, false negative, and true positive. As it can be seen, the values of false negative and false positive versus the true positive detected in this approach are low.

Table 4 shows the mean values of F-measure, recall, and precision in the experiments conducted at a high anomaly rate. The values of all three metrics indicate the proposed approach's appropriate effectiveness when a high anomaly rate occurs. According to the scenario proposed for converting a log data set to a network, almost all node's neighbors are anomalous and connected when the anomaly rate is high. That is why the clustering coefficient metric shows a high value, which contrasts with the approach proposed for anomaly detection. Thus, the only metric that can be used in this case is the closeness centrality.

A node with a high clustering coefficient means that this node has lots of connections with other nodes in the net-

Table 2 Evaluation results of experiment 1

	F-measure	Recall	Precision
Social network anomaly detection (SNAD)	0.9925	0.996875	0.99282

Table 3 Parameters for experiment 2

Number of process instances	Percentage of activity anomaly in each repetition	Percentage of role anomaly in each repetition	Frequency of the experiment repetition
500	Ranging from 50 to 100	Ranging from 10 to 100	15

Table 4 The results of the evaluation by experiment 2

	F-measure	Recall	Precision
Social network anomaly detection (SNAD)	0.8166	0.9643	0.8784

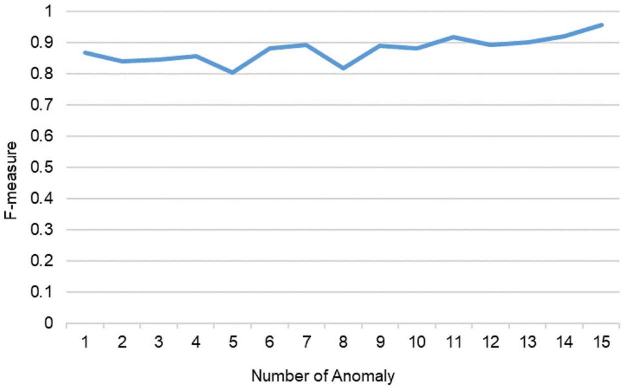


Fig. 7 The diagram of evaluating the proposed approach's performance in logs with a high anomaly rate

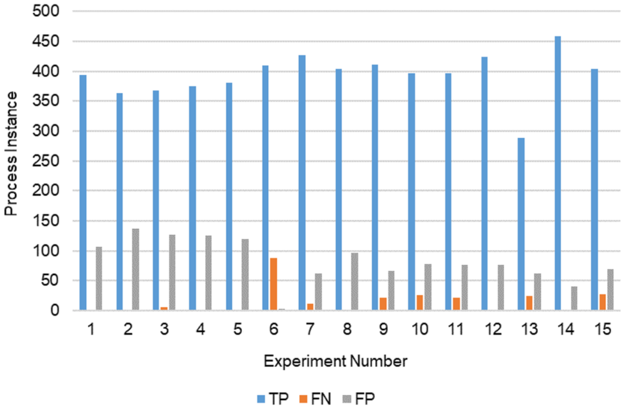


Fig. 8 The true positive, false positive and false-negative rate for the proposed approach in logs with a high anomaly rate

work. When the anomaly rate is high in these logs, the clustering coefficient's value is very high; according to the classifier rule, it considers all nodes normal, which is wrong. Therefore, according to the definition of anomaly, when the anomaly rate is high, the high clustering coefficient indicates that there are many anomalous nodes in this network, and when the anomaly rate is low, the low clustering coefficient indicates that there are many normal nodes in this network. However, Closeness centrality is the only helpful metric. Although, in high anomaly, this metric also produces lots of false positives.

According to the scenario proposed for converting a log data set to a network, almost all node's neighbors are anomalous and connected when the anomaly rate is high. It causes the shortest path from each node to the other nodes in network to be minimum and the closeness centrality to be maximum, though every node has a high closeness centrality and will be detected as an anomaly.

However, the main goal of anomaly detection is to keep false negatives as lower as possible, that in high anomaly rates, the proposed approach showed relatively fewer false positives.

Nevertheless, almost in all real cases, anomalies are infrequent, and high anomaly means that more than half of the users in the system are executing abnormal traces; then, it can be concluded that the current approach has enough efficiency to detect anomalies.

Also, it is should be mentioned that regarding the definition of clustering coefficient metric in cases that the form of network is near to stars, it means that most of the nodes in the network are not connected or based on the definition of the edges in the present paper in each category of roles just one node is anomalous. There will be different star communities in the modeled network. The clustering coefficient of all network nodes is zero since there is no connection between one-step neighbors of nodes. Therefore, the proposed approach will consider all of them as abnormal nodes. However, in this case, i.e., star networks, closeness centrality will have an effective result since all the central nodes in the stars have a connection with other nodes, therefore has the minimum shortest path to all other nodes in the network and subsequently have the high clustering coefficient, which means that these are abnormal nodes. It is evident that what the authors of [28] proposed, finding stars in the networks, will be beneficial in anomaly detection problems in such cases.

According to the concept of edges formed by executing different tasks, the network proposed in this study can be called a conflicts network. In contrast to the conflicts network, a similarity network can be drawn, in which an edge is drawn between both types of users with the same role who do the same things.

In this type of network, in the presence of anomalies, normal nodes will have the most connections, but anomalous nodes are considered as lateral nodes, so it can be said that anomalous nodes have a low clustering coefficient and also, unlike the conflicts network, will have a lower closeness centrality. Furthermore, normal nodes will have a higher closeness centrality and lower clustering coefficient due to more connections with other nodes. Also, in the similarity network, both criteria will have high efficiency in star networks.

Besides, if similarity networks are plotted, it is possible to discard the assumption of systems with a specific survey structure and generalize the research to any system. However, false-positive values due to rare norm behaviors, which will be considered anomalies in this method, will be more. This method will be suggested as future work to determine its effectiveness by performing more accurate evaluations.

6 Conclusions and future work

This work presents an approach to identify anomalies in business process logs using social network analysis and process mining.

Data were collected by extracting possible and authorized traces from a simulated process model of a hypothetical e-commerce website and adding different anomaly types to these traces.

The conclusions of this paper, which could be considered as its main scientific contributions, are:

- Clustering coefficient and closeness centrality were introduced as social network metrics to detect anomalies.
- The proposed approach considered the process instances anomalous if they had a low clustering coefficient or a high closeness centrality.
- Results showed that closeness centrality has the highest functionality in high and low anomaly rates between these metrics, although it suffers from a high computational cost.

The main idea was to prove that social network analysis could detect anomalies in process-aware information systems from organizational and control flow perspectives; despite other studies focusing on execution flow of process instances, it can also detect process instances that follow a correct execution flow but deals with unauthorized roles or users.

The paper's key contribution is mainly the concluded anomaly detection rules and is reflected in the fact that the proposed approach and concluded rules could be used to identify anomalies with more accuracy in different business process logs, especially in e-commerce system logs containing users with different roles and permissions.

The presented anomaly detection approach is limited to the control flow and organizational perspectives. For example, an anomaly may follow a standard flow but produce anomalous data (e.g., huge money). Therefore, data perspective should also be considered to provide more accuracy. Because this study focuses on process-aware systems with a known navigation structure, the proposed approach may not perform well in a dynamic environment without a predefined execution flow, e.g., healthcare systems. Therefore, a similarity network was suggested to overcome this problem in future work.

Only two social network analysis metrics, the clustering coefficient, and closeness centrality, were used to detect anomalies, although examining other metrics such as cliques might effectively detect anomalies. Similar to other previous studies, artificial logs were used to evaluate the approach presented in this study. Therefore, to ensure the proposed approach's performance in real life, it is necessary to use the real logs corresponding to the present research's assumptions to investigate the results.

Funding No funding was received for conducting this study

Availability of data and material Materials and data are available on request

Declaration

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Code availability Code is available on request

References

1. Bezerra, F., Wainer, J., van der Aalst, W.M.P.: Anomaly detection using process mining. In: Enterprise, business-process and information systems modeling, pp. 149–161. Springer (2009)
2. Marlon, D., Wil van der, A., Arthur Ter, H.: Process aware information systems, vol. 1. Wiley Online Library, Hoboken (2005)
3. Bezerra, F., Wainer, J.: Fraud detection in process aware systems. Int. J. Bus. Process. Integr. Manag. **5**(2), 121–129 (2011)

4. Bezerra, F., Wainer, J., et al.: Anomaly detection algorithms in business process logs. In Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS), volume AIDSS, Barcelona, Spain, pp 11–18 (2008)
5. Wil MP Van der, A., Ana Karla A de, M.: Process mining and security: Detecting anomalous process executions and checking process conformance. Elect. Notes Theor. Comput. Sci. **121**, 3–21 (2005)
6. Bezerra, F., Wainer, J.: Anomaly detection algorithms in logs of process aware systems. In: Proceedings of the 2008 ACM symposium on Applied computing, pp. 951–952 (2008)
7. Bezerra, F., Wainer, J.: Towards detecting fraudulent executions in business process aware systems. In: WfPM 2007 Workshop on Workflows and Process Management (2007)
8. Alves De Medeiros, A.K., Weijters, A.J.M.M., Van der Aalst, W.M.P.: Genetic process mining: A basic approach and its challenges. In: International Conference on Business Process Management, pp. 203–215. Springer (2005)
9. de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Min. Knowl. Discov. **14**(2), 245–304 (2007)
10. Bezerra, F., Wainer, J.: Algorithms for anomaly detection of traces in logs of process aware information systems. Inf. Syst. **38**(1), 33–44 (2013)
11. Zhao, W., Zhao, X.: Process mining from the organizational perspective. In: Foundations of intelligent systems, pp. 701–708. Springer (2014)
12. Schönig, S., Cabanillas, C., Jablonski, S., Mendling, J.: Mining the organisational perspective in agile business processes. In: Enterprise, Business-Process and Information Systems Modeling, pp. 37–52. Springer (2015)
13. Van, D.: Process mining discovery, conformance and enhancement of business processes. Springer, Heidelberg (2011)
14. Eckleder, A., Freytag, T.: Woped a tool for teaching, analyzing and visualizing workflow nets. Petri Net Newslett. **75**, 3–8 (2008)
15. Shazia, T., Fabiola SF, P., Sofia, F., João, G.: Social network analysis: An overview. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **8**(5), e1256 (2018)
16. Rozinat, A., Van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Inf. Syst. **33**(1), 64–95 (2008)
17. Rozinat, A., Van der Aalst, W.M.P.: Conformance testing: Measuring the fit and appropriateness of event logs and process models. In: International conference on business process management, pp. 163–176. Springer (2005)
18. Nolle, T., Seeliger, A., Mühlhäuser, M.: Unsupervised anomaly detection in noisy business process event logs using denoising autoencoders. In: International conference on discovery science, pp. 442–456. Springer (2016)
19. Rogge-Solti, A., Kasneci, G.: Temporal anomaly detection in business processes. In: International Conference on Business Process Management, pp. 234–249. Springer (2014)
20. Böhmer, K., Rinderle-Ma, S.: Multi instance anomaly detection in business process executions. In: International Conference on Business Process Management, pp. 77–93. Springer (2017)
21. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Analyzing business process anomalies using autoencoders. Mach. Learn. **107**(11), 1875–1893 (2018)
22. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Binet: Multi-perspective business process anomaly classification. Inform. Syst. 101458 (2019)
23. Nolle, T., Seeliger, A., Mühlhäuser, M.: Binet: multivariate business process anomaly detection using deep learning. In: International Conference on Business Process Management, pp. 271–287. Springer (2018)
24. Marques Tavares, G., Barbon, S.: Analysis of language inspired trace representation for anomaly detection. In: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium, pp. 296–308. Springer (2020)
25. Nascimento, M.C.V.: Community detection in networks via a spectral heuristic based on the clustering coefficient. Discret. Appl. Math. **176**, 89–99 (2014)
26. Salavati, C., Abdollahpouri, A., Manbari, Z.: Ranking nodes in complex networks based on local structure and improving closeness centrality. Neurocomputing **336**, 36–45 (2019)
27. Baesens, B., Van Vlasselaer, V., Verbeke, W.: Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection. Wiley, Hoboken (2015)
28. Akoglu, L., McGlohon, M., Faloutsos, C.: Oddball: Spotting anomalies in weighted graphs. In: Pacific-Asia conference on knowledge discovery and data mining, pp. 410–421. Springer (2010)