© 2018 IJSRSET | Volume 4 | Issue 9 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Segregate Malicious Behavior and Profit Maximization In Cross Organisational Business Process-As-A-Service (COBPAAS) Event Logs

Mrs.G.Sasikala*[1], V. Ramesh[2]

[1]Assistant Professor, PG & Research Department of Computer Science and Applications, Adhiparasakthi College of Arts and Science (Autonomous), G.B.Nagar, Kalavai, Vellore, Tamil Nadu, India

[2]M.Phil (CS) Research Scholar, PG & Research Department of Computer Science and Applications, Adhiparasakthi College of Arts and Science (Autonomous), G.B.Nagar, Kalavai, Vellore, Tamil Nadu, India

ABSTRACT

As of late, web based shopping incorporating third-party payment platforms (TPPs) acquaints new security challenges due with complex communications between Application Programming Interfaces (APIs) of Merchants and TPPs. Malicious users may misuse security vulnerabilities by calling APIs in a subjective request or assuming different roles. To manage the security issue in the beginning times of framework improvement, this paper introduces a formal strategy for displaying and check of web based shopping business forms with malicious behavior patterns method considered in light of Petri nets. We propose a formal model called E-commerce Business Process Net to display a typical internet shopping business process that speak to expected capacities, and malicious behavior patterns representing to a potential attack that violates the security goals at the requirement examination stage. We build up an organized procedure that applies formal strategies while coordinating users through determining value-based prerequisites and choosing configurable highlights. The Binary Decision Diagram (BDD) analysis is then used to confirm that chose configurable features don't damage any limitations. At last, demonstrate checking is connected to confirm the arranged administration against the transactional requirement set we analyze whether a web based shopping business process is impervious to the known malicious behavior patterns. Process mining goes for transforming such event data into significant, noteworthy knowledge, so process execution or consistence issues can be recognized and rectified. Diverse process mining systems are accessible. These incorporate methods for mechanized process disclosure, conformance checking, execution mining and process variation analysis.

Keywords : Verification, Transactional Requirements, Model Checking, Business Process Management, Process Mining, Infrequent Behaviour.

I. INTRODUCTION

ONLINE SHOPPING with a third-party payment plat- form (TPP) has become the new frontier for doing business nowadays, and become increasingly popular in the global economy as more and more business transactions are conducted over the web. Its daily volume is sizable and continues to grow at a rapid pace. It can be successful only if the general public trusts online trading systems. Even if the volume of transactions and the number of users are growing constantly, it appears that many users have not accepted online shopping as the main trading channel. Research has shown that insufficient trust represents a key reason for users to avoid making businesses over the Internet. However, online shopping systems are complex and difficult to be correctly designed. Design-level vulnerabilities are

indeed a major source of security issues. For example, in Microsoft's "security push," about 50% of the security issues had been detected due to design-level flaws. As a distributed application on the web, online shopping business processes are more complex and loosely coupled. Recently, online shopping systems have increasingly integrated TPPs such that a complete online shopping business process has three parties: Shopper, Merchant, and TPP. Their respective business processes construct the entire online shopping process. This integration introduces new security challenges due to the complexity for an application to coordinate its internal states with those of component services and web clients across the Internet. The complex linkages of control and data flows in online shopping business processes may produce very serious problems, e.g., the violation of transaction properties and huge losses of users, and the loose coupling of the different party's results in the lack of mutual understanding among their business processes. Consequently, malicious users can call Application Programming Interfaces (APIs) in an arbitrary order by some special means, and even play several roles to achieve their malicious attack purposes.

As the new security challenges of online shopping business processes are at the application-level, the sufficient protection of online shopping systems from attacks is beyond the capabilities of network-level and operating system-level security approaches, e.g., cryptography, firewall, and intrusion detection. They lack knowledge of application semantics and cannot meet the needs from today's distributed online shopping systems. Engineering software security is essential, and it is important to incorporate the use of assurance techniques throughout development and operation. Thus, one needs to exploit the methodologies to verify the trustworthiness of online shopping business processes. Nevertheless, rigorous and formal methodologies for online shopping business process design remain unavailable. The most pressing challenge is how to verify the trustworthiness of an online shopping process in the conceptual modeling phase such that potential security issues can be addressed before the implementation of an online shopping system. At the requirement analysis and design levels, we need to explicitly identify whether the online shopping systems can be resistant to the possibly malicious behavior patterns. Note that such patterns can be found in many public threat libraries

We focus on the online shopping business process that consists of three parties: Shopper, Merchant and TPP, and verifies it by formal methods at the conceptual modeling phase from the application-level viewpoint. The basic idea is: initially, to construct the functional model according to design specification; then, to choose one malicious behavior pattern and translate it to a malicious behavior model according to the functional model; next, to synthesize them for establishing an online shopping business process able to handle such malicious behavior scenario; at last, verify it and determine whether the online shopping business process can withstand such an malicious behavior pattern. The framework is shown in Fig. 1. Note that we need to verify the designed system subject to all malicious behavior patterns stored in the library according to the above process. After a malicious behavior pattern has been verified, choose another from the library and repeat the process until all are verified.

This work concentrates on the following respects:
a) E-commerce Business Process Net (EBPN). We extend and modify a traditional Petri net to an EBPN by integrating both data and control flows to reflect the data and state information. Thus, data errors and non determinacy of data states during a trading process can be easily described with it. Data information is added, and data states are proposed to reflect the changes of transaction states.
b) Modeling methods of a functional model and malicious behavior model. Based on EBPN features, we propose the methods to build up a functional

model of an online shopping business process and those of malicious behavior patterns, and then, synthesize them to obtain a complete malicious behavior scenario.

c) Formal verification methods. First, we obtain the malicious behavior from the viewpoint of malicious clients according to the malicious behavior model. Next, we analyze the composed EBPN with a malicious behavior sequence, and derive the relation graph of the malicious behavior sequence and legal transitions. At last, by using EBPN's dynamic properties, we determine whether an online shopping business process can withstand malicious behavior patterns.

Using the proposed methodology, designers can identify problems early in a design process and correct them before the system realization, and avoid losses caused by their solution procedure. Thus, one is able to generate more reliable systems faster and at lower costs with the proposed method.

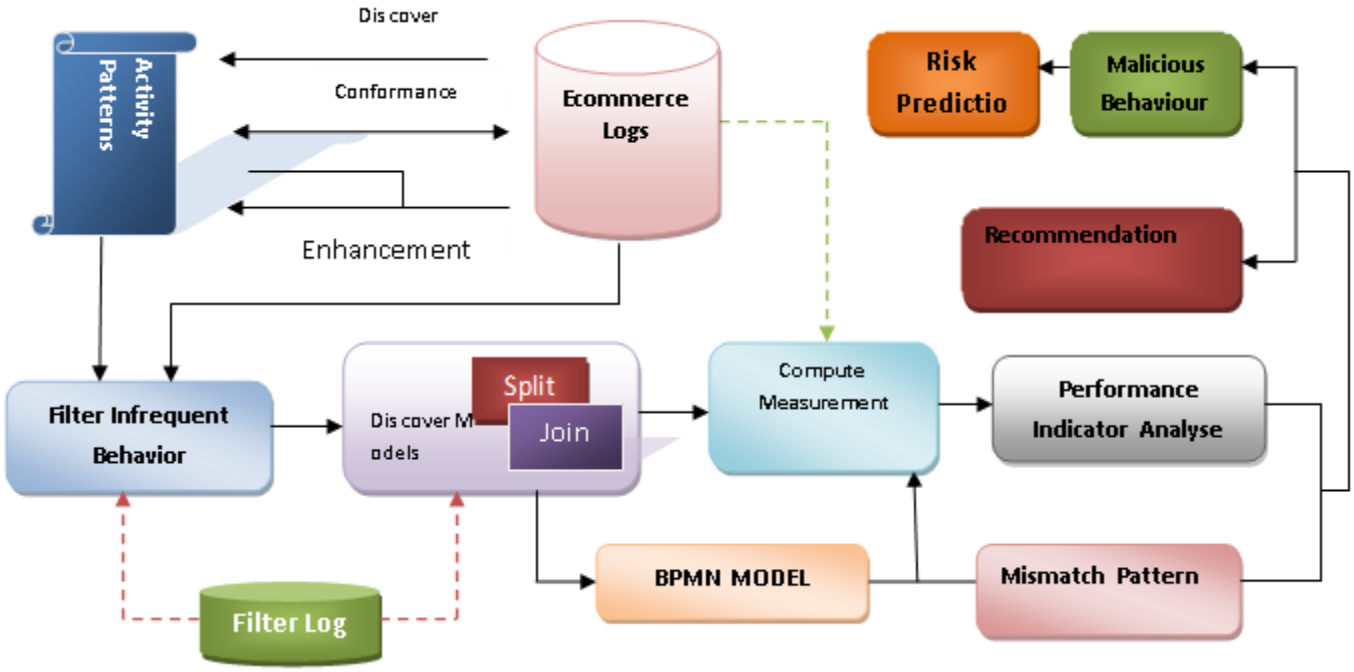

Figure 1. Framework of the proposed solution

II. BACKGROUND

In cross- organizational process mining condition, there is a need to adjust procedures of various associations. In the investigation of Dijkman [13], a gathering of examples to portray visit confounds between the comparative procedure models are displayed. Inside the extent of this investigation, the related confuse designs are characterized in contemplate [13] as takes after:

Skipped Activity An action exists in one process however no proportional action is found in alternate process.

Refined Activity An action exists in one process in any case, as a proportionate, an accumulation of exercises are existing in alternate procedure to accomplish a similar undertaking..

Activities at Different Moments in Processes Set of exercises are attempted with various requests in various procedures.

Different Conditions for Occurrence Set of dependencies are same for two processes; however, occurrence condition is different.

Different Dependencies set of exercises vary in various associations.

Additional Dependencies This pattern is a special case of different dependencies where one set of activities includes the other and results with additional dependencies.

III. METHODOOGY

The approach planned during this examination includes of 4 elementary stages unreal in Figure one. Right off the bat, in method Model Mining, method models area unit aloof from occasion logs for each association with a shopper indicated commotion edge. Also, in Performance Indicator Analysis, occasion logs area unit replayed on method models conjointly, execution markers area unit discovered for each association at that time utilizing these markers, associations area unit clustered seeable of however well they're operating. Thirdly, in mate Pattern Analysis, contrasts between method models of associations area unit freed with entrenched crisscross examples. At long last, in Recommendation Generation, utilizing the execution marker clustering's conjointly, contrasts between method models; a meeting of proposals for every association is made.

3.1 Business Procedure Administration

In this module revelation, wherever the goal is to consequently take away a procedure show from the data, this could originate once in a very whereas voyaged pathways that messiness the procedure demonstrates. This paper introduces a mechanized strategy to the expulsion of occasional conduct from occasion logs. The planned system is assessed well and it's incontestible that its application in conjunction with sure current procedure revealing calculations basically enhances the character of the found procedure models which it scales well to expansive datasets.

3.2 Process Mining

Process mining intends to disentangle important method learning from occasion logs of IT frameworks that area unit frequently accessible in modern associations. One region of enthusiasm for the additional in depth field of method mining is that of procedure revelation that is disturbed regarding the logical thinking of method models from occasion logs. when a while, a scope of calculations are recommended that address this issue.

3.3 Rare Behavior

In this module to restrain these negative impacts, method occasion logs area unit commonly subjected to a pre-handling stage wherever they're physically clean from commotion [3]. This but may be a testing and tedious enterprise, with no assurance on the adequacy of the result, notably with regards to very large logs showing advanced method conduct.

3.4 Visit Behavior

This module proposes the principal flourishing system for winnowing through commotion from method occasion logs. The oddity of the tactic rests upon the choice of demonstrating the rare log separating issue as a machine. This approach empowers the identification of rare method conduct at a fine grain level that prompts the evacuation of individual occasions rather than whole follows (i.e. successions of occasions) from the log.

3.5 Performance Indicator Analysis

Performance indicator analysis prepare concentrates on ascertaining and breaking down the execution esteems utilizing the occasion logs and mined method models. This stage includes of for the foremost half 2 stages as an) arrangement and count of execution markers; and b) bunching of associations in light-weight of their execution esteems. Keeping in mind the tip goal to assess the execution of associate association seeable of their procedure models and past exercises; there are varied markers in time activity, value activity and usage [3]. Be that because it might, during this investigation, method connected execution esteems area unit thought-about since contrasts within the method models area unit examined within the following stages. to the present

purpose, the incidental to execution markers area unit ascertained:

3.6 Average Time Between Activities this is often be a basic but capable execution metric for associations since it can yield the conventional time to end one trip seeable of a starting stage. From the execution purpose of read, associations got to limit traditional time between exercises to expand their turnout [4]. This concept will be characterized as takes after:

Definition 1. Average time between activity A and B in organization i is

Where

$$AvgTime^{i}_{A\rightarrow B} = \frac{\sum_{case\ c\in Eventlog_i} Time\ Between_c\ (A,B)}{|Occurences_{Event\ log_i}(A,B)}$$

Where 1. $Time\ Between_c\ (A,B)$ = $EndTime_c(B) - StartTime_c(A)$

2. $StartTime_c$ (A) is start time of activity A in case c,

3. $EndTime_c$ (B) is end time of activity B in case c,

4. $|Occurences_{EventLogi}$ (A, B)| is number of occurrences of activity A followed by B in Event Log_i.

3.7 Risk Propagation Algorithm

The risk propagation algorithm cascades this information across all currently running instances. To do so, the risk propagation algorithm builds on the PING and estimates the eventuation probability (i.e., the probability that the risk condition of the other instances also evaluates to true) of the detected risk in other instances using similarities, inspired by the signal/collect programming model. The risk propagation algorithm follows a two-phase approach, i.e., initial propagation and re-propagation. If propagation is successful, we refer to the state of the respective instance as "at risk."

Mismatch Pattern Analysis

In order to learn from other organizations, it is necessary to spot the diff erences between process models of diff erent organizations. In this phase, diff erences between process models will be revealed by the mismatch patterns which are defined by Dijkman. Since performance indicators are calculated based on a starting and ending point in the process model, the same approach is applied to locate mismatch patterns. In other words, diff erences of process models are located through a starting activity to an ending activity. With this aim, each mismatch pattern and its analyzers are defined by extending the following definitions. For each organization, mismatch pattern analyzers are pipelined and mismatch patterns are stored for further analysis

Generating Suggestions/Recommendations for Performance Improvement

Recommendation generation stage in the methodology is the final and core stage where all information retrieved from the event logs until now is utilized. In this study, idea of recommendation is based on providing a set of mismatch patterns for each organization so that they can enhance their processes. These mismatch patterns are generated by comparing the process models of other organizations, particularly those that are performing better in terms of their performance indicator values. Recommendation idea and recommendation generation function is defined as following:

IV. ALGORITHMS

Algorithm 1: Recommendation Generation

Input: O organization, C Cluster Analysis Data, P performance difference threshold

Output: Recommendations a set of recommendations

1 Recommendations ← {}

2 i ← C(Assignments(O))

3 for Centroid ∈ C(ClusterCentroidsi) do

4 for Centroid' ∈ C(ClusterCentroidsj) i≠ j do

5 if Centroid (Astart) = Centroid' (Astart) & Centroid (Aend) = Centroid' (Aend) then

6 if (|Centroid (Value) – Centroid' (Value)| ÷ Centroid (Value)) ≥ P then

7 Astart ← Centroid (Astart)

8 Aend ← Centroid (Aend)
9 MismatchPatterns ← {}
10 for O ' ∈ C(Assignments(j)) do
11 Mismatch Patterns ← MismatchPatternAnalysis (O, O', Astart, Aend)
12 Recommendations ← Recommendation (O, Astart, Aend, MismatchPatterns)
13 return Recommendations

V. RESULTS

5.1 SCREENSHOT

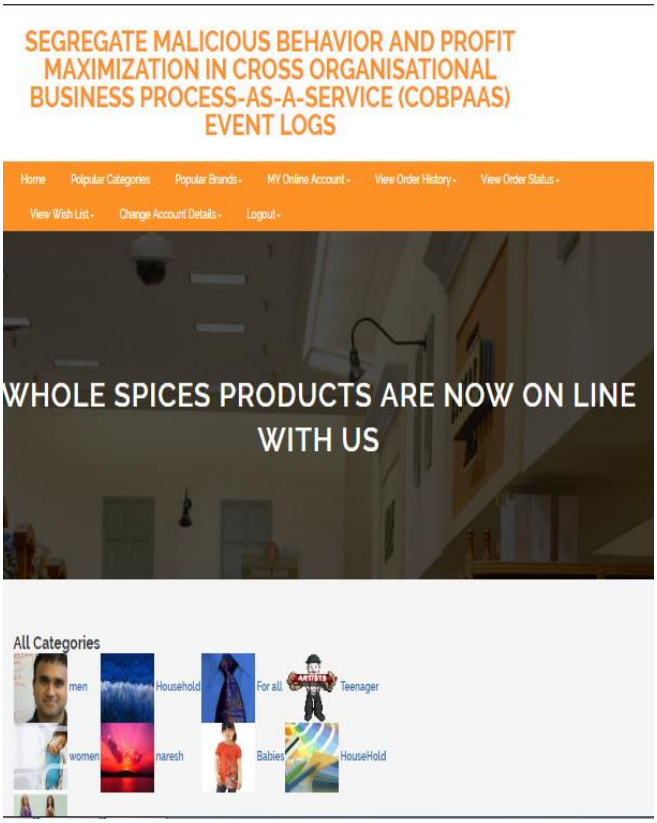

Figure 2. Categories list

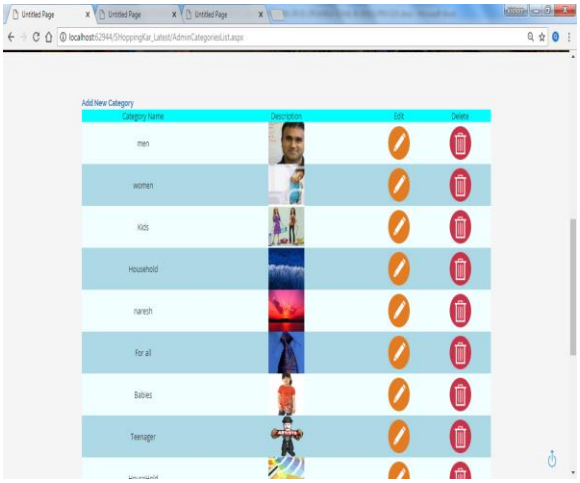

Figure 3. Adding new category

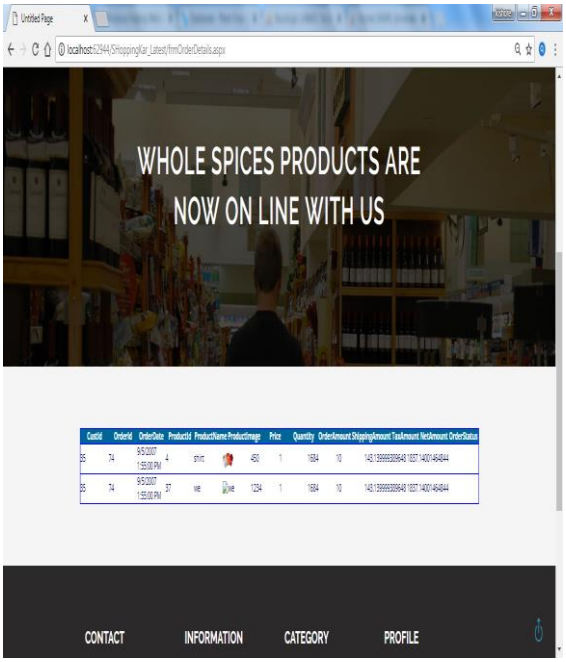

Figure 4. Order details from selected category

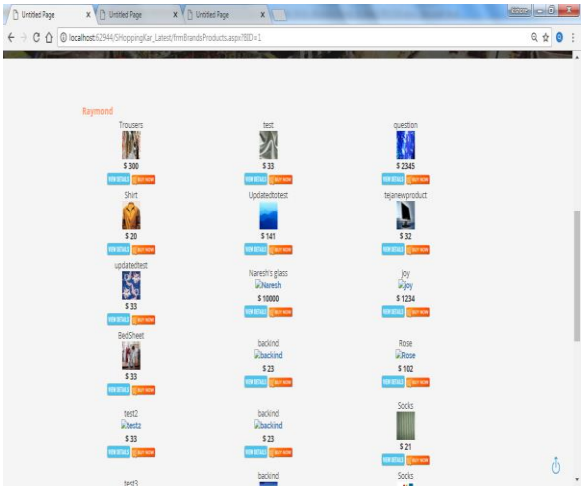

Figure 5. Product details from selected brand

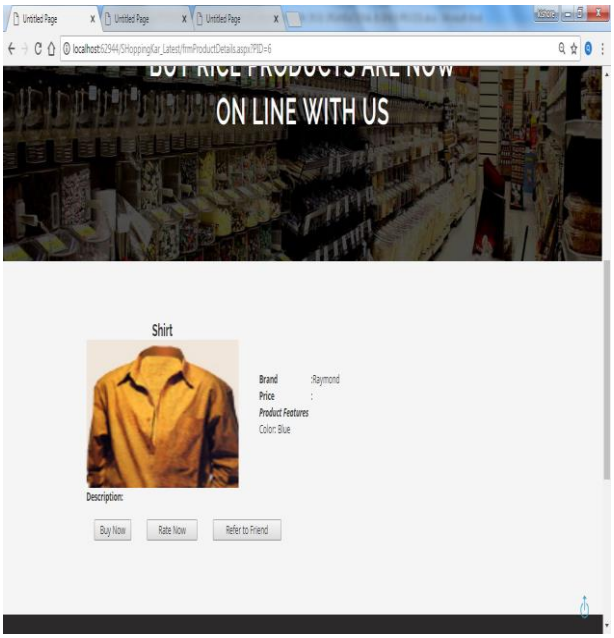

Figure 6. Selected product details

High Frequently Moved Products

Month	ProductId	Name	ProductPrice	Quantity	Total
August	3	Shirts	100	2	200
August	4	shirt	250	1	250
August	5	Trousers	300	1	300
August	3	Shirts	100	1	100
August	4	shirt	250	1	250
August	8	Shirt	750	1	750
August	7	BananaShirts	500	1	500
August	8	Shirt	750	1	750
September	6	Shirt	20	1	20
September	6	Shirt	20	1	20

Figure 7. Frequently moved product details

Low Frequently Moved Products

Month	ProductId	Name	ProductPrice	Quantity	Total
August	3	Shirts	100	2	200
August	4	shirt	250	1	250
August	5	Trousers	300	1	300
August	3	Shirts	100	1	100
August	4	shirt	250	1	250
August	8	Shirt	750	1	750
August	7	BananaShirts	500	1	500
August	8	Shirt	750	1	750
September	6	Shirt	20	1	20
September	6	Shirt	20	1	20

Figure 8 . Less frequently moved product details

VI. CONCLUSION

In this paper we introduced a method for the programmed removal of infrequent behavior from process execution logs. The core idea is to utilize inconsistent direct follows dependencies between event labels as an intermediary for infrequent behavior. These dependencies are detected and removed from an automaton built from the event log, and then the original log is updated accordingly, by removing individual events using alignment-based replay. New approach is proposed and tested for generating recommendations using cross-organizational process mining for process performance improvement. Cross-organizational process mining is applied with the idea of unsupervised learning where predictor variables related to performances of organizations are used in an environment where processes are executed on several organizations. Results show that it is possible to use cross-organizational process mining and mismatch patterns for performance improvement recommendations and able to detect risks earlier. Verification of online shopping business processes against some specific malicious behavior patterns

VII. FUTURE STUDY

For the approach proposed in this study, the following issues can be listed as pointers to future work:

In the process mining stage, instead of *Inductive Miner*, new techniques can be used which can mine complex process models with higher appropriateness levels while keeping the current high fitness values.

In the performance indicator analysis stage, new indicators can be defined based on the business environment, event log attributes and user needs. For instance, personnel and resource allocation indicators can be included as well as cost dimension.

For mismatch pattern analysis, new and business oriented mismatch patterns can be included in the analysis. In addition analyzers can fail when there are loops in the process models in current implementations, therefore more robust implementations for process models with loops can be developed in the future.

For the generated recommendations, quality for business environment is not assessed within the scope of this study. However, when any feedback from a domain expert or BPM people is provided, the learning approach can be converted to semi-supervised learning from unsupervised learning.

VIII. REFERENCES

[1]. A. Adriansyah. Aligning Observed and Modeled Behaviour. PhD thesis, Technische Universiteit Eindhoven, 2014.

[2]. A. Adriansyah, J. Munoz-Gama, J. Carmona, B.F. van Dongen, and W.M.P. van der Aalst. Alignment based precision checking. In Proc. of BPM Workshops, pages 137–149, 2012.

[3]. A. Adriansyah, B.F. van Dongen, and W.M.P. van der Aalst. Conformance checking using cost-based fitness analysis. In Proc. of EDOC, pages 55–64, 2011.

[4]. C.C. Aggarwal. Outlier Analysis. Springer, 2013.

[5]. S. Basu and M. Meckesheimer. Automatic outlier detection for time series: an application to sensor data. KAIS, 11(2):137–154, 2006.

[6]. S. Budalakoti, A.N. Srivastava, and M.E. Otey. Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. IEEE TSMCS, 39(1):101–113, Jan 2009.

[7]. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. IEEE TKDE, 24(5):823–839, May 2012.

[8]. R. Conforti, M. Dumas, L. Garc´ıa-Banuelos, and M. La Rosa. Beyond ˜ tasks and gateways: Discovering BPMN models with subprocesses, boundary events and activity markers. In Proc. of BPM, pages 101–117, 2014.

[9]. K. Das, J. Schneider, and D.B. Neill. Anomaly pattern detection in categorical datasets. In Proc. of ACM SIGKDD, pages 169–176, 2008.

[10]. G. Florez-Larrahondo, S.M. Bridges, and R. Vaughn. Efficient modelling of discrete events for anomaly detection using hidden markov models. In Proc. of ISC, pages 506–514, 2005.

[11]. C.W. Gunther and W.M.P. van der Aalst. Fuzzy mining - adaptive process ¨ simplification based on multi-perspective metrics. In Proc. of BPM, pages 328–343, 2007.

[12]. M. Gupta, C.C. Aggarwal, and J. Han. Finding top-k shortest path distance changes in an evolutionary network. In Proc. of SSTD, pages 130–148. Springer, 2011.

[13]. M. Gupta, J. Gao, C.C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. IEEE TKDE, 26(9):2250–2267, 2014.

[14]. M. Gupta, A. Mallya, S. Roy, J.H.D. Cho, and J. Han. Local Learning for Mining Outlier Subgraphs from Network Datasets, pages 73–81. 2014.

[15]. R. Gwadera, M.J. Atallah, and W. Szpankowski. Reliable detection of episodes in event sequences. KAIS, 7(4):415–437, May 2005.

[16]. S.A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. J. Comput. Secur., 6(3):151–180, August 1998.

[17]. R.M. Karp. Reducibility among combinatorial problems. In Proc. Of CCC, pages 85–103. Springer US, 1972.

[18]. E. Keogh, J. Lin, S.-H. Lee, and H. van Herle. Finding the most unusual time series subsequence: algorithms and applications. KAIS, 11(1):1–27, 2006.

[19]. E. Keogh, S. Lonardi, and B. Chiu. Finding surprising patterns in a time series database in linear time and space. In Proc. of ACM SIGKDD, pages 550–556, 2002.