Statistical verification of process conformance based on log equality test

Hyerim Bae[1], Sung-hyun Sim[1], Yulim Choi[1], and Ling Liu[2]

[1] Department of Industrial Engineering
Pusan National University
Busan, Korea
{hrbae, ssh, flamethrower}@pusan.ac.kr

[2] College of Computing
Georgia Institute of Technology
Atlanta, USA
lingliu@cc.gatech.edu

***Abstract*— In Big data and IoT environments, a huge-sized data is created as the result of process execution, some of which are generated by sensors. The main issue of such application has been to analyze the data in order to suggest enhancements to the process. Evaluation of the conformance of process models is of great importance in this regard. For this purpose, previous studies in process mining approach suggested conformance checking by measuring fitness that uses token replay and node-arc relations based on Petri net. However, fitness thus far has not considered statistical significance, but just offers a numeric ratio. We herein propose a statistical verification based on the Kolmogorov-Smirnov test to judge whether two different log data sets are following the same process model. Our method can also judge that a set of event log data is following a process model by playing out the model and generating event log data from the model. We also propose a new concept of 'Maximum Confidence Dependency' to solve the problem of the trade-off between model abstraction and process conformance. We expect that our method can be widely used in many applications which deal with business process enhancement by analyzing process model and execution log.**

Keywords—Process analysis; Process Conformance; Kolmogorov-Smirnov test; Process Enhancement; Maximum Confidence Dependency

I. Introduction

Process management is a critical issue for enterprise management, as it has a direct effect on an enterprise's competitive power. These days, multiple suppliers and customers make process models complex, which in turn, the enterprise struggle to satisfy the needs of handling complex process models. In order to cope with such complexity, businesses within the service and manufacturing industry develop their own process management methodology. This tendency has accelerated the introduction and utilization of information systems (e.g. IoT, cloud computing, etc.) that collect large-scale and complex datasets. The huge amounts of data that information system managers are obliged to analyze require that process models be simplified. But simplification reduces process conformance, which means that the resultant model explains only a small portion of real process executions [1]. Indeed, there is a trade-off between process abstraction and process conformance. In the process mining [2] field, this decision-making process is highly correlated with process model quality and is typically expressed as conformance checking [1]. When we try to discover a process model with high conformance value from a large scale data, the result turns out a spaghetti process, which obscures intuitive understanding. The easiest approach to the analysis of such data is to simplify the process model for effective representation by controlling the size of the activity or path, which, in turn, may lose conformance value. Another issue is to verify how precisely the discovered model reflects event log data. For this purpose, process mining research has adopted the concept of fitness measure [3]. However, the concept of fitness does not provide any statistical significance on how well a process model conforms to the event log. Therefore, we require a measure to find non-complex process model with high conformance having a certain level of statistical significance.

In the business process management area, there are two types of test.

- Conformance Test (CT): This type of test is carried out between process model and event log to judge if a process model conforms with the log data. This test can be used for evaluation of discovery algorithm and evaluation of process accordance to organization standard.
- Equality Test (ET): This type of test is carried out between two event logs to judge if they are following the same process model. This test can be used to comparing two organizations for benchmarking and BPR purposes.

In process management environments illustrated in Fig. 1, after event log is prepared, a process model (de facto) can be generated by using process mining algorithm. In this setting, we can use CT for conformance checking how well the mining algorithm discovered a process model. If we have standard process model (de jure), we can also use CT for checking how well processes execution is following the standard way. ET can be used for checking if the two sets of data are following the same process model.

The main purpose of this paper is to develop a statistical equality test that provides insight into the process model's abstraction level and its reflection of the original dataset. We use the same event log data schema as typically employed by process mining methods. Event log data can be easily obtained from process execution since many enterprises already have their information system to support their process management in IoT environments. The statistical verification can be used in many comparison scenarios and all of the scenarios can be based on one type of test, ET. This is because that comparison between process model and event log can be transformed into a comparison between event logs by playing out the model and generating event log artificially. Fig. 1 shows possible types of the test using process model and event log.

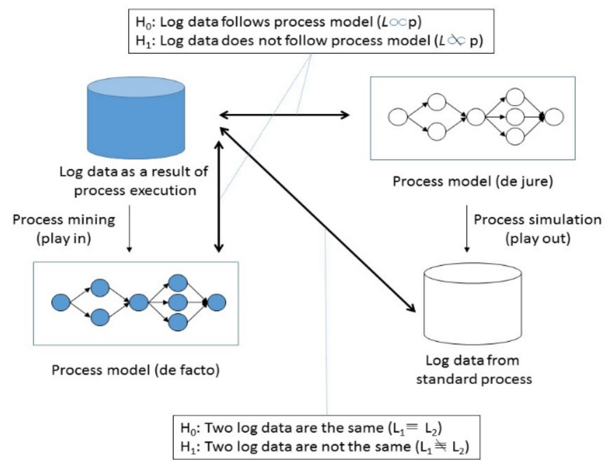


Fig. 1. Application of equality test between process model and event log.

In this paper, to conduct a conformance test by using log data format, we chose to apply a non-parametric statistical equality test, namely the Kolmogorov-Smirnov (K-S) Test [9], to evaluate a process model's conformance with the original dataset. The requirement that the test be non-parametric reflects the fact that process mining data does not assume a given probability and contains mostly text-type variables [7]. For the conformance testing of a process model, we use artificial log data generated by playing out the model. The conformance evaluation entails comparison of the played-out dataset with the original version. This proceeds in four stages. First: Discover the process model. Second: Play-out the log from the process model. Third: Perform an equality test between the original log and the comparative log (extracted from the model). Fourth: Analyze the statistical significance. By this procedure, we can develop a better conformance-checking method according to changes of dataset size and/or complexity.

II. Backgrounds

A. Event Log and Process Discovery

One of the most popular areas of handling both process model and event log is process mining. Process mining techniques are used to extract meaningful knowledge from an event log that is generated as the result of the execution of information systems or the operation of machinery. Among process mining techniques, this paper introduces process discovery methods, which utilize the concepts of event, trace, and event log. We provide definitions of these objects based on the definition of previous research work [2].

Definition 1 (Event, trace, and event log) An event *e* is a unit record of occurrence which has attributes of case, event name, timestamp, and originator. A trace σ is a finite sequence of events, which belong to a single case, such that each event appears at most once. Event log *L* consists of a series of events such that each event appears at most once in the entire log. *L* can be transformed into a set of traces by using case ID.

Event Log Data

Case (Container ID)	Event Name (Activity ID)	Timestamp	Originator (Equipment ID)
C01	*QuayJobDischarge*	2012-05-22 05:22:31 KST	GC04
C01	*YardJobDischarge*	2012-05-22 05:25:59 KST	Y11
C01	*Shuffling*	2012-05-22 18:48:42 KST	Y11
C01	*YardJobGateOut*	2012-05-22 17:48:02 KST	Y11
C02	*QuayJobDischarge*	2012-05-22 05:24:11 KST	GC02
C02	*YardJobDischarge*	2012-05-22 05:34:51 KST	Y51
C02	*ReeferPlugging*	2012-05-22 05:40:00 KST	SYS
C02	*ReeferPlugging*	2012-05-22 11:23:00 KST	SYS
C02	*YardJobGateOut*	2012-05-24 00:38:03 KST	Y51
C03	*QuayJobDischarge*	2012-05-24 00:43:59 KST	GC01
C03	*YardJobDischarge*	2012-05-24 19:18:11 KST	Y51
C03	*YardJobGateOut*	2012-05-24 00:12:05 KST	Y51
C04	*YardJobLoad*	2012-05-24 06:32:24 KST	Y12
C05	*QuayJobLoad*	2012-05-24 06:40:15 KST	GC02
...	...	...	...

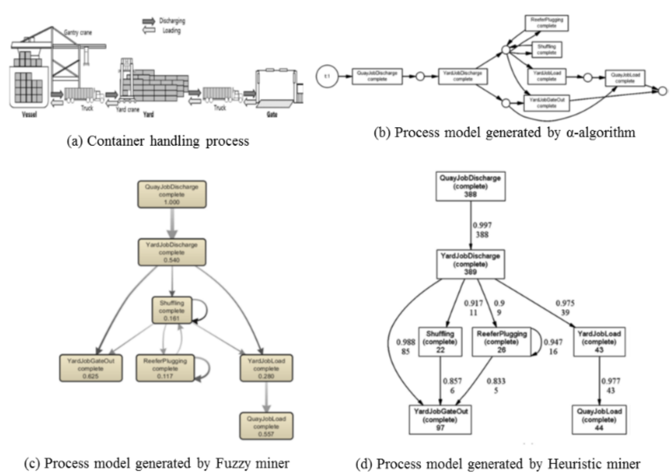
QD: QuayJobDischarge,
YD: YardJobDischarge,
SF: Shuffling,
YO: YardJobGateOut,
YI: YardJobGateIn
RP: ReeferPlugging,
QL: QuayJobLoading,
YL: YardJobLoading

Trace Data

Case ID	Trace
C01	{*QD, YD, SF, YO*}
C02	{*QD, YD, RP, RP, YO*}
C03	{*QD, YD, YO*}
C04	{YL, QL}
...	...

Fig. 2. Process execution log and trace set

Fig. 2 shows an example of process execution log, which is gathered from a container handling process. While a port is operating processes, a large amount of log data is stored in a database. Log data typically consists of a set of structured or semi-structured data including attributes of case, activity, time stamp, and originator. This data can be used for analyzing and enhancing the process.



(a) Container handling process

(b) Process model generated by α-algorithm

(c) Process model generated by Fuzzy miner

(d) Process model generated by Heuristic miner

Fig. 3. Process discovery algorithm.

Process discovery is a basic function of process mining since the other functions such as conformance checking and process enhancement are all based on the discovered process model. In process discovery method, an event log is transformed into a trace dataset as shown in the right bottom of Fig. 2. Using the trace data set, process models shown in Fig 3. (b), (c), and (d) can be discovered by α-algorithm, fuzzy miner, and heuristic miner respectively. From this figure, we know that different process models can be generated from the same event log depending on process discovery algorithm. In this sense, we encounter a necessity of evaluating the quality of process discovery algorithm.

B. Conformance Checking

In process mining, conformance index values are used to evaluate the discovered model. It is important to judge whether the process model derived from an event log is a proper model having sufficient conformance to the log. The existing indicators of process model quality are Fitness, Precision, and Generalization [12]. The most commonly utilized indicator of conformance checking is fitness. The fitness calculation equation [12] is

$$fitness(L, M) = 1 - \frac{fcost(L,M)}{move_L(L) + |L| * move_m(M)} \qquad (1)$$

where L denotes the event log, M the process model, $fcost(L,M)$ is the total alignment cost for L and M, $move_L(L)$ is the total cost of moving through the whole log without ever moving together with the model, and $move_m(M)$ is the total cost of making moves in model only. Previous research on the evaluation of process model conformance including fitness to the event log has entailed checking the degree of node-arc relation or using log replay through token play. These methods, however, are limited, in that they cannot show the statistical relevance of a process model's conformance to event log data.

III. STATISTICAL METHOD FOR PROCESS CONFORMANCE

A. Empirical Distribution Function

In statistics, the most popular estimator of a population's distribution function is the empirical distribution function [9]. Let $X_1, \cdots, X_n$ be an n probability sample from a population with distribution function $F(x)$, where x satisfies $-\infty \le x \le \infty$, and $F_n(x)$ is defined as

$$F_n(x) = \sum_{i=1}^{n} \frac{\varphi(x - X_i) + I\{x = X_i\}}{n}. \qquad (2)$$

$$\varphi_i(x) = \begin{cases} 1, & x \ge 0 \\ 0, & x < 0 \end{cases}, \; I(x = y) = \begin{cases} 1, & x = y \\ 0, & x \ne y \end{cases}$$

Equation (2) indicates that empirical distribution function $F_n(x)$ represents the ratio of the number of observations x smaller or equal to n [8, 10].

B. Kolmogorov-Smirnov Test Procedure

The K-S Test [8] evaluates goodness-of-fit based on the theorem that if two continuous observations' cumulative distribution function is equal, the observation's probability density function is also equal. According to [10], let $F(x)$ be the population distribution function and $F_0(x)$ be the specific distribution function. Then, the hypothesis test is

$$H_0\text{: } F(x) = F_0(x) \text{ for every } x \qquad (3)$$

$$H_1\text{: } F(x) \ne F_0(x) \text{ for some } x$$

The following is the K-S test procedure [8, 9].

Step 1. Let $X_1, \cdots, X_n$ be probabilistic samples of an empirical distribution function $F(x)$.

Step 2. Test statistics, $D = sup_x\{|F_0(x) - F(x)|\}$

Step 3. If $D > d\left(\frac{\alpha}{2}, n\right)$, the null hypothesis is rejected. In $d\left(\frac{\alpha}{2}, n\right)$, α represents the upper-bound 100α percentile, and n is the sample size.

C. Goodness-of-fit test for process model

Suppose that we are evaluating goodness-of-fit for a process model p to a log data L_0 ('original log'). We can generate artificial log L_1 from p by playing out it, which is called 'comparative log'. Then conformance checking problem can be a goodness-of-fit test between the two log sets. Therefore, from now we explain the procedure of equality test between two log data sets.

There are two log sets L_0 and L_1, and we can have a union trace set of distinct traces extracted from the two logs. If the union trace set L ($L_0 \cup L_1$) has n distinct traces ($\sigma_i, i = 1, \ldots n$), $Pr(\sigma_i)$ is the probability of the i^{th} trace in L_0, and $Pr'(\sigma_j)$ is the probability of the j^{th} trace in L_1. The trace occurrence probability vector (TOPV) for the original log and comparative log can represented as

$$E = \{E_1 = Pr(\sigma_1), \cdots, E_\mathrm{i} = Pr(\sigma_\mathrm{i}), \cdots, E_m = Pr(\sigma_n)\}^T \qquad (4)$$

$$H = \{H_1 = Pr'(\sigma_1), \cdots, H_\mathrm{j} = Pr'(\sigma_\mathrm{j}), \cdots, H_n = Pr'(\sigma_n)\}^T$$

Then the order statistics for E and H are $(E_{(1)}, \cdots, E_{(m)})$, and $\left(H_{(1)}, \cdots, H_{(n)}\right)$ respectively. We define $L_0(\sigma_i)$ as the cumulative trace occurrence probability for the empirical distribution of the original log data and $L_1(\sigma_i)$ as for the empirical distribution function of the comparative log, where x is a variable of trace occurrence probability. $L_0(\sigma_i)$ and $L_1(\sigma_i)$ can be obtained using the equation in the following.

$$L_0(\sigma_i) = \sum_{t=1}^{i} E_{(i)} \qquad (5)$$

$$L_1(\sigma_i) = \sum_{t=1}^{i} H_{(i)}$$

Then, the hypothesis test for process model goodness-of-fit is

$$H_0\text{: } L_0(\sigma_i) = L_1(\sigma_i) \text{ for every } i \qquad (7)$$

$$H_1: L_0(\sigma_i) \neq L_1(\sigma_i) \text{ for some } i$$

We define our test statistics as

$$D_C = sup_i\{|L_0(\sigma_i) - L_1(\sigma_i)|\}$$

In our test, a parameter α can be determined according to user preference. With value of α, we test D_c according to the K-S test statistics in the K-S table. Using these statistics, D_c is checked as to whether it follows the inequality $D_c > d\left(\frac{\alpha}{2}, m, n\right)$. Then, based on the result, we will decide whether to accept hypothesis H_1.

D. Maximum Confidence Abstraction Level

One of our purposes in this paper is to suggest a threshold value for finding most abstract process model with satisfying a certain level of conformance. In Heuristic Miner, the abstraction level of process is determined by adjusting dependency threshold. Therefore, we define maximum confidence dependency (MCD). The procedure for MCD determination is related to the analysis of the maximum abstraction level with statistical confidence.

Definition 2 (Maximum Confidence Dependency) Maximum Confidence Dependency (MCD) with the significance level of α is the largest dependency threshold value among the values of D_c, which satisfies

$$D_c \leq d\left(\frac{\alpha}{2}, m, n\right) \quad (9)$$

When the K-S statistics satisfy (9), we say that the process model satisfies goodness-of-fit with the original log data. Therefore, a process with MCD is the most abstract model that satisfies goodness-of-fit with the event log.

E. Verification Procedure and Performance

K-S statics are used to judge whether a log has the same distribution to another log. In order to conduct the verification on that, we have to prepare TOPV for two sets of log data. Then we extract empirical distribution function for K-S test. Based on the K-S test, we can carry out some activities for enhancing the process. For example, if we are benchmarking an organization whose process productivity is excellent, we have to compare our ways of executing the process with that of the target organization. Usually, this activity is relying on comparing process models. However, often we find the execution of a process is far different from the model defined. If there is a statistical evidence that our process execution way is different from the one in the reference process, we indeed have to go for the activity to enhance our process, as illustrated in Fig. 4.

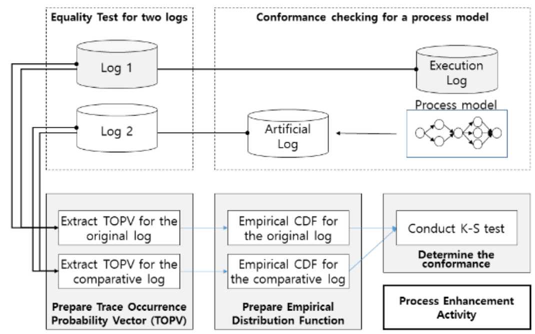


Fig. 4. Statistical verification procedure

To verify the performance of our method, an experiment was carried out to measure the time required for a different number of cases. We repeated each case 1,000 times using artificial experiment data, as shown in Fig. 5. We were able to observe a bell-shaped curve with a low standard deviation value, which confirmed the adequate reliability of our experiment

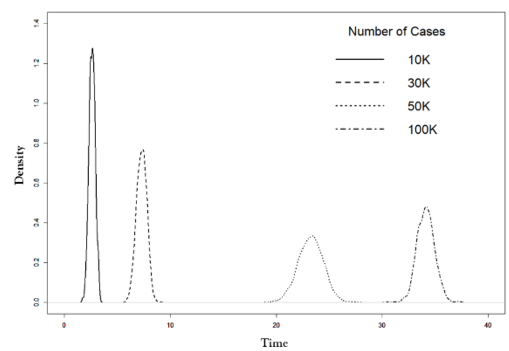


Fig. 5. Performance of the test with the number of cases

Furthermore, this result shows that our algorithm does not incur a proportional increase of processing time with increasing case number. Table I summarizes the result.

TABLE I. RESULT OF THE TIME ANALYSIS FOR THE ALGORITHM

Case size	System time(sec)	
	Mean	Standard deviation
10,000	2.61	0.287
30,000	7.28	0.461
50,000	23.23	1.172
100,000	34.11	0.818

IV. EXPERIMENTS

In order to validate our approach, we carried out experiments using two data sets, one containing a relatively small number of activities and the other a relatively large number of activities. For the analysis of these data, we used a process discovery algorithm, Heuristic Miner to create a process model. After we obtain a process model, we played out the model to generate comparative log data. Then, we evaluated the conformance in two different ways, first calculating the traditional fitness value and then the K-S test value, and compared the results. The datasets used for the experimental are as follows.

- Log data for a small process (*LD*-I): Process execution log in a steel manufacturing company in Korea. The log data contains a small number of activities.
- Log data for a large process (*LD*-II: Building permit application process execution log from five Dutch municipalities (BPIC 2015 data). (Notice that we selected two municipalities among five.)[11]. It contains a large number of activities.

A. *Conformance verification of small process*

The first log data set (*LD*-I) contains 10,313 cases, 89,226 events, and 11 activities, which were generated in the course of the execution of steel manufacturing processes within a Korean company. This dataset has a relatively small number of activities, and so it was easy to understand the overall process by observing the discovered process model. We used Heuristic Miner to obtain a process model, which, as shown in Fig. 6, well describes the process. In Heuristic Miner, we can adjust abstraction level by using dependency threshold. In order to get more abstract process model, we applied a high dependency threshold since it can filter links that are not exceeding the threshold.

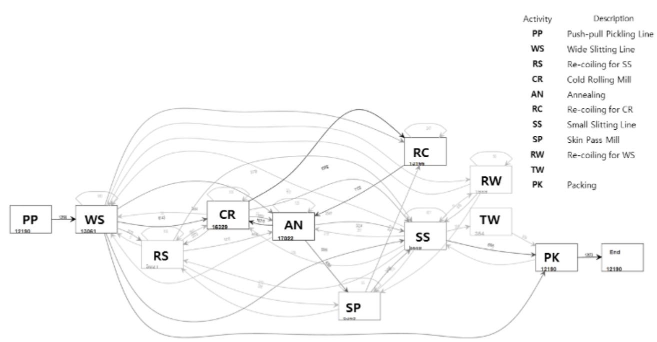


Fig. 6. Steel manufacturing process generated by Heuristic miner

Table II shows the result of a K-S verification test to verify the model with two dependency thresholds: 0.97 and 0.98. With the dependency threshold of 0.97, the *p*-value is almost exactly 1, which means that the model almost perfectly fits the original log data. However, with the dependency threshold of 0.98, the *p*-value is 2.2^{-16}, which means that H_0 cannot be supported statistically. In this case, we cannot say that the process model follows its original dataset. Notwithstanding the similarity of the fitness values for the two cases, there is a serious difference in statistical significance.

TABLE II. CONFORMANCE CHECKING FOR *LD*-I

Dependency	Test statistic	P-value	Fitness
0.97	0.0019	1	0.8723
0.98	0.0687	2.2e-16	0.8635

The difference between the two cases is well illustrated with cumulative distribution functions (CDFs) in Fig. 7. CDF curves in Fig. 7 shows the case that a process model with the dependency of 0.97 conforms to an event log with a statistical significance of 95% and CDF curves of original log and comparative log (process model) almost perfectly fit. However, it shows some gaps between original log distribution and comparative log distribution when the dependency of the discovered model is 0.98.

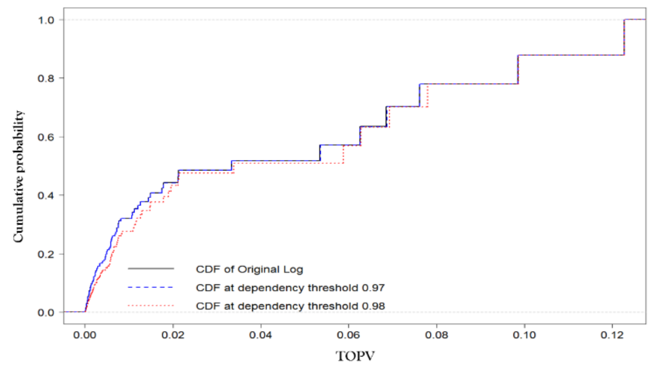


Fig. 7. CDF at dependency threshold 0.97 and 0.98 for *LD*-I data set

From Definition 2, MCD at 95% significance level for LD-I data set is 0.97, and the process model with MCD is depicted in Fig. 8.

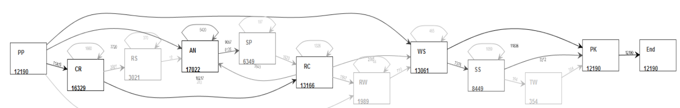


Fig. 8. Steel manufacturing process model with MCD

B. *Conformance verification of large process*

We conducted another experiment using *LD*-II data sets, which is a larger data set than *LD*-I. Since it includes two different regions, we separate it into *LD*-IIa and *LD*-IIb, which are data sets for different municipalities. *LD*-IIa contains 1199 cases, 52217 events, and 398 activities, while *LD*-IIb contains 1053 cases, 47293 events, and 356 activities. In this section, we analyzed *LD*-IIa in ways similar to those for the steel manufacturing data.

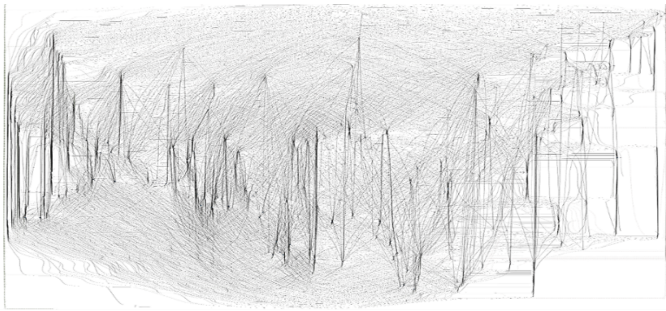


Fig. 9. Building permit application process model generated by Heuristic Miner

Without abstraction, process discovery algorithm generates a spaghetti process as shown in Fig. 9, and such spaghetti process models hinder intuitive understanding of data. We conducted a statistical verification for process models with different abstract levels by adjusting dependency threshold. Table III shows the results of our test for the Heuristic Miner dependency threshold range of 0.93 to 0.98. They show that the *p*-value decreases as the dependency value increases, which indicates that a higher dependency threshold incurs a higher probability of the null hypothesis being rejected. We found that the MCD of this dataset is 0.97 with the 95% significance. Fig. 10 shows the MCD process model.

TABLE III. CONFORMANCE CHECKING FOR *LD*-II

Dependency	Test statistic	P-value	Fitness
0.93	0.0200	0.9838	0.5894
0.94	0.0220	0.9612	0.5791
0.95	0.0229	0.9448	0.5813
0.96	0.0296	0.7462	0.4260
0.97	0.0429	0.2872	0.4291
0.98	**0.0629**	**0.0311**	**0.4268**

Fig. 10. Building permits application process model with MCD

In Fig. 11, we compared two CDF curves with different dependency values with CDF of the original log. The solid line is the CDF curve of the original dataset, the dashed line is that of the comparative log generated from the process model according to a dependency threshold of 0.97, and the dotted line is that for a dependency threshold of 0.98. The line representing the threshold of 0.97 passed the equality test, but the line representing the threshold of 0.98 did not. In this figure, the dashed line at the dependency value of 0.97 shows a small gap from the CDF of the original event log, whereas the dotted line at the dependency value of 0.98 shows a larger gap from the original log. These results confirm that our approach can effectively test a discovered model's conformance to an event log.

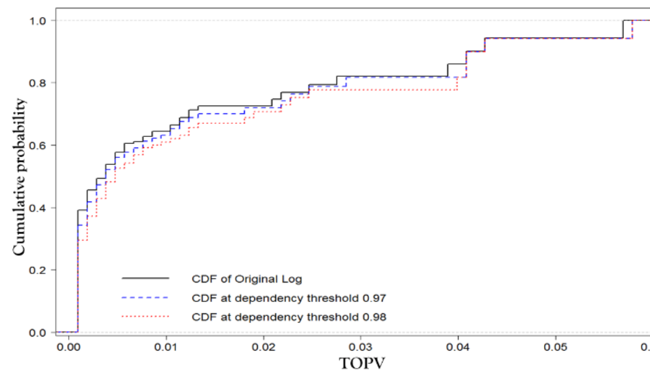


Fig. 11. CDF at dependency threshold 0.97 and 0.98 for *LD*-IIa data

C. Conformance verification of large process

In this section, we compare our method with the existing fitness checking method for heuristic process models. To that end, we evaluated the correlation of traditional fitness and test-statistic with dependency. Table IV shows that our parameters, K-S test-statistics had a higher correlation with the dependency threshold, which is a measure of abstraction level, than did the traditional fitness values.

TABLE IV. CORRELATION TABLE OF *LD*-II

	K-S test	Fitness
Correlation with abstraction level for *LD*-IIa	**0.91**	-0.89
Correlation with abstraction level for *LD*-IIb	**0.94**	-0.90

Furthermore, as shown in Fig. 12 and 13, our parameters are consistent with the varying dependency values. We found that traditional fitness did not show the monotonic decrease with abstraction level, whereas our parameters more consistently reflected the tendency of model simplification.

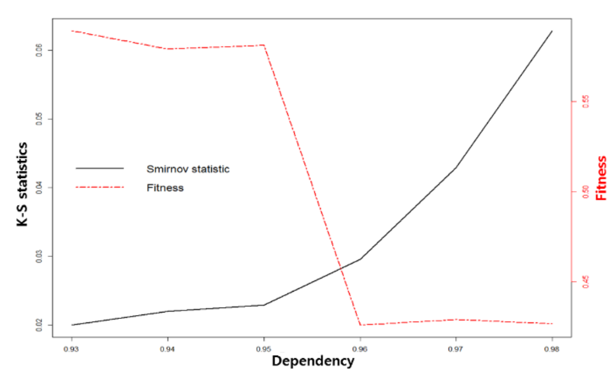


Fig. 12. Correlation plot of *LD*-IIa

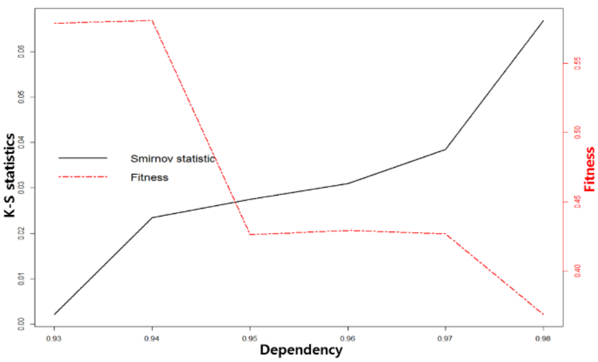


Fig. 13. Correlation plot of *LD*-IIb

V. CONCLUSIONS

In this paper, we develop a method for checking the conformance of a model to the event log by comparison of the statistical variation of two datasets, namely the original log dataset and the comparative log dataset generated by playing out the process model. Additionally, we propose a method for checking process model quality by means of the concept of MCD, which is a threshold value for the satisfaction of the equality condition between two log data sets according to a certain level of statistical significance. Previous research has lacked sufficient criteria to determine whether a model is statistically good enough to fit event log data; in fact, the previous approach employs only intuitive measures based on the node-arc relation. In cases of a very complex process model such as a spaghetti process model, the previous approach cannot provide a precise interpretation of model quality. In order to overcome the limitation of the complex model, simplification was attempted. However, the proper degree of model abstraction could not be determined. In this paper, we suggest a method of determining the optimal level of process model abstraction that conforms to the event log. With this methodology, users can carry out analytics in both intuitive and empirical ways. We verified our approach by conducting experiments with real application data, the results of which showed that our parameter is highly correlated with the conformance of the model to the event log. We expect that our approach will prove to be easily applicable for conformance checking of process models generated by process mining techniques other than Heuristic Miner. Furthermore, our methodology can be extensible to be used for various purposes in BPR and process innovation areas, where evaluation of process quality and test of process equality including conformance are highly needed.

ACKNOWLEDGMENT

This research was partly supported by the IT R&D program of MOTIE / KEIT (Ministry of Trade, Industry, and Energy / Korea Evaluation Institute of Industrial Technology; No. 10054514) and National Research Foundation of Korea (NRF) grant funded by the Korean government(MEST) (No.NRF-2015R1D1A1A09061331). Ling Liu's research is partially sponsored by the National Science Foundation under Grants NSF1547102 and SaTC 1564097 and an IBM Faculty award.

REFERENCES

[1] Rozinat, Anne, M. Veloso, and Wil MP Van der Aalst. "Evaluating the quality of discovered process models." 2nd Intl. Workshop on the Induction of Process Models, Antwerp, Belgium. 2008.

[2] Van Der Aalst, Wil. Process mining: discovery, conformance and enhancement of business processes. Springer Science & Business Media, 2011.

[3] Rozinat, Anne, and Wil MP van der Aalst. "Conformance checking of processes based on monitoring real behavior." Information Systems 33.1 (2008): 64-95.

[4] Weijters, A. J. M. M., Wil MP van Der Aalst, and AK Alves De Medeiros.; Process mining with the heuristics miner-algorithm. In: Technische Universiteit Eindhoven, Tech. Rep. WP 166 pp. 1-34 (2006).

[5] Robert V.Hogg, Joseph W. McKean, Allen T.Craig.; Introduction to Mathematical Statistics 7th Edition

[6] Massey Jr, Frank J.; The Kolmogorov-Smirnov test for goodness of fit. In: Journal of the American statistical Association 46.253 pp. 68-78 (1951).

[7] Facchinetti, Silvia.; A Procedure to Find Exact Critical Values of Kolmogorov-Smirnov Test. In: STATISTICA APPLICATA 21 pp 337-359 (2009).

[8] Hoeffding, Wassily. "A non-parametric test of independence." The Annals of Mathematical Statistics (1948): 546-557.

[9] KVAM, Paul H.; VIDAKOVIC, Brani. Nonparametric statistics with applications to science and engineering. John Wiley & Sons, 2007.

[10] SPRENT, Peter; SMEETON, Nigel C ; Applied nonparametric statistical methods. CRC Press, 2007.

[11] B.F. van Dongen (2015) BPI Challenge 2015. Eindhoven University of Technology.Dataset. http://dx.doi.org/10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1

[12] Buijs, Joos CAM, Boudewijn F. Van Dongen, and Wil MP van der Aalst. "On the role of fitness, precision, generalization and simplicity in process discovery." On the Move to Meaningful Internet Systems: OTM 2012. Springer Berlin Heidelberg, 2012. 305-322

[13] Van Der Aalst, Wil, et al. "Process mining manifesto." Business process management workshops. Springer Berlin Heidelberg, 2011

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

[15] van der Aalst, Wil MP, and C. W. Günth. "Finding structure in unstructured processes: The case for process mining." Application of Concurrency to System Design, 2007. ACSD 2007. Seventh International Conference on. IEEE, 2007.