Expert Systems with Applications 39 (2012) 6061–6068

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Real-time business process monitoring method for prediction of abnormal termination using KNNI-based LOF prediction

Bokyoung Kang [a], Dongsoo Kim [b,*], Suk-Ho Kang [a]

[a] Department of Industrial Engineering, Seoul National University, Republic of Korea
[b] Department of Industrial and Information Systems Engineering, Soongsil University, 369 Sangdoro Dongjak-Gu, Seoul 156-743, Republic of Korea

ARTICLE INFO

Keywords:
Process monitoring
Real-time
Abnormal termination
Local outlier factor (LOF)
Imputation
KNNI (k nearest neighbor imputation)

ABSTRACT

In this paper, we propose a novel approach to real-time business process monitoring for prediction of abnormal termination. Existing real-time monitoring approaches are difficult to use proactively, owing to unobserved data from gradual process executions. To improve the utility and effectiveness of real-time monitoring, we derived a KNNI (k nearest neighbor imputation)-based LOF (local outlier factor) prediction algorithm. In each monitoring period of an ongoing process instance, the proposed algorithm estimates the distribution of LOF values and the probability of abnormal termination when the ongoing instance is terminated, which estimations are conducted periodically over entire periods. Thereby, we can probabilistically predict outcomes based on the current progress. In experiments conducted with an example scenario, we showed that the proposed predictors can reflect real-time progress and provide opportunities for proactive prevention of abnormal termination by means of an early alarm. With the proposed method, abnormal termination of an ongoing instance can be predicted, before its actual occurrence, enabling process managers to obtain insights into real-time progress and undertake proactive prevention of probable risks, rather than merely reactive correction of risk eventualities.

1. Introduction

A business process monitoring system is defined as an information system that provides real-time access to process management indicators (Buytendijk & Flint, 2002). A business process is represented as a set of tasks and their flows orchestrated to achieve a common business goal (Keung & Kawalek, 1997). After process execution, a process instance is generated, which is observed through a set of process- or task-relevant attributes. The monitoring system records such attributes, which include the start and completion times of each task, input and output data, and resources or information from any event occurring during executions (Grigori et al., 2004). By archiving these instance logs and analyzing the relations between process attributes and results, we can extract valuable knowledge that can be utilized for monitoring running instances through observed attributes, which helps to diagnose a current state of running instances and predict their probable issues (Wang & Romagnoli, 2005). Such knowledge is extracted in the form of If-Then rules by means of an inductive data mining technique (Ma & Wang, 2009), which is known as the rule-based approach. A rule is defined as a correlation between a pattern of process attributes and the corresponding result (as given by the pattern). In rule-based monitoring, if the specific condition of attributes is detected, the status of the process is identified, and the predefined operation is provided (Grigori, Casati, Dayal, & Shan, 2001; Grigori et al., 2004). One of the goals of rule-based monitoring is fault detection of infrequent process patterns as compared with the normal, frequent pattern. Various fault detection algorithms have been applied in fields such as municipal solid waste incineration (Chen & Lin, 2008), fraud detection in financial processes (Yue, Wu, Wang, Li, & Chu, 2007), Emergency Department triage (Nie, Zhang, Liu, Zheng, & Shi, 2010), and infrequent image detection in video streams (Medioni, Cohen, Hongeng, Bremond, & Nevatia, 2001).

Local outlier factor (LOF) is one of the most widely used fault detection algorithms. Its operation is based on relative density as measured by how isolated the pattern is with respect to the surrounding neighbors, which indicates the probability of being a fault (Breunig, Kriegel, Ng, & Sander, 2000). Thereby it can detect local faults as well as global faults; indeed, LOF typically achieves the best performance among numerous algorithms with which it is tested (Lazarevic, Ertoz, Ozgur, Srivastava, & Kumar, 2003).

Existing rule-based approaches require that all attributes be obtained. In that sense, identification of a process state is conducted effectively instance by instance, once all attribute conditions are detected instantly or near instantly at the end of the process. However, the existing approaches have shown some limitations when

* Corresponding author. Tel.: +82 2 820 0688; fax: +82 2 825 1094.
E-mail address: dskim@ssu.ac.kr (D. Kim).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.12.007

applied to real-time process monitoring. As a process becomes longer and more complex, execution of a process instance requires more time, so that attributes are detected gradually as the execution period elapses (Kang, Lee, Min, & Cho, 2009). Therefore, when monitoring an ongoing instance in real-time, the monitoring system has to remain idle until all conditions are detected upon its termination (Grigori et al., 2001; Grigori et al., 2004). Even if the instance terminates abnormally, the monitoring system cannot help providing a reactive operation which merely resolves the abnormal termination only after its actual occurrence (Kim, Choi, & Park, 2010). Such limitations require proactive real-time process monitoring systems that predict final outcomes based on the current status at midcourse (Leitner, Wetzstein, Rosenberg, Michlmayr, & Leymann, 2010).

To alleviate these limitations, this paper proposes a novel approach to real-time business process monitoring for fault (especially abnormal termination) prediction using LOF and an imputation method. Over the course of real-time monitoring, unknown attributes are, by imputation, substituted for assumed attributes corresponding to probable results after the current monitoring period. Then, LOF values are computed based on the plausible instances composed of the observed attributes and the imputed attributes from the probable next states as given by the current state. After that, the LOF values, when the ongoing instance is terminated, are estimated probabilistically in each monitoring period. By the proposed method, probable outcomes are predicted over entire monitoring periods, based on the current performance. Therefore, by observing the tendency after real-time progress, an abnormal termination of an instance can proactively be predicted, before its actual occurrence.

The rest of the paper is organized as follows. Section 2 reviews the existing research on LOF algorithms and rule-based monitoring approaches. Section 3 describes the motivation behind and concept of the proposed real-time monitoring method. Section 4 presents details of the real-time business process monitoring scheme using KNNI (k nearest neighbor imputation)-based LOF estimation. In Section 5, the results of experiments conducted with an example scenario are discussed. Finally, Section 6 summarizes conclusions and future work.

2. Background

2.1. Local outlier factor (LOF) algorithm

In this paper, we focus on the LOF (local outlier factor) algorithm among the various unsupervised fault detection algorithms used with rule-based monitoring approaches. For each object, the LOF algorithm computes the degree of being a fault, called the local outlier factor, according to how isolated the object is compared with other, surrounding objects. A higher LOF value indicates that the local density of a data point is smaller than that of its surrounding points. The LOF value for data point q is computed as follows (Breunig et al., 2000).

1. For each object q, compute the $k\text{-}distance(q)$, which is the Euclidean distance to the kth nearest neighbor of q.
2. For each object p, compute a reachability distance from q, as $reach\text{-}dist_k(q,p) = \max\{d(q,p),\ k\text{-}distance(p)\}$, where $d(q,p)$ is the Euclidean distance between q and p.
3. When $N_k(q)$ is a set of k nearest neighbors of q, the local reachability density of q is defined as follows, which corresponds to the surrounding density of q.

$$lrd_k(q) = 1 \Big/ \left(\frac{\sum_{p \in N_k(q)} reach - dist_k(q,p)}{|N_k(q)|} \right)$$

4. Compute the LOF of q, which is defined as the average of the ratio of the lrd of q and those of its k nearest neighbors, as follows.

$$LOF_k(q) = \frac{\sum_{p \in N_k(q)} \frac{lrd_k(p)}{lrd_k(q)}}{|N_k(q)|} \quad (1)$$

For a normal object, its local density is similar to those of surrounding neighbors, so that the LOF, from Eq. (1), converges to 1. However, if a fault is located far from clusters composed of normal objects, its local density is smaller than those of the others, which makes the LOF larger than 1. Therefore, the LOF value increases with the probability of being a fault.

By applying the LOF to business process monitoring for fault detection, we can compute how abnormal a process instance is compared with normal patterns in historical logs. Each process instance is composed of observed attributes during process execution, so that the instance can be regarded as a vector of attributes. Then, the LOF value of each instance can be computed, and the abnormal instance can be detected, with predefined parameters such as the number of k nearest neighbors and the upper limit of the LOF value for normal patterns.

2.2. Existing monitoring approaches

Early researches on process monitoring system focused on recording critical indicators in large data from process executions and visualizing them to process managers. After that, with increasing size and complexity of execution data, expert systems were introduced by adopting knowledge management (Beckett, Wainwright, & Bance, 2000). However, they also had difficulty in autonomous operation and maintenance. To cope with the limitation, rule-based monitoring approaches had been suggested by applying inductive data mining (IDM) (Ma & Wang, 2009). A rule-based monitoring approach with IDM is conducted in two phases, off-line preprocessing and on-line monitoring. During preprocessing, historical process logs are analyzed to extract valuable knowledge, which is represented as a relation between attributes, generated from relevant tasks composing a process, a performance after the process termination. It is usually defined in a form of If (conditions of attributes) – Then (corresponding process status) rule. Based on the predefined rule, a running process instance is observed through occurred attributes during on-line monitoring phase. If observed attributes are satisfied with conditions, we can diagnose the current state of the process and provide an appropriate reaction. In summary, it aims at detecting a set of attributes that leads to probable risks or opportunities. Recently, several rule-based monitoring methods have been suggested by adopting IDM techniques such as Decision Tree (Grigori et al., 2001; Grigori et al., 2004; Kang, Cho, & Kang, 2009; Kang et al., 2009), Support Vector Machine (Widodo & Yang, 2007), Association Rule Mining (Lim & Lee, 2010), Genetic Algorithm (Ma & Wang, 2009) and so on.

However, LOF-based fault detection and rule-based approaches have shown limitations when applied to real-time business process monitoring. In conventional approaches, identification of a process state using the If-Then rule can be conducted only after all process-relevant attributes are generated. Pokrajac, Lazarevic, and Latecki (2007) likewise insisted that outlier detection algorithms determine outliers only once all data records (samples) are present in the dataset. But, when monitoring an ongoing instance in real-time, relevant attributes are observed gradually, not instantly, according to the order of executable tasks composing a process (Kang et al., 2009). Therefore, a monitoring system can observe only partial attributes generated within each monitoring period, which leads to following side effects. First, it is difficult for the monitoring

system to provide proactive operations according to real-time progress. Most existing methods are not embedded in the midcourse of the process, because observations or data records are aggregated at the end of process execution (Curtis, Seshagiri, Reifer, Hirmanpour, & Keeni, 2008). Consequently, identification of the status of an ongoing instance cannot help but be postponed until termination of the process, which fact idles the monitoring system. Therefore, only a reactive correction can be provided after the actual occurrence of a fault corresponding to an abnormal termination, which is too late, rendering real-time monitoring meaningless (Castellanos, Salazar, Casati, Dayal, & Shan, 2006; Rusinov, Rudakova, & Kurkina, 2007). Second, there is no indicator to impart comprehensive and sophisticated intuition about real-time progress to process managers. Rule-based approaches evaluate the state of the executed process deterministically by binary evaluation of whether the observed attributes satisfy the rule conditions or not (Kang et al., 2009). However, because the ongoing instance is still running, unobserved attributes from remained execution parts cannot be determined, which derives the need to measure an uncertainty about next progresses after the monitoring period. Thus it is necessary for indicators to predict outcomes based on the current progress (Grigori et al., 2001). By means of such indicators, process managers can acquire insights into the performance of an ongoing instance and predict the quality of accomplished targeted results, which should be conducted periodically over entire monitoring periods (Rao, Kestur, & Pradhan, 2008).

Whereas conventional process monitoring is focused on identifying the state of process termination instance by instance, real-time monitoring aims at predicting that state as indicated by the current state during real-time progress of one ongoing instance. Therefore, prediction of probable results should be conducted over entire process execution periods (Weller & Card, 2008), and appropriate prevention should be undertaken proactively by real-time feedback at midcourse (Rao et al., 2008).

3. Concept of proposed method

In this section, we present the motivations and concepts behind the proposed method. Fig. 1 schematizes three approaches to LOF-based fault detection for real-time business process monitoring. Let us suppose that, after executing a process model, an ongoing process instance is monitored through a set of m attributes, which are generated gradually with real-time progress and observed by a monitoring system. At monitoring period t at midcourse, only t attributes ($t < m$) have been recorded; the rest have not yet been generated.

In the case of conventional LOF-based fault detection (see Fig. 1(a)), the LOF value cannot be calculated during the initial periods, since not all of the attributes are yet present. At period t, a set of observed attributes is recorded as $(a_1,\ldots,a_t)$, and the rest, unobserved attributes $(a_{t+1},\ldots,a_m)$, are missing. Therefore, during real-time progress, it is difficult to determine whether the ongoing instance will be terminated abnormally or not, and so that determination is postponed until the end.

Compared with a set of recorded attributes from a completed instance, that of an ongoing instance can be considered as *incompletely filled data*. We determined that there exist inevitable missing attributes, unobserved attributes, according to real-time progress. To deal with such missing data, we applied an imputation method. Imputation is defined as "replacing missing or non-sampled measurements for any unit in the population with measurements from another unit with similar characteristics" (Ek, Robinson, Radtke, & Walters, 1997). By using imputation, unobserved attributes are substituted for specific values based on historical instance logs and currently observed attributes. The imputed values correspond to the probable next progress of an ongoing instance given the current progress. Combing observed attributes $(a_1,\ldots,a_t)$ and imputed attributes $(a_{t+1},\ldots,a_m)$, we can generate a plausible instance considered as *completely filled data*. Therefore, the LOF value can be calculated as a single constant value as in Fig. 1(b). Subsequently, it is predicted deterministically whether the ongoing instance will be terminated abnormally or not.

To impute unobserved attributes, we adopted KNNI (k nearest neighbor imputation), one of the most widely used *hot deck* imputation methods. KNNI imputes an average of missing attributes over k nearest neighbors from known attributes (Korhonen & Kangas, 1997). As described in Fig. 1(b), when applying imputation directly, only one plausible instance is generated, which corresponds to merely one of several possible outcomes after the

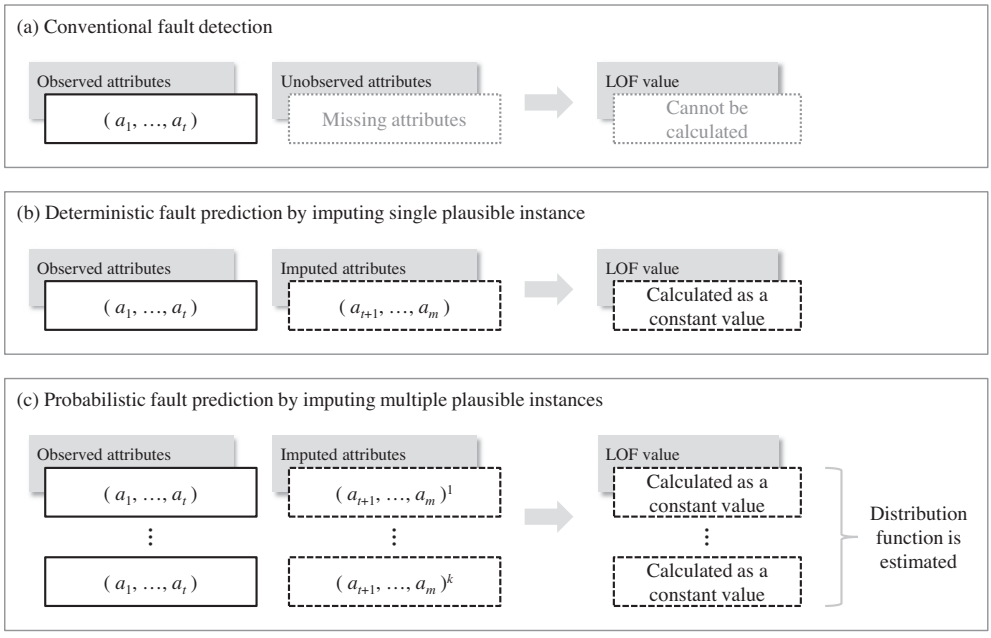

Fig. 1. Three approaches to real-time monitoring for fault detection.

current progress. In our research, we concluded that real-time business process monitoring, in order to cope with probably opportunities or risks, aims to predict various possible outcomes rather than just one.

We modified the original KNNI to make it more suitable to our real-time monitoring purpose. In the probabilistic fault prediction by modified KNNI in Fig. 1(c), we generate k plausible instances by imputing attributes from each of the neighbors. The ith plausible instance is composed of observed attributes $(a_1,\ldots,a_t)$ from the ongoing instance and imputed attributes $(a_{t+1},\ldots,a_m)^i$ from the ith neighbor. Each of the plausible instances represents a unique probable outcome. After computing their LOF values, the probability of terminating abnormally can be predicted by estimating the probability distribution of LOF values.

Also expected LOF values or confidence intervals from the distribution can be utilized as indicators to provide valuable insights, a visualization of a distribution function itself can be more helpful to an understanding of real-time progress when a distribution shows multi-modalities or it is hard to be estimated as one type among already well known distribution functions. Thus, we can predict the outcomes probabilistically according to the current progress. Most importantly, the occurrence of a fault, especially abnormal termination of an ongoing instance, can be predicted proactively by observing the indicators during entire periods.

4. Real-time business process monitoring

4.1. Overall procedures

In this section, we formulate overall procedures for the real-time business process monitoring method using KNNI-based LOF prediction. Fig. 2 shows the procedures as categorized by phase, either preprocessing or real-time monitoring.

Preprocessing aims at defining an upper control limit of an LOF value by analyzing historical process instances as training data. After calculating their LOF values, an upper control limit (UCL) can be derived by applying kernel density estimation (KDE) of LOF values. This preprocessing should be performed periodically to reflect currently recorded instances in updating the UCL.

In the real-time monitoring phase, an ongoing instance is monitored continuously, and attributes are generated gradually and recorded in each monitoring period. From observed attributes, k plausible instances are generated by modified KNNI. After calculating their LOF values, the probability of abnormal termination is predicted by comparing the distribution of LOF values with the UCL. If the probability exceeds the predefined threshold, the monitoring system generates an alarm as notification of the probable occurrence of an abnormality.

4.2. Preprocessing

The preprocessing phase proceeds off-line. Initially, a dataset of historical process instances is defined as $D = [d(0),\ldots,d(n)] \in R^{m\times n}$ with m attributes and n samples. Each $d(i)$ for $1 \leqslant i \leqslant n$ corresponding to one instance is considered as a vector composed of m task attributes $(a_1^i,\ldots,a_m^i)$, which were observed during execution of the instance. For each instance in D, an LOF value set $LOF(D) = [LOF(d(0)),\ldots,LOF(d(n))]$ is calculated by Eq. (1) to measure a degree of being a fault.

$$LOF(d(i)) = \frac{\sum_{p\in N_k(q)} \frac{lrd_k(p)}{lrd_k(d(i))}}{|N_k(d(i))|}$$

Because the $LOF(D)$ is composed of discrete values, it is hard to exploit the entire population. Therefore, an alternative approach to defining a control limit is to use KDE, which is a data-driven technique similar to non-parametric empirical density estimation (Yoo, Lee, Vanrolleghem, & Lee, 2004). KDE makes it possible to extrapolate the continuous probability distribution function (PDF) from a less smooth density estimator. The most widely used kernel function is the Gaussian function. Thus the variance is controlled indirectly through parameter h, that is, the width of each kernel. The KDE of an LOF value using the Gaussian kernel function is described as

$$\hat{f}(x) = \frac{1}{kh}\sum_{i=1}^{k}\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-LOF(d(i)))^2}{2h^2}}. \quad (2)$$

In our research, the UCL from the historical training data was determined to be 90% of the cumulative density function in Eq. (2). If the LOF value of a new instance is higher than the UCL, it is defined as an abnormal instance relative to historical cases.

4.3. Real-time monitoring

In the monitoring phase, an ongoing process instance is monitored through observed attributes. Over entire monitoring periods, a probability that the ongoing instance can be terminated abnormally is estimated periodically by a KNNI-based LOF prediction algorithm.

Monitoring period t is defined as an instant that a new attribute is obtained additionally after the previous period. At t, the ongoing instance is monitored as a set of attributes that have been generated until t, which is defined as $o_t = (a_1,\ldots,a_t)$ for $1 \leqslant t \leqslant m$.

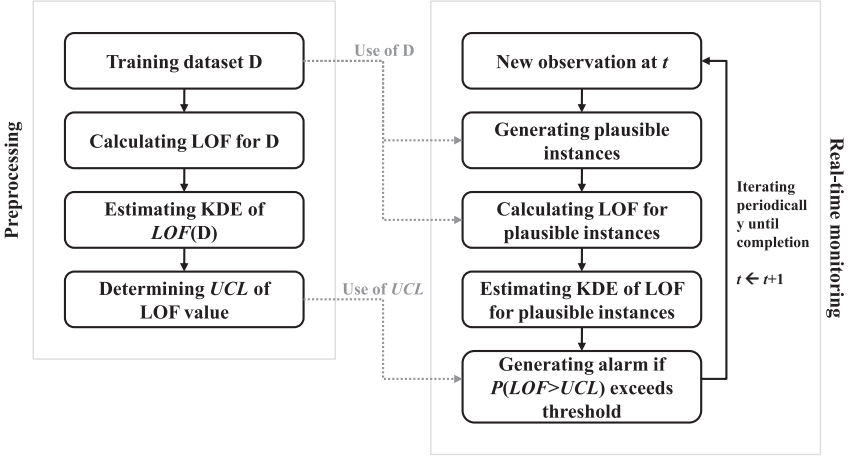

Fig. 2. Overall procedures for real-time business process monitoring.

To generate k plausible instances of o_t, we first compute the Euclidean distance in t-dimensions between o_t and all historical cases $d(i)_t = (a_1^i, \ldots, a_t^i)$ for $1 \leqslant i \leqslant n$, as in the following equation.

$$d(o_t, d(i)_t) = \sqrt{\sum_{j=1}^{t} (a_j - a_j^i)^2} \tag{3}$$

By sorting the distances in Eq. (3), we can extract the set of k nearest neighbors of o_t. Then, k plausible instances can be derived by imputing the attributes of each nearest neighbor for unobserved attributes of o_t. Their set is defined as $Imputed(o_t) = [Imputed\_o_1, \ldots, Imputed\_o_k]$. Each $Imputed\_o_i$ is considered as the plausible instance that o_t can be progressed if it is executed continuously until completion. When a_j is the jth attribute of o_t and a_j^i is the jth attribute of the ith nearest neighbor of o_t, the $Imputed\_o_i$ for $1 \leqslant i \leqslant k$ is defined as follows.

$$\begin{aligned} Imputed\_o_i &= (\text{original attributes of } o_t, \\ &\qquad \text{imputed attributes from } i\text{th nearest neighbor}) \\ &= (a_1, \ldots, a_t, a_{t+1}^i, \ldots, a_m^i) \end{aligned}$$

A set of plausible k LOF values, $LOF(Imputed(o_t)) = [LOF(Imputed\_o_1), \ldots, LOF(Imputed\_o_k)]$, is also computed using Eq. (1) from the set $\{D \cup Imputed\_o_i\}$. Using KDE, the PDF of $LOF(Imputed(o_t))$ is estimated as the distribution of plausible LOF values, which is defined by using KDE as in Eq. (4).

$$\hat{f}(LOF(o_t)) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi} h_i} e^{-\frac{(LOF(o_t) - LOF(Imputed\_o_i))^2}{2h_i^2}} \tag{4}$$

The variance of each kernel h_i is determined by the distance from the referred neighbor in Eq. (3). To give more weight to the kernel of $Imputed\_o_i$ according to how close to o_t the ith nearest neighbor is, h_i is in inverse proportion to the distance from o_t. This means that a probability that o_t can be progressed similarly to $d(i)$ becomes larger to the extent that $d(i)$ moves closer to o_t. When $d(i)$ is the ith nearest neighbor of o_t, h_i is defined as follows.

$$h_i = \frac{\frac{1}{d(o_t, d(i))}}{\sum_{i=1}^{k} \frac{1}{d(o_t, d(i))}}$$

Finally, using Eq. (4) and the UCL from preprocessing, the probability that o_t can be terminated abnormally is estimated as follows.

$$P(LOF(o_m) > UCL | o_t) = \int_{UCL}^{\infty} \hat{f}(LOF(o_t)) \tag{5}$$

When o_t is progressed until completion, entire attributes can be obtained as o_m, and an exact LOF value can be computed. However, instead of waiting until termination, we can predict the conditional probability of becoming an abnormal instance given only partial information such as o_t. If the conditional probability is higher than the predefined threshold, the proposed monitoring scheme proactively sends an early alarm to process managers, notifying them of a probable fault. Otherwise, it continues monitoring until the next monitoring period $t + 1$, and KNNI-based LOF prediction is conducted periodically.

5. Experiments

5.1. Experimental design

We conducted experiments with an example scenario to describe how the proposed method can be applied to real-time business process monitoring. The visualized indicators included the expected LOF value, confidence intervals and the probability of abnormal termination. Then, the error of expected LOF value was analyzed through entire monitoring periods in order to observe real-time progress with decreasing uncertainty. Finally, an early alarm was generated by comparing the probability of abnormal termination with the predefined threshold, of which the accuracy and the earliness were analyzed.

The following was assumed of the process model. It was composed of 6 tasks, one relevant attribute being generated after the execution of each task, and real-time monitoring being phased over 6 gradual periods. In the 6th period, an ongoing instance was terminated, after which it was clearly determined whether the instance had been terminated abnormally or not.

The following was taken for the process-relevant attributes. Each attribute was an independent, identically distributed random variable. Two attributes, a_1 and a_6, were chosen from the standard normal distribution. Attribute a_2 was chosen uniformly from the interval $[0, 1]$, attribute a_3 was chosen from the mixture of the standard normal distribution and the uniform distribution, and finally, attributes a_4 and a_5 were chosen from the mixture of two standard normal distributions. After randomly generating 10,000 instances composed of 6 attributes from these distributions, we divided them into 8000 historical process instances and 2000 ongoing instances. These were conducted 5 times by the 5-folding technique.

5.2. Results and discussion

After executing the process, an ongoing instance was monitored periodically through observed attributes. In each monitoring period, a probability of abnormal termination was estimated by the KNNI-based LOF prediction algorithm. Fig. 3 visualizes the results from the real-time monitoring with abnormal termination. The left side of Fig. 3 describes the PDFs of $LOF(o_t)$ over six monitoring periods, which were estimated by Eq. (4). From the PDFs, the expected value of $LOF(o_t)$ and its 90% confidence intervals could be estimated for each period, as illustrated by the error-bar on the right side of Fig. 3. Then, we could estimate the probability of abnormal termination from Eq. (5) (see the dotted-line in Fig. 3 at the right). Similarly, Fig. 4 shows the monitoring results for a normal termination case.

Regardless of the termination type, the variance and modality of the PDF showed common properties over the monitoring periods. They were increased in the initial periods and then decreased in the later periods. Subsequently, the confidence intervals showed similar properties. According to the termination type, the expected value of $LOF(o_t)$ and the probability of abnormal termination showed a particular trend. For abnormal termination, they became higher as the period elapsed, and for normal termination, they showed the opposite trend. It could be concluded that these values reflected properties pertinent to the real-time progress and the expected termination type; this means that such values can be used as real-time indicators of the state of an ongoing instance.

We analyzed the accuracy of the expected value of $LOF(o_t)$ over 10,000 monitored samples. The error of the expected value of $LOF(o_t)$, e_t, was computed as

$$e_t = |E[LOF(o_t)] - LOF(o_6)|,$$

where $LOF(o_6)$ was the exact LOF value when the ongoing instance was entirely terminated, so that all of the attributes were observed at the final monitoring periods. Fig. 5 shows the expected value and the variance of e_t through the entire periods. As a period elapsed, both the expected value and the variance decreased. During the real-time progress, observing additional attributes decreased the number of probable outcomes, so that the estimated values became more accurate with reduced uncertainty.

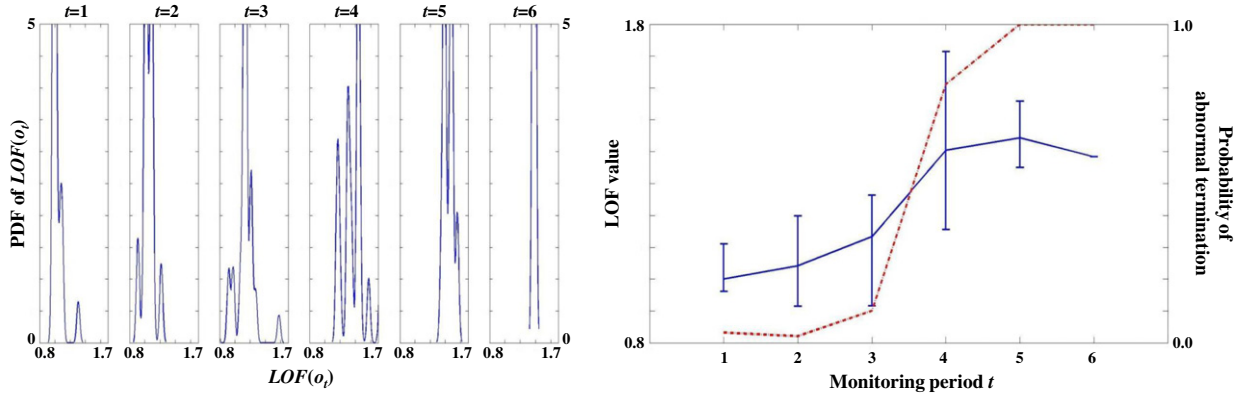

Fig. 3. Real-time monitoring of ongoing instance with abnormal termination.

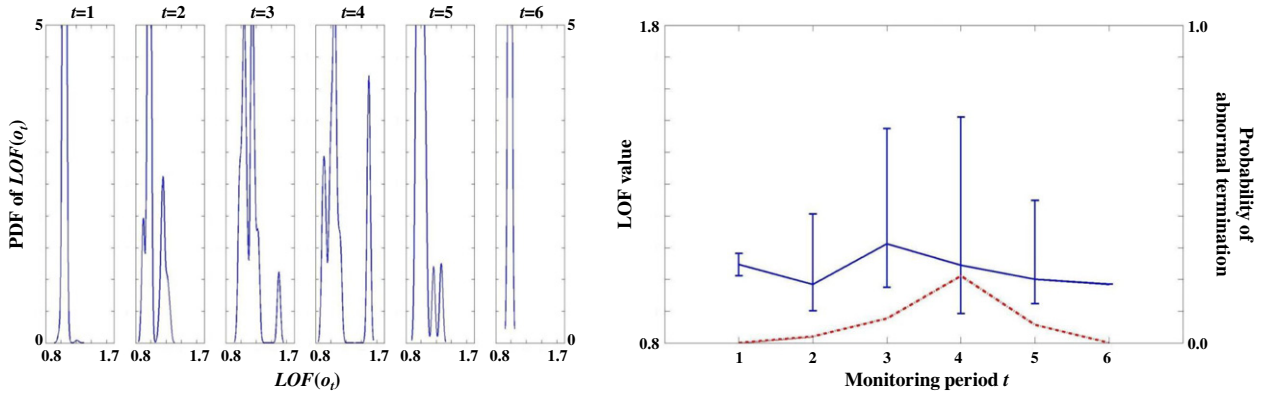

Fig. 4. Real-time monitoring of ongoing instance with normal termination.

Figs. 3 and 4 showed that the probability of abnormal termination can be used as an indicator of real-time progress. Using the probability as the predictor of termination type, we performed the early alarm strategy. We first defined the threshold for the probability. Then, during real-time monitoring, we generated the early alarm when the probability exceeded the threshold in period $t (t < 6)$. The accuracy of the early alarm was measured by two indexes, the *precision* and the *recall*. The precision equaled '1 – false alarm rate,' and the recall, '1 – missing alarm rate.' For $\{F\}$ as a set of instances with abnormal termination and $\{A\}$ as a set of instances for which early alarms are generated, the precision and the recall can be defined as

$$precision = \frac{|\{A\} \cap \{F\}|}{|\{A\}|}, \quad recall = \frac{|\{A\} \cap \{F\}|}{|\{F\}|}.$$

Additionally, we measured the *earliness* of the early alarm for instances with a true early alarm in $\{A\} \cap \{F\}$. The earliness was defined as how proactively the occurrence of abnormal termination is detected before the actual occurrence. Then, the earliness was defined as $6 - t^*$ when the early alarm was generated first in period t^*.

Fig. 6 shows the precision, the recall and the expected value of earliness for threshold 0 to 0.9 and 0.1 intervals. At the top of Fig. 6, the dotted-line is the precision, and the polygonal line is the recall.

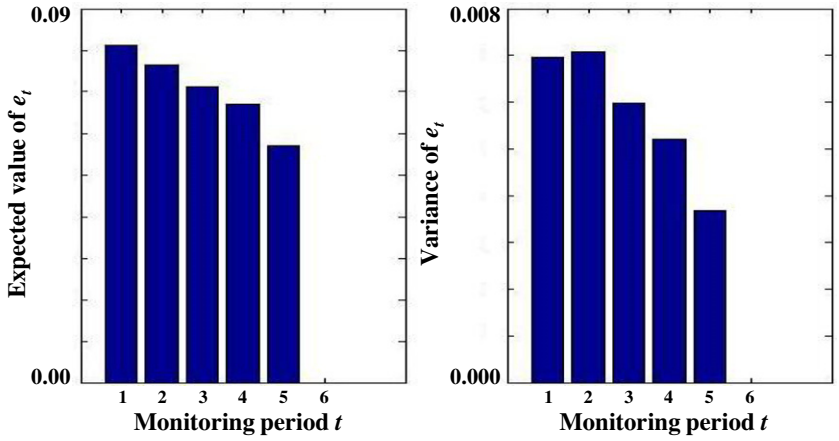

Fig. 5. Expected value and variance of e_t.

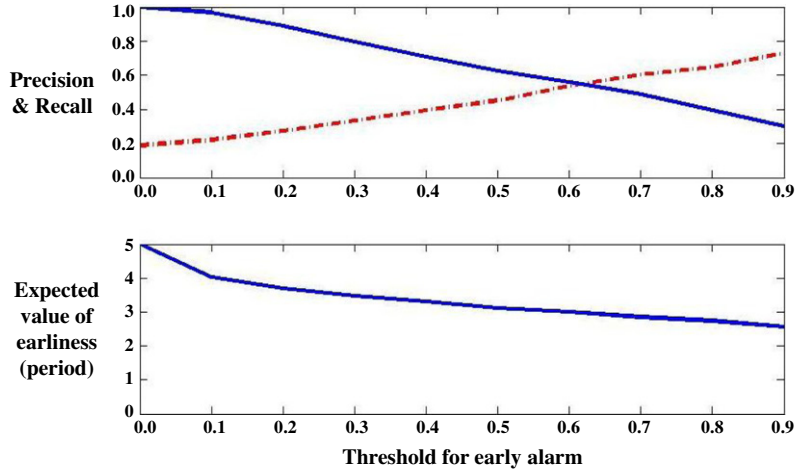

Fig. 6. Precision, recall and earliness of early alarm.

The bottom of Fig. 6 shows the expected values of earliness. According to changes in threshold, following properties were observed. (1) With threshold 0, early alarms were generated for all of the ongoing instances in the first period, which made the overall earliness 5. Therefore, true early alarms were generated for all of the abnormal terminations, and the recall became 1. At the same time, because alarms also were provided for normal terminations, the precision was very low. (2) As the threshold rose, early alarms for abnormal termination rose, and so the precision also rose. Oppositely, the recall became lower. Because the tendency according to termination type became remarkable as the period was close to termination, the earliness diminished with decreasing threshold. (3) When the threshold was defined as 1, alarms were generated for most of the abnormal terminations in the final period. Because these alarms are true alarms only for abnormal terminations, the precision and the recall of the alarm, not the early alarm, became 1. However, the termination type was exactly determined in the last period, so that the early alarm was useless and the earliness was 0.

6. Conclusion

In this paper, we proposed a novel approach to real-time business process monitoring for prediction of abnormal termination. To realize this monitoring method, we devised a KNNI-based LOF prediction algorithm. The conventional rule-based approach, especially LOF-based fault detection, is inefficient as applied to real-time monitoring, and indeed shows limitations such as no indicator or late alarm, due to inevitably unobserved attributes according to the monitoring period. To improve these limitations, we adopted an imputation method that generates a plausible instance by substituting unobserved attributes. Moreover, we extended simple KNNI-based fault prediction by way of a probabilistic fault prediction algorithm, which predicts probable outcomes according to the current progress in each monitoring period. We conducted experiments with an example scenario, showing how the proposed method can be applied to real-time monitoring.

By our method, an ongoing instance is monitored with several indicators including the PDF of LOF values, the expected value and confidence intervals, and the probability of abnormal termination. These are updated gradually as the monitoring period elapses. Besides, based on the above indicators, an early alarm can be provided for notification of a probable abnormal termination. Thereby, process managers can obtain insights into real-time progress and undertake proactive prevention of probable risks, rather than merely react to risk eventualities.

Adopting imputation enables effective and successful application of LOF-based fault detection to real-time process monitoring. Most importantly, the imputation method can be used regardless of algorithms or functions, so that it can also be integrated with any other rule-based approaches.

User-defined parameters in KNNI-based LOF prediction should be optimized by analyzing the accuracy and costs of executing or compensating the ongoing instance, which goal can be recommended for future work. For the accuracy of the early alarm, the threshold of the probability of abnormal termination should be determined by considering losses from false and missing alarms. For example, if a loss from a missing alarm is much larger than that from a false alarm, lowering the threshold can be more profitable from an aspect of entire gains and losses. Moreover, by additionally considering profits and losses from termination types, an early stopping strategy can be incorporated into the monitoring system to complement the early alarm.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0020943).

References

Beckett, A. J., Wainwright, C. E. R., & Bance, D. (2000). Implementing an industrial continuous improvement system: a knowledge management case study. *Industrial Management & Data Systems, 100*(7), 330–338.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density based local outliers. *ACM SIGMOD Record, 29*(2), 93–104.

Buytendijk, F. & Flint, D. (2002). How BAM can turn a business into a real-time enterprise. *Gartner Research*, AV-15-4650.

Castellanos, M., Salazar, N., Casati, F., Dayal, U., & Shan, M.-C. (2006). Predictive business operations management. *International Journal of Computational Science and Engineering, 2*(5/6), 292–301.

Chen, J.-C., & Lin, K.-Y. (2008). Diagnosis for monitoring system of municipal solid waste incineration plant. *Expert Systems with Applications, 34*(1), 247–255.

Curtis, B., Seshagiri, G. V., Reifer, D., Hirmanpour, I., & Keeni, G. (2008). The cases for quantitative process management. *IEEE Software, 25*(3), 24–28.

Ek, A. R., Robinson, A. P., Radtke, P. J., & Walters, D. K. (1997). Development and testing of regeneration imputation models for forests in Minnesota. *Forest Ecology and Management, 94*, 129–140.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., & Shan, M.-C. (2004). Business process intelligence. *Computers in Industry, 53*(3), 321–343.

Grigori, D., Casati, F., Dayal, U., & Shan, M.-C. (2001). Improving business process quality through exception understanding, prediction, and prevention. In *Proceedings of the 27th very large data base endowment conference, Roma, Italy* (pp. 159–168).

Kang, B., Cho, N. W., & Kang, S.-H. (2009). Real-time risk measurement for business activity monitoring (BAM). *International Journal of Innovative Computing, Information and Control, 5*(11A), 3647–3657.

Kang, B., Lee, S. K., Min, Y., Kang S.-H., & Cho, N. W. (2009). Real-time process quality control for business activity monitoring. In *Proceedings of the 2009 international conference on computational science and its applications, Yongin, Korea* (pp. 237–242).

Keung, P., & Kawalek, P. (1997). Goal-based business process models: Creation and evaluation. *Business Process Management Journal, 3*(1), 17–38.

Kim, K., Choi, I., & Park, C. (2010). A rule-based approach to proactive exception handling in business processes. *Expert Systems with Applications, 38*(1), 394–409.

Korhonen, K. T., & Kangas, A. (1997). Application of nearest-neighbor regression for generalizing sample tree information. *Scandinavian Journal of Forest Research, 12*, 97–101.

Lazarevic, A., Ertoz, L., Ozgur, A., Srivastava, J., & Kumar, V. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the third SIAM international conference on data mining, San Francisco, CA, USA* (pp. 25–36).

Leitner, P., Wetzstein, B., Rosenberg, F., Michlmayr, A., Dustdar S., & Leymann, F. (2010). Runtime prediction of service level agreement violations for composite services. *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops, Lecture notes in computer science* (Vol. 6275, pp. 176–186).

Lim, A. H. L., & Lee, C-S. (2010). Processing online analytics with classification and association rule mining. *Knowledge-Based Systems, 23*(3), 248–255.

Ma, C. Y., & Wang, X. Z. (2009). Inductive data mining based on genetic programming: Automatic generation of decision trees from data for process historical data analysis. *Computers and Chemical Engineering, 33*, 1602–1616.

Medioni, G., Cohen, I., Hongeng, S., Bremond, F., & Nevatia, R. (2001). Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*(23), 873–889.

Nie, G., Zhang, L., Liu, Y., Zheng, X., & Shi, Y. (2010). Decision analysis of data mining project based on Bayesian risk. *Expert Systems with Applications, 36*(3), 4589–4594.

Pokrajac, D., Lazarevic, A., & Latecki, L. J. (2007). Incremental local outlier detection for data streams. In *Proceedings of IEEE symposium on computational intelligence and data mining (CIDM), Honolulu, HI* (pp. 504–515).

Rao, U. S., Kestur, S., & Pradhan, C. (2008). Stochastic optimization modeling and quantitative project management. *IEEE Software, 25*(3), 29–36.

Rusinov, L. A., Rudakova, I. V., & Kurkina, V. V. (2007). Real time diagnostics of technological processes and field equipment. *Chemometrics and Intelligent Laboratory Systems, 88*(1), 18–25.

Wang, D., & Romagnoli, J. A. (2005). Robust multi-scale principal components analysis with applications to process monitoring. *Journal of Process Control, 15*(8), 869–882.

Weller, E., & Card, D. (2008). Point argument: Applying SPC to software development: Where and Why. *IEEE Software, 25*(3), 48–51.

Widodo, A., & Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing, 21*(6), 2560–2574.

Yoo, C. K., Lee, J.-M., Vanrolleghem, P. A., & Lee, I.-B. (2004). On-line monitoring of batch processes using multiway independent component analysis. *Chemometrics and Intelligent Laboratory Systems, 71*, 151–163.

Yue, D., Wu, X., Wang, Y., Li, Y., & Chu, C.-H. (2007). A review of data mining-based financial fraud detection research. In *Proceedings of 2007 international conference on wireless communications, networking and mobile computing, Shanghai, PR China* (pp. 5514–5517).