Expert Systems With Applications 141 (2020) 112950

Contents lists available at ScienceDirect

Expert Systems With Applications

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

SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining

Hoonseok Park, Jae-Yoon Jung*

Department of Industrial and Management Engineering, 1732, Deogyeong-daero, Giheung-gu, Yongin, Gyeonggi 17104, Republic of Korea

A R T I C L E I N F O

Article history:
Received 26 February 2019
Revised 8 August 2019
Accepted 12 September 2019
Available online 13 September 2019

Keywords:
Multivariate time series
Event pattern discovery
Inverse normal transformation (INT)
Symbolic aggregate approximation (SAX)
Association rule mining (ARM)

A B S T R A C T

The discovery of event patterns from multivariate time series is important to academics and practitioners. In particular, we consider the event patterns related to anomalies such as outliers and deviations, which are important factors in system monitoring for manufacturing processes. In this paper, we propose a method for discovering the rules to describe deviant event patterns from multivariate time series, called SAX-ARM (association rule mining based on symbolic aggregate approximation). Inverse normal transformation (INT) is first adopted for converting the distribution of time series to the normal distribution. Then, symbolic aggregate approximation (SAX) is applied to symbolize time series, and association rule mining (ARM) is used for discovering frequent rules among the symbols of deviant events. The experimental results show the discovery of informative rules among deviant events in a multivariate time series from a die-casting manufacturing process that has ten variables with 1,437 lengths. We also present the results of sensitivity analysis, which demonstrates that significant rules can be discovered with different settings of the SAX parameters. The results describe the usefulness of the proposed method to identify deviant event among multivariate time series with high complexity.

1. Introduction

Advanced manufacturing environments, so-called smart factories, introduce the information-gathering technologies collecting data from various sensors on equipment and production lines for efficient operations management (Zuehlke, 2010). In this context, monitoring and analyzing manufacturing systems using data-driven techniques such as artificial intelligence, data mining, and knowledge discovery are also arising. Various studies have been conducted for quality improvement, process optimization, and equipment maintenance, based on data collected from many components of a manufacturing system (Choudhary, Harding, & Tiwari, 2009; Harding, Shahbaz, & Kusiak, 2006; Köksal, Batmaz, & Testik, 2011).

Process control is a traditional research area that deals with the variance of process variables collected from manufacturing processes (Montgomery, 2009). The process variables such as pressure, temperature, and velocity have typical structures of time series data since sensors record their values through time. For stable process control, it is necessary to monitor whether process variables observed inside of control limits. If an observation outside the control limits is detected, the operator should recognize it as an outlier or a deviant event that could negatively affect product quality and take appropriate action to resolve the related problem.

In complex manufacturing processes, the process control systems should control hundreds of variables to operate the process. Moreover, they should detect the correlations and the variabilities among multiple process variables and account for the inherent relationships among process variables for effective responses. The data collection systems also become complicated, and the methods of finding patterns within multivariate data are needed (Yin, Gao, & Kaynak, 2014). However, dealing with multivariate time series is challenging in terms of computation time and information extraction (Esling, 2012; Keogh & Mueen, 2011).

Nevertheless, to derive meaningful information, several data mining techniques such as classification, clustering, rule discovery, and visualization have been applied to time series (Fu, 2011). In particular, rule discovery techniques are useful for finding meaningful patterns inherent in time series and represent discovered information as the form of interpretable rules. Association rule mining (ARM) algorithms have been effectively applied to discovery rules for the multivariate time series (Mitsa, 2010; Morchen & Ultsch, 2007). This approach is suitable for the process control since operators usually need to identify and understand relation-

* Corresponding author.
E-mail address: jyjung@khu.ac.kr (J.-Y. Jung).

https://doi.org/10.1016/j.eswa.2019.112950
0957-4174/

ships among deviant events from process variables. For example, a relationship could be “When the temperature is very low, the low-speed stroke is also very low.”

In this paper, we propose a method for discovering patterns among deviant events in the multiple process variables that compose multivariate time series. The proposed method, called SAX-ARM, involves association rule mining with inverse normal transformation (INT) and symbolic aggregate approximation (SAX). INT is applied to map the distribution of each variable to the normal distribution, and then SAX converts the multivariate time series into consecutive symbolic representations. From the symbolized multivariate time series, a set of symbol baskets is generated, which is an itemset of the deviant events. As a final step, the ARM algorithm discovers rules from the symbol baskets that contain relationships among deviant events of multivariate time series such as process variables.

The remainder of this paper is structured as follows. In Section 2, we introduce the related work on the representation and the rule discovery of multivariate time series. In Section 3, we introduce the framework and techniques of the proposed method SAX-ARM. Section 4 describes the experimental results by applying the method to multivariate time series collected from a die-casting manufacturing process. In Section 5, we discuss conclusions and future work.

2. Related work

2.1. Time series representation and discretization

Various techniques in data mining have been applied to discover meaningful knowledge and patterns in data. The discovery of knowledge within different data types has received constant interest from several industries, including science, medicine, finance, economics, and manufacturing (Han, Pei, & Kamber, 2011; Witten, Frank, Hall, & Pal, 2016). In particular, time series including information about time are special because they have the characteristics of large data size, high dimensionality, and continuous change. Many review papers have described the features, current status, and taxonomy of time series data mining (Aghabozorgi, Shirkhorshidi, & Wah, 2015; Fu, 2011; Torkamani & Lohweg, 2017).

Several studies of rule discovery in time series deal with techniques for data representation and discretization simultaneously (Das, Lin, Mannila, Renganathan, & Smyth, 1998; Jiang & Gruenwald, 2006; Morchen & Ultsch, 2005). The rule discovery techniques were applied after time series had been converted into the appropriate format, because time series usually have numerical values, which make difficult to descriptive rule expression. The discovered rules can be represented by sequences of symbols or words for a defined interval, and they can be described using a relationship between them. Therefore, it is necessary to investigate related studies about both the representation and discretization of time series and rule discovery in time series.

Time series representation and discretization are mainly applied for dimension reduction, which is an important factor for handling high dimensional time series. A wide variety of time series representation techniques have been proposed, and Wang et al. (2013) compared the effects of key techniques with several similarity measures.

Representation techniques for dimension reduction in time series improve the efficiency of data mining processes such as querying, classification, and clustering. The highest priority of time series representation and discretization for rule discovery should be improving the interpretation of the derived rule. Therefore, symbolic expressions and discretization methods are applied in natural language, sentences, or word forms that humans can understand (Mitsa, 2010).

The initial form of symbolic representation is described in Das et al. (1998); they constructed subsequences by dividing all the time series by the time window size. Thereafter, clustering was performed, and letters were assigned to the representative form of the cluster. A representative symbolic technique is SAX, proposed by Lin, Keogh, Lonardi, and Chiu (2003). SAX reduces the dimension of time series using piecewise aggregate approximation (PAA) (Keogh, Chakrabarti, Pazzani, & Mehrotra, 2001), dividing the quantiles and assigning symbols to data points within the same quantiles using the standard normal distribution.

In an initial study of shape-based representation for the changing shapes in observed values, Rakesh, Giuseppe, Edward, and Mohamed (1995) proposed the Shape Definition Language. They represented time series data by defining eight types of events {up, Up, down, Down, appears, disappears, stable, zero}. Martinez-de-Pison, Sanz, Martinez-de-Pison, Jimenez, and Conti (2012) and Xue, Zhang, Chen, Le, and Lavassani (2016) defined and discretized events using the amount and type of change in the data over a certain period of time.

2.2. Rule discovery from time series

Discovering rules and detecting events from the database have widely been studied in many areas such as social media (Weng & Lee, 2011), video (Chang, Ma, Lin, Yang, & Hauptmann, 2017; Chang, Yu, Yang, & Xing, 2017), file system (Parkinson, Somaraki, & Ward, 2016) and security (Khan & Parkinson, 2018). Methods based on association rule discovery (Agrawal, Imieliński, & Swami, 1993; Hipp, Güntzer, & Nakhaeizadeh, 2000) and sequential rule discovery (Agrawal & Srikant, 1995; Mooney & Roddick, 2013) have been proposed to find rules in time series. In particular, Jiang and Gruenwald (2006) introduced association rule discovery techniques for time-sequenced data streams.

Rule discovery methods have been proposed for various manufacturing subdivisions, such as product quality and equipment maintenance. Martinez-de-Pison et al. (2012) addressed topics similar to our concerns in this study. They used an episode concept that considers the sequence between events after disassembling the time series. Then they applied the Eclat algorithm to find frequent itemsets and derived association rules between the parameters of a plating production line.

Kamsu-Foguem, Rigal, and Mauget (2013) organized each event in production operation data according to time and found sequential rules. In addition, they found the rules for the delay patterns that occurred during production and evaluated their importance. They then analyzed the relationship between process delays and the factors affecting those delays by interpreting the rules.

Concerning product failure, Chen, Tseng, and Wang (2005) applied association rule discovery techniques to analyze the relationship between production equipment used in a manufacturing process and defective products. They bundled machines that work simultaneously at each stage into a single transaction and used the data about defective products and poor coverage that each machine set included during rule discovery.

Previous studies have mainly focused on finding rules that describe general changes in the overall shape of time series as above. Despite its importance, few studies tried to discover the interpretable rules which describe outliers or deviant events in multivariate time series data. In this study, we focus on deviant events rather than on the changing shape of time series data. In addition, we consider a method for simultaneously analyzing multiple time series rather than a single time series. Detecting the undesirable event and identifying the causes are crucial tasks in the perspec-

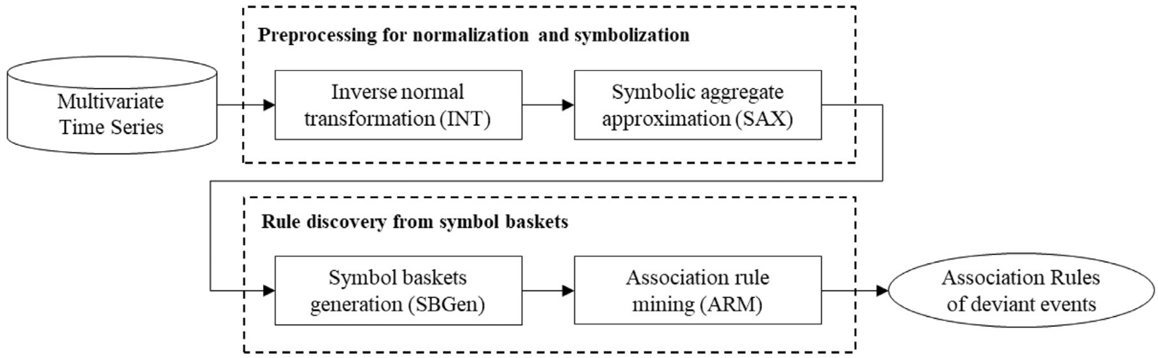

Fig. 1. SAX-ARM: A framework for discovering deviant event patterns from multivariate time series.

tive of manufacturing process control. This research could help to provide useful information for the tasks.

3. Deviant event pattern discovery from multivariate time series

3.1. Framework

We here describe the SAX-ARM framework for discovering association rules among deviant events in multivariate time series from manufacturing processes. Fig. 1 shows the overall steps of the SAX-ARM framework. The first step is a data preprocessing which transforms the raw multivariate time series into the appropriate form for the rule discovery. For each variable in the preprocessed data, INT converts the distribution of time series to the normal distribution and SAX provides a symbolic representation of the time series in the form of letters. In the next stage, the symbols of interest, letters, are extracted to represent deviant events (out of control characteristics). The symbol basket is constructed as a form for the itemset of symbols of interest. Finally, ARM algorithm extracts frequent patterns and rules that imply relationships among deviant events in process variables.

3.2. Preprocessing and representation

3.2.1. Inverse normal transformation

Time series representation methods such as PAA and SAX assume that the empirical distribution of time series follows a specific probability distribution. For example, values in time series are discretized to a set of regions by mapping on the quantiles of the uniform distribution. However, distributions of time series collected from the real-world follow various probability distributions such as normal, lognormal, gamma, Weibull, and Poisson distributions (Wang, Yang, & Hao, 2016). The discordance between the empirical distribution of time series and the reference distribution makes the distortion of time series representation. In this research, appropriate normalization and transformation of time series having various distributions are necessary before applying time series representation based on SAX (Keogh & Kasetty, 2003).

The representative normalization techniques for time series, the z-score normalization preserves the characteristics of time series and outliers well (Mitsa, 2010). However, if the distribution of time series is skewed, the z-score normalized distribution is also skewed. It means that the quantiles of z-score normalized distribution for high skewed time series are not matched on the quantiles of the standard normal distribution. To transform a time series that has the skewed distribution, the Box-Cox transformation (Box & Cox, 1964), which is a general form of the power transformation, is a useful approach to make the data normally distributed. Although the Box-Cox transformed distribution looks like the normal distribution, it is hard to explain that the quantiles of the original distribution are retained. It is because the power function makes it difficult to interpret the quantile of a transformed data point against its quantile at the raw data directly (Chatifield, 2003). After applying the Box-Cox transformation, the SAX that uses quantiles of the normal distribution would return the distorted representations for the original time series.

Inverse normal transformation (INT), which is adopted in this research, is one of nonparametric transformations to convert the sample distribution to the normal distribution (Beasley, Erickson, & Allison, 2009). The main idea of INT is mapping the quantiles from cumulative distribution function (CDF) of the time series to the inverse normal function. The approaches to obtain the empirical quantiles for CDF are presented by Beasley et al. (2009) and Pyrcz and Deutsch (2018). In this research, we specifically adopt the deterministic rank-based INT method since it is relatively intuitive and simple to approximate the empirical quantiles of the sample data based on the ordinary rank of observations (Peterson & Cavanaugh, 2019). The *i*th transformed observation x_i^{INT} in the time series is given by:

$$x_i^{INT} = \Phi^{-1}\left(\frac{r_i - c}{N - 2c + 1}\right) \quad (1)$$

where Φ^{-1} denotes the inverse normal (or probit) function, r_i is the ordinary rank of the *i*th observation in the length N time series, and the constant value c is set to 3/8 (Blom, 1958). Thus, the quantile of transformed value can be directly matched to the its original quantile if there are no tied values.

We note that INT can preserves the quantiles of the original distribution during the transformation procedure. And INT has an advantage that it approximately transforms arbitrary distribution to the normal distribution. These features of INT can alleviate the condition to apply the normality-based discretization methods such as SAX, for the non-normal distributed time series and applications. An example of INT for a sample time series is illustrated in Fig. 2. The histogram and the Q-Q plot show that the distribution of a sample time series is skewed at its tails. After applying INT to the sample time series, the skewness is nearly removed as shown in the second Q-Q plot. Nevertheless, the shape of the time series is also preserved well.

3.2.2. Symbolic aggregate approximation

SAX is a method for representing time series data with alphabetic symbols. It effectively summarizes and represents the time series while maintaining the characteristics of the source data. Therefore, it efficiently reduces high dimensional time series data for application in various types of data mining algorithms (Lin, Keogh, Wei, & Lonardi, 2007).

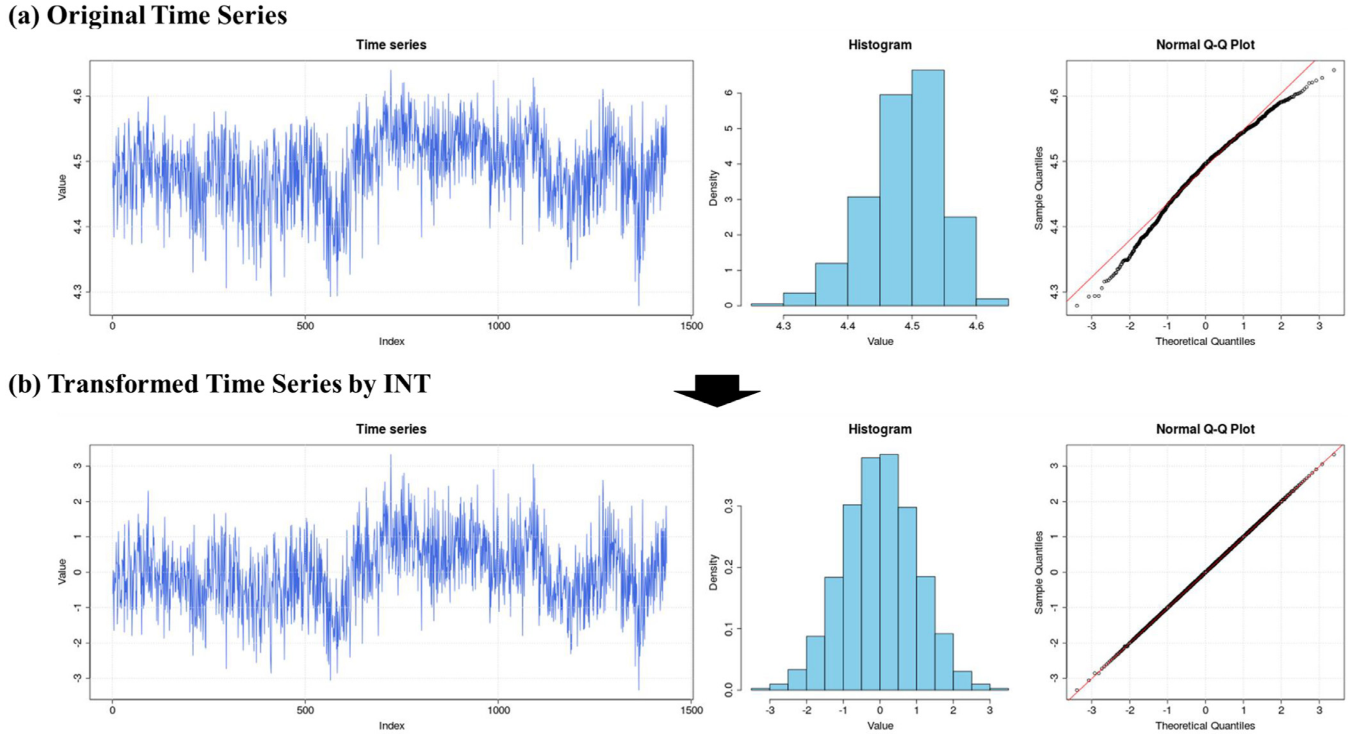

Fig. 2. Transformation of a sample time series to normal distribution through INT.

The procedure of SAX is depicted in Fig. 3. The first step is dimensionality reduction using PAA, which is introduced by Keogh et al. (2001). In PAA, a time series is divided into segments of a certain length, and each segment is summarized with the mean value of the values that it obtains, as shown in Fig. 3(a). By the PAA, the time series $X = [x_1, \ldots, x_N]$ with length N is divided into a PAA vector $\bar{\boldsymbol{X}} = [\bar{x}_1, \ldots, \bar{x}_W]$ with W segments. The ith element $\bar{x}_i$ of the PAA vector is calculated by:

$$\overline{x_i} = \frac{W}{N} \sum_{j=\frac{N}{W}(i-1)+1}^{\frac{N}{W}i} x_j \tag{2}$$

It means that the values in a segment from $\frac{N}{W}(i-1)+1$ to $\frac{N}{W}i$ are averaged and it becomes the compressed representation to the segment of the time series. In this part, we name a constant N/W as *time segment size*, for using the parameter of PAA in the following experiment. It is more intuitive because *time segment size* describes the number of values in a segment.

The second step is discretization, which is depicted in Fig. 3(b). In this step, a PAA vector $\bar{\boldsymbol{X}} = [\bar{x}_1, \ldots, \bar{x}_W]$ is converted to a symbol vector $\hat{\boldsymbol{X}} = [\hat{x}_1, \ldots, \hat{x}_W]$. Each element of $\bar{\boldsymbol{X}}$, $\bar{x}_1$, can be mapped to one of discretization regions according to its value, and then the element is converted to an alphabet symbol, which will become the value of its corresponding element of $\hat{\boldsymbol{X}}$, $\hat{x}_j$. In this way, SAX assigns an appropriate alphabet to each segment of the time series according to the mean value of the segment. Finally, the area of the time series values that follow the standard normal distribution are broken up into the discretization regions of the equal size to produce symbols satisfying the equiprobability (Lin et al., 2003). Herein, the number of discretization regions is called the *alphabet size*. Note that in this research the *alphabet size* can be adjusted according to the precision of the deviant events that are wanted to investigate the process. For example, Fig. 3(b) shows a case with an *alphabet size* of 3, that is, $z = \pm 0.446$ are the breakpoints of separation and 33.3% area of the standard normal distribution is covered by each alphabet in {*a, b, c*}. It means that 33.3% in the tails are considered deviant events. Finally, the given time series is converted to a symbolic sequence as shown in Fig. 3(c).

In this study, we focus only on deviant events beyond certain limits, not all events in the time series. Therefore, we need only two symbols to represent two tails of the normal distribution. In other words, we consider only the first and the last alphabets in the SAX vector. The number of deviant events that should be monitored depends on the characteristics of the process and data. The *alphabet size* can also control it. For example, if top 10% and bottom 10% tails are required to monitor, the *alphabet size* will be set to 10. In the example in Fig. 3, the first and the third letters, *a* and *c*, are considered deviant events in the process.

3.2.3. Symbol baskets

Before applying ARM algorithms, which find rules among frequently occurring items from the transaction data (Agrawal & Srikant, 1994; Han et al., 2011), it is necessary to prepare transaction-like structured data. Therefore, we propose a new data structure called symbol basket as the transaction-like data structure, designed for containing the symbolized multivariate time series. A sequence of symbol baskets is constructed along time axis, and each symbol basket involves the deviant events occurred at the same time segment from all variables. Fig. 4 depicts the procedure of constructing symbol baskets from a multivariate time series discretized by SAX when the *alphabet size* is 10. The letters, *a* and *j*, represent the first and the last letters, respectively, for ten equally-sized areas under the standard normal distribution.

The procedure for generating symbol baskets is presented in the SBGen algorithm (see Algorithm 1). As input, the algorithm uses a symbolized multivariate time series $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \ldots, \hat{\mathbf{x}}_M]$ for M variables when each time series has length W. Also, it is assumed that the letters of interest are the first and the last, denoted by sym^{top} and sym^{btm}. The algorithm creates baskets $\boldsymbol{b}_t$ for each time segment t by collecting the letters of interest observed in multiple time series during the same time segment (in lines 4 to 9). Herein, only two symbols for deviant events (e.g., *a* and *j* when the *al-*

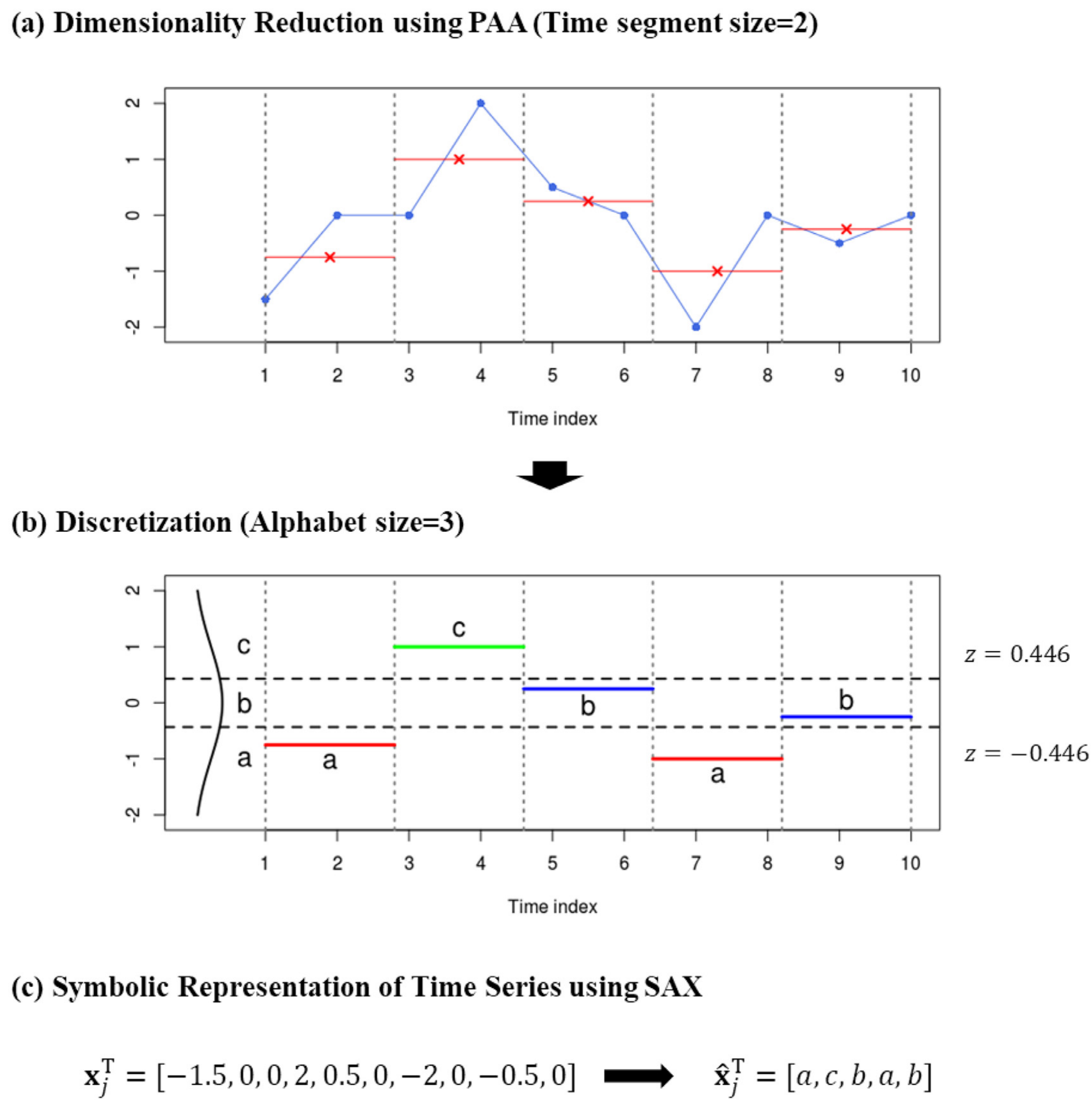

Fig. 3. Procedure of generating a symbolized time series.

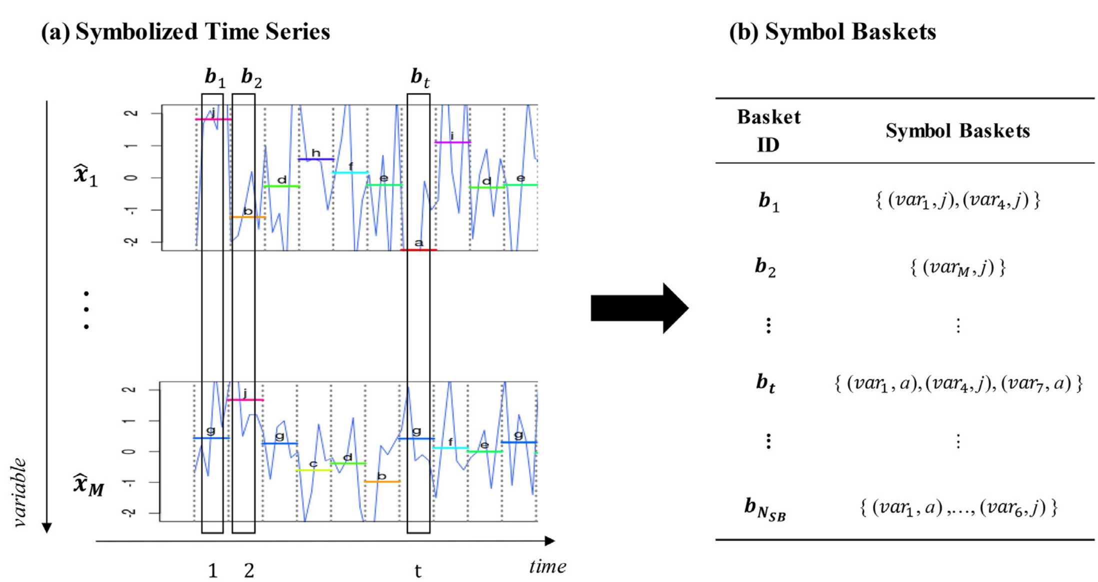

Fig. 4. Symbol baskets constructed from multiple symbolized time series by SAX.

Algorithm 1 SBGen - Symbol Basket Generation Algorithm.

SBGen($\hat{\mathbf{X}}$, sym^{top}, sym^{btm})
Input: Symbolized multivariate time series $\hat{\mathbf{X}}$ that has M variables with length W
Symbols of interest, sym^{top} and sym^{btm}
Output: Symbol baskets $\boldsymbol{B}$
1: Initialize $\boldsymbol{B}=\emptyset$
2: **for** time t=1 to W **do**
3: symbol basket $\boldsymbol{b}_t=\emptyset$
4: **for** each time series $\hat{\mathbf{x}}_j$ in $\hat{\mathbf{X}}$ **do**
5: $sym \leftarrow \hat{x}_{tj}$, the tth element of $\hat{\mathbf{x}}_j$
6: **if** $sym \in \{sym^{top}, sym^{btm}\}$ **then**
7: add a deviant event e=(var_j, sym) into $\boldsymbol{b}_t$
8: **end if**
9: **end for**
10: add $\boldsymbol{b}_t$ to $\boldsymbol{B}$
11: **end for**
12: **return** $\boldsymbol{B}$

phabet size = 10) are placed in basket $\boldsymbol{b}_t$ (in lines 6 to 7). When the occurrence of deviant events, sym^{top} and sym^{btm}, is infrequent, basket $\boldsymbol{b}_t$ may be null. The algorithm builds symbol baskets $\boldsymbol{B}$ with the size of W. Note that the number of symbol baskets, N_{SB}, is the same as the number of time segments W because some baskets $\boldsymbol{B}$ are empty to maintain the number of time series segments.

3.3. Association rule mining using deviant events

In this paper, we apply the association rule mining technique to discover frequent patterns of deviant events that occur simultaneously among process variables. The results of association rule discovery are co-occurrence patterns acquired using several measures (Tan, Kumar, & Srivastava, 2002). In this research, the association rules among deviant events extracted from the symbol basket data provide useful information about which deviant events in one variable are related to deviant events in another variable. For example, it could be noticed that when variable *var_1* is out of the lower limit, variable *var_2* is often out of the upper limit at the same time.

The form of the association rule among deviant events is $A \Rightarrow B$, where A and B are itemsets of the given deviant event e. A deviant event, denoted by $e = (var_j,\ code^{percent})$, is defined as the pair of a variable name and a deviation code with its associated percentage, where the code can be *top* or *btm*. For example, $e_1 = (temperature,\ top^{10\%})$ means a deviant event that the *temperature* value lies in the top 10% range of the standard normal distribution. Moreover, $\{(high\_velocity\_time,\ btm^{10\%}), (biscuit\_thickness,\ btm^{10\%})\} \Rightarrow \{(maximum\_velocity,\ btm^{10\%})\}$ indicates that when the values of both *high_velocity_time* and *biscuit_thickness* lie in the bottom 10% range, the value of *maximum_velocity* also lies in the bottom 10% range.

To discover association rules, there are a few well-known algorithms: Apriori (Agrawal & Srikant, 1994), Eclat (Zaki, Parthasarathy, Ogihara, & Li, 1997), and FP-Tree (Han, Pei, & Yin, 2000). The algorithms first create frequent itemsets, which are considered symbol baskets in this research, and then generate candidate rules among them, eventually selecting meaningful rules based on a few measures. For the details of algorithms for building association rules, see Chapter 6 of Tan et al. (2006).

Representative measures for evaluating the generated rules are support, confidence, and lift. Suppose A and B are the sets of deviant events on the antecedent and the consequent sides of an association rule $A \Rightarrow B$. The three evaluation measures are calculated as follows (Hahsler, 2015; Han et al., 2011). The *support* of a rule is the relative frequency of the co-occurrence of two deviant events, A and B, and so it is calculated by dividing the frequency of baskets containing both A and B, $f(A \cup B)$, by the number of time segments W.

$$supp(A \Rightarrow B) = \frac{f(A \cup B)}{W} \tag{3}$$

The *confidence* of a rule measures the likelihood of the occurrence of the consequent event B in all symbol baskets that contain the antecedent event A. This measure provides the reliability of the association rule. It can be transformed into the ratio between the support for the rule and the support for antecedent event A.

$$conf(A \Rightarrow B) = \frac{f(A \cup B)}{f(A)} = \frac{supp(A \cup B)}{supp(A)} \tag{4}$$

The *lift* of a rule calculates the ratio between the confidence of the rule and the support for A; that is, it compares the likelihood of B under the occurrence of A and the likelihood of B under no assumption. It can be transformed into another objective measure called the *interest factor*, which is the ratio between the support of the rule and the multiplication of the supports of A and B.

$$lift(\mathrm{A}, \mathrm{B}) = \frac{conf(A \Rightarrow B)}{supp(B)} = \frac{supp(A \cup B)}{supp(A)supp(B)} \tag{5}$$

The significance of candidate association rules can be evaluated using those evaluation measures (Tan et al., 2006). As a result, the discovered association rules can reveal the relationships among the deviant events that occur frequently in the multivariate time series.

4. Experimental results

4.1. Case study: multivariate time series from die-casting manufacturing process

4.1.1. Dataset and data preprocessing

For empirical validation of the proposed method, a manufacturing process dataset collected from a die-casting machine was used. This dataset is a typical multivariate time series. The data were recorded from July 17, 2015, to July 18, 2015, with 1437 lengths, and ten process variables. The conditions of the process appear to the variables. And each variable has a different range of value and distribution relatively.

As the first step, to preprocess of the data, INT was applied to the dataset as shown in Fig. 5. The blue lines describe the value of each variable through time, and the red dotted lines describe the boundary to determine the deviant events. The boundary values ±1.28 indicate the quantiles for area of top 10% and bottom 10% of the normal distribution. We considered that values beyond these boundaries as deviant events. Although INT was applied, certain variables were skewed from the normal distribution because of their tied vales. However, the empirical quantiles of the variables were matched to the quantiles of the normal distribution.

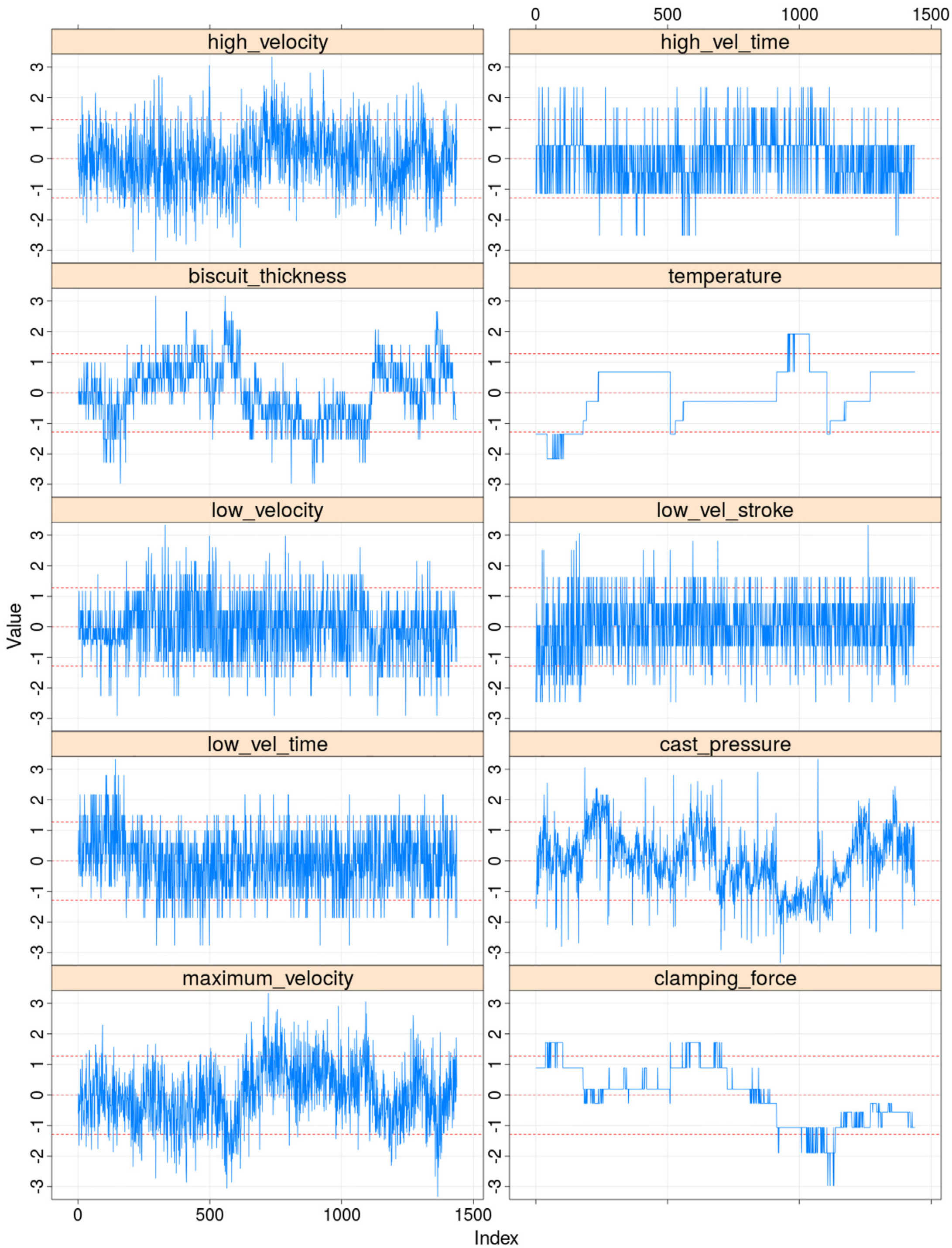

Fig. 5. Time series plot of the transformed data in the manufacturing process.

The next step is SAX for the dimensionality reduction and the discretization. As we described in Section 3.2.2, SAX has the two parameters such as *time segment size* and *alphabet size*. In this experiment, we set the *time segment size* 2 to reduce the dimensionality of the time series because this value does not overly diminish the effect of deviant events (i.e., the data points in each segment becomes the average of the values in the two-time indices). Therefore, the length of the time series was reduced from 1437 to 719.

The *alphabet size* was set to 10 because we considered the values beyond the top 10% and bottom 10% of quantiles as the deviant events. Each letter represents an area corresponding to 10% of the distribution; that is, two letters, *a* and *j*, represent the deviant events, respectively, while the other letters represent the normal state. For example, if the letter of any segment in the variable '*high_velocity*' is *g*, which is the sixth letter, it represents the segment value corresponding to the range between 60% and 70%. Likewise, when the letter of any segment in the variable '*temperature*' is *a*, the value of that segment is less than 10% quantile. In this example, the variable '*high_velocity*' is in the normal state, and the variable '*temperature*' is in the deviant state.

After applying SAX, the symbol baskets that describe the relationships among co-occurring deviant events were generated by

Table 1
Symbol baskets from symbolized multivariate time series.

No.	Symbol baskets
1	{(*temperature*, $btm^{10\%}$), (*low_velocity_stroke*, $btm^{10\%}$), (*cast_pressure*, $btm^{10\%}$)}
2	{(*temperature*, $btm^{10\%}$)}
...	...
230	{(*high_velocity*, $top^{10\%}$), (*biscuit_thickness*, $btm^{10\%}$), (*max_velocity*, $top^{10\%}$)}
231	{(*cast_pressure*, $btm^{10\%}$)}
...	...
462	{(*high_velocity*, $btm^{10\%}$)}
463	{(*high_velocity*, $top^{10\%}$)}

Table 2
Support for deviant events in a single variable.

No.	Variable	supp(var, $top^{10\%}$)	supp(var, $btm^{10\%}$)	Sum
1	*temperature*	0.057	0.146	0.203
2	*cast_pressure*	0.086	0.085	0.171
3	*clamping_force*	0.090	0.067	0.157
4	*biscuit_thickness*	0.061	0.058	0.120
5	*max_velocity*	0.049	0.053	0.102
6	*high_velocity*	0.043	0.038	0.081
7	*low_velocity*	0.036	0.031	0.067
8	*low_velocity_time*	0.035	0.025	0.060
9	*high_velocity_time*	0.024	0.025	0.049
10	*low_velocity_stroke*	0.013	0.029	0.042

using the SBGen algorithm. The symbol baskets listed in Table 1 only consist of the segments containing one or more letters representing deviant events. We re-expressed the letter *a* as '$btm^{10\%}$' and the letter *j* as '$top^{10\%}$' to help the interpretation. As a result, 432 non-empty symbol baskets were generated from all the 719 time segments.

4.1.2. Discovered association rules

The association rules depend on the frequency of events in an itemset. Thus, it is meaningful to check the support of each type of event before applying ARM. Table 2 presents the support of deviant events ($top^{10\%}$ and $btm^{10\%}$) for each variable and their sum from the symbol baskets. The variable '*temperature*' was shown which have the largest proportion of deviant events. It implies that this variable could be frequently included the discovered rules and considered as the significant factor to control the process.

We used an implementation of the Apriori algorithm by Hornik, Grün, and Hahsler (2005) for applying ARM. To find meaningful association rules using the Apriori algorithm, it is necessary to set appropriate parameters. We set the minimum support to 0.01 because we wanted to find rules about deviant events that occur more frequently in multiple variables than in a single variable. And we set the minimum confidence to 0.5 to find rules that hold with more than 50% probability in the presence of any left side condition.

The association rules corresponding to the deviant events are presented in Table 3. We found six association rules and sorted them in descending order of support. Among ten process variables included in the symbol baskets, the rules related to the deviant events of each of variable were found, except '*clamping_force*', '*high_velocity*', and '*low_velocity*'. It means that the deviant events of the three variables are relatively independent for the other variables. We noticed that the discovered rules can be classified into two categories based on the meaning of process variables that constitute the rules.

The first type of rules are explaining the relationship between deviant events in process variables that have the related characteristics in the operation. For example, 'Rule 6′' describes that '*high_vel_time*' and '*max_velocity*' frequently have deviant events in the bottom area at the same time. Both variables have measured the values related to the velocity of the screw in the die-casting machine. The support of this rule is about 0.013, which accounts for about 1.3% of the symbol baskets. The confidence of this rule is about 0.9, which means that in the 90% of the time segments where the value of '*high_vel_time*' is in the bottom area, the value for '*max_velocity*' is also in the bottom area.

The second type of rules contain more interesting information about the deviant events from multiple time series variables. These association rules describe deviant events in the process variables with semantically different information. For example, 'Rule 5′' explains that deviant events in the bottom area of '*high_vel_time*' frequently occur at the same time as deviant events in the top area of '*biscuit_thickness*'. Although these two variables have measured different values, a latent relationship between the occurrences of the deviant events can be inferred. The confidence of this particular rule is 1.0, which indicates that deviant events in the right-side variable always occur at the same time as deviant events in the left-side variable. In other words, this rule describes the useful pattern in the viewpoint of process control; when '*high_vel_time*' was relatively low, '*biscuit_thickness*' of the products was thicker than the normal state.

Additional information could be gained from the association rules. That is, the direct and indirect relationships among the deviant events in the process variables are indicated in the rules. 'Rule 1′', 'Rule 2′', 'Rule 3′', 'Rule 4′', and 'Rule 5′' all contain variables with different meanings. However, some process variables are shared among these rules. For example, the deviant events of '*low_vel_time*' and '*low_vel_stroke*' have co-occurred with deviant events of '*temperature*' in 'Rule 3′' and 'Rule 4′' directly. And the deviant events of *'biscuit_thickness'* can occur at the same time with the deviant events of *'high_vel_time'* through *'max_velocity'* in 'Rule 1′' and 'Rule 6′'.

These relationships can be expressed more intuitively with the graph-based visualization techniques (Zhao & Bhowmick, 2003). The graph for the discovered rules shown in Fig. 6 was generated by using the algorithm introduced in Hahsler and Chelluboina (2011). A circle represents a rule, and a rectangle represents the variable contained in the rule. The rectangle at which an arrow starts is on the left side of a rule, and the rectangle which an arrow points is on the right side of a rule. The size of the circle increases proportionally to the support of the rule, and the darkness of the circle also increases to the confidence of the rule. In the lower part of Fig. 6, three rules and three variables are connected directly or indirectly, illustrating the relationships described in the preceding paragraph.

4.2. Sensitivity analysis for SAX parameters

The proposed method uses two preprocessing techniques, INT and SAX, to find association rules among deviant events in multivariate time series. Therefore, the quality of the association rules is greatly affected by the output from the preprocessing step. In particular, SAX transforms the original time series into a symbolic representation through dimensionality reduction and discretization.

Table 3
Association rules about deviant events from process variables (*time segment size* = 2 and *alphabet size* = 10).

No.	Rule	Support	Confidence	Lift
1	$\{(max\_velocity,\ btm^{10\%})\} \Rightarrow \{(biscuit\_thickness,\ top^{10\%})\}$	0.033	0.632	10.321
2	$\{(temperature,\ top^{10\%})\} \Rightarrow \{(cast\_pressure,\ btm^{10\%})\}$	0.032	0.561	6.612
3	$\{(low\_vel\_time,\ top^{10\%})\} \Rightarrow \{(temperature,\ btm^{10\%})\}$	0.026	0.760	5.204
4	$\{(low\_vel\_stroke,\ btm^{10\%})\} \Rightarrow \{(temperature,\ btm^{10\%})\}$	0.017	0.571	3.913
5	$\{(high\_vel\_time,\ btm^{10\%})\} \Rightarrow \{(biscuit\_thickness,\ top^{10\%})\}$	0.014	1.000	16.341
6	$\{(high\_vel\_time,\ btm^{10\%})\} \Rightarrow \{(max\_velocity,\ btm^{10\%})\}$	0.013	0.900	17.029

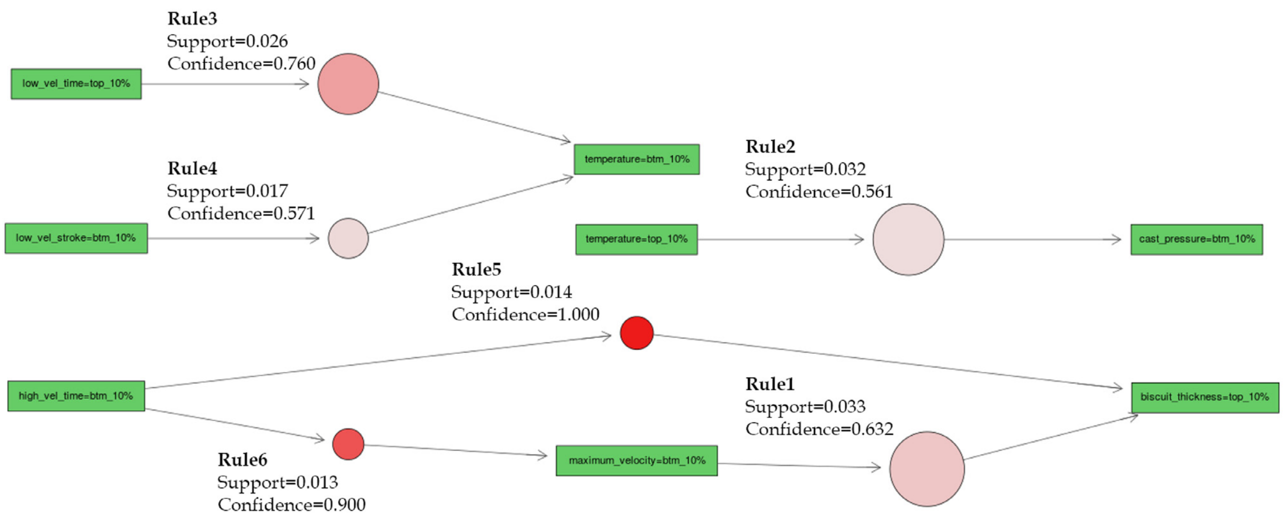

Fig. 6. Graph of discovered association rules for deviant events.

Table 4
Number of discovered rules per *alphabet size* and *time segment size*.

Alphabet size	Time segment size			Subtotal
	2	4	8	
4	849	421	230	1500
6	311	142	49	502
8	176	72	32	280
10	122	56	27	205
Total	1458	691	338	2487

The abstraction level of SAX depends on parameters such as *alphabet size* and *time segment size*. To identify the performance of the proposed method along these parameters, we primarily conducted the running time analysis for SBGen algorithm. Moreover, we also performed sensitivity analysis for the quality of the resulted association rules with different values of SAX parameters. We ran this analysis on Intel® Xeon® CPU E5-2640 2.5 GHz and 160GB RAM using R 3.44 (R core team, 2018).

SBGen algorithm generates symbol baskets of variables over time from multivariate time series data. When the number of the variables is fixed, only the *time segment size* affects its running time. We repeated the measurement of running times ten times per time segment size to measure the running time, and this result is shown in Fig. 7. The highest average of the running time is 0.022 s at *time segment size* 2, and the lowest value is 0.083 s at *time segment size* 8. If the *time segment size* of a time series increases, the number of symbol baskets decreases. As a result, the running time is also shortened as presented in Fig. 7.

The number of discovered rules with the different values of two SAX parameters is presented in Table 4. For this experiment, we set the minimum support to 0.001 and the minimum confidence to 0.1 in the parameters of the Apriori algorithm to discover as many

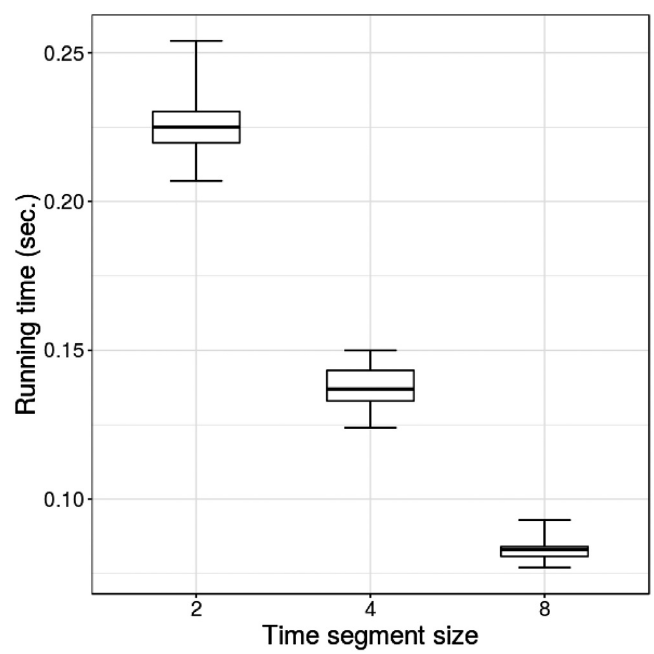

Fig. 7. Box-plot of running times of SBGen algorithm. The ends of whiskers mean minimum and maximum values of the ten-time running experiments.

rules as possible. In general, more rules were discovered when the *time segment size* became smaller for dimensionality reduction or the *alphabet size* decreased for discretization. It means that deviant events were not diluted by the normal state values when the value of the *time segment size* was small. Likewise, the normal state val-

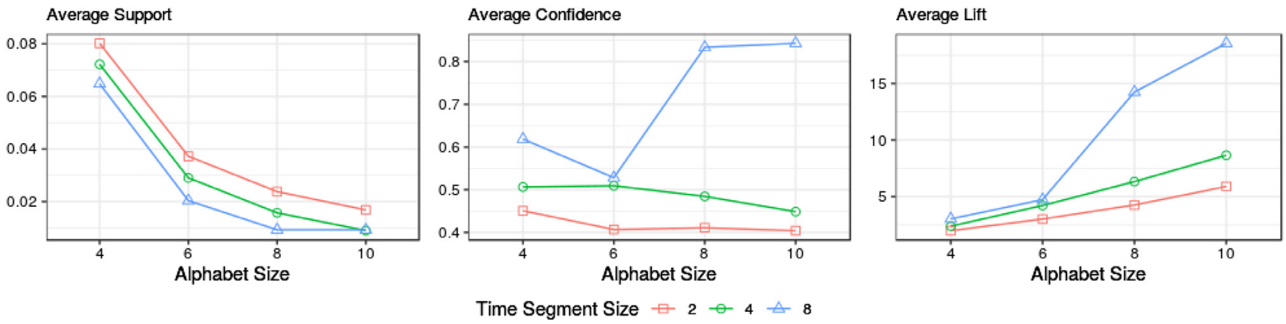

Fig. 8. Average performance of example rules per *alphabet size*.

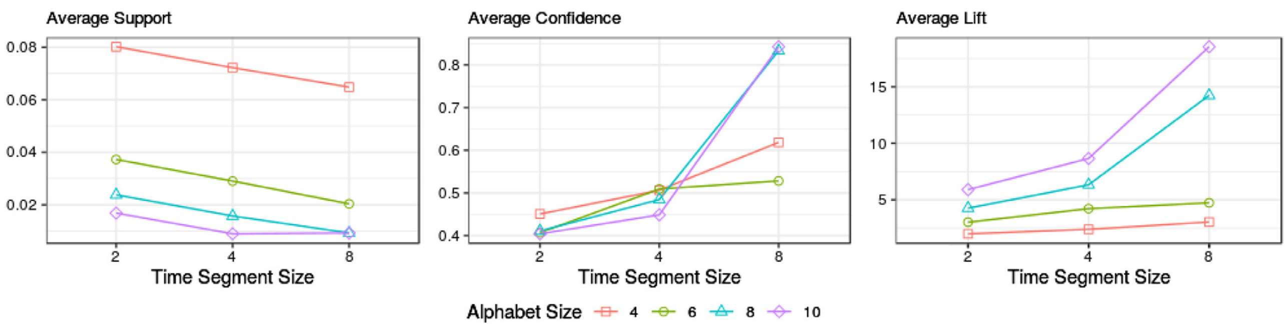

Fig. 9. Average performance of example rules per *time segment size*.

ues were less captured in the deviant event area when the value of the *alphabet size* was small.

It is also important to analyze how SAX parameters affect the quality of rules. This sensitivity analysis was conducted by observing changes in the performance measures such as support, confidence, and lift for each rule. We have observed nine rules discovered in all different settings of the SAX parameters, which are listed in Table 5. We could identify the trends of the performance measures for association rule A ⇒ B corresponding to the settings of the *time segment size* and the *alphabet size*.

First, we analyzed a case that the *time segment size* is fixed and the *alphabet size* increases. In this case, the number of the discretization area increased and its area shrank. As a result, the proportion of time segments that include deviant events in the symbol basket decreased. Therefore, $supp(A)$, $supp(B)$, and $supp(A{\Rightarrow}B)$ decreased monotonically. However, the confidence and the lift fluctuated for each rule. Because the decreasing rates of $supp(A)$ and $supp(B)$ were irregular in the viewpoint of the *alphabet size*.

Fig. 8 shows the average values of performance measures per *alphabet size* for each *time segment size* for rules in Table 5. As the *alphabet size* increases, the support decreases and the confidence shows irregular patterns, as described in the preceding paragraph. However, the lift shows increasing patterns for all values of *time segment sizes* when the *alphabet size* increases. It happens that the relative correlation indicated by the lift increases since the number of deviant events included in the symbol baskets decreases.

Next, we increased the *time segment size* while keeping the *alphabet size*. As the time segment size increases, each time segment contains more observations. The observations in each time segment are usually in the normal state more often than in the deviant state. Therefore, as the *time segment size* increases, the ratio of the time segments that include deviant events in the symbol basket will decrease. In this case, $supp(A)$, $supp(B)$, and $supp(A{\Rightarrow}B)$ will generally decrease. However, the confidence and lift gradually increase.

Fig. 9 shows the values of performance measures per *time segment size* for each *alphabet size* for the rules in Table 5. Excepting when the *alphabet size* is 10, the support decreases as the *time segment size* increases. The confidence increases with the *time segment size*. When the *time segment size* is 8, the order of confidence changes. The lift also shows an increasing pattern with increasing *time segment size*. This is similar to the previous case because the number of time segments including deviant events decreases as the *time segment size* increases.

In both cases described above, the drifts are observed in the confidence values when the *time segment size* was 8. As the *time segment size* passes a certain level, the symbol baskets come to include the time segments which contain more deviant events than normal states. Therefore, the deviant events momentarily occurred might not belong to the symbol baskets. Furthermore, the reduction rate in the support of the left side in the rule was larger than the right side when the *time segment size* was 8. Likewise, this phenomenon also happened by the value of *alphabet size* because the support for the left side increases larger than the right side as the *alphabet size* increases.

Therefore, we empirically identified that the appearance of rules, support, confidence, and lift are affected by the SAX parameters. In practice, if we want to discover many rules or rules with high support, we can set both the *alphabet size* or the *time segment size* to low values. On the other hand, if we want to discover rules with high confidence, we can consider increasing the *time segment size*. In this way, it will be able to discover meaningful rules by setting SAX parameters appropriately.

5. Conclusions and future work

In this study, we proposed a method for finding and analyzing rules about the outliers that frequently occurred in multivariate time series from manufacturing processes. By applying INT and SAX to the data preprocessing, we achieved effective dimensionality reduction and discretization. We proposed SBGen algorithm to make a set of symbol baskets as a data structure for effective rule discovery.

Table 5
Performance of example rules per *alphabet size* and *time segment size*. Rules are sorted by average support.

No	Rule	Time segment size	2				4				8				Avg.
		Alphabetsize (percent)	4 (25%)	6 (16.7%)	8 (12.5%)	10 (10%)	4 (25%)	6 (16.7%)	8 (12.5%)	10 (10%)	4 (25%)	6 (16.7%)	8 (12.5%)	10 (10%)	
1	$\{(max\_velocity,\ btm^{percent})\} \Rightarrow$	Supp	0.135	0.065	0.043	0.033	0.122	0.069	0.031	0.022	0.111	0.056	0.033	0.033	**0.063**
	$\{(biscuit\_thickness,\ top^{percent})\}$	Conf	0.708	0.610	0.608	0.632	0.721	0.781	0.611	0.615	0.800	0.667	1.000	1.000	**0.729**
		Lift	2.737	4.026	6.243	10.321	3.019	5.740	7.857	11.660	3.512	5.455	15.000	20.000	**7.964**
2	$\{(max\_velocity,\ top^{percent})\} \Rightarrow$	Supp	0.102	0.042	0.024	0.010	0.097	0.031	0.017	0.006	0.106	0.017	0.006	0.006	**0.038**
	$\{(biscuit\_thickness,\ btm^{percent})\}$	Conf	0.483	0.353	0.327	0.200	0.530	0.367	0.316	0.222	0.633	0.250	0.167	0.250	**0.342**
		Lift	2.033	2.616	3.220	3.424	2.121	2.694	3.920	4.000	2.533	2.045	2.143	4.091	**2.903**
3	$\{(low\_vel\_time,\ top^{percent})\} \Rightarrow$	Supp	0.071	0.046	0.035	0.026	0.061	0.033	0.025	0.017	0.056	0.028	0.006	0.006	**0.034**
	$\{(temperature,\ btm^{percent})\}$	Conf	0.436	0.611	0.694	0.760	0.629	0.857	0.900	0.857	1.000	1.000	1.000	1.000	**0.812**
		Lift	2.035	4.145	4.710	5.204	2.939	5.822	6.231	6.050	4.737	6.923	7.200	7.500	**5.291**
4	$\{(high\_velocity,\ btm^{percent})\} \Rightarrow$	Supp	0.097	0.043	0.025	0.018	0.083	0.028	0.011	0.006	0.072	0.017	0.006	0.006	**0.034**
	$\{(biscuit\_thickness,\ top^{percent})\}$	Conf	0.579	0.508	0.500	0.481	0.625	0.667	0.500	0.500	0.684	0.750	1.000	1.000	**0.650**
		Lift	2.236	3.352	5.136	7.868	2.616	4.898	6.429	9.474	3.004	6.136	15.000	20.000	**7.179**
5	$\{(high\_velocity,\ top^{percent})\} \Rightarrow$	Supp	0.071	0.042	0.024	0.017	0.094	0.017	0.011	0.006	0.067	0.011	0.006	0.006	**0.031**
	$\{(max\_velocity,\ top^{percent})\}$	Conf	0.425	0.476	0.415	0.387	0.607	0.286	0.400	0.500	0.706	0.286	1.000	1.000	**0.541**
		Lift	2.024	4.028	5.733	7.952	3.312	3.429	7.579	20.000	4.235	4.286	30.000	45.000	**11.465**
6	$\{(high\_velocity,\ btm^{percent})\} \Rightarrow$	Supp	0.085	0.038	0.024	0.018	0.064	0.031	0.017	0.006	0.056	0.017	0.006	0.006	**0.030**
	$\{(max\_velocity,\ btm^{percent})\}$	Conf	0.504	0.443	0.472	0.481	0.479	0.733	0.750	0.500	0.526	0.750	1.000	1.000	**0.637**
		Lift	2.646	4.133	6.657	9.110	2.828	8.250	15.000	13.846	3.789	9.000	30.000	30.000	**11.272**
7	$\{(max\_velocity,\ btm^{percent})\} \Rightarrow$	Supp	0.067	0.028	0.021	0.013	0.058	0.031	0.017	0.011	0.056	0.017	0.011	0.011	**0.028**
	$\{(cast\_pressure,\ top^{percent})\}$	Conf	0.350	0.260	0.294	0.237	0.344	0.344	0.333	0.308	0.400	0.200	0.333	0.333	**0.311**
		Lift	1.636	1.906	2.858	2.747	1.823	2.813	3.529	4.260	1.946	1.714	4.000	6.000	**2.936**
8	$\{(low\_vel\_time,\ top^{percent})\} \Rightarrow$	Supp	0.047	0.015	0.007	0.007	0.036	0.008	0.008	0.006	0.028	0.017	0.006	0.006	**0.016**
	$\{(biscuit\_thickness,\ btm^{percent})\}$	Conf	0.291	0.204	0.139	0.200	0.371	0.214	0.300	0.286	0.500	0.600	1.000	1.000	**0.425**
		Lift	1.222	1.510	1.368	3.424	1.486	1.574	3.724	5.143	2.000	4.909	12.857	16.364	**4.632**
9	$\{(high\_velocity,\ btm^{percent})\} \Rightarrow$	Supp	0.047	0.017	0.013	0.010	0.033	0.014	0.006	0.003	0.033	0.006	0.006	0.006	**0.016**
	$\{(cast\_pressure,\ top^{percent})\}$	Conf	0.281	0.197	0.250	0.259	0.250	0.333	0.250	0.250	0.316	0.250	1.000	1.000	**0.386**
		Lift	1.312	1.443	2.429	3.007	1.324	2.727	2.647	3.462	1.536	2.143	12.000	18.000	**4.336**
Avg.		**Supp**	**0.080**	**0.037**	**0.024**	**0.017**	**0.072**	**0.029**	**0.016**	**0.009**	**0.065**	**0.020**	**0.009**	**0.009**	
		Conf	**0.451**	**0.407**	**0.411**	**0.404**	**0.506**	**0.509**	**0.484**	**0.449**	**0.618**	**0.528**	**0.833**	**0.843**	
		Lift	**1.987**	**3.018**	**4.262**	**5.895**	**2.385**	**4.216**	**6.324**	**8.655**	**3.033**	**4.735**	**14.244**	**18.551**	

We demonstrated the use case of our work with real-life die-casting process data. We found the useful association rules that describe the relationships among deviant events frequently occurred from process variables. Through sensitivity analysis, we also provided a guidance to use the proposed method. However, we note that it depends on data characteristics and the purpose of the analysis.

Monitoring and analyzing the time series data collected in the field carry many difficulties. The proposed method provides frequent patterns for deviant events in multivariate time series in a form that is easy to understand. Therefore, we expect that practitioners will be able to efficiently and effectively manage their process using the rules found through the proposed method. Furthermore, the process can be improved by interpreting the information provided by the variables included in each rule.

In future work, if class information could be obtained, the rules that describe the interesting relationship between variables and classes could be discovered. For example, the rules represent the relationship that the defeat of products is related to the deviant event of the variables. It would be helpful to practitioners in manufacturing industry.

Furthermore, the rule discovery considering the sequential property of time series is also an interesting research subject. In this study, we explored the association patterns that only can describe the occurrence of deviant events at the same time segment. However, if the rules express some temporal patterns, it would be more useful to predict the dynamics among deviant events over time. This approach could reveal the hidden information in the time domain.

Acknowledgment

This work was supported by the Smart Factory Advanced Technology Development Program of MOTIE/KEIT (No. 10054508) and the Advanced Training Program for Smart Factory of MOTIE/KIAT (No. N0002429).

Credit authorship contribution statement

Hoonseok Park: Methodology, Formal analysis, Software, Validation, Writing - original draft, Visualization. **Jae-Yoon Jung:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision.

References

Aghabozorgi, S., Shirkhorshidi, A. S., & Wah, T. Y. (2015). Time-series clustering–A decade review. *Information Systems, 53*, 16–38.
Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Record, 22*(2), 207–216.
Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In Proceedings of the 20th international conference on very large data bases (VLDB*): 1215* (pp. 487–499).
Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the 17th international conference on data engineering* (pp. 3–14).
Beasley, T. M., Erickson, S., & Allison, D. B. (2009). Rank-based inverse normal transformations are increasingly used, but are they merited? *Behavior Genetics, 39*(5), 580.
Blom, G. (1958). *Statistical estimates and transformed beta-variables*. New York: Wiely.
Box, G. E., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological), 26*(2), 211–243.
Chang, X., Ma, Z., Lin, M., Yang, Y., & Hauptmann, A. G. (2017). Feature interaction augmented sparse learning for fast kinect motion detection. *IEEE Transactions on Image Processing, 26*(8), 3911–3920.
Chang, X., Yu, Y., Yang, Y., & Xing, E. (2017). Semantic pooling for complex event analysis in untrimmed videos. *IEEE Transactions of Pattern Analysis and Machine Intelligence, 39*(8), 1617–1632.
Chatfield, C. (2003). *The analysis of time series: An introduction*. Chapman and Hall/CRC.
Chen, W., Tseng, S., & Wang, C. (2005). A novel manufacturing defect detection method using association rule mining techn Niques. *Expert Systems with Applications, 29*(4), 807–815.
Choudhary, A. K., Harding, J. A., & Tiwari, M. K. (2009). Data mining in manufacturing: A review based on the kind of knowledge. *Journal of Intelligent Manufacturing, 20*(5), 501.
Das, G., Lin, K., Mannila, H., Renganathan, G., & Smyth, P. (1998). Rule discovery from time series. In *Proceedings of the 4th international conference on knowledge discovery from databases (KDD): 98* (pp. 16–22). (1).
Esling, P., & Agon, C. (2012). Time-series data mining. *ACM Computing Surveys (CSUR), 45*(1), 12.
Fu, T. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence, 24*(1), 164–181.
Hahsler, M. (2015). *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*, http://michael.hahsler.net/research/association_rules/measures.html.
Hahsler, M., & Chelluboina, S. (2011). Visualizing association rules: Introduction to the R-extension package arulesViz. *R Project Module*, 223–238.
Han, J., Pei, J., & Kamber, M. (2011). Data mining: Concepts and techniques. Elsevier.
Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM SIGMOD Record, 29*(2), 1–12.
Harding, J. A., Shahbaz, M., & Kusiak, A. (2006). Data mining in manufacturing: A review. *Journal of Manufacturing Science and Engineering, 128*(4), 969–976.
Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining—A general survey and comparison. *ACM SIGKDD Explorations Newsletter, 2*(1), 58–64.
Hornik, K., Grün, B., & Hahsler, M. (2005). arules-A computational environment for mining association rules and frequent item sets. *Journal of Statistical Software, 14*(15), 1–25.
Jiang, N., & Gruenwald, L. (2006). Research issues in data stream association rule mining. *ACM SIGMOD Record, 35*(1), 14–19.
Kamsu-Foguem, B., Rigal, F., & Mauget, F. (2013). Mining association rules for the quality improvement of the production process. *Expert Systems with Applications, 40*(4), 1034–1045.
Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2001). Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record, 30*(2), 151–162.
Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery, 7*(4), 349–371.
Keogh, E., & Mueen, A. (2011). Curse of dimensionality. In *Encyclopedia of machine learning* (pp. 257–258). Springer.
Khan, S., & Parkinson, S. (2018). Eliciting and utilising knowledge for security event log analysis: An association rule mining and automated planning approach. *Expert Systems with Applications, 113*, 116–127.
Köksal, G., Batmaz, İ., & Testik, M. C. (2011). A review of data mining applications for quality improvement in manufacturing industry. *Expert Systems with Applications, 38*(10), 13448–13467.
Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 2–11).
Lin, J., Keogh, E., Wei, L., & Lonardi, S. (2007). Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery, 15*(2), 107–144.
Martinez-de-Pison, F. J., Sanz, A., Martinez-de-Pison, E., Jimenez, E., & Conti, D. (2012). Mining association rules from time series to explain failures in a hot-dip galvanizing steel line. *Computers & Industrial Engineering, 63*(1), 22–36.
Mitsa, T. (2010). *Temporal data mining*. CRC Press.
Montgomery, D. C. (2009). *Introduction to statistical quality control*. John Wiley & Sons.
Mooney, C. H., & Roddick, J. F. (2013). Sequential pattern mining- approaches and algorithms. *ACM Computing Surveys (CSUR), 45*(2), 19.
Morchen, F., & Ultsch, A. (2005). Optimizing time series discretization for knowledge discovery. In Proceedings of the 11th ACM international conference on knowledge discovery and data mining (SIGKDD) (pp. 660–665).
Morchen, F., & Ultsch, A. (2007). Efficient mining of understandable patterns from multivariate interval time series. *Data Mining and Knowledge Discovery, 15*(2), 181–215.
Parkinson, S., Somaraki, V., & Ward, R. (2016). Auditing file system permissions using association rule mining. *Expert Systems with Applications, 55*, 274–283.
Peterson, R. A., & Cavanaugh, J. E. (2019). Ordered quantile normalization: A semiparametric transformation built for the cross-validation era. *Journal of Applied Statistics*, 1–16.
Pyrcz, M.J., .& Deutsch, C.V. (.2018). Transforming data to a gaussian distribution. In J. L. Deutsch (Ed.), Geostatistics Lessons. Retrieved from http://www.geostatisticslessons.com/lessons/normalscore
R Core Team. (2018). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*. Vienna, Austria URL https://www.R-project.org/ .
Rakesh, A., Giuseppe, P., Edward, L. W., & Mohamed, Z. (1995). Querying shapes of histories. In Proceedings of the 21th international conference on very large data bases *(VLDB)* (pp. 502–514).
Tan, P., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In Proceedings of the 8th ACM international conference on knowledge discovery and data mining (SIGKDD) (pp. 32–41).
Tan, Steinbach, & Kumar (2006). *Introduction to data mining*. Addison Wesley.

Torkamani, S., & Lohweg, V. (2017). Survey on time series motif discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(2).

Wang, H., Yang, J., & Hao, S. (2016). Two inverse normalizing transformation methods for the process capability analysis of non-normal process data. *Computers & Industrial Engineering, 102*, 88–98.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., & Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 1–35.

Weng, J., & Lee, B. (2011). Event detection in twitter. In *Proceedings of the 5th international AAAI conference on weblogs and social media* (pp. 401–408).

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Xue, R., Zhang, T., Chen, D., Le, J., & Lavassani, M. (2016). Sensor time series association rule discovery based on modified discretization method. In *Proceedings of the IEEE international conference on computer communication and the internet (ICCCI)* (pp. 196–202).

Yin, S., Gao, H., & Kaynak, O. (2014). Data-driven control and process monitoring for industrial applications—Part I. *IEEE Transactions on Industrial Electronics, 61*(11), 6356–6359.

Zaki, M. J., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of association rules. In Proceedings of the 3rd international conference on knowledge discovery from databases *(*KDD*): 97* (pp. 283–286).

Zhao, Y., & Bhowmick, S. S. (2003). *Association rule mining with R- A Survey*. CAIS, Nanyang Technological University Technical ReportNo. 2003116.

Zuehlke, D. (2010). SmartFactory—Towards a factory-of-things. *Annual Reviews in Control*, *34*(1), 129–138.