VTT INFORMATION TECHNOLOGY

RESEARCH REPORT TTE1-2000-42

FUME Project

MAPS: A Method for Identifying And Predicting Aberrant Behaviour in Time Series

Version 1.0

November 2000

Evangelos Kotsakis
Antoni Wolski

Version history

Version	Date	Author(s)	Reviewer	Description
0.1-1	9.11.2000	E.K., A.W		First draft
1.0	15.11.2000	E.K., A.W.		Submitted for publication

Contact information

A. Wolski
VTT Information Technology
P.O. Box 1200, FIN-02044 VTT, Finland
Street Address: Tekniikantie 4 B, Espoo
Tel. +358 9 4561, fax +358 9 456 7024
Email: antoni.wolski@vtt.fi
Web: http://www.vtt.fi/tte/fume

Last modified on 5 January, 2001
T:\Fume\Papers\TTE RR-cover.doc

MAPS: A Method for Identifying and Predicting Aberrant Behaviour in Time Series

Evangelos Kotsakis[*]
Joint Research Centre (JRC)
Space Application Institute (SAI)
TP 261, I-21020 Ispra (VA), Italy
Email: kotsakis@acm.org

Antoni Wolski
Technical Research Centre of Finland (VTT)
VTT Information Technology
P.O. Box 1201, 02044-VTT Espoo, Finland
Email: antoni.wolski@vtt.fi

Abstract - We present a method for inducing a set of rules from time series data, which is originated from a system monitoring process. The proposed method is called MAPS (**M**ining **A**berrant **P**atterns in **S**equences) and it may be used in decision support or in control to identify faulty system states. It consists of four parts: training, identification, mining and prediction. In order to improve the flexibility of the event identification, we employ fuzzy sets and propose a method that extracts membership functions from statistical measures of the time series. The proposed approach integrates fuzzy logic and mining in a seamless ways. Some of the existing mining algorithms have been modified to accommodate the need of discovering fuzzy event pattern.

Keywords: modelling behaviour, rule extraction, mining event structures Mining Aberrant Patterns in Sequences

1. Introduction

An industrial process is a series of operations performed in manufacturing or some other industrial activities. Monitoring is used in time dependent industrial processes in order to ensure that the process is effective. Monitoring is particularly important in aligning the process with other processes and ensuring that the process operates according to the specifications. Monitoring usually involves several indicators, which are used to depict the level of a variety of temporal attributes of the process. Many variables are measured during monitoring. Such variables may include pressure, temperature, humidity etc. Measurements are made on some constituent parts of the industrial process, which are considered critical for the operation and stability of the process. Those attributes participating in monitoring are chosen carefully, so that the monitoring is effective and able to identify important aspects of the process. The choice of the attributes, which need to be monitored, is critical and human experts mainly perform it. The monitoring is accomplished by frequently sampling the values of the temporal attributes. The frequency of sampling is user defined and it can range from a few milliseconds to many days.

Suppose we are given such a process *P* with *n* temporal attributes $a_1, a_2, ..., a_n$, each of which is sampled every τ time units (one time granule) at time points $t_1, t_2, ..., t_j, ...$, where $t_j=j\tau$. Then, the observation data may be viewed as time series like the ones shown in Figure 1. Each attribute observation constitutes a single time series. A data element $d(i,j)$ indicates the value of the attribute a_i at time point t_j.

One of the challenges is to discover important events hidden in this time series data. Consequently the question becomes: how is it possible to utilise this time series data in such a way that we are able (1) to identify abnormal events in a process and (2) to describe or predict the behaviour of a sequence of such abnormal events that may lead to failures?

Therefore, the main goal focuses on the development of a systematic way to identify and predict abnormal events and event patterns in time sequences. We are particular interested in methods that require no a priori knowledge and allow us to obtain the above goal by studying monitoring information acquired by measuring time varying attributes.

[*] Research supported by the fellowship program of the European Research Consortium for Informatics and Mathematics (ERCIM).

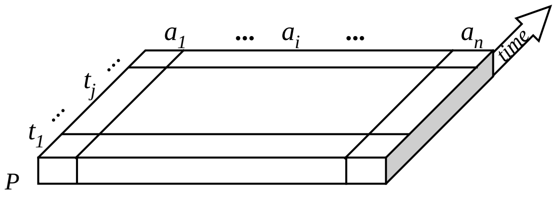

Figure 1. Temporal data

This paper describes a method (named MAPS - Mining Aberrant Patterns in Sequences) that determines the necessary steps, we need to perform, in order to develop a knowledge-based system, which is able to identify abnormal events and deduce rules for describing or predicting the behaviour of such abnormalities. The knowledge is acquired from observed (aberrant) behaviour during the lifetime of the industrial process. No a priori knowledge is required. The proposed method provides fully adapted mechanisms that allow the knowledge base to be enriched during the lifetime of the process.

Our main goal focuses on discovering knowledge concerning abnormal behaviour of a process from time series data. Towards this goal, we start directly with multiple data sequences, which represent measurements of various attributes of the monitored process. Then, the data sequences are sensed and abnormal events are identified. Most of previous works have assumed that row data are somehow processed to generate event sequences. The MAPS method employs a fuzzy set technique to identify aberrant events and generate event sequences. An event sequence is then mined for interesting patterns and rules are extracted from these patterns. Prediction is obtained by matching the most recent events of the sequence with the antecedent part of such rules.

The rest of the paper is organised as follows: Section 2 presents an outline of the MAPS method and discusses briefly its constituent components. Section 3 discusses related work. In section 4, we present how regular behaviour is captured in time series by employing fuzzy sets; and then how these fuzzy sets are utilised for identifying abnormal events. Section 5 discusses the event mining technique we use to discover frequent fuzzy event patterns as well as the method for extracting rules that involve event patterns. Section 6 presents a method for predicting aberrant events. We conclude in section 7 by summarising the contributions and pointing out some further work to be done.

2. Outline of the Method

The proposed method employs fuzzy sets for identifying aberrant events and mining techniques for extracting rules from the event sequences. The method is divided into four steps. Each step constitutes a separate unit. The outline of the MAPS method is shown in Figure 2. The four units are as follows:

Training unit: It collects statistical information from measurements. Based on this information, the training module defines the membership functions of fuzzy sets that describe the regularity of the attribute values. The membership functions of the fuzzy set are stored in the database of the training unit.

Identification unit: The membership functions of the fuzzy sets in the training unit are used to identify abnormal values that may cause aberrant events. Each aberrant event is identified by its type, its occurrence time and its intensity. An aberrant event is a fuzzy event whose intensity specifies the level of abnormality. Sequences of such fuzzy events are stored in the database of the identification unit.

Event mining unit: This unit has two constituent parts; the first one identifies frequent event patterns in the event sequences stored in the identification unit and the second one extract rules from the set of frequent patterns. Once the most frequent patterns are known, they are used to obtain rules that associate close related event patterns. The rules are stored in the rule base of the event-mining unit.

Prediction unit: It predicts a possible future event pattern by studying the occurrence of the event patterns during the last few observation windows. That is, if an event pattern occurs in one of the most recent windows, then the predicting module checks whether such a pattern matches the premise of a rule in the rule repository; and if so, it suggests the conclusion of the matched rule as a possible future behaviour.

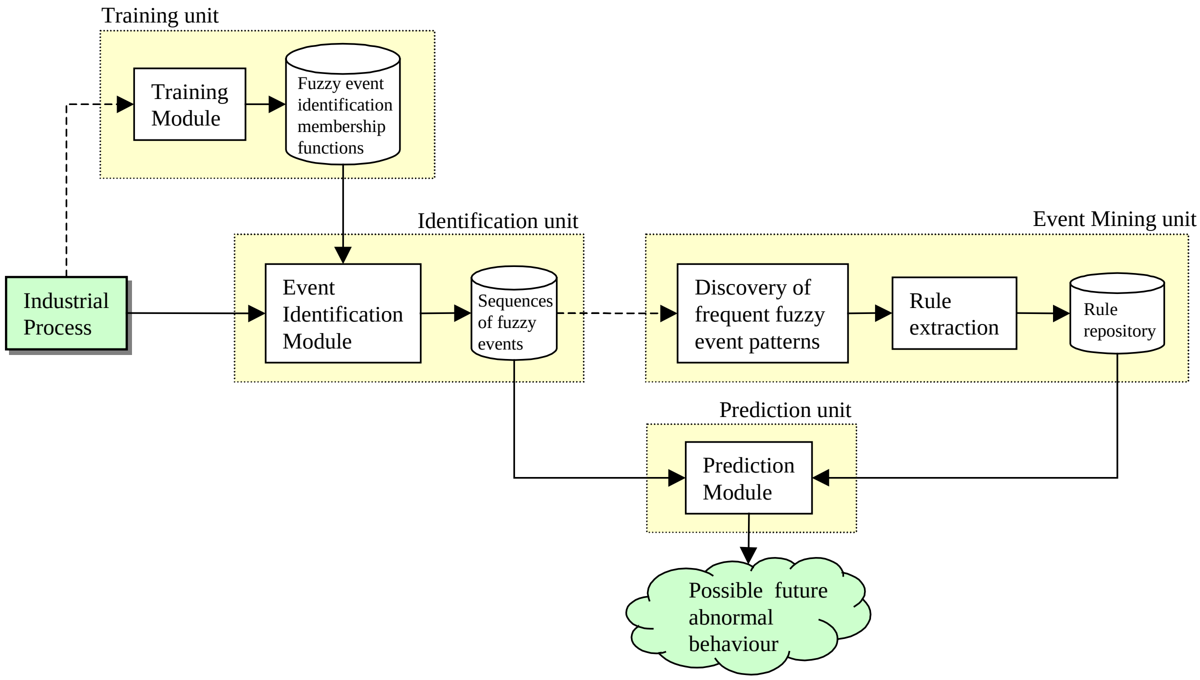
Training unit
Training Module
Fuzzy event identification membership functions
Identification unit
Industrial Process
Event Identification Module
Sequences of fuzzy events
Event Mining unit
Discovery of frequent fuzzy event patterns
Rule extraction
Rule repository
Prediction unit
Prediction Module
Possible future abnormal behaviour

Figure 2. Layout of the MAPS method.

The arrows in Figure 2 represent data or information flows. The dotted arrows depict temporal flows, which are active only when the receiving module needs data for accomplishing its task. Information does not continuously run over dotted flows. For example, the flow connecting the industrial process and the training unit is active only during a training session when measurement data is used for determining the membership functions of the fuzzy sets. Upon completion of the training, this flow becomes inactive. Data flows from the identification unit if further mining in event sequences is necessary to update or refine the rule repository. Solid arrows depict continuous flow of data between connected modules.

3. Related Work

A lot of work has been done in the area of Artificial Intelligence for discovering patterns in sequential data (see for example [3, 6]). In the context of databases, the problem has been studied in a number of recent papers [1, 8, 2]. Our work with respect to mining event sequence is more related to [8] where event sequences are searched for frequent patterns. We use the same concept of windows for estimating the frequency of the patterns. However, we have extended the notion of pattern occurrence by introducing the concept of fuzzy events and fuzzy patterns. In order to deal with fuzzy patterns, we have introduced a new definition of pattern frequency. Our algorithms for finding the set of the most frequent patterns are based on those in [8]. However, the algorithms have been adapted to deal with fuzzy events and fuzzy patterns. In [1] the problem of discovering sequential patterns is considered over a customer transaction database. The strategy in [1] is similar to that in [8] and it starts with simple sub-patterns and incrementally builds longer sequence candidates for the discovery process. In [2] the discovery process considers more complex patterns where events may be in terms of different granularities and patterns may include temporal distances between the events in the patterns.

In the context of event identification, there has been recently an effort in Knowledge Discovery and Data mining (KDD) research area, which aims at generating such event sequences [4, 5, 7]. In [4], the problem of activity monitoring is discussed and a framework for evaluating activity monitoring is proposed for applications such as cellular phone fraud detection. However, in [4] the focus is to defining the functions used for scoring false alarms. The proposed approach in [5] focuses on how to detect events from phenomena, which exhibit a dynamic behaviour. This seems to be close to our goal, which focuses on identifying aberrant events. However, [5] investigates the potential to identify the time points at which the behaviour changes (change-point detection). Towards this, [5] suggests that there is a need to determine the number of change points and then to find the functions that match the behaviour between two successive change-points. In our approach change points are identified automatically by using the minimum, maximum and statistical mean values of an attribute, which are obtained during a training session. The idea of using fuzzy sets for mining association rules and frequent episodes is also discussed in [7]. However, the approach in [7] is based on attributing the events in a similar way as a customer transaction is attributed by items. In that sense, an event is not atomic, as in our approach, but rather a vector quantity whose attributes are assigned a membership degree. This approach is close to detecting abnormal behaviour (fraud or intrusion) over a customer transaction database. Our approach considers the incoming data as time series data and assigns event types to each time series. Then, an abnormal event is generated by the occurrence of an abnormal value of a single attribute. This helps to quickly identify the abnormality in the industrial process by studding the type of the event occurred and it may be used to suggest possible future abnormal events that might be caused by abnormal values of the corresponding attributes.

4. Fuzzy identification of abnormal behaviour

The identification of abnormal events is obtained by way of training the system using measurement data. The result of a training session is a group of fuzzy sets used to identify abnormal values. The membership functions of the fuzzy sets are particularly employed for identifying these abnormal values, which consequently yield the occurrence of aberrant fuzzy events whose membership values determines the level of abnormality.

4.1. Training the system for event identification

The aim of training is to extract fuzzy sets from observation data. Each fuzzy set describes the normal behaviour of an attribute of the industrial process. The method described in this section constitutes the main procedure of the training module in Figure 2. The observation data of each attribute is viewed as a time series. The duration of the training period is *M* time granules and it is called *training session*. During the training session every single value of each attribute is observed and registered in a time sequence. Figure 3 shows the time series obtained by observing the attribute a_i during a training session of *M* time granules. For each attribute a_i, three values are maintained at the end of the training session. These are the *minimum* value (a_i^{min}), the *maximum* value (a_i^{max}) and the *mean* value (a_i^{mean}) of the attribute a_i, which are defined as follows:

$$a_i^{\min} = Min_{j=1}^{M}\{a_i(j\tau)\}$$

$$a_i^{\max} = Max_{j=1}^{M}\{a_i(j\tau)\}$$

$$a_i^{mean} = \sum_{j=1}^{\mathrm{M}} \frac{a_i(j\tau)}{M}$$

These values form an indicator that is used to identify acceptable values of an attribute. In some industrial processes we encounter cases where some exceptional values occur beyond the maximum or below the minimum value. These exceptional values are called outliers and they are treated as noise in our case. A pre-processing is needed to remove this noise during the training session. The set of values $\{a_i^{min}, a_i^{max}, a_i^{mean}\}$ of an attribute a_i is called the *index* of a_i. Each attribute index is stored as an entry in a relation in the database of the training unit (see Figure 2). The schema of the relation, which is used for storing attribute indices is the following:

AIndex(a:AttributeName, a^{min}*:AttributeMinimum,* a^{max}*:AttributeMaximum,* a^{mean}*:AttributeMean).*

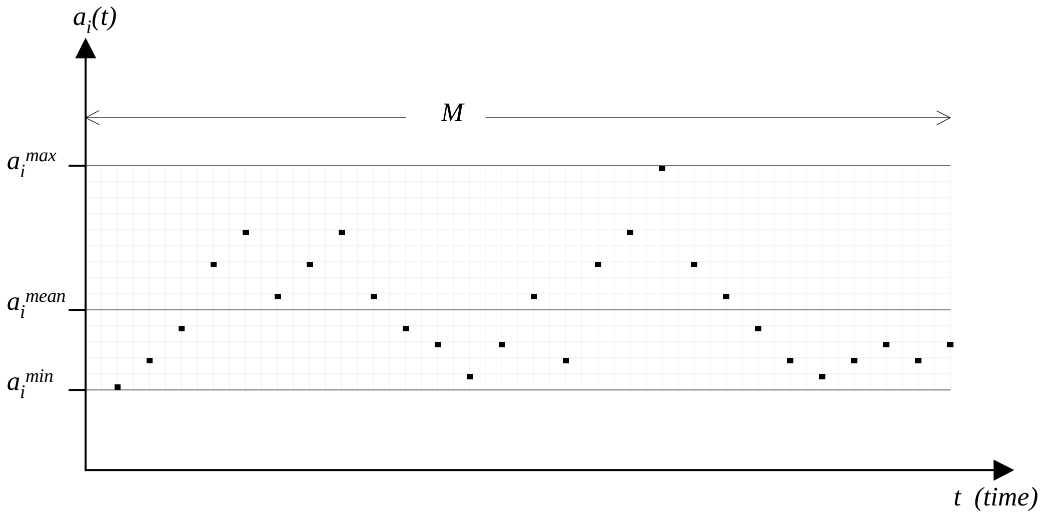

Figure 3. Observation of the attribute a_i during a training session of M time granules.

The index of the attribute a_i is used to describe acceptable values of a_i. For example, values of a_i, which are within the interval [a_i^{min}, a_i^{max}] are considered perfectly normal, while other values of a_i are consider abnormal. The level of abnormality is expressed by a fuzzy membership function, which assigns a number between 0 and 1 to such an abnormal value of a_i. The more distant the value of a_i is from the interval [a_i^{min}, a_i^{max}], the more abnormal the value of a_i is. The generation of such a membership function is performed as follows:

First, a fuzzy set that describes the normal behaviour of the attribute a_i is defined by using the index of the attribute a_i and a threshold $0 \leq \vartheta \leq 1$. The threshold ϑ determines how the intensity that characterises abnormal values changes with respect to a_i. Knowing the index of an attribute a_i (i.e. a_i^{min}, a_i^{max}, a_i^{mean}) and the threshold ϑ, one can form a fuzzy set A_i, which describes the regularity of the values of the attribute a_i. Figure 4 shows a triangular membership function of the fuzzy set A_i. In fact, the membership function of the fuzzy set A_i may be any unimodal function like trapezoidal or bell typed. For the sake of clarity we employ only triangular membership functions in this paper. The triangle A_i is formed by considering the following points: (a_i^{min}, ϑ), (a_i^{mean},1) and (a_i^{max}, ϑ).

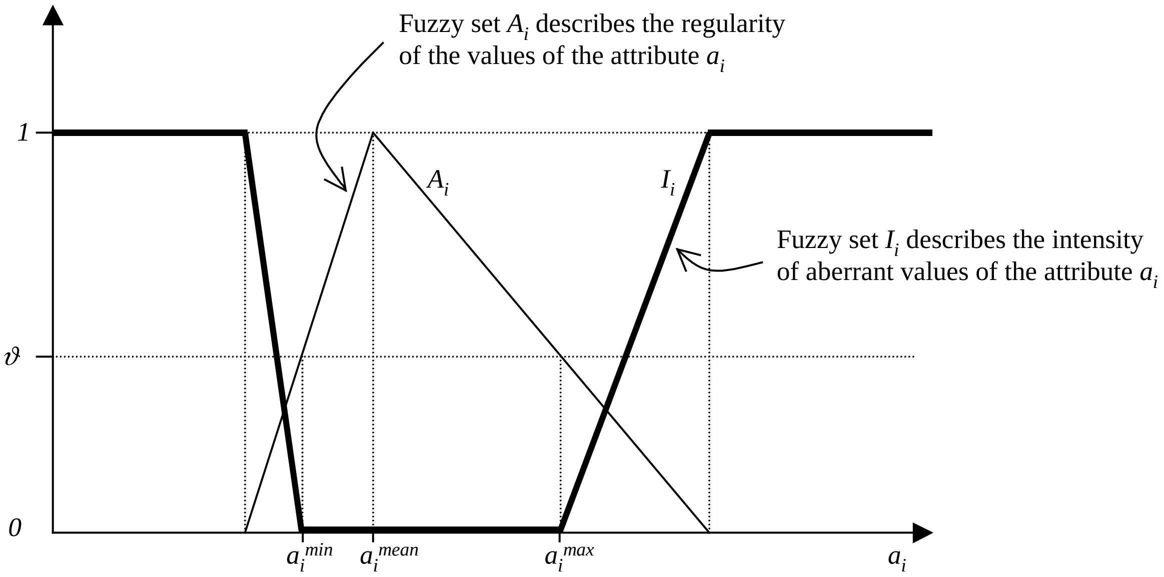

Figure 4. Fuzzy sets A_i and I_i when $0 < \vartheta < 1$. I_i is used to identify abnormal values of the attribute a_i . The majority of regular values of a_i is between a_i^{min} and a_i^{max} . The more distant a value of a_i is from the interval [a_i^{min} , a_i^{max}], the more intense the aberrant event is.

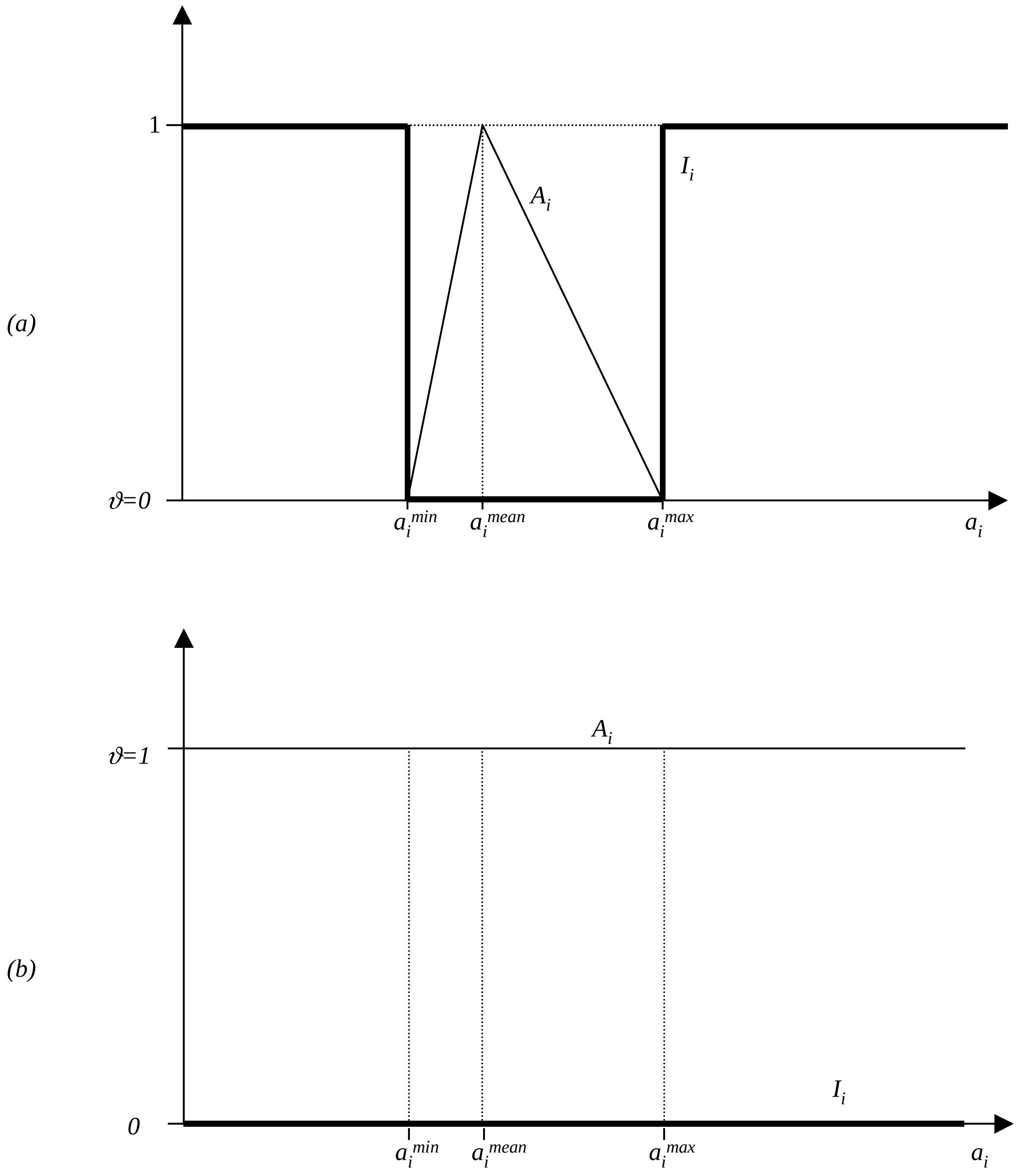

Figure 5. Fuzzy sets A_i and I_i when (*a*) ϑ=0 and (*b*) ϑ=1.

Then, a second fuzzy set I_i that describes the intensity of an abnormal value of a_i is defined by using the following membership function:

$$I_i\left(a_i(t)\right) = \begin{cases} 0 & \text{if } A_i\left(a_i(t)\right) \geq \vartheta \\ 1 - \dfrac{A_i\left(a_i(t)\right)}{\vartheta} & \text{if } A_i\left(a_i(t)\right) < \vartheta \end{cases}$$

Where, $a_i(t)$ is the value of the attribute a_i at time t, $I_i(a_i(t))$ is the membership value that expresses the level of abnormality of $a_i(t)$, and $A_i(a_i(t))$ is the membership value that expresses the level of regularity of $a_i(t)$. The above definition is for $0< \vartheta <1$. In case ϑ=0, $I_i(a_i(t))$ is given by

$$I_i\left(a_i(t)\right) = \begin{cases} 0 & \text{if } A_i\left(a_i(t)\right) > 0 \\ 1 & \text{if } A_i\left(a_i(t)\right) = 0 \end{cases}$$

Figure 5-*a* shows the membership functions of A_i and I_i when ϑ =0. In this particular case, I_i fuzzy set converges into a crisp set, which identifies all the values of a_i, which are outside of the interval [a_i^{min}, a_i^{max}] as fully abnormal (with membership value equal to 1).

If $\vartheta =1$, then constantly, for all *t*, $A_i(a_i(t))=1$ and consequently $I_i(a_i(t))=0$ (see Figure 5-*b*). This means that none of the values of a_i is abnormal when $\vartheta =1$. The parameter ϑ specifies the sensitivity of the system in identifying abnormal values. When $\vartheta =1$ the system is fully indifferent; none of the values is identified as abnormal, no matter how far away from the normal level the value is. When $\vartheta =0$ the system is fully sensitive, which means that if a value of the attribute a_i appears to be outside the interval [a_i^{min} , a_i^{max}], no matter how far away, then the value of a_i is considered abnormal with maximum intensity (i.e. 1). The value of ϑ usually is between 0 and 1 excluding the end points.

As we have previously mentioned, the aim of the training module is to observe the attribute time sequences for a given time period (training session) and then derive the attribute indices at the end of the training session. The threshold ϑ and the attribute indices are then used to define the membership functions of the fuzzy sets A_i and I_i ($1 \le i \le n$). The fuzzy set I_i is then used to identify the intensity of abnormal events associated with the attribute a_i. The following section discuses how this identification is performed.

4.2. Event identification

Let's assume that the industrial process can be observed by monitoring a set of *n* temporal attributes $A=\{a_1, a_2, ..., a_n\}$. At the end of the training session, we are able to specify a set of membership functions $I=\{I_1, I_2, ..., I_n\}$, where I_i ($1 \le i \le n$) describes the extent to which a value of a_i is abnormal. Let $E=\{e_1, e_2, ..., e_m\}$ be a set of event types, which represent distinct kinds of abnormal behaviour. We may define a mapping *f* from *A* to *E* that assigns one or more attributes to one or more events. For example, if $f=\{(a_1, e_3), (a_1, e_4), (a_2, e_3), (a_3, e_2), ...\}$, it means that an abnormal value of a_1 causes a concurrent occurrence of the events e_3, e_4 and the event e_3 may occur due to an abnormal value of either a_1 or a_2. A fuzzy event is identified by using the following criterion:

A value $a_i(t)$ of the attribute a_i causes the occurrence of an event e_j if only if $(a_i, e_j) \in f$ and $I_i(a_i(t))>\omega$, where $0 \le \omega \le 1$.

The threshold ω indicates the minimum intensity that an abnormal value must have in order to cause the occurrence of an event. This has been introduced to disallow the occurrence of events of a very low intensity. However, if we want to allow the occurrence of events of any intensity level, then we may define $\omega=0$. Upon the occurrence of an event, the system registers the data relevant to the event in a database in the identification unit (see Figure 2). The event record includes the following data: (a_i, e_j, t, I_e), where a_i is the attribute whose abnormal value causes the occurrence of the event of type e_j (i.e. $e_j=f(a_i)$), *t* is the time of occurrence, which may be an integer indicating the time granule at which the event occurred and $I_e= I_i(a_i(t))$ is the event intensity. It suffices to represent such a fuzzy event as a triple (e_j, t, I_e). The database in the identification unit stores all the event occurrences forming in that way sequences of fuzzy events. Each event in a sequence has occurred due to abnormal attribute measurements of the industrial process.

The next step is to examine this sequence of events and extract useful knowledge from it by identifying frequent event patterns. This is discussed in the following section.

5. Event Mining Unit

The event-mining unit concerns itself with the discovering of rules that associate fuzzy event patterns. It is divided into two parts; the first one deals with the discovering of the most frequent fuzzy event patterns and the second one deals with the extraction of rules that associate close related fizzy event patterns.

5.1. Discovering frequent event patterns

A fuzzy event is a triple (e_j, t, $I(e_j)$), where $e_j \in E$ is an event type, such that $e_j=f(a_i)$, t is the time at which the event occurs and $I(e_j)=I_i(a_i(t))$ is the intensity of the event. An event sequence is a triple (s, T_s, T_e), where s is an ordered set of events whose first event occurs at T_s and the last event occurs at T_e ($T_e \geq T_s$). An event pattern p is a partial order (B,<), where $B \subseteq E$. A pattern p matches a sequence s, if all of the events in p occur in s in an order respecting the partial order. The event sequence (s_1, T_{s1}, T_{e1}) is contained in the sequence (s_2, T_{s2}, T_{e2}) if only if $T_{s1} \geq T_{s2}$, $T_{e1} \leq T_{e2}$ and $s_1 \subseteq s_2$. Two typical sequences of events, which one contains the other are shown in Figure 6.

The algorithms used to identify frequent event patterns in an event sequence are based on the *sliding window* algorithms [8]. However, we use a different definition of the frequency of the event patterns. Our definition takes in to account the intensity of the fuzzy events. In [8], the events are crisp (there is no indication of the event intensity, an event is fully present if it occurs) and consequently the frequency of an event pattern (episode) is defined as the fraction of the windows in which the pattern occurs. Our definition aims at estimating the uncertainty introduced by the occurrence of the fuzzy events. In our case, the events are not crisp, but they occur with an intensity, which is a measure of the abnormality of the attribute value. So, consecutive events appearing in two different patterns may have different effects if they occur with different intensities. It is obvious that the definition of the pattern frequency as the fraction of the windows in which the pattern occurs, does not suffices in the case of patterns consisting of fuzzy events. Therefore, we propose a new definition of pattern frequency, which is based on the intensity of the fuzzy events. Given a window of width l, the frequency of a pattern p in a sequence of fuzzy events s is defined as follows:

$$fr_p(s,l) = \frac{\sum_{w_i} I_p(w_i)}{|W(s,l)|}$$

Where w_i is the *i-th* window in s, $I_p(w_i)$ is the intensity of the pattern in the *i-th* window and $|W(s,l)|$ is the total number of windows on the sequence s. The width l of the window is an integer specifying the number of time granules The intensity of a pattern p is given by

$$I_p(w_i) = \underset{(e_j \in p) \wedge (e_j \in w_i)}{Min} \{I(e_j)\}$$

Where $I(e_j)$ is the intensity of the event e_j in the pattern p in the window w_i. This means that the intensity of a pattern, which occurs in a window w_i is equal to the minimum intensity of the events, which occur in the pattern of this window. In the case where the fuzzy events are crisp (i.e. $\vartheta=0$), then the above definition converges to the one given in [8], since the numerator converges to the number of the windows where the pattern occurs. In this sense, the above definition is viewed as a more general one that extends the notion of pattern occurrence in order to deal with uncertainty, which is expressed through the pattern intensity.

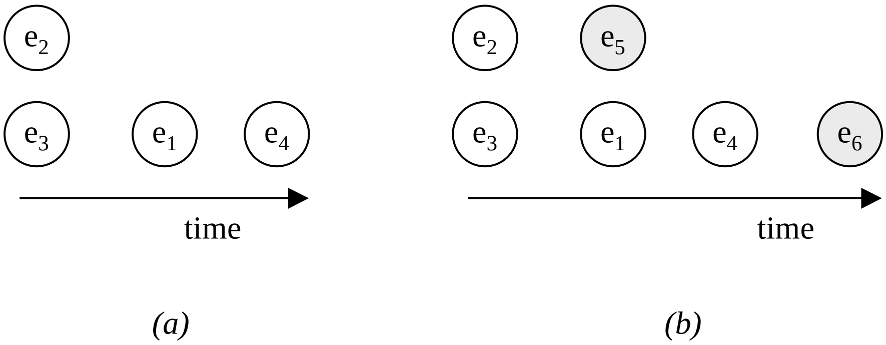

Figure 6. Two event sequences. Sequence (*b*) contains sequence (*a*)

An event pattern *p* is considered frequent if its frequency fr_p is greater than or equal to a threshold σ, which is known as the *minimal support*. The identification of frequent event patterns is based on the principle that if an event pattern *p* is frequent, then all of its sub-patterns are frequent as well with a frequency, which is greater than or equal to that of the pattern *p*. For example, if the frequency of the pattern in Figure 6-*b* is $f_b \geq \sigma$, then the pattern in Figure 6-*a* is also frequent and its frequency $f_a \geq f_b \geq \sigma$. This principle is used to prune out non-frequent event patterns. The following algorithm is used to extract the set of frequent event patterns from a sequence of fuzzy event.

Algorithm 1. Extraction of frequent patterns

Input: An event sequence S
Output: The set F containing all frequent fuzzy event patterns in S
Method:

```
    Let T be a temporary set of fuzzy event patterns.
    Let $C_i$ be the candidate set of fuzzy event patterns containing $i$ fuzzy events. First i=1 and
    $C_1$ is the set containing singleton fuzzy event patterns. Each of such a singleton fuzzy event
    patterns contains an event type occurring in the sequence S.
    While $C_i \neq \emptyset$ do
       $T=\emptyset$
       for all $x \in C_i$ do
            compute $fr_x$ in S and if $fr_x \geq \sigma$ then
               find all the pattern that extend x by a single event and add these patterns in T.
       $C_i=T$
```

The set of frequent patterns is used to obtain rules. The following section discusses how a rule base may be constructed by a set of frequent patterns.

5.2. Rule Extraction

Once the frequent patterns are known, they are used to obtain rules. A rule associates two close related event patterns where one contains the other. Let *F* be the set of frequent event patterns. The frequency of a pattern $p \in F$ is greater than or equal to the minimal support σ. Let *A* and *B* be two frequent event patterns, $A \in F$ and $B \in F$, such that *A* is contained in *B* (i.e. $A \subseteq B$). Then the confidence of the rule $A \rightarrow B$ is given by the fraction $c_{A\rightarrow B} = \frac{fr_B}{fr_A}$. Literally, $c_{A\rightarrow B}$ is the strength of the rule and it is an estimate of the conditional probability of occurrence of *B* in a window, given that *A* occurs in this window. The rule is considered valid and it is added to the rule base if its confidence is greater than or equal to a confidence threshold ε, which is known as the *minimal confidence*. The following algorithm is used to extract rules from the set of frequent event patterns *F*.

Algorithm 2. Rule extraction

Input: set of frequent patterns *F*
Output: set of rules *R* (rule base)
Method:

```
    $R=\emptyset$
    for all $p \in F$ and $q \in F$ such that $p \subseteq q$ do
       $c_{p\rightarrow q} = fr_q/fr_p$
       if $c_{p\rightarrow q} \geq \varepsilon$ then
          add the rule $p \rightarrow q$ to the set R with confidence $c_{p\rightarrow q}$
    od
```

The set of rules *R* is stored in the rule repository in the event mining unit (see Figure 2). The prediction unit guesses future events by using the rules of this repository. The prediction of future events is discussed in the following section.

6. Predicting aberrant events

Given a set of rules and a recent event sequence of *k* windows width, the goal is to guess future possible event patterns and to form a fuzzy set whose membership function assigns a likelihood value to each of the possible future event patterns. Let's denote *antecedent*(r_j) the premise of the rule r_i, *consequence*(r_j) the conclusion of r_j and $c(r_j)$ the confidence of the rule r_j. The algorithm for predicting event patterns is the following:

Algorithm 3. Prediction of event patterns

Input: the most recent *k* windows in the sequence
Output: the fuzzy set G of event patterns that are likely to occur in the future
Method:

```
for all the most recent windows $w_i$ do
    for all rules $r_j$ in the rule base do
        if the antecedent($r_j$) is present in the window $w_i$ then
            add the pattern consequence($r_j$) to G with confidence c($r_j$).
```

The members of the fuzzy set G are the event patterns guessed whose membership values represent the likelihood that the event pattern will occur.

7. Conclusions

Given a time series, whose data is originated from an industrial monitoring process, MAPS method is able to identify marginal cases in the sequential data and automatically extract the fuzzy sets that describes the regularity of data. These fuzzy sets are then used to identify aberrant behaviour and generate relevant events. Such events are registered and they usually form event sequences. Event mining is then used over these sequences to discover frequent fuzzy event patterns and deduce rules that associate close related frequent fuzzy event patterns. The antecedent part of such rules is used as a knowledge pattern that may match current event sequences. In case of a matching, we are able to predict future sequences of aberrant events that may affect the industrial process. The MAPS method employs fuzzy sets and seamlessly integrates fuzzy logic and data mining providing a more flexible way for identifying and predicting abnormal events. The contributions of this paper are summarised as follows:

- It introduces fuzzy logic for identifying abnormal events in observation data
- It provides a new definition for frequent event patterns, which is based on the intensity of the event patterns. The intensity of events defined as a membership value of the fuzzy set that describes the abnormal behaviour.
- It points out the way to define fuzzy membership functions in terms of some statistical measures (minimum, maximum and mean) of the sequential data.
- It modifies existing event pattern mining algorithms in order to accommodate the fuzzy metrics.
- It shows how prediction may be applied by utilising the rule base obtained from mining and the most recent event patterns in the sequence.
- It seamlessly integrates event identification, event pattern mining and event pattern prediction into a single system.

In this paper, we have utilised some statistical measures to construct fuzzy sets that describe the regularity of a sequence. However, other statistical measures such as standard deviation and variance may be also used in shaping the membership functions. Another important issue that may be considered in extracting such fuzzy sets is the utilisation of higher order metrics of the sequence (first derivative etc.). In rapidly changing time sequences, we may need to take into account such metrics in order to capture more accurately the regular behaviour of the sequence.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant: Mining Sequential Patterns. In *Proceedings of the Eleventh IEEE International Conference on Data Engineering (ICDE'95),* pp. 3-14, March 6-10, 1995, Taipei, Taiwan.

[2] Claudio Bettini, Xiaoyang Sean Wang, Sushil Jajodia, Jia-Ling Lin: Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* **10**(2): 222-237 (1998)

[3] Thomas G. Dietterich and Ryszard S. Michalski. Discovering Patterns in Sequences of Events. *Artificial Intelligence* **25**(2): 187-232 (1985).

[4] Tom Fawcett, Foster J. Provost: Activity Monitoring: Noticing Interesting Changes in Behavior. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99),* pp. 53-62, August 15-18, 1999, San Diego, CA, USA.

[5] Valery Guralnik, Jaideep Srivastava: Event Detection from Time Series Data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99),* pp. 33-42, August 15-18, 1999, San Diego, CA, USA.

[6] Philip Laird. Identifying and Using Patterns in Sequential Data. In *Proceedings of the Fourth International Workshop on Algorithmic Learning Theory (ALT'93),* LNCS 744, pp. 1-18, Tokyo, Japan, November 8-10, 1993.

[7] Jianxiong Luo, Susan M. Bridges. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems (IJIS)*, **15**(8): 687-703 (2000).

[8] Heikki Mannila, Hannu Toivonen, A. Inkeri Verkamo: Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery* **1**(3): 259-289 (1997)