UDC 004.65.021:519.246.8
Paper by invitation
Preliminary communication

ALGORITHMIC METHODS FOR SEGMENTATION OF TIME SERIES: AN OVERVIEW

Miodrag Lovrić, PhD[1]

Marina Milanović, MSc[2]

Milan Stamenković[3]

Abstract

Adaptive and innovative application of classical data mining principles and techniques in time series analysis has resulted in development of a concept known as time series data mining. Since the time series are present in all areas of business and scientific research, attractiveness of mining of time series datasets should not be seen only in the context of the research challenges in the scientific community, but also in terms of usefulness of the research results, as a support to the process of business decision-making. A fundamental component in the mining process of time series data is time series segmentation. As a data mining research problem, segmentation is focused on the discovery of rules in movements of observed phenomena in a form of interpretable, novel, and useful temporal patterns. In this Paper, a comprehensive review of the conceptual determinations, including the elements of comparative analysis, of the most commonly used algorithms for segmentation of time series, is being considered.

Keywords: *time series, data mining, segmentation, piecewise linear approximation, algorithm, approximation error*

JEL Classification: C22, C81, C82

Introduction

The development of information technology, which necessarily implies the presence of computers and networking in virtually all areas of business and science, as well as the need for extraction of information, and knowledge discovery in large collections of (structured and unstructured) data, led to the development of the data mining concept. Data mining, as an interdisciplinary research area, acts as an interface between the broad disciplines, such as: statistics, computer science and artificial intelligence, machine learning, database management, etc. Therefore, as a result of a different research focus on different aspects of data mining, data mining can be defined in different ways. A common definition of data mining, often cited in literature, is (Hand

[1] Full time Professor, Faculty of Economics, University of Kragujevac and Visiting Professor at Federal University of Pernambuco, Brazil, mlovric@kg.ac.rs; miodrag.lovric@de.ufpe.br

[2] Teaching Assistant, Faculty of Economics, University of Kragujevac, milanovicm@kg.ac.rs

[3] Teaching Assistant Faculty of Economics, University of Kragujevac, m.stamenkovic@kg.ac.rs

1999, p.433): *“Data mining is the discovery of interesting, unexpected, or valuable structures in large data sets.”* Thus understood, the term data mining is the basis for the perception of data mining, as a process of identifying the significant patterns or models in data, for making, inter alia, crucial business decisions.

Many data mining problems include time aspects, and the most frequent form of presenting temporal data is time series. Consequently, adaptive and innovative application of classical data mining principles and techniques in time series analysis resulted in development of a time series data mining concept (TSDM), (Antunes and Oliviera 2001; Keogh 2010; Last et al. 2004; Liu 2009; Mörchen 2006; Povinelli 1999; Ratanamahatana et al. 2005). Unlike the traditional techniques for analysis of time series and their limiting assumptions, methods that are based on TSDM network can successfully be used for identification of features of large time series datasets, that are, essentially, high-dimensional.

In the relevant literature (e.g. Chundi and Rosenkrantz 2009; Fu 2011; Keogh 2010; Lin et al. 2005; Lin et al, 2007; Mörchen 2006), related to the analysis of time series, based on the application of data mining methodology, the following, essentially and methodologically correlated, typical TSDM tasks are emphasised: *pre-processing*, *similarity search*, *representation*, *clustering*, *classification*, *segmentation*, *anomaly detection*, *motif discovery*, *prediction* and *visualisation*. In this classification of TSDM tasks, the role of segmentation of time series (as a pre-processing step in time series analysis applications) in dimensionality reduction, and patterns and rules discovery in behaviour of observed phenomenon, is emphasised (for details, see Fu 2011). In general, time series can contain hundreds or thousands of observations. The main goal of segmentation is the extraction of time segments with similar observations, or different from the rest of the time period. In that manner, the decomposition of time series into a small number of homogeneous pieces is performed.

Time series segmentation is discussed in the literature in different contexts, by many authors, and therefore the segmentation problem can be referred to: as a pre-processing step and core task for variety of data mining tasks, as a trend analysis technique, as a discretisation problem in function of dimensionality reduction, as a component in data mining applications in various fields, etc. For detailed discussions regarding the segmentation problem in these contexts (which are beyond the scope and objectives of this Paper) interested researches are referred to the references (Chung et al. 2004; Fu et al. 2001; Fu 2011; Gionis and Mannila 2003; Haiminen and Gionis 2004; Han et al. 2007; Himberg et al. 2001; Park et al 2001; and Shatkay and Zdonik 1996), which contain excellent and extensive theoretical and/or empirical surveys.

For the purpose of this study and analysis of the problem of time series segmentation, the excellent reviews of time series segmentation, presented by Keogh et al.(2004), Bingham et al. (2006), Chundi and Rosenkrantz (2009), and Gionis and Mannila (2005), have been considered and analysed. In relevant literature, many algorithms have been proposed for representation of time series into their segmented forms, and determination of adequate number of homogeneous (or alternatively heterogeneous) segments (Hiisilä 2007). In addition, in the literature, many methods for creating time series representations can be found (Mörchen 2006; Lin et al. 2007). The approximation of original time series in the form of straight lines is the essence of Piecewise Linear Representation (PLR). The three most common segmentation algorithms, based on the PLR, are as follows: Top-Down, Bottom-Up, and Sliding Window algorithm (Keogh et al. 2004). Based on the empirical research and testing of the performances of these algorithms, as a combination of the Bottom-Up and Sliding Window algorithm, the SWAB algorithm has been proposed by Keogh et al. (2004). Issues regarding the efficiency and properties of the good segmentation algorithm are analysed in Lemire (2007) and Terzi and Tsaparas (2006).

The exploration of time series in a data mining context is a relatively new and ever changing research field (Keogh 2010). Scientific and technical papers published in top-tier conference proceedings (KDD - bringing together data mining, data science and analytics community 2014), such as: *ACM Knowledge Discovery in Data and Data Mining*, *IEEE International Conference on Data Mining*, and *IEEE International Conference on Data Engineering*, are recommended to the researchers interested in TSDM issues. In addition, the attractiveness of knowledge discovery in high-dimensional time series, should not be seen only in the context of the research challenges in the scientific community, but also in terms of usefulness of the obtained results in the function of supporting the process of business decision–making, because, as a nugget of a gold is hidden beneath the earth or water, the nugget of (business) information is hidden in the data, (Milanovic and Stamenkovic 2011, p.10).

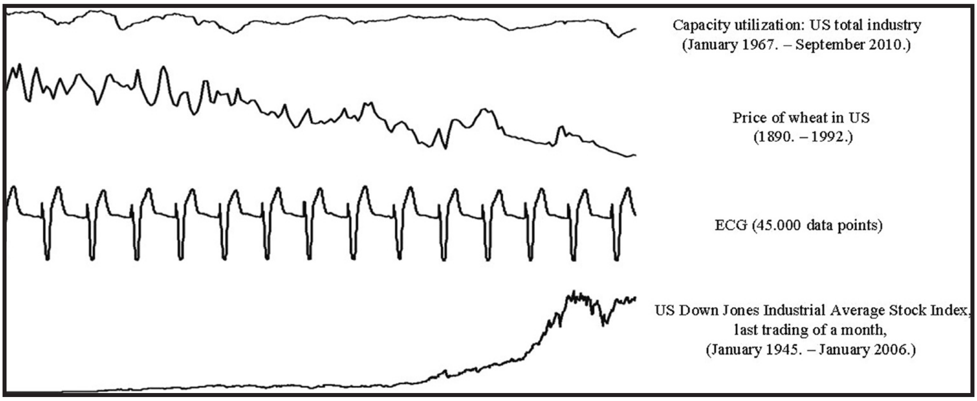
Previously presented findings define two main goals of this Paper. Since TSDM is an extensive research area, the primary goal emerged from focusing on the segmentation of time series, as one of the key components in the structure of TSDM tasks. In other words, the primary goal is to emphasise the importance of segmentation in the process of extraction of relevant information from large time series datasets. At the same time, additional effects in the form of a secondary goal of the Paper, are focused on promotion and popularisation of application of methods included in TSDM network. Generally, TSDM is an extensive, very heregoneneous, research area, so that the complete (full) coverage of all relevant aspects cannot be implemented and presented in the form of one, particular, research paper. Therefore, in this Paper, a detailed review of one particular, significant, segment of this area, related to the key characteristics, as well as the ways of functioning (in the form of flowcharts) of the major algorithmic methods for segmentation of time series, is provided. The results of the study, presented in this Paper, should provide, to the interested researchers, a detailed insight into the overall picture regarding the segmentation approaches and their role within the TSDM framework.

Accordingly, the results of the research conducted by examining the available relevant literature, with an introduction (*Section 1*) and conclusion (*Section 8*), are presented in the following parts of this Paper. The considerations in *Section 2* cover the conceptual framework of segmentation. In *Section 3*, the phases of the segmentation process are presented; and in *Section 4*, properties of major segmentation algorithmic methods for univariate time series in the context of piecewise linear representation. Due to their different properties, the quality evaluation of the segmentation algorithms is discussed in *Section 5*. The results of empirical comparison of the performances of segmentation algorithms are given in *Section 6*. The importance of segmentation, especially in economic research, is highlighted in *Section 7*.

The conceptual framework of segmentation

Expanding interest in data mining researchers for mining of temporal data, and connection of data mining techniques with the analysis of time series datasets, have resulted in the application of a whole range of specific algorithmic TSDM methods for identification, analysis, and prediction of characteristics of non-periodic, nonlinear, irregular, and chaotic time series. In Figure 1, the examples of high-dimensional stationary and non-stationary time series from different areas are illustrated. In general, crucial aspects of mining time series data focus on identifying the movements and/or components, which exist within the data.

Figure 1. Examples of time series



Source: authors` representation
Sources of data (respectively):
http://research.stlouisfed.org/fred2/series/TCU/downloaddata?cid=3;
http://robjhyndman.com/tsdldata/data/9-9.dat;
http://www.cs.ucr.edu/~eamonn/discords/qtdbsel102.txt;
http://www.forecasts.org/data/data/djiaM.htm.

Time series segmentation is a fundamental component in the process of analysis and research of time series data. Its relevance should especially be viewed in the context of implications on the creation of a valid model of time series. As a data mining research problem, segmentation focuses on dividing the time series into appropriate, internally homogenous segments, so that the structure of time series, through pattern and/or rule discovery in the behaviour of the observed variable, could be revealed. Therefore, understanding the conceptual framework of segmentation requires the prior, clear definition and delineation of the basic terms that are important for segmentation, and also related to the classical concept of time series.

Generally, time series, T, is defined as a set of real values of the observed variable that are arranged by chronological order in successive time periods (year, quarter, month, week, day, hour, etc.), (Kovačić 1995). In the context of segmentation, it is useful to define the time series as a sequence of time dependent values of the observed variable. Symbolically, the time series, as a set of n pairs of data (data points) is represented as follows: $T = \{(y_1, t_1), (y_2, t_2), \dots , (y_n, t_n)\}$, where, (for $i = 1,2,...,n$), y_i- value of observed variable, t_i- time, (y_i, t_i)- pair of data or data point. Since segmentation is division of the time series into pieces, segments can be defined as consecutive parts of time series, which are, in a way, the time series for themselves. In other words, segmentation, S, provides representation of time series, T, of length n, in a form of a set of k non-overlapping, consecutive segments, S_j, (for $j=1, 2,..., k$). Symbolically, $S = (S_1, S_2, ..., S_k)$. Each segment, S_j, is composed of a certain number of pairs of data, (y_i, t_i), i.e. data points s_{ij}, where i denotes the order of point in time series, and j denotes the affiliation to the certain segment. In segmented series, points belonging to each segment can be represented by: (a) one specific value that represents them (e.g. mean, or median of a segment), or (b) model that is suited to data, i.e. data points within the segment.

The process of conversion of the actual time series into its segmented version leads to reduction of the dimensionality of original values, which is important for pattern discovery in the structure of time series. However, this process should also be viewed in terms of the achieved level of accuracy of approximate representation of the original time series. Accuracy of approximate representation of time series is measured by using the appropriate error function. The most commonly used measure of accuracy is Euclidean distance, E_p, (or, some of its derived forms). Euclidean distance is based on the difference between the actual values in the time series and the values given by the approximation (modelled values). Minimizing the sum of squares of these distances the Euclidean distance also minimizes, which is, precisely, defined as the square root of the sum of squared distances between empirical and approximated values. Therefore, the essence of segmentation problem is reflected in finding the optimal approximation for which the error function, E_p, is minimal. In fact, the optimal segmentation of time series, T, for the defined parameters is defined as the segmentation that results in the lowest segmentation error in relation to other possible combinations of segmentation, $S_{n,k}$, or symbolically, (Terzi and Tsaparas 2006):

$$S_{opt}(T,k) = arg \min_{S \in S_{n,k}} E_p(T,S) \quad (1)$$

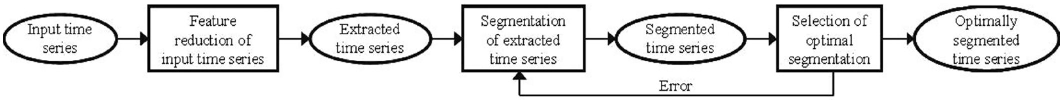
Besides determination of the error function as a total error or segmentation error, which represents the measure of the overall difference between approximated and actual values, determination of segment error is also a component of the analysis of accuracy of approximate representation. Unlike the segmentation error that refers to the entire time series, the segment error is a measure of the differences between approximated and actual values on the segment level.

Segmentation process

Given the specificity of requirements in the analysis of large time series data sets, and variety of requirements that can be defined in a form of segmentation problem, finding the optimal solution, *Sopt(T;k)*, is a complex iterative process composed of a series of sub-processes, phases, and activities that transform inputsinto output elements. Figure 2, shows the generalisation of the process approach to the segmentation problem and grouping of elements of the process in the following phases:

- feature reduction of input time series;
- segmentation of extracted time series;
- selection of the optimal segmentation.

Figure 2. Time series segmentation process


Input time series
Feature reduction of input time series
Extracted time series
Segmentation of extracted time series
Segmented time series
Selection of optimal segmentation
Optimally segmented time series
Error

Source: authors` representation

Feature reduction phase of the segmentation process conducts the conversion of input time series data in an extracted form of time series, which is, in relation to the input form, not only more suitable for segmentation process, but also better adopted to the general requirements of the segmentation problem and relevant characteristics of specific TSDM application. With quality improvement of outputs of this phase, through the preliminary dimensionality reduction and elimination of white noise, the preconditions for achieving better performance, with reduction of processing effort in the phases that follow, are created. In the ***segmentation phase***, in accordance with the specificities of the objectives in the time series analysis, the nature of research topic, and the level of expert knowledge of the data miners, the main segmentation parameters are defined, along with the selection, and then, the implementation of the appropriate algorithmic methods that result in a segmented, approximate form of time series. In the last phase of the segmentation process, by comparison of user-defined parameters and results of conducted segmentation, the ***selection of the optimal segmentation*** is performed.

The selection of the best solution is based on the identification of the size of the error, i.e. difference between the original and the approximated version of time series. The activities in the segmentation phase are repeated as long as the error is (much) above the threshold defined by the data mining researcher. The output of this phase is an optimally segmented time series.

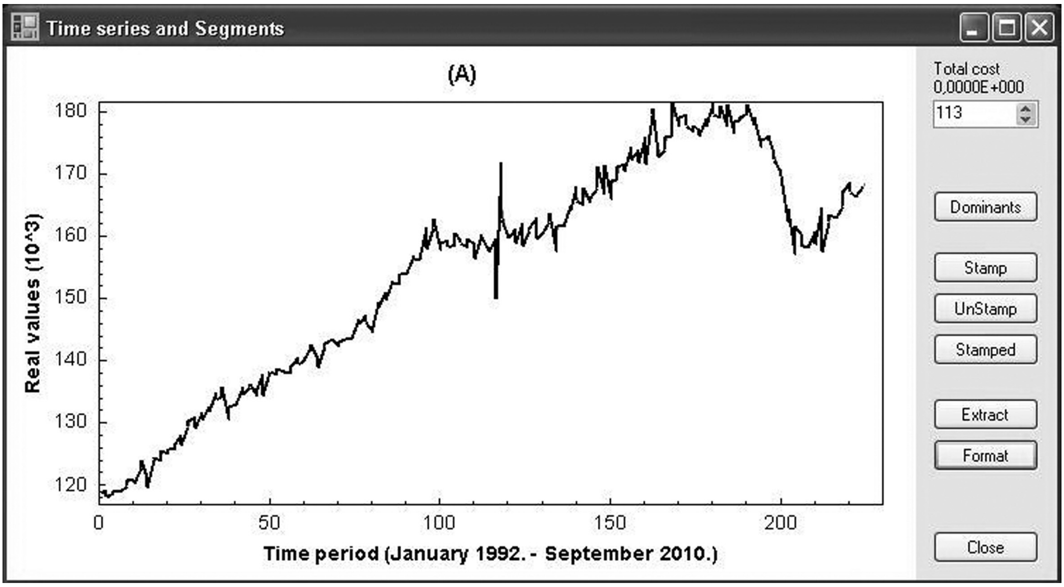
From the above noted, the two primary tasks in the segmentation process can be clearly defined as follows:

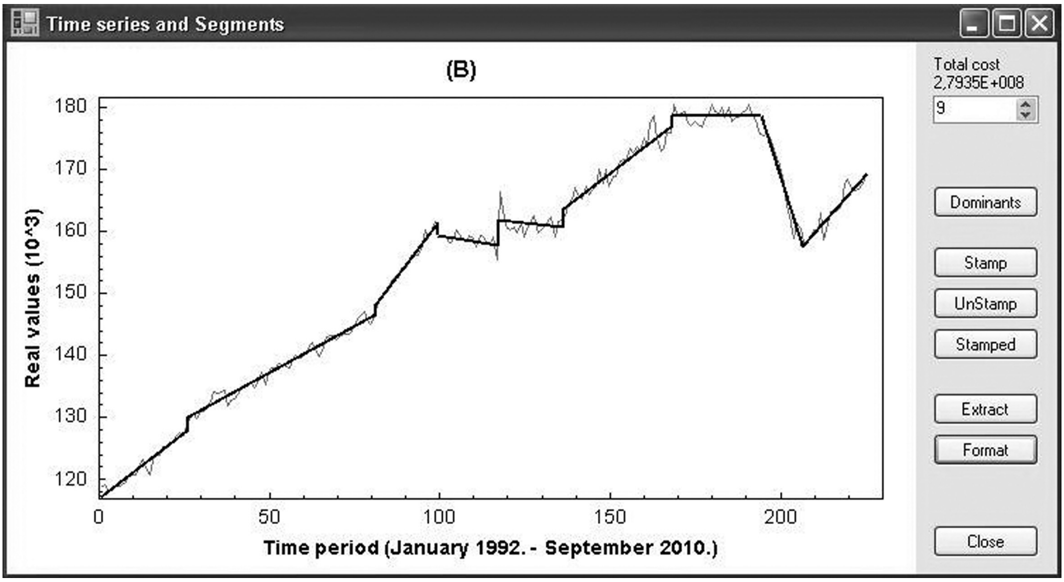
▸ first, produce the best representation of each segment, so that the maximal error for any segment (segment error) is not above some user-specified threshold, called the max error.

▸ second, produce the best representation of the entire time series, so that the combined error of all segments (segmentation error) is not above some user-specified threshold, which is called the total max error.

Generally, the basic idea of representing the original time series data sets in segmented forms is to provide concise display and clear insight into their basic characteristics through the appropriate approximation forms, with minimal loss of relevant information (visit, *http://www.cs.ucr.edu/~eamonn/*). For example, one of the most commonly used time series representations, since it can be determined more easily than the quadratic or other higher degree representations, is Piecewise Linear Approximation (PLA). This representation, as a result of segmentation procedure, approximates time series, *T*, of length *n*, over *k*, non-overlapping segments, presented in the form of *k* straight lines, which can be determined by using linear interpolation or linear regression (Keogh et al. 2004).

Figure 3. The original time series (A) and its segmented approximation (B)





Total cost
2,7935E+008
9

Source: authors` representation, Segmenter 2.1 software output
Source of data: http://research.stlouisfed.org/fred2/series/RRSFS.

Representation of time series, as a sequence of segments represented as straight lines, is illustrated in Figure 3, where curve (A) refers to the original time series – *real retail and food services sales in USA, within the time period from January 1992 to September 2010*, and curve (B) refers to its segmented version, or piecewise linear approximation.

Segmentation algorithms

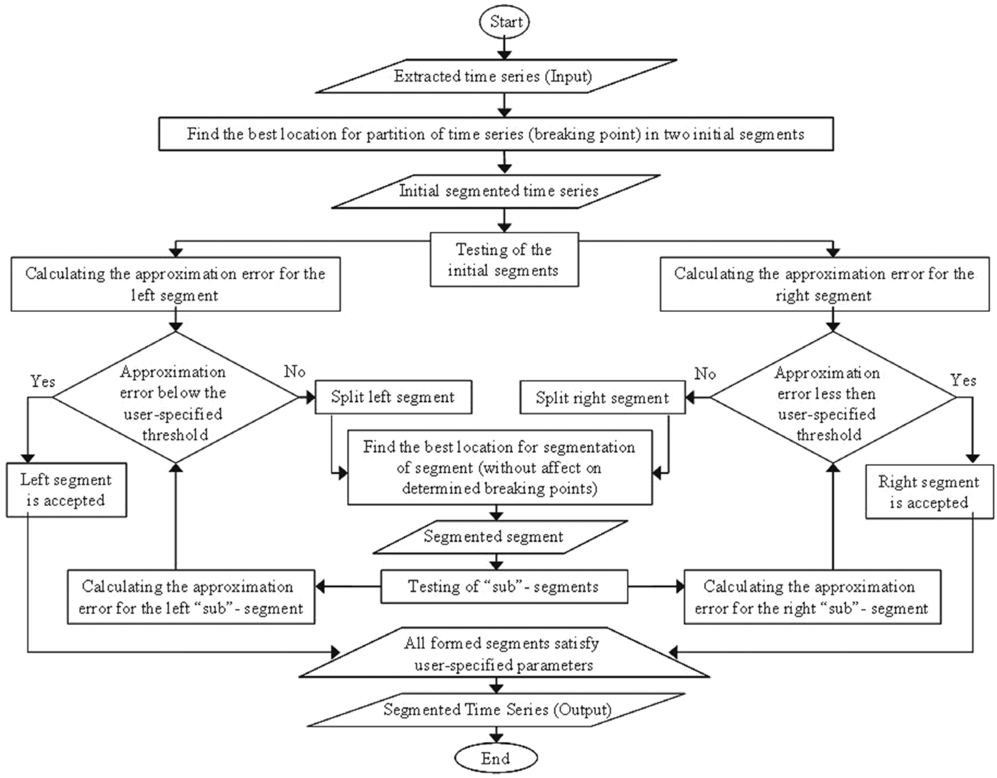
As already noted, in order to ensure efficient handling and processing of data it is necessary to have reduced and concise representation of data at disposal. Given the complexity of the segmentation problem, conducting the segmentation process manually is, not only, extremely slow, and inaccurate, but in the analysis of time series databases that kind of an “attempt” can be characterised as *Sisyphus work* (Milanovic and Stamenkovic 2010). Therefore, it is logical why the automated, precise segmentation should be the integral part of every mining process in temporal data. Automated segmentation is based on an entire range of algorithmic methods and procedures that can be described as a well-defined set of instructions and rules for conducting the transformation of the original time series into segmented approximation. In other words, the main purpose of a segmentation algorithm is to find the best segmentation to represent the given time series, but with efficient running time (execution time). However, in accordance with the variety of ways of formulating the segmentation problem in terms of defining the key parameters (number of segments, segmentation starting point, length of segments, error function, user-specified threshold, etc.), it is obvious that the universal algorithm, which in all cases results in optimal solution, does not exist. In addition, the selection and application of an adequate algorithmic method for segmentation may include not only the application of already existing algorithms, but also their modification, or creation of entirely new conceptual designs as well, in function of obtaining more representative results of the analysis.

Generally, basic classification of algorithmic methods for segmentation implies their division into two categories (Aksoy et. al, 2008): ▸*offline*, and ▸*online* algorithms. The essence of the offline segmentation is contained in the scanning and division of entire time series $T = \{(y_1, t_1), (y_2, t_2),...,(y_n, t_n)\}$, into a number of segments, while the available number of data (data points) remains unchanged during the execution time of the algorithm. On the other hand, in the online segmentation para llelwith the execution time of the algorithm, the uploading of new data points is being performed, one at the time, *$(y_1, t_1), (y_2, t_2),...,(y_{n-m}, t_{n-m}),...$*, and at every time step t_i, (for *i = 1,2,...,n,* where *n* is growing potentially forever) it must be decided whether the obtained data belong to the previous segment, or should be assigned to a new segment, which starts at t_i. Based on the essential context of the offline and online segmentation, numerous algorithms for segmentation of time series are created. In spite of their differences in terms of efficiency, application complexity, and specific implementation details, as well as the names under which they have been listed in the reference literature, segmentation algorithms, which convert original time series into PLA segmented version are usually classified into one of the following categories of algorithms: ▸ Top–Down algorithm; ▸Bottom–Up algorithm; and ▸Sliding Window algorithm. As a result of combining Bottom–Up and Sliding Window procedures, a new, precise, streaming SWAB algorithm, has been developed (Keogh et al. 2004).

The Top–Down algorithm, often called "*divide and conquer*" or "*binary split*", starts with conditional observing of non-segmented time series as one major segment. Based on the consideration of all possible variants for initial division, the best location for placing the boundary (breaking point) that splits original time series in two segments, S_1 (left) and S_2 (right), is identified, but in such a way that the difference between those two segments is maximal. Both of these segments are then tested from the aspect of the level of the approximation error. If the approximation error of the observed segment is below the user-defined threshold, the segmentation procedure stops, and the tested segment is accepted. On the other hand, if the approximation error is above the user-defined threshold, further division of the tested segment into two new (sub) segments is performed. For each of the newly formed segments, the process of division into two new segments is repeated in an identical manner, without effect on the location of the breaking point determined in the previous iteration (step). The algorithm repeats these steps until some of the defined stopping criteria [a) *k* number of segments, and/or, b) approximation error < the user-specified threshold] is satisfied, i.e. when further division no longer contributes to the minimisation of the segment or segmentation error.

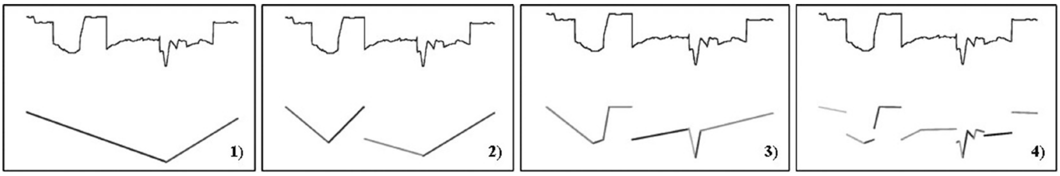
Its potential inflexibility is often cited as the main disadvantage of this algorithm, caused by the fact that the breaking points (boundaries) determined in the previous iterations remain unchanged until the end of the segmentation process. In fact, the probability of obtaining the optimal solution in the first few iterations is very small, and the boundaries marked as "*the best location*", set at the beginning of the segmentation, need not, necessarily, prove to be the optimal in the later phases of the implementation of the process. Graphical representation of the Pseudo-Code for the Top–Down algorithm is shown in Figure 4. In the literature, numerous, modified versions of this algorithm can be found. An especially interesting version for support of the concurrent Text mining and TSDM, including a novel stopping criterion, based on the Student's t-test, is proposed by Lavrenko et al. (2000). The Top–Down algorithm belongs to the category of the offline algorithms, and Figure 5 illustrates the flow of the second phase of the segmentation process with applied Top–Down algorithm.

Figure 4. The flowchart for the Top–Down algorithm


Start
Extracted time series (Input)
Find the best location for partition of time series (breaking point) in two initial segments
Initial segmented time series
Testing of the initial segments
Calculating the approximation error for the left segment
Calculating the approximation error for the right segment
Yes
Approximation error below the user-specified threshold
No
Split left segment
Split right segment
No
Approximation error less then user-specified threshold
Yes
Left segment is accepted
Find the best location for segmentation of segment (without affect on determined breaking points)
Right segment is accepted
Segmented segment
Calculating the approximation error for the left "sub"- segment
Testing of "sub"- segments
Calculating the approximation error for the right "sub"- segment
All formed segments satisfy user-specified parameters
Segmented Time Series (Output)
End

Source: authors` representation

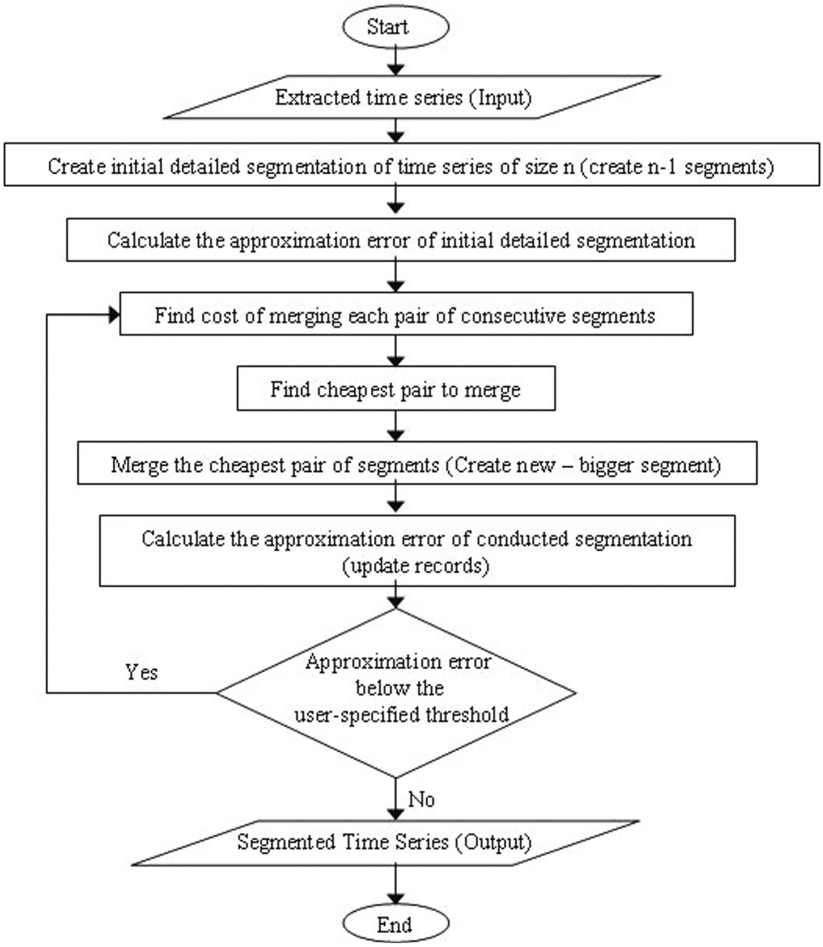
Figure 5. How does the Top–Down algorithm work?



Source: authors` representation (adapted from Keogh et al. 2001)

The Bottom–Up algorithm, often called "*iterative merge*", as a natural complement to the Top–Down algorithm (Keogh et al. 2004), begins by dividing the original time series, of length *n*, into a large number of very small segments with equal lengths. In the next step, based on the comparison of each pair of consecutive segments (including left and right neighbour), the pairs that cause the smallest increase in the error are being identified, and consequently merged in one new, bigger segment.

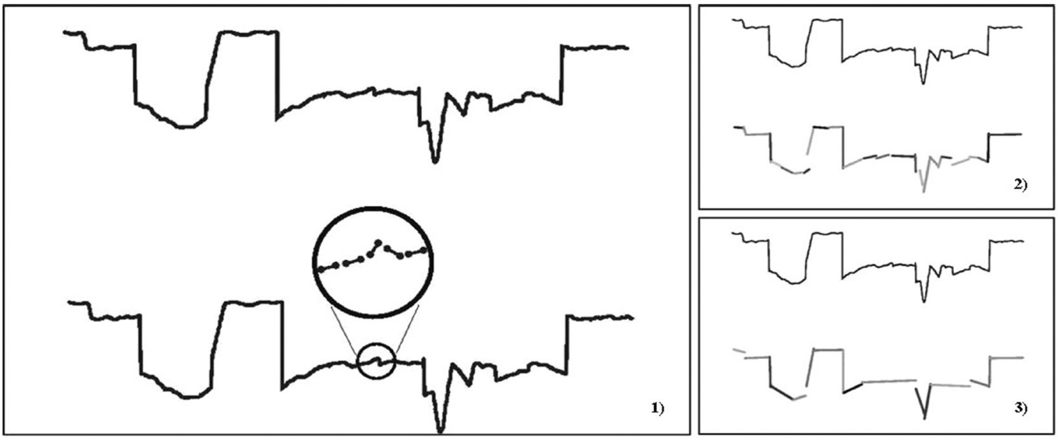
Figure 6. The flowchart for the Bottom-Up algorithm


Start
Extracted time series (Input)
Create initial detailed segmentation of time series of size n (create n-1 segments)
Calculate the approximation error of initial detailed segmentation
Find cost of merging each pair of consecutive segments
Find cheapest pair to merge
Merge the cheapest pair of segments (Create new – bigger segment)
Calculate the approximation error of conducted segmentation (update records)
Approximation error below the user-specified threshold
Yes
No
Segmented Time Series (Output)
End

Source: authors` representation

The algorithm repeats these steps until some of the defined stopping criteria [a) *k* number of segments, and/or, b) approximation error > specified threshold] is satisfied. Graphical representation of the Pseudo-Code for the Bottom–Up algorithm is shown in Figure 6. The Bottom–Up algorithm belongs to the category of the offline algorithms, and Figure 7 illustrates the flow of the second phase of the segmentation process with applied Bottom–Up algorithm.

Figure 7. How does the Bottom–Up algorithm work?

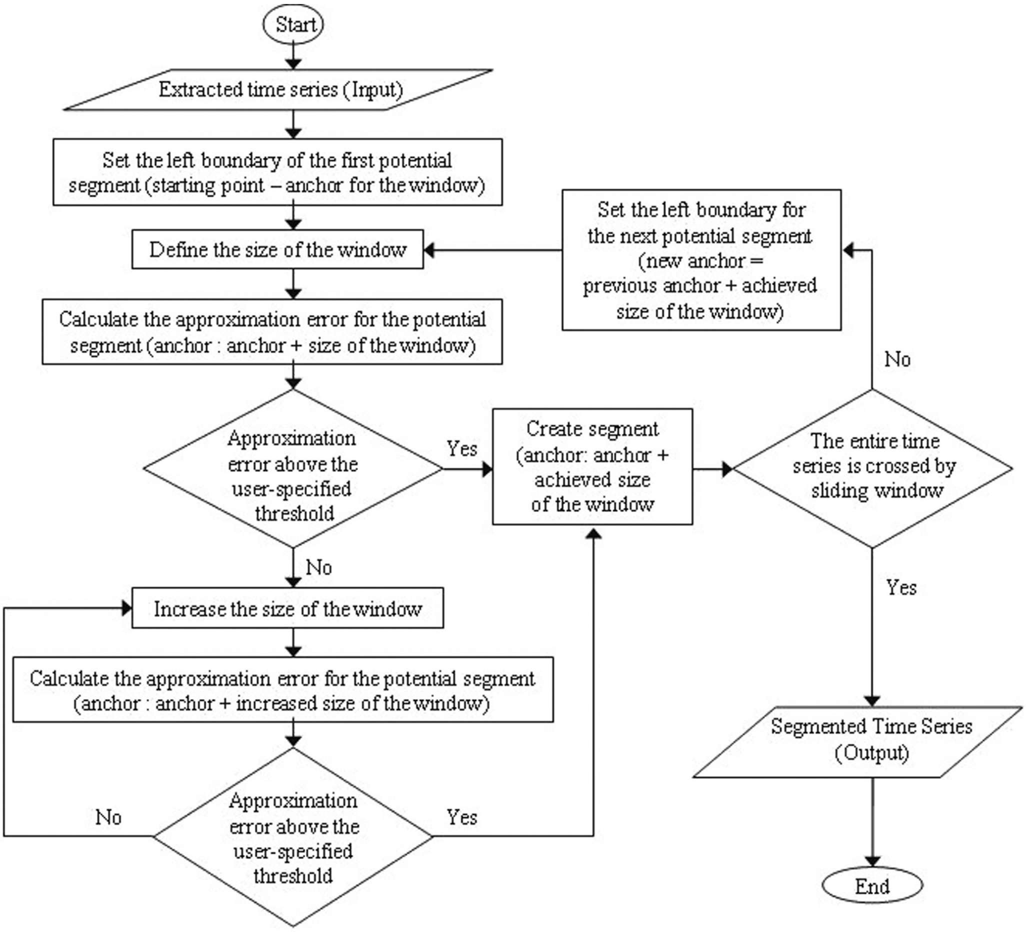


Source: authors` representation (adapted from Keogh et al. 2001)

The process of segmentation by using the ***Sliding Window algorithm***, often called "*brute force*" or "*one-pass algorithm*", begins by determining the left boundary (anchor) of the first potential segment (usually the first data point of a time series), which is also the starting point for the window which slides (to the right) along the time series, and in that manner provides identification and selection of segments that satisfy predefined segmentation criterion (user-specified threshold). While sliding down the sequence, the size of the window gradually increases, since all the visited data points on its journey, automatically become potential elements of potential segment, until the error of the potential segment does not become greater than the user-specified threshold. At this point, the right boundary of the moving window ceases to be unknown. In that manner, the length of the certain segment is being determined, and the stopping point of the newly formed segment becomes the new anchor, i.e. the starting point of the next potential segment. The algorithm, this process of forming the segments, repeats until the entire time series is converted into a PLA representation. Graphical representation of the Pseudo-Code for the Sliding Window algorithm is shown in Figure 8. Unlike the previous two algorithms, the Sliding Window algorithm belongs to the category of the online algorithms, and Figure 9, illustrates the flow of the second phase of the segmentation process with applied Sliding Window algorithm.

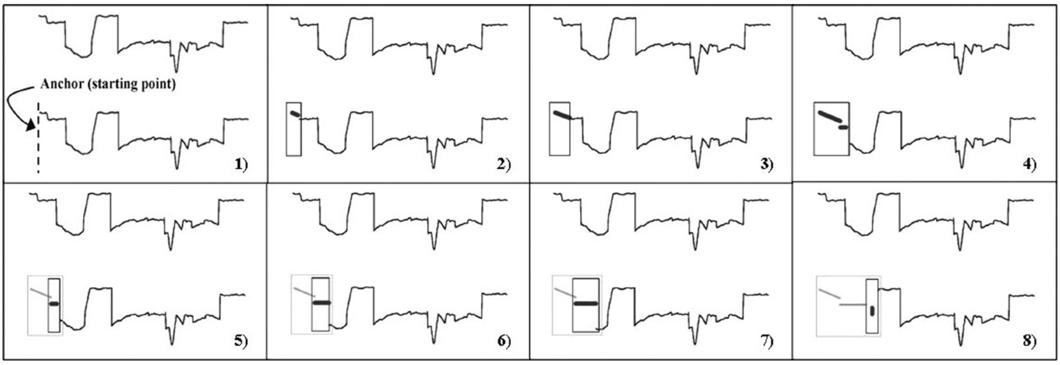
The procedural simplicity and its online nature make the Sliding Window algorithm particularly attractive from the point of view of the application, which resulted in creation of a numerous, modified versions of this algorithm, especially in the field of medical research (because, by its nature, monitoring of the patient's health condition is an online process).

Figure 8. The flowchart for the Sliding Window algorithm


Start
Extracted time series (Input)
Set the left boundary of the first potential segment (starting point – anchor for the window)
Define the size of the window
Calculate the approximation error for the potential segment (anchor : anchor + size of the window)
Approximation error above the user-specified threshold
Yes
No
Increase the size of the window
Calculate the approximation error for the potential segment (anchor : anchor + increased size of the window)
Approximation error above the user-specified threshold
No
Yes
Create segment (anchor: anchor + achieved size of the window
The entire time series is crossed by sliding window
No
Set the left boundary for the next potential segment (new anchor = previous anchor + achieved size of the window)
Yes
Segmented Time Series (Output)
End

Source: authors` representation

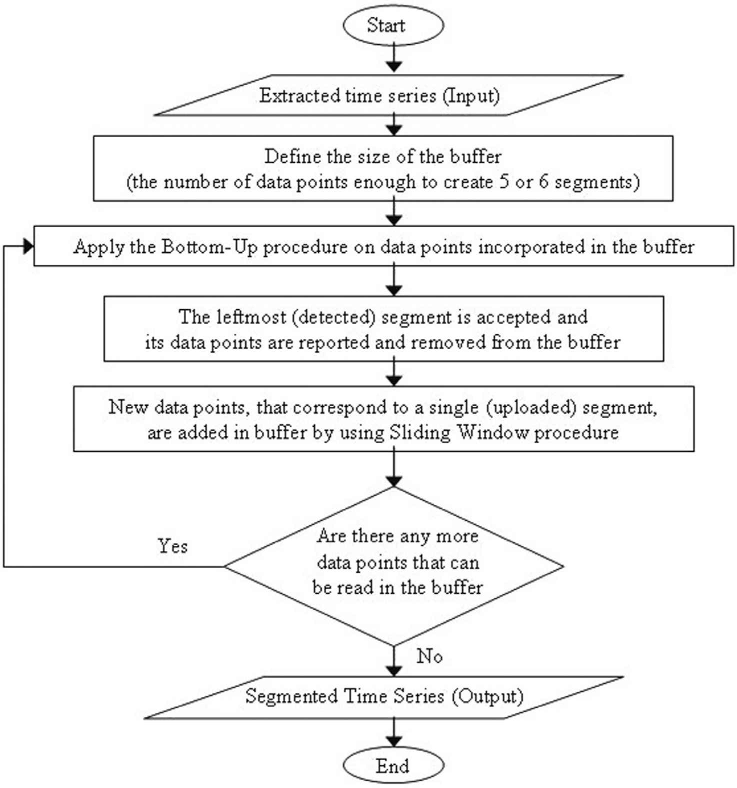
Figure 9. How does the Sliding Window algorithm work?



Source: authors` representation (adapted from Keogh et al. 2001)

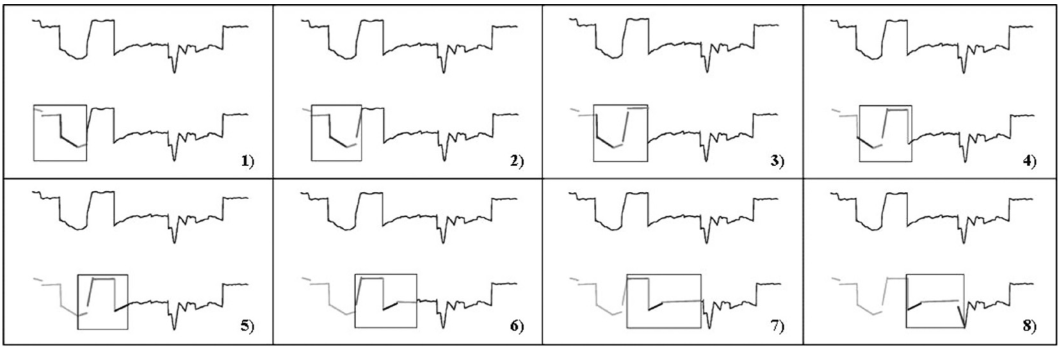
Based on the results of the conducted empirical research (see Section 6 for more details), and comparative analysis of the generic properties, advantages, and disadvantages of the above described algorithms, Keogh et al. (2004), propose new, more accurate algorithm for segmentation of data streams, called ***SWAB algorithm*** (***S***liding ***W***indow ***A***nd ***B***ottom–Up algorithm). Its name indicates that it is the approach that is based on a combination of the Sliding Window and Bottom–Up algorithm. In addition, an improved version of the approximation form of a time series, obtained by the SWAB algorithm, is a direct result of the combination of superior quality of approximation of the Bottom–Up approach, with the online nature and capability of the Sliding Window. SWAB segmentation algorithm begins by defining and selecting one, initial size of the buffer that is big enough to contain enough data to create 5 or 6 segments. In the next step, the Bottom–Up procedure is applied to the data (data points) inside the buffer. In that way, the leftmost segment is detected, and the data points that correspond to the detected segment are reported and removed from the buffer. Then, in the buffer, using classical Sliding Window procedure for defining new entry points, new data points are added. It follows re-application of the Bottom–Up procedure in the buffer.

Figure 10. The flowchart for the SWAB algorithm


Start
Extracted time series (Input)
Define the size of the buffer
(the number of data points enough to create 5 or 6 segments)
Apply the Bottom-Up procedure on data points incorporated in the buffer
The leftmost (detected) segment is accepted and
its data points are reported and removed from the buffer
New data points, that correspond to a single (uploaded) segment,
are added in buffer by using Sliding Window procedure
Are there any more
data points that can
be read in the buffer
Yes
No
Segmented Time Series (Output)
End

Source: authors` representation

Figure 11. How does the SWAB algorithm work?



Source: authors` representation (adapted from Keogh et al. 2001)

Obviously, the described process of incorporating new data points in the buffer can be repeated as long as the data arrive in continuous streams, potentially indefinitely. For the efficient functioning of this algorithm, based on the combined Sliding Window and Bottom–Up procedures, the question of defining the size of the initial buffer is of particular importance. If it is allowed for a buffer size to become very large, the SWAB segmentation outcome would be the same as the outcome of the implemented Bottom–Up procedure, while, on the other hand, the small size of the buffer would reduce the segmentation outcome to the outcome of the Sliding Window procedure. Therefore, the buffer size is fixed to a size that is needed for creating 5 or 6 segments. Graphical representation of the Pseudo-Code for the SWAB algorithm is shown in Figure 10. The use of the buffer creates conditions for implementation of the Bottom–Up algorithm, with a "semi-global" view on the dataset, and Figure 11, illustrates the flow of the second phase of the segmentation process with applied SWAB algorithm.

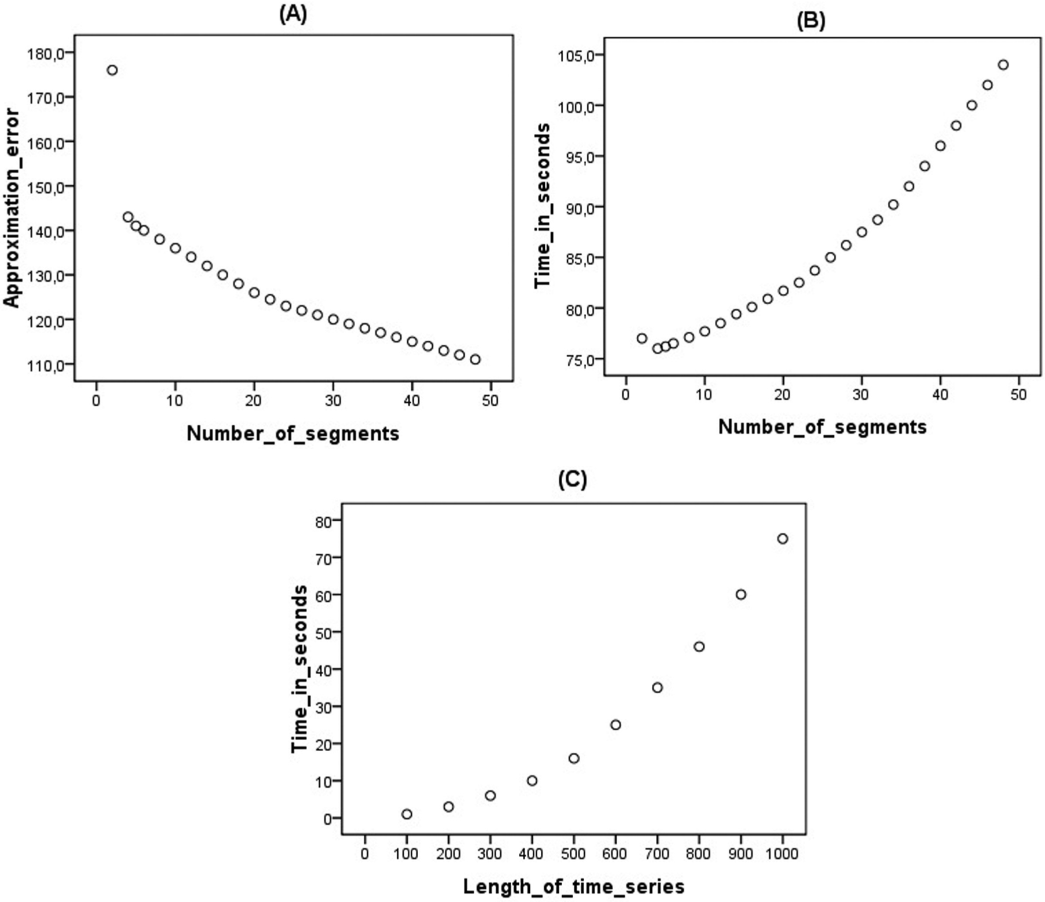
Quality evaluation of the segmentation algorithms

The selection of the segmentation algorithm which will in particular application provide and ensure discovering of interpretable, novel, and useful patterns, is not an easy task. Hence, data mining researches must take into account a number of issues, which directly or indirectly determine the specific choice of the segmentation algorithm. A brief overview of key issues and their mutual influences, and relationships of different direction and intensity, will clearly emphasise the multidimensional nature of the problem of selection of the segmentation algorithm.

The first issue refers to the knowledge of qualitative characteristics and nature of the phenomenon, which is the subject of the analysis, including dimensions, which determine the type of the observed time series: stationary/non-stationary, noisy/smooth, cyclical/non-cyclical, symmetric/asymmetric, etc. Since the algorithms differ among themselves, from the point of view of input parameters, the second issue refers to procedural algorithmic peculiarities and possibilities of specifying the desired level of quality of the approximation by defining the number and length of segments, max segment error, and total error of approximation, (e.g. Sliding Window allows specifying max segment error, but, because of the online nature, not the number of segments). It is important to point out that, if the size of the approximation error is observed in the function of the number of segments, there is an inverse correlation (see, Figure 12 (A)).

The third issue involves consideration of advantages and disadvantages of the online and offline approach, i.e. global view on the entire dataset and continuous arrival of streams of data. Last, but not least, is the issue regarding execution time of segmentation and its improvement. Algorithms that trade execution time for the quality of approximation are called anytime algorithms (Ueno et al. 2006). In general, anytime algorithm in every moment provides a *best-so-far* solution, and the quality of approximation, in terms of reduction of approximation error, improves with the increase of the execution time (positive correlation). Data miner can analyse this solution, and accordingly, decide whether to stop the work of algorithm or allow its work until the end. Logically, if the length of the time series increases, the execution time of the segmentation will also increase (see, Figure 12 (C)). Furthermore, if the execution time of the segmentation is observed in the function of the number of segments, there is a positive correlation (see, Figure 12 (B)).

Figure 12. Inverse correlation between the approximation error and the number of segments (A); Positive correlation between the execution time of the segmentation and the number of segments (B); and Positive correlation between the execution time of the segmentation and the length of the sequence (C).



Source: authors` representation (adapted from Gionis and Mannila, 2005)

Starting from the already emphasised variety of the segmentation algorithms, consequently, in order to achieve holistic insight into the complexity of the problem of selection of the appropriate segmentation algorithm, there is a need to specify and define the characteristics of a good algorithm. Quality evaluation of the segmentation algorithm is based on the consideration of the following three criteria (Lemire, 2007): ▸accuracy (quality of the approximation model), ▸efficiency (running time of the algorithm – the amount of time or effort required to conduct the segmentation), and ▸repeatability. Therefore, a good time series segmentation algorithm must be accurate and fast with the possibility of repeating satisfied segmentation results performed on different types of time series.

The quality of segmentation algorithm performance can be measured in different ways. One of the most commonly used measure is the error ratio, a ratio based on the segmentation errors of the algorithms that are compared (Euclidean distances between original time series and its segmented approximations). However, quality evaluation cannot be conducted on the basis of results obtained by the experiment performed on only one time series dataset, although, in order to formulate the valid conclusions, it is necessary to explore and analyse a variety of cases, on which the compared algorithms are applied (cross-validation analysis).

Empirical comparison of the performances of the segmentation algorithms

“Good” or “poor” segmentation algorithm? “Good” or “poor” approximation? Giving the answers to these (dichotomous) questions greatly exceeds the simplicity of their formulation. In other words, respecting the fact, which refers to the number of segmentation algorithms and their modified forms, as well as the variety of specificities of the particular research phenomenon, comprehensive analysis that would result in an answer to these questions, is practically impossible. However, partial simplification of the mentioned problem can be achieved with the experimental research and comparative analysis of the performances of the generic algorithms. Description, methodological aspects, as well as the results of one such empirical study, conducted by Keogh et al. (2004), are presented in this Section of the Paper.

Empirical research was conducted on 10 datasets, i.e. collections of data, that are different, not only from the aspect of the area to which they refer and phenomena the behaviour of which they represent (finance, medicine, manufacturing, and science), but also from the aspect of their dominant characteristics, i.e. dimensions that characterise the type of time series. It is also taken into account that the performance of the algorithm depends on the value of the user-defined threshold, which refers to the segment error. In addition, depending on the value of the user-defined threshold, a distinction is made between three types of segmentation output: ▸*too fine an approximation* (when user-defined threshold tends to zero), ▸*correct approximation* (for some “reasonable” value of user-defined threshold), and ▸*too coarse an approximation* (when user-defined threshold becomes very large the approximation is reduced, ultimately, to a single best-fit line). Accordingly, in the research, 6 different, fixed values of user-defined threshold for segment error are defined, as the only input parameter that can be defined for all three segmentation algorithms, in order to avoid the subjectivity in defining the initial parameters, and ensure accuracy of formulated conclusions. Hence, the comparison of the resulting piecewise linear approximations (piecewise linear representation based on the linear regression was used in the study) is performed for each time series dataset (10), for each of six different levels of user-defined thresholds (60 approximations per algorithm, or 180 in total), by calculating the relative indicators of the quality of approximation.

The procedure for calculating the relative indicators of the quality of approximation (for a particular dataset and for a particular threshold level), includes the following steps:

- calculation of the total error of entire approximation (segmentation error) separately for Top–Down, Bottom–Up, and Sliding Window algorithm;
- identification of the max total error (i.e. identification of the algorithm with the lowest quality of approximation);
- calculation of the relative indicator, as a ratio of the total error of each algorithm and the max total error.

The results of the empirical research, generally, can be summarised as follows (see, Figure 13 (A)):

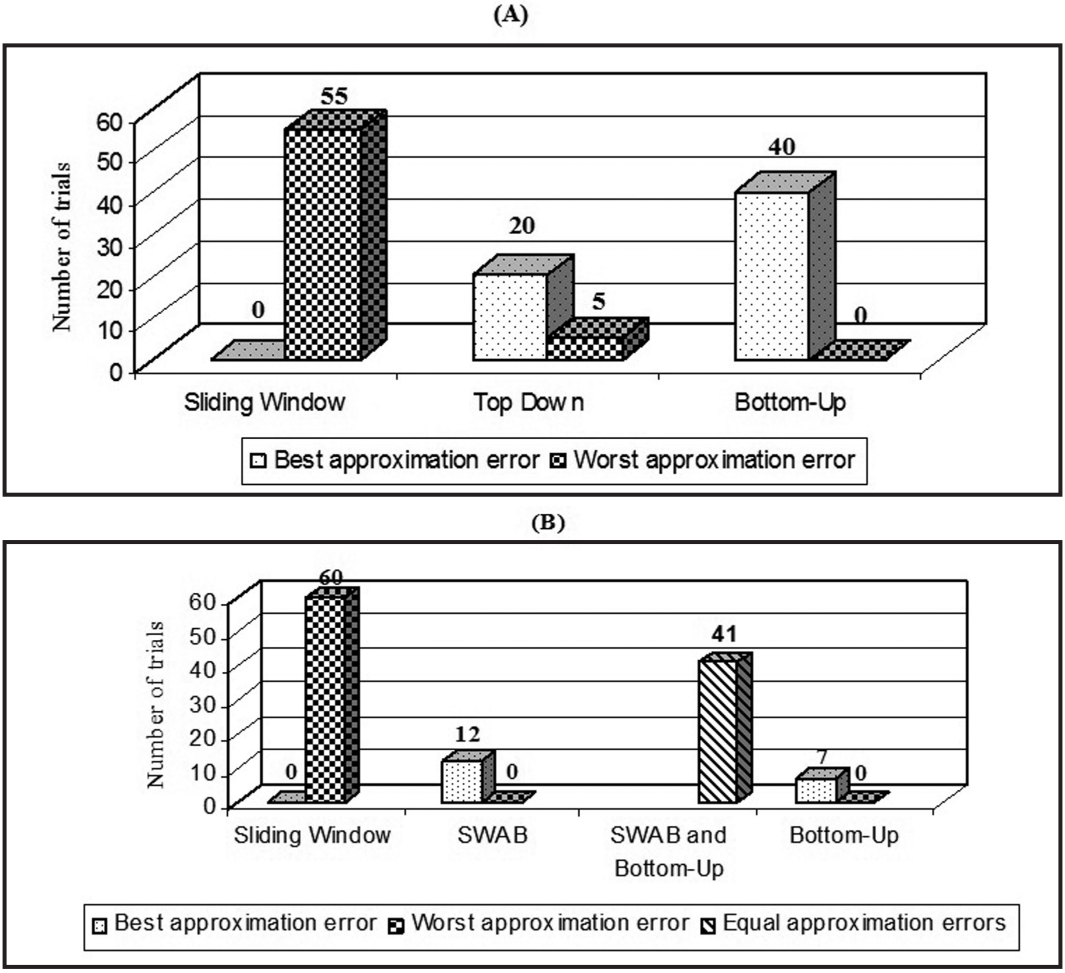
- in most trials, Sliding Window algorithm proved to be the algorithm with the worst performances;
- in several trials, Top–Down algorithm showed slightly better results than the Bottom–Up algorithm;

- in most trials, Bottom–Up algorithm with its results significantly exceeded results obtained by the Top–Down, and especially Sliding Window.

The conducted comparative analysis and its results have inspired the above mentioned authors to investigate the possibilities for creating a new, alternative algorithm, which will integrate the strengths and mitigate the disadvantages of generic algorithms. This idea has been embodied in the algorithm, which is called the SWAB algorithm. The basic definitions of this algorithm are presented in Section 3. In order to test its properties on the same datasets conceptually identical empirical comparison was carried out, comparing the quality of SWAB, Bottom–Up, and Sliding Window approximations. By testing and comparing the performances of these three algorithms, the following results were obtained (see, Figure 13 (B)):

- in all trials, Sliding Window algorithm proved to be the algorithm with by far the worst
- in exactly seven trials, Bottom–Up algorithm showed slightly better results than the SWAB algorithm, but in twelve trials, recorded advantage belongs to the SWAB algorithm;
- in most trials (41), the determined quality of the approximations obtained by the Bottom–Up and SWAB algorithm were essentially identical.

Figure 13. The results of empirical comparison of the performances of the segmentation algorithms



Source of data: empirical research conducted by Keogh et al. (2004)

Presented considerations, once again, reaffirm the previously highlighted complexity of the optimisation of the problem of the selection of segmentation algorithm. Proper selection includes not only knowledge of the Pseudo-Code of algorithms, but also the essence of the analysed problem.

Segmentation of time series – why and how?

Methodological complexity of segmentation (determination of the number and length of the segments, algoritmisation, the accuracy of the approximation forms, methodological correlation with the techniques of other tasks of TSDM), along with the implications on dimensionality reduction of the original values of the time series, has been discussed in the previous sections of the Paper. However, a complete understanding of segmentation also includes highlighting its importance in terms of the analysis of phenomena in various fields, from natural phenomena to scientific, engineering, and medical experiments. In this context, the role of segmentation, from the standpoint of scientific researches of economic phenomena, is clear and meaningful. In fact, the inherent property of the economic phenomena is high dimensionality. Through the analysis in many economic domains (such as, for example, stock market analysis, budgetary analysis, analysis of fluctuations in the level of product quality, analysis of market segmentation, inventory forecasting, sale forecasting analysis, etc.), by reducing the actual dimensionality of huge amount of raw data to much lower level, through high quality representations and abstractions, the key and relevant patterns and rules are identified.

Accordingly, the importance of segmentation procedure can be viewed from the following two aspects:
a) from the standpoint of its contribution to the improvement of quality of time series analysis, and
b) from the standpoint of solutions to business problems and realization of business goals converted into the form of data mining tasks.

In fact, in the analysis of time series in the focus of research interest are the following analyses: trend analysis, the analysis of cyclical and seasonal fluctuations, the analysis of similarity within the segments of the series, the analysis of correlations between both time series and its parts, the analysis and identification of connections that exist between time series and its parts and the appropriate corresponding market trends, the analysis of autocorrelation etc. Therefore, all aspects of mining time series data are focused on identification of rules and patterns in the movement of the components that characterize the time series, in order to predict the future behavior of the observed variable on the basis of understanding the past.

Over time, the level of methodological complexity and hence the level of accuracy in predicting, have increased from classical decomposition models over the simple moving averages and exponential smoothing methods to the inclusion of the elements of regression analysis and sophisticated ARIMA models. Today, at the time of information–technological revolution, the entire arsenal of highly sophisticated algorithmic methods and procedures with appropriate software support and solutions, which are based on a combination of data mining methods and time series analysis (from various transformation techniques for reduction of dimensionality to neural networks and fuzzy logic) has appeared. This combination allows the extraction of the key characteristics of time series relevant in terms of creating a better predictive models.

In this context, the segmentation methodology is a very important topic in the processing of time (temporal) information contained in long, high-dimensional series. As already noted, the segmentation procedure provides a concise representation of the characteristics of segments of time, as well as complete time series. The resulting segmented series are a starting point for the creation of complex models and implementation of sophisticated analysis based on algorithms such as statistical methods, clustering, decision tree, market basket analysis, and linking them and combining additional possibilities, in terms of extraction of knowledge, can be provided and created.

From the above stated follows that depending on the purposes of analysis, the time series can be viewed at the level of individual segments, number of segments, or as the entire series. Through the extraction and creation of internally homogeneous segments in the series (which can be represented by statistic parameters such as average, standard deviation, dynamic model, etc.), segmentation makes it possible to locate the stable periods of time, to identify significant changes in the behaviour of the observed variable, the “change-points” or “cutting-points”, as well as the time of their occurrence. Generally, the segmentation process functions as follows: first, the algorithm searches for the known data behaviour, and then uses some additional knowledge to define precisely the time duration of each segment. Subsequently, on a series of shorter sequences, obtained as a result of the application of the particular segmentation algorithm and transformation of longer sequences, more thorough and detailed analysis is carried out by running (initiating) the appropriate algorithms for classification, assessment and prediction of behaviour of the observed data (observations). For example, for the extraction of relevant patterns, by running the classification algorithm, each segment is classified according to its average, variance, slope tendency, presence of outliers, etc.

Therefore, the segmentation process is important for the identification of structural changes in the data, i.e. changes in the time series data characteristics. The importance (and attractiveness) of the application of the algorithmic segmentation methods here is demonstrated through the analysis of financial time series, specifically, stock price time series. The goal of segmentation is decomposition of non-stationary stock price time series into a small number of homogenous pieces, i.e. a set of stationary segments. The extraction and definition of interpretable, novel, and useful patterns is based on the detection of: ▸special events, and key cutting and extreme points in price movements, and ▸stable (flat) periods of time, as well as the periods with decreasing or increasing trends (uptrend and downtrend) in terms of similar statistical properties. For example, if the objective of the potential investor is to establish the most appropriate moment for buying or selling the stocks in financial market, the determination of this moment is possible on the basis of analysis of (massive) time series and their average daily prices. In this context, the segmentation is required for discovering patterns and rules in movements of average daily prices during the entire period of time, as well as some parts of that period (segments). In that mannert he segments, which are characterised by the highest and lowest average daily price, the highest or lowest oscillations of average price, etc. are identified. Discovered rules and patterns of price movements in the past period, in constellation with analysis of other relevant factors (variables), provide prediction of future movements of prices which will determine not only the choice of the most appropriate moment for buying or selling of stocks, but also, according to the assessed risk, the sum and structure of funds for trading in financial market.

Hence, in addition to identification of important data points, in the stock prices analysis, primarily when it comes to prediction, it is necessary to consider the impact of the number of other variables that affect the frequent changes in financial markets. Therefore, the segmentation of financial time series “corresponds to splitting the time into different phases for the economy, such as recession, recovery, expansion, market behaviour after terrorist attack, etc.”(Bingham et al. 2006, p.370). Tendencies in stock prices movements can be explained by the regression model and the influence of explanatory variables, such as the oil price, the general state of the economy and its various sectors, the measures of economic policy, etc., the direction and degree of influence of which can be different in different periods of time (i.e. phases).

A very interesting model (with unusual explanatory variable) for predicting trends in stock prices is based on the impact of the content of news stories on forthcoming trends in the stock prices (see, Lavrenko et al. 2000, for details). In addition, news stories are viewed from two angles. On the one hand, news stories are seen as a factor that influences behaviour and decisions of the actors in financial markets, and consequently, the stock prices. News stories, on the other hand, can also be seen as a source of current information about fluctuations that already occurred in financial markets. For successful connection of the news stories and trends in stock prices time series, through an analysis of all these influences, special language models have been created, based on Text mining and TSDM.

However, extraction of relevant information and knowledge from large time series datasets, regardless of the research area to which they relate, is practically impossible without the application of appropriate software solutions. Popular software packages (as a collection of data mining algorithms), which include toolboxes or modules for time series analysis are: *WEKA* (*W*aikato *E*nvironment for *K*nowledge *A*nalysis), *Rapid Miner* (formerly Yale), *IBM PASW* (formerly SPSS-Clementine), *SAS* (Enterprise Miner), and *MATLAB* (*mat*rix *lab*oratory). The mentioned software solutions support only some aspects of statistical time series analysis, and this is considered to be their main fault. Therefore, the discovery of the structure of time series datasets must be based on a compilation of available software solutions and,of course (by default), expert knowledge of data miners.

Conclusion

It is generally known that the time series, as essential high-dimensional collections of data, permeate all areas of business and scientific research. Therefore, the time series analysis is continuously active and attractive research area. Knowledge discovery in large datasets of time series by identifying interpretable, novel and useful temporal patterns is practically not possible without the application of data mining methodology in time series analysis. In general, all aspects of mining time series data are focused on identification of rules and patterns in the movement of the components that characterise the time series (including trend movements, seasonal variations, cyclical variations, and random movements), in order to, on the basis of understanding the past, predict the future behaviour of the observed variable.

The key component in the data mining process of discovering the structure in time series data is the segmentation of time series. As a data mining research problem, it is focused on the division of time series into adequate internally homogeneous segments, which are, in a way, a time series for themselves, that are composed of several consecutive data points. In addition, the process of conversion of the actual time series into its segmented approximation and reduction of dimensionality provides compact representation of the underlying time series data, which is more suitable for discovering the relevant and interesting information in data.

Time series segmentation is often used as a pre-processing step in time series data mining applications, so that the quality of the output of the segmentation process is the key factor that determines the quality of the analysis and validity of the time series models. The implementation of the segmentation is based on the entire range of the algorithmic methods. As a precisely defined set of instructions and rules for conducting the transformation of the original time series into its segmented version, they should provide a finding of the best segmentation to represent the given time series, and, in terms of well-defined segmentation criteria, ensure that the data can speak for themselves.

In this Paper, the family of algorithms for segmentation of time series, which are based on the Piecewise Linear Representation, is presented and described, on the basis of the insight into the reference literature.The future work will include empirical research and discovery of hidden information in the structures of real economic time series datasets using appropriate software applications.

References

Antunes, C. and Oliveira A. (2001) Temporal Data Mining: An Overview, in: Proceedings of the Workshop on Temporal Data Mining at the 7th International Conference on Knowledge Discovery and Data Mining, ACM Press, San Francisco, California, pp. 1-13.

Aksoy, H., Gedikli, A., Unal, N. E. and Kehagias, A. (2008) Fast Segmentation Algorithms for Long Hydrometeorological Time Series, Hydrological Processes, 22, pp. 4600-4608.

Bingham, E., Gionis, A., Haiminen, N., Hiisilä, H., Mannila, H. and Terzi E. (2006) Segmentation and Dimensionality Reduction, in: Proceedings of the 2006 SIAM International Conference on Data Mining, pp. 370–381. Retrieved from:<http://www.cs.helsinki.fi/u/mannila/>

Chundi, P. and Rosenkrantz D. (2009) Segmentation of Time Series Data, in J. Wang (Ed.), Encyclopaedia of Data Warehousing and Mining, Information Science Reference, New York, USA, pp. 1753–1758.

Chung, L., Fu, T. C. and Luk, R. (2004) An evolutionary approach to pattern-based time series segmentation, IEEE Transactions on Evolutionary Computation, IEEE Press, Vol. 8, Issue 5, pp. 471-489.

Das, G., Lin, I. K., Mannila, H., Renganathan, G. and Smyth, P. (1998) Rule Discovery from Time Series, in: Proceedings of the Fourth Annual Conference on Knowledge Discovery and Data Mining, pp. 23–29.Retrieved from:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.3240&rep=rep1&type=pdf

Eamonn Keogh <http://www.cs.ucr.edu/~eamonn/> (Accessed: December 20, 2013)

Fu, C., Chung, F. L., Ng, V. and Luk, R. (2001) Evolutionary segmentation of financial time series into sub-sequences, in: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, pp. 426-430.

Fu, Tak-chung (2011) A review on time series data mining, Engineering Applications of Artificial Intelligence, Vol. 24, Issue 1, pp.164-181.

Gionis, A. and Mannila, H. (2003) Finding recurrent sources in sequences, in: Proceedings of the 7th annual international conference on research in computational molecular biology (RECOMB 2003), pp. 123–130.

Gionis, A. and Mannila, H. (2005) Segmentation Algorithms for Time Series and Sequence Data, in: Tutorial at 5th SIAM International Conference on Data Mining.

Gionis, A. and Terzi, E. (2007) Segmentations with Rearrangements, in: Proceedings of the 2007 SIAM International Conference on Data Mining, pp. 285–296. Retrieved from: <http://cs-people.bu.edu/evimaria/papers/swaps.pdf>

Haiminen, N. and Gionis, A. (2004) Unimodal segmentation of sequences, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 106–113.

Han, W. S., Lee, J., Moon, Y. S. and Jiang, H. (2007) Ranked subsequence matching in time-series databases, in: Proceedings of the 33rd International Conference on Very Large Databases, pp. 423–434.

Hand, D. J. (1999) Why Data Mining is more than Statistics Writ Large, Bulletin of the International Statistical Institute, 52nd Session, Vol. 1, pp. 433-436.

Hiisilä, H. (2007) Segmentation of Time Series and Sequences Using Basic Representations, Ph.D. Thesis, Helsinki University of Technology, Laboratory of Computer and Information Science, Helsinki, Finland, Retrieved from: <http://cis.legacy.ics.tkk.fi/heli/lisuri.pdf>

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J. and Toivonen, H. T. (2001) Time series segmentation for context recognition in mobile devices, in: Proceedings of the IEEE International Conference on Data Mining (ICDM’01), California, USA, pp. 203-210.

KDD - bringing together data mining, data science and analytics community <http://www.sigkdd.org/index.php> (Accessed: December 2013)

Keogh, E. (2010) Data Mining Time Series Data, in M. Lovrić(Ed.), *International Encyclopaedia of Statistical Science*. New York, USA: Springer.

Keogh, E., Chu, S., Hart, D. and Pazzani, M. (2001) An Online Algorithm for Segmenting Time Series, in: *Proceedings of the 2001 IEEE International Conference on Data Mining*, Retrieved from: <http://www-scf.usc.edu/~selinach/publications.html>

Keogh, E., Chu, S., Hart, D. and Pazzani, M. (2004) Segmenting Time Series: A Survey and Novel Approach, in M. Last, A. Kandel, H. Bunke (Eds.), *Data Mining in Time Series Databases*, Singapore: World Scientific Publishing Co.,pp. 1–21.

Kovačić, Z.(1995)*Time series analysis*, Faculty of Economics, University of Belgrade.

Kugiumtzis, D. and Tsimpiris, A. (2010) Measures of Analysis of Time Series (MATS): A MATLAB Toolkit for Computation of Multiple Measures on Time Series Data Bases, *Journal of Statistical Software*, Vol. 33, Issue 5, Retrieved from: <http://www.jstatsoft.org/>

Last, M., Kandel, A. and Bunke, H., (Editors) (2004)*Data Mining in Time Series Databases*. Singapore: World Scientific Publishing Co.

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D. and Allan, J. (2000) Mining of Concurrent Text and Time Series, in: *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining,* pp. 37-44.

Lemire, D. (2007) A Better Alternative to Piecewise Linear Time Series Segmentation, in: *Proceedings of the Seventh SIAM International Conference on Data Mining2007*, pp. 545-550. Retrieved from: <http://arxiv.org/PS_cache/cs/pdf/0605/0605103v8.pdf>

Lin, J., Keogh, E. and Lonardi, S. (2005) Visualizing and Discovering Non-Trivial Patterns in Large Time Series Databases,*Information Visualization Journal*, Vol. 4, Issue 2, pp. 61–82.

Lin, J., Keogh, E., Wei, L. and Lonardi, S. (2007) Experiencing SAX: a Novel Symbolic Representation of Time Series, *Data Mining and Knowledge Discovery*, 2, pp. 107–144. Retrieved from: <http://www.citeulike.org/user/lenov/article/2821475>

Liu, Lon-Mu (2009) *Time Series Analysis and Forecasting, Second edition*, Scientific Computing Associates Corp., USA.

Milanović M. and Stamenković, M. (2010) Data Mining and Segmentation of Time Series for Knowledge Discovery, in: *Proceedings of theInternational Scientific Conference – The Challenges of Economic Science and Practice in the 21st Century*, Faculty of Economics, Niš, pp. 679-689.

Milanović M. and Stamenković, M. (2011) Data Mining in Time Series, *EkonomskiHorizonti*, YU ISSN:1450-863 X, Faculty of Economics, University of Kragujevac, Vol. 13, Issue 1, pp. 5-25, (005.94; 005.511:519.246.8; original scientific article)

Mörchen, F. (2006)*Time Series Knowledge Mining*, Ph.D. Thesis, Philipps-University Marburg, Germany, Retrieved from: <http://www.mybytes.de/papers/moerchen06tskm.pdf>

Park, S., Kim, S. W. and Chu, W. W. (2001) Segment-Based Approach for Subsequence Searches in Sequence Databases, in: *Proceedings of the 16th ACM Symposium on Applied Computing,* (*SAC'01*), pp. 248-252.

Povinelli, R. J. (1999)*Time Series Data Mining: Identifying Temporal Patterns for Characterization and Prediction of Time Series Events*, Ph.D. Thesis, Marquette University, Faculty of the Graduate School, Milwaukee, Wisconsin.
Retrieved from: http://povinelli.eece.mu.edu/publications/papers/dissertation.pdf

Ratanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M. and Das, G. (2005) Mining Time Series Data, in: *Data Mining and Knowledge Discovery Handbook*, Springer, pp.

1069-1103.

Shatkay, H. and Zdonik, S. B. (1996) Approximate queries and representations for large data sequences, in: *Proceedings of the International Conference on Data Engineering* (*ICDE*),pp. 536–545.

St. Louis Fed: Economic Research (2010). Retrieved October 10, 2010, from:
<http://research.stlouisfed.org/fred2/series/TCU/downloaddata?cid=3>

St. Louis Fed: Economic Research (2010). Retrieved October 10, 2010, from:
<http://research.stlouisfed.org/fred2/series/RRSFS>

Terzi, E. and Tsaparas, P. (2006) Efficient Algorithms for Sequence Segmentation, in: *Proceedings of the 2006 SIAM International Conference on Data Mining,*pp. 314-325. Retrieved from: <http://cs-people.bu.edu/evimaria/papers/TT06.pdf>

The Financial Forecast Centre (2010). Retrieved October 10, 2010, from:
<http://www.forecasts.org/data/data/djiaM.htm>

Time Series Data Library (2010). Retrieved October 8, 2010, from:
<http://robjhyndman.com/tsdldata/data/9-9.dat>

Ueno, K., Xi, X., Keogh, E. and Lee, J. (2006) Anytime Classification Using the Nearest Neighbour Algorithm with Applications to Stream Mining, in: *Proceedings of the 6th IEEE International Conference on Data Mining*, pp. 623–632. Retrieved from <http://www.cs.ucr.edu/~eamonn/selected_publications.htm>