

A Cluster-based Algorithm for Anomaly Detection in Time Series Using Mahalanobis Distance

Erick Giovanni Sperandio Nascimento^{1a}, Orivaldo de Lira Tavares¹, and Alberto Ferreira De Souza¹

Erick@lcad.inf.ufes.br, tavares@inf.ufes.br, alberto@lcad.inf.ufes.br

¹Department of Informatics, Federal University of Espirito Santo, Vitória, ES, Brazil

^aCorresponding Author

Abstract - We propose an unsupervised learning algorithm for anomaly detection in time series data, based on clustering techniques, using the Mahalanobis distance function. After a brief review of the main and recent contributions made in this research field, a formal and detailed description of the algorithm is presented, followed by a discussion on how to set its parameters. In order to evaluate its effectiveness, it was applied to a real case, and its results were compared with another technique that targets the same problem. The obtained results suggest that this proposal can be successfully applied to detect anomaly in time series.

Keywords: time series, anomaly detection, clustering, unsupervised learning, mahalanobis distance, pattern recognition

1 Introduction

Nowadays, many processes, such as industrial plants, meteorological monitoring stations or stock markets, generate relevant time series data continuously. In general, these data are collected and stored by specific hardware, and later are analyzed and maintained by specialized people, who learn about these processes using the data, but are also responsible for verifying its correctness in representing the real state of the processes.

In many situations, it is critical for the process to identify unusual patterns that could be generated by unexpected behavior. And such unwanted behavior may be due to any problem that the related process might be experiencing. For example, an industry may monitor some variables of its current productive process to diagnose bottlenecks, violations of quality requisites, violation of environmental requisites such as a specific pollutant emitted to the environment over the permitted by law, or any other situation that could be harmful to its business. Another example is one certain environmental institute or agency, whom would need to monitor some meteorological or air

quality parameters in order to evaluate the air quality of an urban area, might experience that some equipment were presenting failure, which could lead to misunderstand data monitoring. Either, a credit card company may monitor each user transaction to look for unusual behaviors that could point to fraudulent operations. These unusual, unwanted behaviors are often called as *anomalous* behaviors, and might be induced in the data due to a variety of reasons, all of them presenting a certain degree of importance to the analyst. And it is important that this analysis could take into account any changes in the parameter's behavior to identify opportunities to improve, prevent or correct any situation.

In this context we present an unsupervised learning algorithm based in clustering techniques using the Mahalanobis distance as its distance function targeting the problem of anomaly detection in time series, here called as *C-AMDATS*, which stands for *Cluster-based Algorithm using Mahalanobis distance for Detection of Anomalies in Time Series*. The paper is organized as follows: the remainder of this section presents a brief review of the recent research regarding anomaly detection in time series. Section 2 presents the foundations of the algorithm, and a detailed and formal description of the algorithm. Section 3 presents a real case with anomalous patterns that was evaluated in order to assess the ability of the *C-AMDATS* approach to detect these anomalies, in conjunction with a comparison with other technique targeting the same problem, i.e., the detection of anomalies in time series. Section 4 presents a conclusion and recommendations for future works. And in Section 5 we acknowledge our main contributors.

1.1 Related Work

Several works have been developed to identify patterns in time series data, and some of them were specialized to detect anomalous patterns in time series. We will briefly present a review about the most recent works in anomaly detection in time series in order to identify

whether our proposed technique is introducing a new contribution to the community.

Some works uses distance-based techniques, like in [2,3,4,5,6,7,8] to detect outliers or anomalies in time series. Other works uses sliding windows and discretization techniques. In some cases, a single time series is converted to a time series database through the use of a sliding window incrementally [9,10,11,12,17] or in discrete steps according to a known period [13,14]. Specifically in [17], the authors present a technique, called *SAX*, which addresses anomaly detection using time series discords, and applies it to real cases. We chose to compare our technique with this one because there is a graphic visualization tool and user interface, called *VizTree* [18], which implements the technique. Unlike methods that seek anomalies of a pre-specified length, the method presented in [15] looks for anomalies at varying levels of granularity (i.e., day, month, year), using a tree structure called *TSA-Tree* that contains pre-computed trend and anomaly information in each node. The InfoMiner technique [13, 14] detects “surprising” patterns on periodic event sequence data. Thus, the data is already discretized, and the known period allows the authors to treat a single continuous time series as a set of smaller one period time series. The work presented in [1] introduces a technique to identify patterns in time series data using an algorithm called by them as *PCAD* – Periodic Curve Anomaly Detection, which is a clustering-based algorithm built above the basis of the k-means algorithm, that outputs a ranked list of both global and local anomalies. The technique developed in [16] proposes an approach that employs a kernel matrix alignment method to capture the dependence relationships among variables in the time series in order to detect anomalies.

Some of these works have been extracted from [19], which brings a literature survey about clustering time series data stream that we recommend to be read as a supplementary reference about the related work. We chose some of the most recent researches regarding anomaly detection in time series data. In the next section we present our proposal of an algorithm based in clustering techniques with some enhancements built to let it recognize anomalies in a single time series data. The results of its application will be further evaluated in this paper.

2 The Proposed Algorithm

Our algorithm, presented in Box 1, is a dynamic clustering procedure that, given (i) a time series $T = \{t_1, t_2, \dots, t_{|T|}\}$ of real-valued, time-indexed variables sampled at a certain frequency and ordered by time, (ii) the initial clusters' size τ , and (iii) the clustering factor φ , computes a

set of anomalous patterns in T , $\mathbf{P} = \{P_1, P_2, \dots, P_{|\mathbf{P}|}\}$, where $P_j = \{C_1, C_2, \dots, C_{|P_j|}\}$ is an anomalous pattern of T , which is composed of a set of disjoint clusters $C_k = \{t_a, t_{a+1}, \dots, t_b\}$, $1 \leq a \leq b \leq |T|$, that are ordered subsets of T .

C-AMDATS (T, τ, φ)

```

1.  $\mathbf{C} \leftarrow \text{ComputeInitialClusters}(T, \tau);$ 
2. while changes in  $\mathbf{C}$  happen do
3.    $\mathbf{C}' \leftarrow \mathbf{C}$ 
4.   for  $i \leftarrow 1$  to  $|T|$  do
5.     Move  $t_i$  from its cluster in  $\mathbf{C}$  to the
6.     nearest cluster in  $\mathbf{C}$  according to
        $f(\mathbf{C}')$ 
7.   endfor
8. endwhile
9. repeat
10.   Add  $C_1$  to  $\mathbf{P}$ ;
11.   Remove  $C_1$  from  $\mathbf{C}$ ;
12.    $k \leftarrow 1$ ;
13.   while  $k \leq |\mathbf{C}|$  do
14.     for  $j \leftarrow 1$  to  $|\mathbf{P}|$  do
15.       if  $P_j$  is similar to  $C_k$  then
16.         Add  $C_k$  to  $\mathbf{P}_j$ ;
17.         Remove  $C_k$  from  $\mathbf{C}$ ;
18.       else
19.          $k \leftarrow k + 1$ ;
20.       endif
21.     endfor
22.   endwhile
23. until  $|\mathbf{C}| = 0$ 
24. for  $i \leftarrow 1$  to  $|\mathbf{P}|$  do
25.   Compute the Anomaly Rank  $r(\mathbf{P}_i)$ 
26. endfor
27.  $\text{SortByAnomalyRank}(\mathbf{P})$ ;
28. return  $\mathbf{P}$ ;
```

Box 1 – C-AMDATS Pseudo-Algorithm

The algorithm starts in line 1 by computing the set of equal sized initial clusters, \mathbf{C} . In this step, the set T is split into a set of sets, \mathbf{C} , where each subset $C_k = \{t_a, t_{a+1}, \dots, t_b\}$ has size $|C_k| = \tau$, i.e., $b - a = \tau$ (except the last cluster, in cases where $|T|$ is not divisible by τ). After that, in lines 2-8, the algorithm rebuilds \mathbf{C} iteratively using its copy, \mathbf{C}' , computed before the iteration (line 3). For that, in lines 5-6, the algorithm uses $f(\mathbf{C}')$ to determine which cluster in \mathbf{C} is the nearest to t_i . In the algorithm, $f(\mathbf{C}')$ computes the average of the time series segments within each cluster C'_k , or the centroid of each C'_k , $m_k = (t_a + t_{a+1} + \dots + t_b) / (b - a)$ (see **Fig. 1**), and the distance, $d(t_i, m_k)$, from t_i to each m_k , for $1 \leq k \leq |\mathbf{C}'|$. Using these distances, in lines 5-6 the algorithm moves t_i from its current cluster in \mathbf{C} to the cluster C_k , where k is the index of the cluster C'_k whose centroid m_k is the nearest to t_i according to $d(\cdot, \cdot)$. Our choice of distance function $d(\cdot, \cdot)$ is the Mahalanobis distance [20]. We explain this choice below but, in our

experiments, we also use the Euclidian distance for comparison.

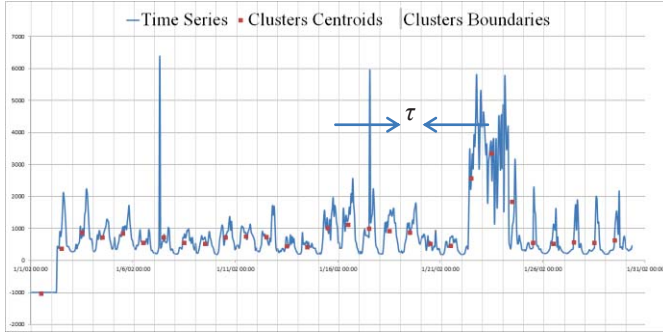


Fig. 1. A time series T split into equal-sized clusters C'_k , each of which of size t . The red dots are the centroids, m_k , of each cluster C'_k .

The loop of lines 2-8 terminates when no sample t_i is moved in lines 4-7. After this loop, we have the set of sets, C (see **Fig. 1**), composed of clusters of samples, C_k , that better group the samples according to their sample values distributed over time (please compare **Fig. 1** with **Fig. 3**; note that the size of each cluster C_k in **Fig. 3** is not the same). This happens because of our choice of distance function. In clustering problems, it is common to use the Euclidean distance function. Its use leads to clusters with circular format, due to the fact it does not take into account the variance of each dimension of the data set. However, it is possible that this circular shape may not be suitable to represent the cluster's shape. To solve this problem, another distance function should be used to build clusters that take in consideration variances in the x and y axes. The Mahalanobis distance differs from the Euclidean distance in that it takes into account the variances of each dimension (see **Fig. 2**). The **Equation (1)** presents the formulation of the Mahalanobis distance [20]:

$$d_m(x, \mu) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (1)$$

In **Equation (1)**, $x = (x_1, x_2, \dots, x_n)^T$ is a specific variable in the data set, where n is the number of dimensions of the variables, $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$ is a certain cluster centroid and S is the covariance matrix relative to that cluster.

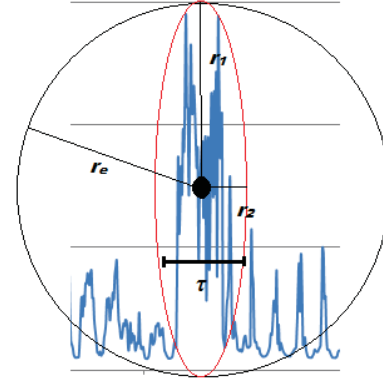


Fig. 2. A sample of a time series illustrating the differences between the application of the Euclidean (forming the circle) and Mahalanobis (forming the ellipse) distance functions.

In **Fig. 2**, τ is the initial clusters' size, r_e is the radius of the circle that fits the cluster, and r_1 and r_2 correspond to the radii of the ellipse that fits the same cluster as well. The r_e value is the Euclidean distance of farthest point in the cluster to its centroid, being big enough to embrace all the points in the cluster, while the r_1 and r_2 values are obtained by the application of the Mahalanobis distance. As we can note, the shape that best fits the cluster is the ellipse, while the circle is grouping regions that do not fall into the cluster. It is due to the fact that the Mahalanobis distance function takes into account both dimensions simultaneously, not separately. In order to show the real impact of using this distance function rather than the Euclidean distance, we will present the results of applying both to a real case in the Section 3.

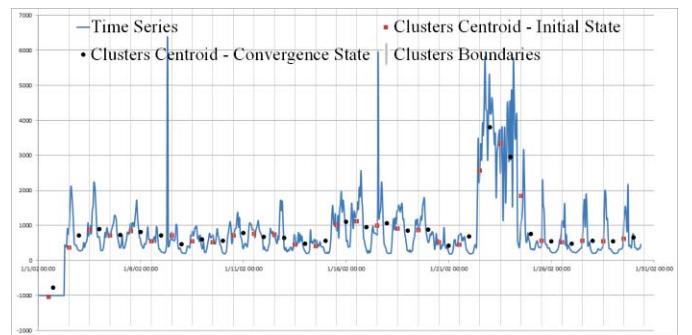


Fig. 3. A time series T split into clusters C'_k , each of which of variable size. The red dots are the centroids at the initial state of the algorithm, and the black dots are the centroids after the iterative process at lines 2-8

The following step (lines 9-23) performs the task of finding the final patterns P in the time series T . After all clusters have been found, the algorithm verifies which

clusters are *similar*. This similarity is calculated using the standard deviation σ_y of the real-values of the variables in T , the y-coordinate of each cluster and the *clustering factor* ϕ . If the modulus of the difference between the y-coordinate of the centroids of two clusters is less than or equal to σ_y times ϕ , then these clusters can be *merged*, meaning that they will represent the same pattern P . This task is performed till all the clusters have been analyzed.

In the last step (lines 24-27), the algorithm performs the detection of the anomalies. An anomaly is a pattern that does not conform to an expected behavior in T , i.e. an anomalous pattern. This detection is done by computing the anomaly score r for each pattern P found in the previous step, which is calculated as the ratio of the size of the entire time series by the summation of the sizes of the clusters present in P . The anomaly score (or rank) r is a measure of how much P is interesting in terms of being an anomaly. Following, the entire set P is ordered by r in descending order, and the anomalous patterns will be those with the highest anomaly score values. The higher the anomaly score value for a pattern P , the greater is its chance to be an anomaly in T . In **Fig. 4** we present the final state of the algorithm: all similar clusters have been merged into a pattern, as stated by the criteria described above. Three patterns have been found, and according to their anomaly score, the most anomalous are those highlighted in red and green color, while the blue pattern is the least.

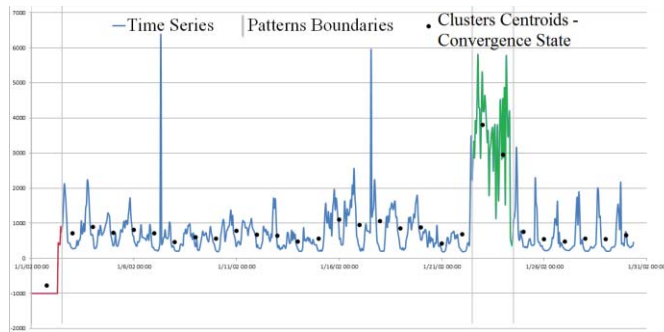


Fig. 4. A time series T divided into three patterns, at the final of the execution of the algorithm. The green and red are the most anomalous.

The complexity of the $C\text{-AMDATS}$ is $O(nkz)$, where n is the number of variables, k is the number of initial clusters, and z is the number of iterations till the convergence state. Since it is a derivation of the k-means algorithm, it can also be classified as a NP-Hard problem [21], meaning that the algorithm will stop at the z iteration due to its stop criterion, but there is no guarantee that the absolute minimum of the objective function can be reached.

In Section 1.1 we presented a review of the related work. To the best of our knowledge, it was not possible to find any other clustering algorithm for anomaly detection in time series data that could be even similar to this technique here presented.

3 C-AMDATS – Applications and Results

To verify its ability to analyze real time series data, this technique was applied to real cases. Hence, a real case episode was selected. It will be further presented and discussed, as well as the results of the application of the $C\text{-AMDATS}$ algorithm. During the tests, two versions of the algorithm were developed: one using the Euclidean distance ($C\text{-AMDATS}_E$) and the other using the Mahalanobis distance ($C\text{-AMDATS}_M$). The experiments showed that the application of the Mahalanobis distance led to better results, but it took more CPU time than the application of the Euclidean distance function due to the need to compute the inverse of the covariance matrix for each cluster, at each iteration step. We will present results using both distance functions.

To assess the algorithm's performance with respect to its ability to identify the same anomalous patterns identified by the human specialists, its results were compared to those patterns using the *precision*, *recall* and *accuracy* methodologies [22]. Also, a *confusion matrix* was built to show the differences between each approach.

3.1 Real Case – Carbon Monoxide

This case refers to the measurement of carbon monoxide during two months in the year of 2002. The data is hourly sampled, and was collected in a metropolitan area, by an automated monitoring system maintained. For reference, we will use its chemical representation, CO . Its cycle's length is of 24 hours.

The **Fig. 5** shows the result of the $C\text{-AMDATS}_M$ approach for this case. Three major patterns are highlighted: the red, green and blue, in order of anomaly score, which led us to select the patterns highlighted with red and green color as the anomalous patterns. In **Table 1** we present the confusion matrix for this case. The patterns are also delimited by vertical lines.

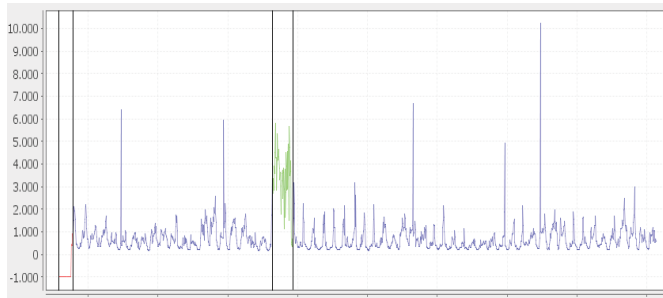


Fig. 5. Time plot for CO highlighting the anomalous patterns found by $C\text{-}AMDATS_M$

Table 1. Confusion matrix for this case

	Anomalous Pattern		Precision	Recall	Accuracy
$C\text{-}AMDATS_M$	82	2	0.9762	0.9111	0.9931
	1 348	8			
$C\text{-}AMDATS_E$	66	7	0.9041	0.7333	0.9785
	1 343	24			
SAX	40	8	0.8333	0.4444	0.9597
	1 342	50			

The values of the parameters for $C\text{-}AMDATS$ were: *initial clusters' size* of 24 hours and *clustering factor* of 1.2. For SAX, we spent about 2 hours looking for a best combination of its parameters, and we found that a *window length* of 24h, *number of symbols per window* of 3 and *alphabet size* of 4 performed the best. Moreover, we also had to set one advanced option in the VizTree tool, called “No Overlapping Windows”, which led to the best results we could experiment.

Similarly, both approaches were able to find, some partially, the region of the anomaly subsequence. However, the SAX approach was just able to give a clue about the second anomaly, as we can see in **Fig. 6**, and the $C\text{-}AMDATS_M$ could give a good result in comparison with the others.

For SAX, we extracted the most meaningful branches regarding these anomalous patterns, which corresponded to “ccc” and “acc”.

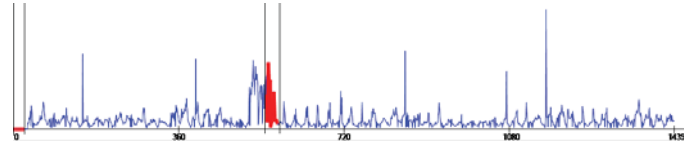


Fig. 6. Time plot for CO highlighting, in red, the anomalous region found by the SAX technique

4 Conclusions and Future Work Recommendations

In this work we presented a proposal of an algorithm for anomaly detection in time series. We showed that there is a plenty of applications, and also many contributions made to this research field. We identified the main contributions presented recently and analysed them to identify whether the algorithm proposed by these authors is indeed a new contribution to the community. We verified that there is no similar technique. We presented the concepts behind this work, and then we described our proposal. Finally, we applied the algorithm to a real data case, and have identified that it performed good results in comparison with the other approach, which shows that it can be applied as a tool to leverage the specialists' job in analyzing and identifying anomalies in time series data.

There are several future works to be developed in the following:

- to write a full paper of this work, describing in more details some issues regarding the review of the related work, the description of the algorithm and the methods for assessing its performance, applying the algorithm to other real cases and compare with the same technique (SAX);
- to use $C\text{-}AMDATS$ in an operational environment, where the algorithm would be set up to work continuously, with the jobs of analysing time series data, finding patterns, and sending status report to specialists. Then, these specialists would be able to verify the results of the algorithm at real time;
- to implement a function to analyze correlated parameters at once to find anomalies between them, e.g. ozone and solar radiation, which are different parameters but have an intrinsic correlation. This recommendation would demand creating a derivation of the $C\text{-}AMDATS$ algorithm to be applied to analyze various time series data at the same execution;
- to design a learning module to learn from the user what are the best values computed at a certain moment, based upon past applications of the algorithm that have been validated by the user.

Based on the results here presented, we think this work could be successfully applied in several areas of this research field to improve the way time series data are analyzed in order to detect anomalies.

5 Acknowledgements

We acknowledge the Espirito Santo Research Support Foundation (*FAPES*), which partially funded this work. We also acknowledge our contributors, who gave us access to real data and let us publish the results. Namely, we thanks the Environmental Institute of the state of Espirito Santo, Brazil – *IEMA* for the data that is related to the real case.

6 References

- [1] Umaa Rebbapragada, Pavlos Protopapas, Carla Brodley, Charles Alcock. “Finding Anomalous Periodic Time Series: An Application to Catalogs of Periodic Variable Stars”, *Spring Machine Learning Journal* (Springer), Vol. 74, Issue 3, 281-313, Mar 2009. DOI: 10.1007/s10994-008-5093-3;
- [2] Edwin Knorr, Raymond Ng. “Algorithms for Mining Distance-Based Outliers in Large Datasets”, In: *Proceedings of the 24th International Conference on Very Large Data Bases – VLDB, VLDB International Conference*, pp. 392–403, 1998;
- [3] Sridhar Ramaswamy, Rajeev Rastogi, Kyuseok Shim. “Efficient Algorithms for Mining Outliers from Large Datasets”, In: *SIGMOD '00: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD, pp. 427–438, 2000;
- [4] Fabricio Angiulli, Carla Pizzuti. “Fast Outlier Detection in High Dimensional Spaces”, In: *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 15–26, 2002;
- [5] Mingxi Wu, Christopher Jermaine. “Outlier Detection by Sampling with Accuracy Guarantees”, In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 767–772, 2006;
- [6] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, Jörg Sander. “LOF: Identifying Density-Based Local Outliers”, In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000;
- [7] Wen Jin, Anthony Tung, Jiawei Han. “Mining Top-N Local Outliers In Large Databases”, In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 293–298, 2001;
- [8] Dongmei Ren, Baoying Wang, W. Perrizo. “RDF: A Density-based Outlier Detection Method Using Vertical Data Representation”, In: *Proceedings of the 4th IEEE International Conference on Data Mining*, pp. 503–506, 2004;
- [9] Dipankar Dasgupta, Stephanie Forrest. “Novelty Detection in Time Series Data Using Ideas from Immunology”, In: *Proceedings of the International Conference on Intelligent Systems*, pp. 82–87, 1996. DOI: 10.1.1.57.3894;
- [10] Eamonn Keogh, Stefano Lonardi, Bill Yuan-chi Chiu. “Finding Surprising Patterns in a Time Series Database in Linear Time and Space”, In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 550–556, 2002;
- [11] Junshui Ma, Simon Perkins. “Online Novelty Detection on Temporal Sequences”, In: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 613–618, 2003;
- [12] Li Wei, Nitin Kumar, Venkata Lolla, Eamonn Keogh, Stefano Lonardi, Chotirat Ratanamahatana. “Assumption-Free Anomaly Detection in Time Series”, In: *SSDBM'2005: Proceedings of the 17th International Conference on Scientific and Statistical Database Management*, pp. 237–240, 2005;
- [13] Jiong Yang, Wei Wang, Philip Yu. “Infominer: Mining Surprising Periodic Patterns”, In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 395–400, 2001;
- [14] J. Yang, W. Wang, P. S. Yu. “Mining Surprising Periodic Patterns, Data Mining and Knowledge Discovery”, *Data Mining and Knowledge Discovery (ACM)*, Vol. 9, Issue 2, 189–216, Sep 2004;
- [15] Cyrus Shahabi, Xiaoming Tian, Wugang Zhao. “TSA-Tree: A Wavelet-Based Approach to Improve the Efficiency of Multilevel Surprise and Trend Queries on Time-Series Data”, In: *Proceedings of the 12th*

International Conference on Statistical and Scientific Database Management, pp. 55–68, 2000;

[16] Haibin Cheng, Pang-Ning Tan, Christopher Potter, Steven Klooster. “Detection and Characterization of Anomalies in Multivariate Time Series”, In: Proceedings of the 9th SIAM International Conference on Data Mining, pp. 413–424, 2009;

[17] Eamonn Keogh, Jessica Lin, Ada Fu. “HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence”, In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), pp. 226–233, 2005;

[18] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey Lankford, Daonna Nystrom. “VizTree: a Tool for Visually Mining and Monitoring Massive Time Series Databases”, In: Proceedings of the 30th International Conference in Very Large Data Bases, pp. 1269-1272, 2004;

[19] V. Kavitha, M. Punithavalli. “Clustering Time Series Data Stream – A Literature Survey”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 8, Issue 1, pp. 289-294, Apr 2010;

[20] Julius T. Tou, Rafael C. Gonzalez. “Pattern Recognition Principles”, Addison-Wesley, 1974;

[21] Meena Mahajan, Prajakta Nimbhorkar, Kasturi Varadarajan. “The Planar k-Means Problem is NP-Hard”, In: Proceedings of the 3rd International Workshop on Algorithms and Computation, Springer-Verlag, pp. 274–285, 2009. DOI: 10.1007/978-3-642-00202-1_24;

[22] David L. Olson, Dursun Delen. “Advanced Data Mining Techniques”, Springer, 2008.