Time Series Analysis Indexing Methods: Survey

T V Divya [1], Baddepaka Prasad [2], Kathi Venkata Ramana[3]
[1, 2] Assistant Professor in KMIT, Hyderabad
Research Scholar in Madhav University, Rajasthan
3

Abstract- *In almost every scientific field, measurements are performed over time. These observations lead to a collection of organized data called time series. The purpose of time-series data mining is to try to extract all meaningful knowledge from the shape of data. Even if humans have a natural capacity to perform these tasks, it remains a complex problem for computers. In this article we intend to provide a survey of the techniques applied for time-series data mining. The first part is devoted to an overview of the tasks that have captured most of the interest of researchers. Considering that in most cases, time-series task relies on the same components for implementation, we divide the literature depending on these common aspects, namely representation techniques, distance measures, and indexing methods. The study of the relevant literature has been categorized for each individual aspects. Four types of robustness could then be formalized and any kind of distance could then be classified. Finally, the study submits various research trends and avenues that can be explored in the near future.*

I. INTRODUCTION

A time series represents a collection of values obtained from sequential measurements over time. Time-series data mining stems from the desire to reify our natural ability to visualize the shape of data. Humans rely on complex schemes in order to perform such tasks. We can actually avoid focusing on small fluctuations in order to derive a notion of shape and identify almost instantly similarities between patterns on various time scales. Major time-series-related tasks include query by content [Faloutsos et al. 1994], anomaly detection [Weiss 2004], motif discovery [Lin et al. 2004], prediction [Weigend and Gershenfeld 1994], clustering [Lin and Keogh 2005], classification [Bakshi and Stephanopoulos 1994], and segmentation [Keogh et al. 2003a]. Despite the vast body of work devoted to this topic in the early years, Antunes and Oliveira [2001] noted that "the research has not been driven so much by actual problems but by an interest in proposing new approaches". However, with the ever-growing maturity of time-series data mining techniques, this statement seems to have become obsolete. Nowadays, time-series analysis covers a wide range of real-life problems in various fields of research. Some examples include economic forecasting [Song and Li 2008], intrusion detection [Zhong et al. 2007], gene expression analysis [Lin et al. 2008], medical surveillance [Burkom et al. 2007], and hydrology [Ouyang et al. 2010]. Time-series data mining unveils numerous facets of complexity. The most prominent problems arise from the high dimensionality of time-series data and the difficulty of defining a form of similarity measure based on human perception. With the rapid growth of digital sources of information, time-series mining algorithms will have to match increasingly massive datasets. These constraints show us that three major issues are involved.

—Data representation. How can the fundamental shape characteristics of a time-series be represented? What invariance properties should the representation satisfy? A representation technique should derive the notion of shape by reducing the dimensionality of data while retaining its essential characteristics.

—Similarity measurement. How can any pair of time-series be distinguished or matched? How can an intuitive distance between two series be formalized? This measure should establish a notion of similarity based on perceptual criteria, thus allowing the recognition of perceptually similar objects even though they are not mathematically identical. —Indexing method. How should a massive set of time-series be organized to enable fast querying? In other words, what indexing mechanism should be applied? The indexing technique should provide minimal space consumption and computational complexity.

These implementation components represent the core aspects of time-series data mining systems. However, these are not exhaustive as many tasks will require the use of more specific modules. Moreover, some of these are useless for some specific tasks. Forecasting (refer to Section 3.5) is the most blatant example of a topic that requires more advanced analysis processes as it is more closely related to statistical analysis. It may require the use of a time-series representation and a notion of similarity (mostly used to measure prediction accuracy) whereas model selection and statistical learning are also at the core of forecasting systems. The components that are common to most time-series mining tasks have therefore been analyzed and other components found in related tasks have been briefly discussed. The following part of this article has been organized as follows: first introducing the fundamental concepts of time-series data mining (Section 2); then presenting an overview of the tasks to which most of the research in this field has been devoted (Section 3); then reviewing the literature based on the three core components for implementation (Section 4) and finally reviewing the research trends for future work in this field (Section 5)

3. TASKS IN TIME-SERIES DATA MINING This section provides an overview of the tasks that have attracted wide research interest in time-series data mining. These tasks are usually just defined as theoretical objectives though concrete applications may call for simultaneous use of multiple tasks. 3.1. Query by Content Query by content is the most active area of research in time-series analysis. It is based on retrieving a set of solutions that are most similar to a query provided by the user. Figure 1 depicts a typical query by content task, represented on a two-dimensional search space. We can define it formally as follows.

In former times, time-series mining was almost exclusively devoted to this task (refer to seminal work by Agrawal et al. [1993]). In this article, the representation was based on a set of coefficients obtained from a Discrete Fourier Transform (DFT) to reduce the dimensionality of data. These coefficients were then indexed with an R*-tree [Beckmann et al. 1990]. False hits were removed in a postprocessing step, applying the Euclidean distance to complete time series. This paper laid the foundations of a reference framework that many subsequent works just enlarged by using properties of the DFT [Rafiei and Mendelzon 1998] or similar decompositions such as Discrete Wavelet Transform (DWT) [Chan and Fu 1999], that has been shown to have similar efficiency depending on the dataset at hand

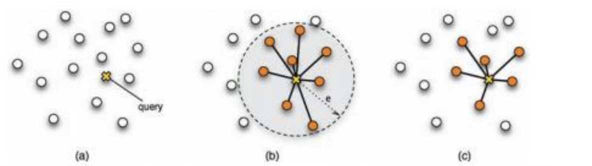

Fig. 1. Diagram of a typical query by content task represented in a two-dimensional search space. Each point in this space represents a series whose coordinates are associated with its features. (a) When a query is entered into the system, it is first transformed into the same representation as that used for other datapoints. Two types of query can then be computed. (b) A ϵ*−range query* will return the set of series that are within distance ϵ of the query. (c) A *K−Nearest Neighbors query* will return the *K* points closest to the query.

[Popivanov and Miller 2002]. The Discrete Cosine Transform (DCT) has also been suggested [Korn et al. 1997] but it appeared later that it did not have any advantage over other decompositions [Keogh et al. 2004]. Several numeric transformations—such as random projections [Indyk et al. 2000], Piecewise Linear Approximation (PLA) [Shatkay and Zdonik 1996], Piecewise Approximate Aggregation (PAA) [Keogh et al. 2001b; Yi and Faloutsos 2000], and Adaptive Piecewise Constant Approximation (APCA) [Keogh et al. 2001a]—have been used as representations. Symbolic representations have also been widely used. A shape alphabet with fixed resolution was originally proposed in Agrawal et al. [1995]. Other symbolic representations have been proposed, such as the bit-level approximation [Ratanamahatana et al. 2005] or the Symbolic Aggregate approXimation (SAX) [Lin et al. 2003]; the latter one has been shown to outperform most of the other representations [Stiefmeier et al. 2007]. We will find shortly a detailed overview of representations distance measures and indexing techniques.

II. CLUSTERING

Clustering is the process of finding natural groups, called clusters, in a dataset. The objective is to find the most homogeneous clusters that are as distinct as possible from other clusters. More formally, the grouping should maximize intercluster variance while minimizing intracluster variance. The algorithm should thus automatically locate which groups are intrinsically present in the data. Figure 2 depicts some possible outputs of a clustering algorithm. It can be seen in this figure that the main difficulty concerning any clustering problem (even out of the scope of time-series mining) usually

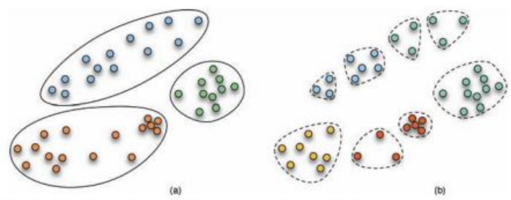

Fig. 2. Two possible outputs from the same clustering system obtained by changing the required number of clusters with (a) $N = 3$ and (b) $N = 8$. As we can see, the clustering task is a nontrivial problem that highly depends on the way parameters are initialized and the level of detail targeted. This parameter selection issue is common to every clustering task, even out of the scope of time-series mining.

lies in defining the correct number of clusters. The time-series clustering task can be divided into two subtasks.

Clustering can be applied to each complete time series in a set. The goal is thus to regroup entire time series into clusters so that the time series are as similar to each other as possible within each cluster.

There have been numerous approaches for whole series clustering. Typically, after defining an adequate distance function, it is possible to adapt any algorithm provided by the generic clustering topic. Clustering is traditionnally performed by using Self-Organizing Maps (SOM) [Chappelier and Grumbach 1996], Hidden Markov Models (HMM) [Smyth 1997], or Support Vector Machines (SVM) [Yoon et al. 2005]. Gaffney and Smyth [1999] proposed a variation of the Expectation Maximization (EM) algorithm. However, this model-based approach has usually some scalability problems and implicitly presupposes the existence of an underlying model which is not straightforward for every dataset. Using Markov Chain Monte Carlo (MCMC) methods, Frohwirth-Schnatter and Kaufmann [2008] make an estimation about the appropriate ¨ grouping of time series simultaneously along with the group-specific model parameters. A good survey of generic clustering algorithms from a data mining perspective is given in Berkhin [2006]. This review focuses on methods based on classical techniques that can further be applied to time series. A classification of clustering methods for various static data is proposed in Han and Kamber [2006] following five categories: partitioning, hierarchical, density based, grid based, and model based. For the specificities of time-series data, three of these five categories (partitioning, hierarchical, and model based) have been applied [Liao 2005]. Clustering of time series is especially useful for data streams; it has been implemented by using clipped data representations [Bagnall and Janacek 2005], Auto-Regressive (AR) models [Corduas and Piccolo 2008], k−means [Vlachos et al. 2003], and—with greater efficiency—k-center clustering [Cormode et al. 2007]. Interested readers may refer to Liao [2005] who provides a thorough survey of time-series clustering issues by discussing the advantages and limitations of existing works as well as avenues for research and applications.

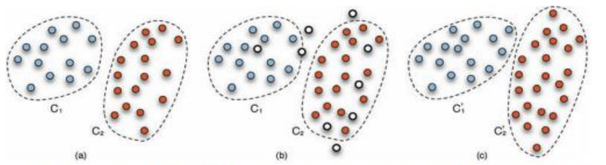

Fig. 3. The three main steps of a classification task. (a) A training set consisting of two prelabeled classes C_1 and C_2 is entered into the system. The algorithm will first try to learn what are the characteristic features distinguishing one class from another; they are represented here by the class boundaries. (b) An unlabeled dataset is entered into the system that will then try to automatically deduce to which class each datapoint belongs. (c) Each point in the set entered has been assigned to a class. The system can then optionally adapt the class boundaries.

In Hebrail and Hugueney [2000], the series are sliced into nonoverlapping windows. Their width is chosen by investigating the periodical structure of the time series by means of a DFT analysis. This approach is limited by the fact that, when no strong periodical structure is present in the series, nonoverlapping slicing may miss important structures. A straightforward way to extend this approach can therefore be to extract shorter overlapping subsequences and then cluster the resulting set. However, this overlapping approach has been shown to produce meaningless results [Keogh et al. 2003b].

III. CLASSIFICATION

The classification task seeks to assign labels to each series of a set. The main difference when compared to the clustering task is that classes are known in advance and the algorithm is trained on an example dataset. The goal is first to learn what are the distinctive features distinguishing classes from each other. Then, when an unlabeled dataset is entered into the system, it can automatically determine to which class each series belongs. Figure 3 depicts the main steps of a classification task.

There are two types of classification. The first one is the time-series classification similar to whole series clustering. Given sets of time series with a label for each set, the task consists in training a classifier and labeling new time series. An early approach to time-series classification was presented in Bakshi and Stephanopoulos [1994]. However, it is based on simple trends whose results are therefore hard to interpret. A piecewise representation was later proposed in Keogh and Pazzani [1998]; it is robust to noise and weighting can be applied in a relevance feedback framework. The same representation was used in Geurts [2001]; it is apparently not too robust to outliers. To overcome the obstacle of high dimensionality, Jeng and Huang [2008] used singular value decomposition to select essential frequencies. However, it implies higher computational costs. In a recent study, Rodriguez and Kuncheva [2007] compared three types of classifiers: nearest neighbor, support vector machines, and decision forests. All three methods seem to be valid, though highly depending on the dataset at hand. 1-NN classi- fication algorithm with DTW seems to be the most widely used classifier; it was shown highly accurate [Xi et al. 2006], though computing speed is significantly affected by repeated DTW computations. To overcome this limitation Srisai and Ratanamahatana [2009] proposed a template construction algorithm based on the Accurate Shape Averaging (ASA) technique. Each training class is represented by only one sequence so that any incoming series is compared only with one averaged template per class. Several other techniques have been introduced, such as ARMA models [Deng et al. 1997] or HMM [Zhong and Ghosh 2002]. In the context of clinical studies, Lin et al. [2008] enhanced HMM approaches by using discriminative HMMs in order to maximize interclass differences. Using the probabilistic transitions between fewer states results in the patients being aligned to the model and can account for varying rates of progress. This approach has been applied in Lowitz et al. [2009], in order to detect postmyocardial infarct patients. Several machine learning techniques have also been introduced such as neural networks [Nanopoulos et al. 2001] or Bayesian classification [Povinelli et al. 2004]

IV. SEGMENTATION

The segmentation (or summarization) task aims at creating an accurate approximation of time series, by reducing its dimensionality while retaining its essential features. Figure 4 shows the output of a segmentation system. Section 4.2 will show that most time-series representations try to solve this problem implicitly.

The objective of this task is thus to minimize the reconstruction error between a reduced representation and the original time series. The main approach that has been undertaken over the years seems to be Piecewise Linear Approximation (PLA) [Shatkay and Zdonik 1996]. The main idea behind PLA is to split the series into most representative segments, and then fit a polynomial model for each segment. A good review on the most common segmentation methods in the context of PLA representation can be found in Keogh et al. [2003a]. Three basic approaches are distinguished. In sliding windows, a segment is grown until it exceeds some error threshold [Shatkay and Zdonik 1996]. This approach has shown poor performance with many real-life datasets [Keogh et al. 2003a]. The top-down approach consists in recursively partitioning a time series until some stopping criterion is met [Li et al. 1998]. This approach has time complexity O(n2) [Park et al. 1999] and is qualitatively outperformed by bottom-up. In this approach, starting from the finest approximation, segments are iteratively merged [Keogh and Pazzani 1998]. Himberg et al. [2001a] present fast greedy algorithms to improve previous approaches and a statistical method for choosing the number of segments is described in Vasko and Toivonen [2002].

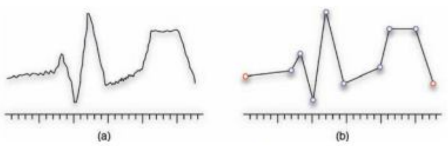

Fig. 4. Example of application of a segmentation system. From (a) usually noisy time series containing a very large number of datapoints, the goal is to find (b) the closest approximation of the input time series with the maximal dimensionality reduction factor without losing any of its essential features.

Several other methods have been introduced to handle this task. Palpanas et al. [2008] introduced a representation of time series that implicitly handles the segmentation of time series. They proposed user-specified amnesic functions reducing the confi- dence to older data in order to make room for newer data. In the context of segmenting hydrological time series.

V. PREDICTION

Time series are usually very long and considered smooth, that is, subsequent values are within predictable ranges of one another [Shasha and Zhu 2004]. The task of prediction is aimed at explicitly modeling such variable dependencies to forecast the next few values of a series. Figure 5 depicts various forecasting scenarios.

Prediction is a major area in several fields of research. Concerning time series, it is one of the most extensively applied tasks. Literature about this is so abundant that dozens of reviews can focus on only a specific field of application or family of learning methods. Even if it can use time-series representations and a notion of similarity evaluate accuracy, it also relies on several statistical components that are out of the scope of this article, for example, model selection and statistical learning. This task will be mentioned because of its importance but the interested reader willing to have further information may consult several references on forecasting.

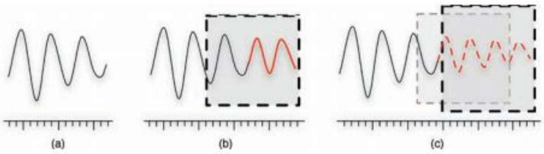

Fig. 5. A typical example of the time-series prediction task. (a) The input time series may exhibit a periodical and thus predictable structure. (b) The goal is to forecast a maximum number of upcoming datapoints within a prediction window. (c) The task becomes really hard when it comes to having *recursive prediction*, that is, the long-term prediction of a time series implies reusing the earlier forecast values as inputs in order to go on predicting.

Recent improvements for time-series forecasting have been proposed; Pesaran et al. [2006] proposed a Bayesian prediction for time series subject to discrete breaks, handling the size and duration of possible breaks by means of a hierarchical HMM. A dynamic Genetic Programming (GP) model tailored for forecasting streams was proposed in Wagner et al. [2007] by adapting incrementally based on retained knowledge. The prediction task seems one of the most commonly applied in real-life applications, considering that market behavior forecasting relies on a wealth of financial data. Bai and Ng [2008] proposed to refine the method of factor forecasting by introducing "targeted predictors" selected by using a hysteresis (hard and soft thresholding) mechanism. The prediction task has also a wide scope of applications ranging from tourism demand forecasting [Song and Li 2008] to medical surveillance.

VI. IMPLEMENTATION COMPONENTS

In this section, we review the implementation components common to most time-series mining tasks. As said earlier, the three key aspects when managing time-series data are representation methods, similarity measures, and indexing techniques. Because of the high dimensionality of time series, it is crucial to design low-dimensional representations that preserve the fundamental characteristics of a series. Given this representation scheme, the distance between time series needs to be carefully defined in order to exhibit perceptually relevant aspects of the underlying similarity. Finally the indexing scheme must allow to efficiently manage and query ever growing massive datasets.

2.1. Preprocessing

In real-life scenarios, time series usually come from live observations [Reeves et al. 2009] or sensors [Stiefmeier et al. 2007] which are particularly subject to noise and outliers. These problems are usually handled by preprocessing the data. Noise filtering can be handled by using traditional signal processing techniques like digital filters or wavelet thresholding. In Himberg et al. [2001b], Independent Component Analysis (ICA) is used to extract the main mode of the series. As will be explained in Section 4.2, several representations implicitly handle noise as part of the transformation.

The second issue concerns the scaling differences between time series. This problem can be overcome by a linear transformation of the amplitudes [Goldin and Kanellakis 1995]. Normalizing to a fixed range [Agrawal et al. 1995] or first subtracting the mean (known as zero mean/unit variance [Keogh et al. 2001a]) may be applied to both time series, however, it does not give the optimal match of two series under linear transformations [Argyros and Ermopoulos 2003]. In Goldin et al. [2004] the transformation is sought with optional bounds on the amount of scaling and shifting. However, normalization should be handled with care. As noted by Vlachos et al. [2002], normalizing an essentially flat but noisy series to unit variance will completely modify its nature and normalizing sufficiently small subsequences can provoke all series to look the same.

2.2. Representation

As mentioned earlier, time series are essentially high-dimensional data. Defining algorithms that work directly on the raw time series would therefore be computationally too expensive. The main motivation of representations is thus to emphasize the essential characteristics of the data in a concise way. Additional benefits gained are efficient storage, speedup of processing, as well as implicit noise removal. These basic properties lead to the following requirements for any representation:

—significant reduction of the data dimensionality;
—emphasis on fundamental shape characteristics on both local and global scales;
—low computational cost for computing the representation;
—good reconstruction quality from the reduced representation; —insensitivity to noise or implicit noise handling.

Many representation techniques have been investigated, each of them offering different trade-offs between the properties listed before. It is, however, possible to classify these approaches according to the kind of transformations applied. In order to perform such classification, we follow the taxonomy of Keogh et al. [2004] by dividing representations into three categories, namely non data adaptive, data adaptive, and model based.

2.1.1 Nondata Adaptive:

In nondata-adaptive representations, the parameters of the transformation remain the same for every time series regardless of its nature. The first nondata-adaptive representations were drawn from spectral decompositions. The DFT was used in the seminal work of Agrawal et al. [1993]. It projects the time series on a sine and cosine functions basis [Faloutsos et al. 1994] in the real domain. The resulting representation is a set of sinusoidal coefficients. Instead of using a fixed set of basis functions, the DWT uses scaled and shifted versions of a mother wavelet function [Chan and Fu 1999]. This gives a multiresolution decomposition where low frequencies are measured over larger intervals thus providing better accuracy [Popivanov and Miller 2002]. A large number of wavelet functions have been used in the literature like Haar [Chan et al. 2003], Daubechies [Popivanov and Miller 2002], or Coiflets [Shasha and Zhu 2004]. The Discrete Cosine Transform (DCT) uses only a cosine basis; it has also been applied to time-series mining [Korn et al. 1997]. However, it has been shown that it does not offer any advantage over previously cited decompositions [Keogh et al. 2004]. Finally, an approximation by Chebychev polynomials [Cai and Ng 2004] has also been proposed but the results obtained have later been withdrawn due to an error in implementation.

2.2.2 Data Adaptive:

This approach implies that the parameters of a transformation are modified depending on the data available. By adding a data-sensitive selection step, almost all nondata-adaptive methods can become data adaptive. For spectral decompositions, it usually consists in selecting a subset of the coefficients. This approach has been applied to DFT [Vlachos et al. 2004] and DWT [Struzik et al. 1999]. A data-adaptive version of PAA has been proposed in Megalooikonomou et al. [2004], with vector quantization being used to create a codebook of recurrent subsequences. This idea has been adapted to allow for multiple resolution levels [Megalooikonomou et al. 2005]. However, this approach has only been tested on smaller datasets. A similar approach has been undertaken in Stiefmeier et al. [2007] with a codebook based on motion vectors being created to spot gestures. However, it has been shown to be computationally less efficient than SAX.

2.2.3. Model Based:

The model-based approach is based on the assumption that the time series observed has been produced by an underlying model. The goal is thus to find parameters of such a model as a representation. Two time series are therefore considered similar if they have been produced by the same set of parameters driving the underlying model. Several parametric temporal models may be considered, including statistical modeling by feature extraction [Nanopoulos et al. 2001], ARMA models [Kalpakis et al. 2001], Markov Chains (MCs) [Sebastiani et al. 1999], or HMM [Panuccio et al. 2002]. MCs are obviously simpler than HMM so they fit well shorter series but their expressive power is far more limited. The time-series bitmaps introduced in Kumar et al. [2005] can also be considered as a model-based representation for time series, even if it mainly aims at providing a visualization of time series.

VII. CONCLUSION

The time-series data mining, an incredible wealth of systems and algorithms has been proposed. The ubiquitous nature of time series led to an extension of the scope of applications simultaneously with the development of more mature and efficient solutions to deal with problems of increasing computational complexity. Time-series data mining techniques are currently applied to an incredible diversity of fields ranging from economy, medical surveillance, climate forecasting to biology, hydrology, genetics, or musical querying. Numerous facets of complexity emerge with the analysis of time series, due to the high dimensionality of such data, in combination with the difficulty to define an adequate similarity measure based on human perception

As for most scientific research, trying to find the solution to a problem often leads to raising more questions than finding answers. We have thus outlined several trends and research directions as well as open issues for the near future. The topic of time-series data mining still raises a set of open questions and the interest of such research sometimes lies more in the open questions than the answers that could be provided.

REFERENCES

[1] ABONYI, J., FELL, B., NEMETH, S., AND ARVA, P. 2003. Fuzzy clustering based segmentation of time-series. In Proceedings of the 5th International Symposium on Intelligent Data Analysis (IDA 03). Springer, 275–285.

[2] AGRAWAL, R., FALOUTSOS, C., AND SWAMI, A. 1993. Efficient similarity search in sequence databases.In Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms. Springer, 69–84.

[3] AHMED, N., ATIYA, A., EL GAYAR, N., EL-SHISHINY, H., AND GIZA, E. 2009. An empirical comparison of machine learning models for time series forecasting. Econometr. Rev. 29, 5, 594–621.

[4] AN, J., CHEN, H., FURUSE, K., OHBO, N., AND KEOGH, E. 2003. Grid-Based indexing for large time series databases.In Intelligent Data Engineering and Automated Learning. Lecture Notes in Computer Science, vol. 1983. Springer 614–621.

[5] ANTUNES, C. AND OLIVEIRA, A. 2001. Temporal data mining: An overview. In Proceedings of the KDD Workshop on Temporal Data Mining.1–13.ARGYROS, T. AND ERMOPOULOS, C. 2003.

[6] Efficient subsequence matching in time series databases under time and amplitude transformations. In Proceedings of the 3rd IEEE International Conference on Data Mining.481–484.

[7] ASSENT, I., KRIEGER, R., AFSCHARI, F., AND SEIDL, T. 2008. The TS-tree: Efficient time series search and retrieval. In Proceedings of the 11th International Conference on Extending Database Technology.25–29.

[8] ASSENT, I., WICHTERICH, M., KRIEGER, R., KREMER, H., AND SEIDL, T. 2009. Anticipatory DTW for efficient similarity search in time series databases. Proc. VLDB Endowm. 2, 1, 826–837.

[9] AßFALG, J., KRIEGEL, H., KROGER, P., KUNATH, P., PRYAKHIN, A., AND RENZ, M. 2006. Similarity search on time series based on threshold queries.In Proceedings of the 10th International Conference on Extending Database Technology. 276.

[10] BAGNALL, A. AND JANACEK, G. 2005.Clustering time series with clipped data. Mach. Learn. 58, 2, 151–178.

[11] BAGNALL, A., JANACEK, G., DE LA IGLESIA, B., AND ZHANG, M. 2003.Clustering time series from mixture polynomial models with discretised data.In Proceedings of the 2nd Australasian Data Mining Workshop.105–120.

[12] Bruce L. Bowerman, Richard T. O' Connell, & Anne B. Koehler, "Forecasting, time series, and regression: an applied approach," 4th ed. The United States of America: Thomson Brooks, 2005.

[13] Makridakis, S., S.C. Wheelwright and R.J. Hyndman, "Forecasting: Methods and Applications," 3 ed. Wiley, Inc., New York, 1998.

[14] Box, G.E.P. and G. Jenkins, "Times Series Analysis Forecasting and Control," Holden-Day, San Francisco, CA, 1976.

[15] [Online]. Available: http://www.rstudio.com/

[16] [Online]. Available: http://cran.us.r-project.org/

[17] Jonathan D. Cryer& Kung-Sik Chan, “Time series analysis: with applications in R,” 2nd ed. New York: Springer, 2008.
[18] Robert H. Shumway & David S. Stoffer, “Time Series Analysis and Its Applications with R Examples,” 3rd ed. New York: Springer, 2011.
[19] Oleg Nenadic, Walter Zucchini, “Statistical Analysis with R - a quick start -,” Retrieved November 10, 2012, from http://www.scribd.com/doc/114263559/Statistical-Analysis-With-RA-Quick-Start.
[20] Samer Saab, ElieBadr and George Nasr, “Univariate modeling and forecasting of energy consumption: the case of electricity in Lebanon,” Energy, vol.26, 2001, pp. 1-14.