Time series reconstruction analysis

Vladimir Kurbalija, Brankica Bratić
Faculty of Sciences
University of Novi Sad
Novi Sad, Serbia
Email: {kurba, brankica.bratic}@dmi.uns.ac.rs

***Abstract*—The dimensionality of time series data is usually very large, so it must often be reduced before applying certain data mining tasks upon it. Dimensionality reduction is achieved by creating appropriate time series representation that is actually new time series of lower dimensionality obtained from the original one by preserving only the important features. I addition to that, the reconstruction of original time series from its representation is inevitable task in many practical applications. The main objective of this paper is the comparison of different time series representations in the term of their reconstruction accuracies on a number of freely available data sets. Reconstructed time series are compared with the original ones, using several state-of-the-art similarity measures, in order to measure the quantity of information loss. Additionally, we measured the correlations between several data set properties and their reconstruction errors, which will give a deeper insight in the problem of choosing of appropriate representation technique for a particular data set.**

***Keywords*—time series;time series representations;time series reconstructions;similarity measures**

I. Introduction

Main data mining algorithms such as classification, clustering and indexing, often become useless when applied upon large datasets with high dimensional data. This issue applies particularly to probably most interesting task in time-series mining: forecasting [1], [2], [3], [4], [5]. The first problem is execution time. Many algorithms process data by putting it in main memory. If dataset does not fit main memory, swapping is the only option. As a consequence most of the execution time is spent by moving data from disk into a main memory. Second problem with high dimensional data is well known curse of dimensionality [6], [7], [8]. Various data mining algorithms showed poor results while operating upon high dimensional data. The problem arises because data in high dimensional spaces is very sparse. This sparsity is the consequence of the fact that as data dimensionality increases, volume of the space also rapidly increases.

Described problems in the domain of time series mining are usually solved with time series representations [9], [10], [11], [12], [13], [14]. Main purpose of time series representations is to decrease data dimensionality while keeping the important characteristics of the original time series. If Q is time series of dimensionality n, and Q' is its representation of dimensionality k, than k must be much lower than n ($k \ll n$). This process, of course, includes some information loss, so the main objective is to keep crucial information. By keeping important information, representation preserves time series features, behaving just like original time series in data mining algorithms. As a consequence of reduced dimensionality many real-world forecasting problems produce more efficient solutions [15], [16], and in some cases more accurate results [17].

If the original data is dismissed after creation of suitable representation, than there might be a need to reconstruct original data from time series representation. There are numerous reasons to do that, for example: if there is an interest in certain value(s) recorded at certain point(s) in time; if there is a need for visual presentation of certain time series; if there is a need for comparison of two time series that are both compressed with different representations; if there exists a need for a time series in original space which is particularly important in similarity based time series forecasting [18]; etc. Since reconstructions of original data are sometimes inevitable, it would be helpful to know which representation behaves the best when it comes to that. That is actually the main intention of this paper. We want to examine to what degree are different representations suitable for data reconstruction.

In section II we will give overview of the similarity measures and time series representations. Section III describes experimental settings that we relied on during this analysis. Section IV analyzes results of experiments that examined reconstruction accuracy of time series representation. Section V includes analysis of similarity measures that are applied on reconstructions from various time series representations. Section VI concludes the analysis and gives directions for future research.

II. Background and Related Work

Comparisons between different types of time series representations are presented both in introductory papers [9], [10], [11], [12], [13], [14] and in a few review papers [19]. However, those papers present pairwise comparisons between very limited number of time series representations, omitting global picture. On the other hand, authors of paper [20] made comprehensive comparisons of many time series representations showing that all representations are similar in terms of tightness of lower bound. But, as far as we know, analysis of time series representations in terms of information loss during the reconstruction process is never fully conducted. So, we wanted to expand former analysis by applying several state-of-the-art similarity measures upon reconstructed data, and also to give overview of correlations between data set properties and results of mentioned analysis of different representations. So the goal is not just to evaluate suitability of certain representations in terms of information loss during reconstruction process, but is also to explore in what conditions certain representation gives good results. We also wanted to detect similarity measures that adapt well on reconstructed data. Similarity measure adapts well if time series' that it declared similar in original data set, stay similar in reconstructed dataset also.

In further text we will give brief overview of all similarity measures and representations that will be examined in this paper.

A. Similarity Measures

L_p-norms is one of the most popular classes of similarity measures that are used in data mining algorithms. Its definition is given in formula (1) where T and S are time series of length N, t_i is the value of T at time point i, and s_i is value of S at time point i. Variable p denotes the norm that is used. For example, $p = 1$ corresponds to the Manhattan distance, $p = 2$ is Euclidean distance, and $p = \infty$ is Chebyshev distance.

$$L_p(T, S) = \Big(\sum_{i}^{N} |t_i - s_i|^p \Big)^{\frac{1}{p}} \tag{1}$$

One of the main disadvantages of Lp norms is sensitivity to scaling and shifting along the time axis. If two time series are very similar, except that one is shifted along time axis, then Lp norms will not detect the similarity between these two time series. That is why elastic distance measures are introduced. All measures that are presented in further text are elastic distance measures.

Dynamic Time Warping (DTW) [21], [22] is similarity measure that allows non-linear alignment of two time series. That means that DTW introduces comparisons between points that are not identically positioned on time axis. Lets denote two time series with symbols T and S, and suppose that length of time series T is N, while the length of S is M. DTW algorithm starts with creation of $N \times M$ matrix, whose element on position (i, j) corresponds to the $d(t_i, s_j)$, where d is some distance measure, t_i is i$^{\text{th}}$ point of T, and s_j is j$^{\text{th}}$ point of S. The idea is to find a path through that matrix with minimal sum of the contained elements. That path is called warping path, and the sum of the elements that this path contains is actually DTW value.

Another elastic similarity measure is **Longest Common Subsequence (LCS)** [23]. This similarity measure is one of the edit distances. The similarity of two time series is presented as the length of their longest common subsequence.

Edit Distance on Real Sequence (EDR) [24] is another edit distance that found its application in time series data mining. EDR value is actually minimal number of edit operations (insertions, substitutions and deletions) that need to be applied upon one time series in order to transform it to another time series. This similarity measure proved to be very robust to noise.

Main problem with all mentioned elastic measures is that they do not satisfy the triangle inequality, so they cannot be used in indexing algorithms. In order to overcome this problem, **Edit distance with Real Penalty (ERP)** [25] was introduced. This similarity measure combines L_1-norm with DTW and EDR, dismissing segments of those measures that are causing them not to satisfy triangle inequality.

B. Time Series Representations

One of the most important time series representations is **Discrete Fourier Transform (DFT)**. DFT is actually used in many fields (such as digital signal processing, image processing, solving of partial differential equations, etc.), and has also found its application in time series data mining. Basic idea of DFT is that any signal can be represented by the super position of finite number of sine/cosine waves. Each such wave is called Fourier coefficient, and is presented as a complex number. After applying DFT, dimensionality of time series stays unchanged, so to reduce data dimensionality, we need to dismiss some Fourier coefficients. It is observed that only the first few coefficients appear to be dominant and therefore the rest can be omitted without great information loss [9]. In that way data series dimensionality can be efficiently decreased.

Discrete Wavelet Transform (DWT) [10] is very similar to DFT. Wavelets are functions that represent time series in terms of the sum and difference of prototype function, called mother wavelet. Unlike DFT, in which case each coefficient carries out only global information, DWT coefficients hold local information. To be more precise, each wavelet holds information of some time series segment. First coefficients are more global, while the each next group of coefficients refines global picture by adding more information to certain segments of time series. That is why they are very suitable for decreasing time series dimensionality. By taking first few DWT coefficients, we are actually taking global time series characteristics, and that is exactly what time series representation should hold.

Completely different dimensionality reduction method is **Piecewise Aggregate Approximation (PAA)** [11]. Underlying idea is very simple. Original time series is divided in N equal segments. For each segment mean value is calculated and stored. In that way dimensionality of original time series is decreased to N. Although very simple, this method proved to be reasonably good when used in data mining algorithms.

Similar to PAA is **Piecewise Linear Approximation (PLA)** [12]. Just like in PAA, time series is divided in N segments (not necessarily of equal size). Each segment is then presented by a line. There are two ways of defining that line: it can be linear interpolation between start and end point of corresponding segment, or it can be linear regression that takes into account all points that are contained in the segment. Each segment is then presented by two numbers (a and b coefficients of line equation $y = ax + b$), so if dimensionality of final representation needs to be N, time series is divided into $N/2$ segments. As noted at the beginning, time series segments by definition do not have to be of equal size. Segments lengths are determined by different kind of algorithms whose main aim is to create representation of minimal reconstruction error. The problem with that approach is that distance measure upon this kind of representation does not support lower bounding, which is necessary in order to use representation in indexing algorithms. That is why **Indexable Piecewise Linear Approximation (IPLA)** was introduced [13]. IPLA modifies PLA in two aspects. According to first modification, IPLA does not allow segments of different lengths. Second modification introduces reset of x (time) component of each time series segment.

Symbolic Aggregate Approximation (SAX) [14] is representation that is built upon PAA. Idea of SAX is to convert data into a discrete format which is based on predefined alphabet. Before discretization, time series is transformed via PAA algorithm. In order to convert PAA coefficients to alphabet symbols, we must first define breakpoints that divide the distribution space into S equiprobable regions, where S is the size of alphabet. Each defined region is mapped into one alphabet symbol. Transformation of PAA representation is now trivial each coefficient is mapped and transformed into one alphabet symbol.

III. Analysis Setting

This analysis was conducted upon 85 data sets from [26]. That collection contains data sets of various structures. Data sets sizes vary from 40 to 16637, while the time series' dimensionalities are between 24 and 2709. Diversity between data sets is very important in this kind of analysis, since behavior of data mining algorithms is very influenced by structure of the data. So, in order to get global picture, analysis must be conducted on various data set types.

Framework upon which we built this research was FAP [27]. FAP contains implementations of all mentioned similarity measures [28], [29] and time series representations. As such, it was suitable choice for this analysis.

First step of this analysis was to create time series representations. For each time series of each data set, we created its DFT, DWT, PAA, IPLA and SAX representations. Each of these representations was created in four different dimensions: 4, 6, 8 and 10. Additionally, in creation of DWT representation we used Haar function as a mother wavelet, and for SAX we used alphabet of 256 symbols. Next step was to create time series reconstructions upon all created representations, which is inverse process from the previous one. Dimensionality of reconstructed time series must be the same as it was before creation of corresponding representation. After this step we got reconstructed data sets for each representation type and for each representation length. These reconstructed data sets present the basis of further analysis.

IV. Reconstruction Accuracy

One of the goals of this analysis is to calculate the amount of information loss that appeared as a consequence of dimensionality reduction and time series reconstruction processes. We want to see how much reconstructed time series' are different from original ones. In order to determine that, we calculated reconstruction accuracy as a root-mean-square-deviation (RMSD). If T is original time series, and T' is it's reconstruction, than RMSD value is calculated by using equation 2. As it can be seen, it is actually Euclidean distance divided by the square root of time series dimensionality. Besides that, we also applied other famous similarity measures upon each time series and its reconstruction, which gave us additional information about similarity between the two.

$$\mathrm{RMSD}(T, T') = \sqrt{\frac{\sum_{i=1}^{N}(t'_i - t_i)^2}{N}} \tag{2}$$

Averaged RMSD values can be seen in Figure 1. Results showed that all representations have quite similar RMSD values. Talking about representation dimensionality, it can be seen that increase of dimensionality implies better RMSD values. However, there is one exception to that in DFT representation in which dimensionality 4 gives better results than dimensionality 6. Although RMSD values are in general very similar to each other, there is still one thing to note. With lower representation dimensionalities, DFT produced better results, while the other representations overperforms DFT with greater dimensionalities.

As it is mentioned, beside overviewing RMSD values, we also calculated average distances between original time series' and their reconstructions by using famous similarity measures. Results are shown in Figure 2, and also in Table I. The most evident conclusion from these results is that DFT representation produces reconstructions that are far from their original time series' according to all similarity measures. All the other representations are very similar to each other. Interesting fact is also that LCS and EDR similarity measures don't give extremely high distance values for DFT, as the other similarity measures does.

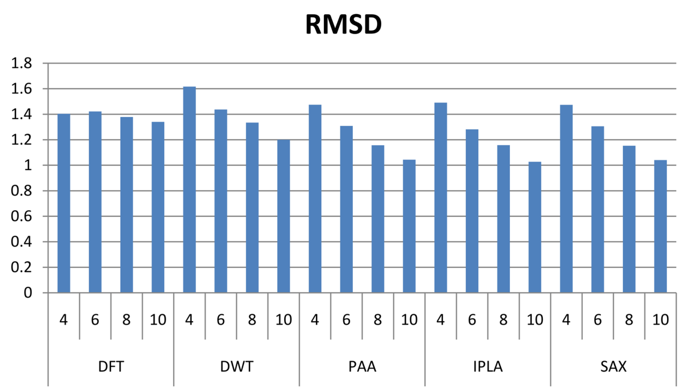


Fig. 1: Averaged root-mean-square-deviation of each time series and its reconstruction. Results are calculated for each data set, and averaged afterwards.

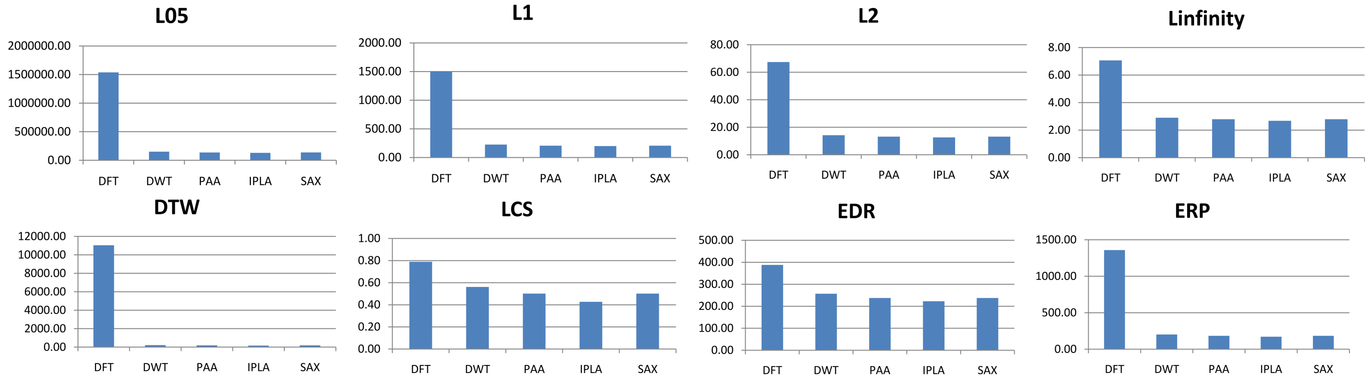


Fig. 2: Averaged distances from each time series to its reconstruction. Results are calculated for each data set, and averaged afterwards.

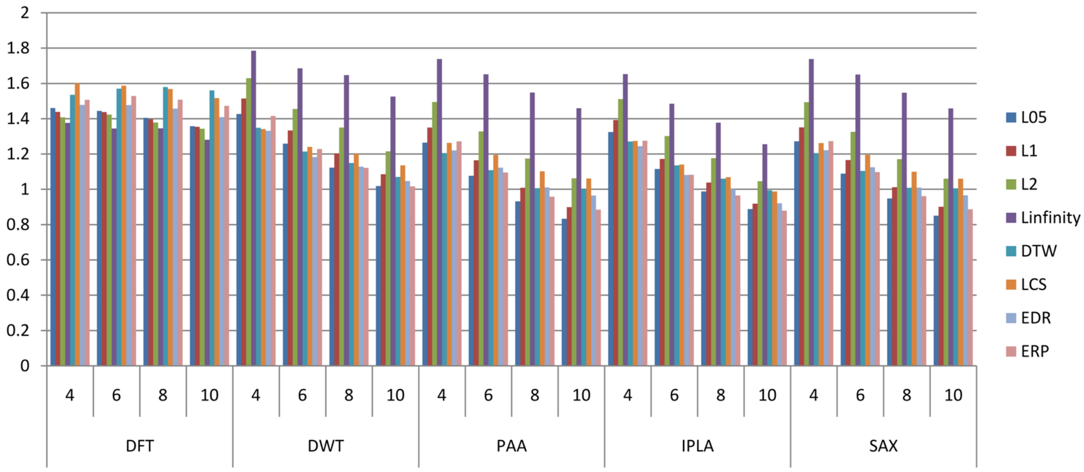


Fig. 3: Figure of averaged SMRE values.

		$L_{0.5}$	L_1	L_2	L_∞	DTW	LCS	EDR	ERP
DFT	4	2021037.33	1817.46	77.43	7.32	15690.94	0.83	393.74	1675.87
	6	1608721.02	1577.61	70.54	7.29	12651.04	0.80	388.97	1434.95
	8	1284678.76	1359.90	63.30	6.93	8769.07	0.77	385.83	1214.06
	10	1236883.27	1257.20	58.52	6.73	7065.21	0.75	383.05	1109.87
DWT	4	180200.93	267.43	16.37	3.08	280.48	0.67	294.34	242.98
	6	154848.16	229.74	14.41	2.92	219.58	0.57	261.74	205.46
	8	143954.05	212.44	13.51	2.85	194.20	0.53	245.82	187.93
	10	124395.98	192.42	12.58	2.76	172.56	0.48	225.37	171.10
PAA	4	169196.53	250.71	15.56	3.06	253.85	0.60	275.79	225.25
	6	146646.66	215.70	13.69	2.85	196.98	0.52	246.31	189.96
	8	123190.80	188.24	12.24	2.68	164.76	0.46	224.55	167.87
	10	109266.65	169.51	11.29	2.57	145.79	0.42	202.90	151.07
IPLA	4	166580.95	251.05	15.36	2.94	231.81	0.56	276.37	216.87
	6	137188.55	208.71	13.19	2.76	168.19	0.45	236.44	178.02
	8	120124.42	178.89	11.42	2.56	140.19	0.36	201.25	153.79
	10	99341.55	159.50	10.56	2.45	122.71	0.33	176.53	137.98
SAX	4	169465.58	250.76	15.56	3.06	253.85	0.60	275.78	225.30
	6	146957.59	215.76	13.69	2.85	196.98	0.52	246.30	190.02
	8	123528.24	188.31	12.24	2.68	164.76	0.46	224.59	167.95
	10	109601.47	169.59	11.29	2.57	145.79	0.42	202.91	151.15

TABLE I: Averaged distances between original and reconstructed time series'

V. Similarity Measures on Reconstructed Time Series'

Besides calculating reconstruction accuracy, we also wanted to see how does information loss affect effectiveness of famous similarity measures, or in the other words, we wanted to see how similarity measures adapt on time series reconstructions. Measurement of this includes selection of 1000 random time series pairs from each data set, which we will denote as $(T_1, S_1), ..., (T_n, S_n)$, where $n = 1000$ corresponds to random sample size. For each pair (T_i, S_i) there is one corresponding pair $(T_{R,i}, S_{R,i})$ per each reconstructed data set R (there is one reconstructed data set for each analysis setting). $T_{R,i}$ is point from data set R that originates from time series T_i, while the $S_{R,i}$ originates from S_i. For all similarity measures we firstly calculated distances between T_i and S_i, and then also between $(T_{R,i}, S_{R,i})$ pairs for $i \in [1, 1000]$, and for all reconstructed data sets. Those distance values are normalized, so that we can compare results of different distance measures. Lets now denote distance between time series' of pair (T_i, S_i) with d_i, and distance between its corresponding pair from reconstructed data set with $d_{R,i}$. We are calculating effectiveness of similarity measure by using root-mean-square-deviation formula. Equation 3 shows how this formula is adapted for described purpose. We will call that value SMRE which stands for Similarity Measure Reconstruction Error.

$$\mathrm{SMRE}(R) = \sqrt{\frac{\sum_{i=1}^{\mathrm{n}} (d_i - d_{R,i})^2}{n}} \tag{3}$$

First part of this analysis is to give global overview of the results. For that purpose we averaged SMRE values of each separate data set, and the final values for each input parameter are given in Table II and shown in Figure 3. First thing to note is that PAA and SAX representations give almost identical results. Reason for that is that reconstructions from those two representations are almost identical, so it seams that SAX alphabet was large enough not to include information loss during the PAA values discretization step. Next interesting phenomenon is that except for the DFT, L_∞ adapts quite badly. Its SMRE values are very large comparing to other similarity measures, and that can be seen on Figure 4.

		$L_{0.5}$	L_1	L_2	L_∞	DTW	LCS	EDR	ERP
DFT	4	1.46	1.44	1.41	1.38	1.54	1.60	1.48	1.51
	6	1.44	1.44	1.42	1.34	1.57	1.59	1.48	1.53
	8	1.40	1.40	1.38	1.35	1.58	1.57	1.46	1.51
	10	1.36	1.35	1.34	1.28	1.56	1.52	1.41	1.47
DWT	4	1.43	1.51	1.63	1.79	1.35	1.34	1.33	1.42
	6	1.26	1.33	1.45	1.69	1.21	1.24	1.18	1.23
	8	1.12	1.20	1.35	1.65	1.15	1.20	1.13	1.12
	10	1.02	1.09	1.21	1.53	1.07	1.14	1.05	1.02
PAA	4	1.26	1.35	1.49	1.74	1.20	1.26	1.22	1.27
	6	1.08	1.16	1.33	1.65	1.11	1.20	1.12	1.10
	8	0.93	1.01	1.17	1.55	1.01	1.10	1.01	0.96
	10	0.83	0.90	1.06	1.46	1.00	1.06	0.96	0.89
IPLA	4	1.33	1.39	1.51	1.65	1.27	1.27	1.24	1.27
	6	1.12	1.17	1.30	1.48	1.14	1.14	1.08	1.08
	8	0.99	1.04	1.18	1.38	1.06	1.07	1.00	0.97
	10	0.89	0.92	1.05	1.26	1.00	0.99	0.92	0.88
SAX	4	1.27	1.35	1.49	1.74	1.21	1.26	1.22	1.27
	6	1.09	1.17	1.33	1.65	1.10	1.19	1.12	1.10
	8	0.95	1.01	1.17	1.55	1.01	1.10	1.01	0.96
	10	0.85	0.90	1.06	1.46	1.01	1.06	0.97	0.89

TABLE II: Averaged SMRE values

From the general point of view, representations on which similarity measures adapt the best are PAA, IPLA and SAX, while the DWT and especially DFT gives the worst results, which all can be seen on Figure 4. Talking about similarity measures, interesting fact is that in most cases L_p-norms give better results for smaller value of p. Elastic similarity measures give very similar results between each other. Slightly better then the others is EDR, and slightly worse then the others is LCS. But, since difference in results is not evident enough, we cannot reliably compare elastic measures between each other. What we can say is that in general they give better results than L_p-norms for $p > 1$.

What is true for averaged results, doesn't seem to be valid for each separate representation. It is said that L_p-norms give worse results for larger p value. That statement is valid for all representations except for DFT (see Figure 5a). In case of DFT the opposite is true - L_p-norms with larger p adapt better. DFT produces different results for elastic measures, too. Elastic measures do not adapt well on reconstructions from DFT representation. So, in case of DFT reconstructions, L_∞ preserves its original behavior the most.

In comparison of results of different representation dimensionalities we got expected results (see Figure 5b). As representation dimensionality increases, results get better since more information about original time series' is preserved. Still, there is one more thing to note. DFT representation does not have significant fall of SMRE value as dimensionality increases. Because of that, it might be a waste of resources to increase representation dimensionality in case of DFT, since results in that case would be just slightly better.

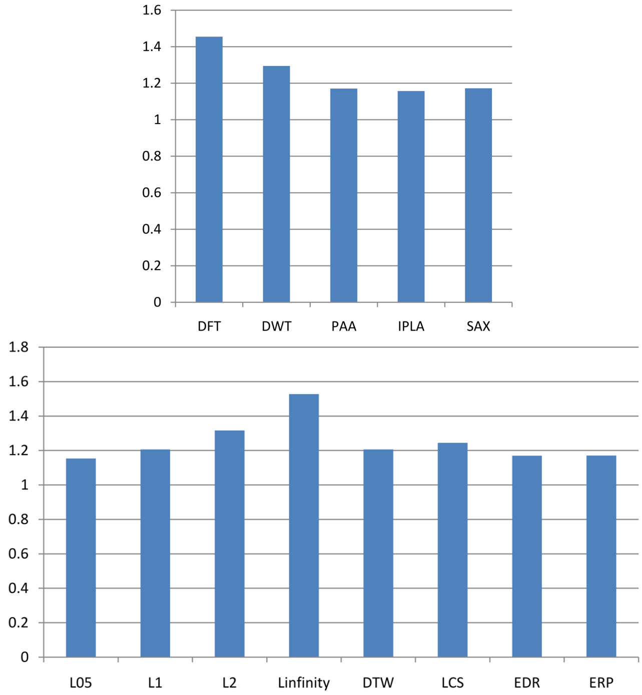
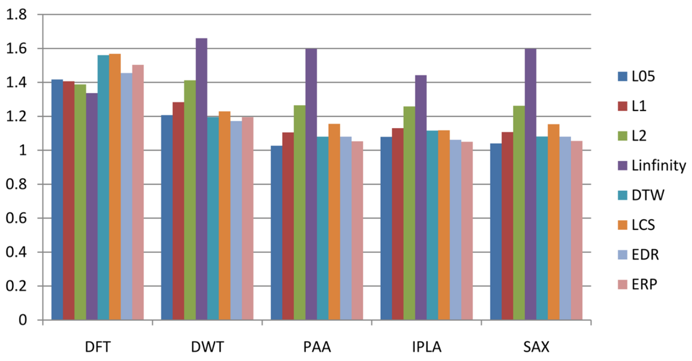


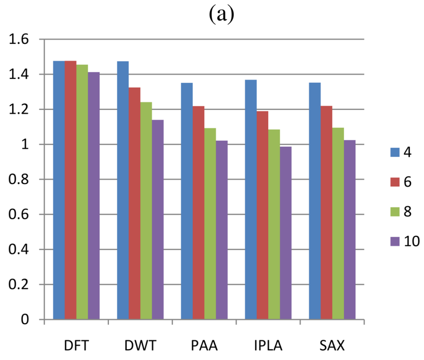
Fig. 4: Averaged SMRE values per representation (top) and per similarity measure (bottom).

All of the mentioned results are giving us global picture, which is not necessary valid for certain instance of data. So the next part of the analysis has the purpose to tell us what type of data sets are more suitable to what representations and similarity measures. In order to test that, we recorded a few properties that describe each data set, and than we calculated correlation between each such property and each SMRE value among all data sets. The properties that we examined are: number of classes that characterize data set instances, size of data set, dimensionality of time series, average standard deviation of time series' that data set is contained of and average distance between randomly sampled data set points. As we expected, correlations between SMRE and number of classes, as well as between SMRE and data set size, are minimal and not significant. Slight increase of correlation is manifested in DFT representation, but it is still bellow 0.2. One more interesting thing about that is that correlations are mostly positive - negative values are real rareness. That implies that there actually exists small tendency in results getting worse as the number of classes or data set size increase. But still, correlation coefficient is small enough that we cannot rely on such conclusion.

Unlike for number of classes and data set size, correlation values are in some cases significantly high for remaining data set properties (Figure 6). We will start from property

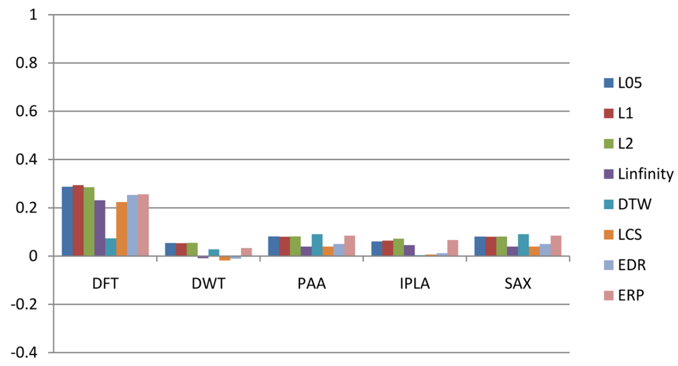


(a)

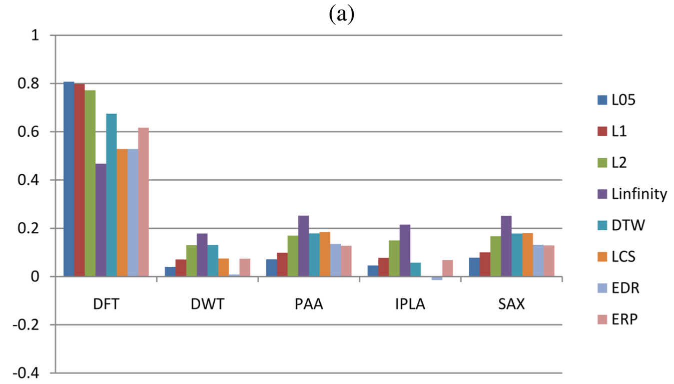


(b)

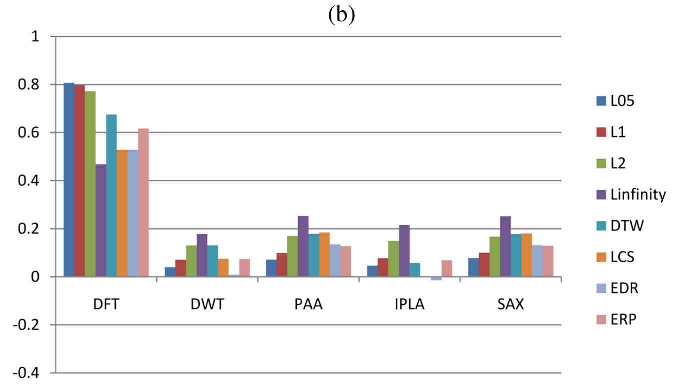
Fig. 5: Averaged SMRE values for each similarity measure inside each representation (figure a) and for each time series dimensionality setting inside each representation (figure b).



(a)



(b)



(c)

Fig. 6: (a) Correlation values between SMRE and data set dimensionality. (b) Correlation values between SMRE and averaged time series' standard deviation. (c) Correlation values between SMRE and averaged distances of randomly sampled data set points.

that corresponds to data set dimensionality (Figure 6a). In this case, the only significant correlations exist for DFT representation, excluding DFT in combination with L_∞-Norm. Those correlation values are not very large, they are about 0.25, but they still indicate that DFT representation doesn't react well when dimensionality of data set is higher. We will now analyze correlations between SMRE and average standard deviation of time series' (Figure 6b). DFT has very high correlation values, and the correlation values of the other representations are also not negligible. In case of DFT, we have very certain information about very high influence of time series' standard deviation on similarity measure effectiveness upon reconstructed data. As standard deviation gets larger, similarity measures upon DFT reconstructions will deviate more from their original behavior. Very similar thing is detected for property that tells about average distance of randomly sampled data set points. This property determines if data set contains points that are distanced from each other, or it contains points that are all very similar. High correlation in case of DFT representation indicates that DFT gives worse results as distances between data set points increase.

VI. Conclusion and Future Work

During this research we paired reconstructions of different time series representations with different similarity measures, and we concluded what does and what does not fit together. First thing to note is that most unstable time series representation is DFT. Its behavior varies a lot, and is affected by different kind of parameters. It can produce results that are significantly better from the results of the other representations, but it can also produce results that are much worse. So one has to be careful when choosing DFT as target representation.

Next thing to note is that elastic similarity measures are less adaptive to DFT reconstructions than L_p-Norms are. For all the other representations the opposite is true - elastic measures adapts better.

DFT representation is also affected by data set dimensionality, average standard deviation of time series, and by pairwise time series' distances. As those values increase, DFT representation gives worse results. The other representations are also affected by these data set properties, but much less.

Future work could be directed to discovering how exactly and why those phenomenons happen. Also, one direction of future research could be examining how reconstructed data sets, together with different similarity measures, affect classification and other data mining algorithms.

Acknowledgments

This work was partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, through project no. OI174023: Intelligent techniques and their integration into wide-spectrum decision support. The authors would like to thank Eamonn Keogh for collecting and making available the UCR time series data sets, as well as everyone who contributed data to the collection, without whom the presented work would not have been possible.

References

[1] Y. Ren, P. Suganthan, N. Srikanth, and G. Amaratunga, "Random vector functional link network for short-term electricity load demand forecasting," *Information Sciences*, pp. –, 2016.

[2] C. Zecchin, A. Facchinetti, G. Sparacino, G. D. Nicolao, and C. Cobelli, "A new neural network approach for short-term glucose prediction using continuous glucose monitoring time-series and meal information," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 5653–5656.

[3] M. Zhu and L. Wang, "Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–5.

[4] S. Gupta and L. Wang, "Stock forecasting with feedforward neural networks and gradual data sub-sampling," *Austr. J. Intelligent Information Processing Systems*, vol. 11, 2010.

[5] R. Ghazali, A. J. Hussain, D. Al-Jumeily, and P. Lisboa, "Time series prediction using dynamic ridge polynomial neural networks," in *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*, 2009, pp. 354–363.

[6] R. Bellman, "Dynamic programming." Princeton, New Jersey: Princeton University Press. XXV, 342 p. (1957)., 1957.

[7] M. Verleysen and D. François, *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, ch. The Curse of Dimensionality in Data Mining and Time Series Prediction, pp. 758–770.

[8] W. Wang and J. Yang, *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2005, ch. Mining High-Dimensional Data, pp. 793–799. [Online]. Available: http://dx.doi.org/10.1007/0-387-25465-X_37

[9] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient similarity search in sequence databases," in *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, 1993, pp. 69–84.

[10] K.-p. Chan and A. Wai-chee Fu, "Efficient time series matching by wavelets," in *Proceedings of the 15th International Conference on Data Engineering*, 1999, pp. 126–133.

[11] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, no. 3, pp. 263–286, 2001.

[12] E. Keogh and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in *Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, 1998, pp. 239–241.

[13] Q. Chen, L. Chen, X. Lian, Y. Liu, and J. Xu Yu, "Indexable pla for efficient similarity search," in *Proceedings of the 33rd International Conference on Very Large Data Bases*, 2007, pp. 435–446.

[14] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[15] L. Wang, K. K. Teo, and Z. Lin, "Predicting time series with wavelet packet neural networks," in *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, vol. 3, 2001, pp. 1593–1597 vol.3.

[16] K. K. Teo, L. Wang, and Z. Lin, *Computational Science - ICCS 2001: International Conference San Francisco, CA, USA, May 28—30, 2001 Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, ch. Wavelet Packet Multi-layer Perceptron for Chaotic Time Series Prediction: Effects of Weight Initialization, pp. 310–317.

[17] D. C. Park, "A time series data prediction scheme using bilinear recurrent neural network," in *2010 International Conference on Information Science and Applications*, 2010, pp. 1–7.

[18] V. Kurbalija, M. Ivanovi, and Z. Budimac, "Case-based curve behaviour prediction," *Software: Practice and Experience*, vol. 39, no. 1, pp. 81–103, 2009.

[19] Y.-L. Wu, D. Agrawal, and A. El Abbadi, "A comparison of dft and dwt based similarity search in time-series databases," in *Proceedings of the Ninth International Conference on Information and Knowledge Management*, ser. CIKM '00. ACM, 2000, pp. 488–495.

[20] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," in *Proceedings of the VLDB Endowment*, vol. 1, no. 2, 2008, pp. 1542–1552.

[21] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD Workshop*, 1994, pp. 359–370.

[22] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[23] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings of 18th International Conference on Data Engineering*, 2002, pp. 673–684.

[24] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005. ACM SIGMOD International Conference on Management of Data*, 2005, pp. 491–502.

[25] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the 30th International Conference on Very Large Data bases, Volume 30*, 2004, pp. 792–803.

[26] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," 2015. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data/

[27] V. Kurbalija, M. Radovanović, Z. Geler, and M. Ivanović, "A framework for time-series analysis," in *Proceedings of the 14th International Conference, AIMSA*, 2010, pp. 42–51.

[28] V. Kurbalija, M. Radovanović, Z. Geler, and M. Ivanović, "The influence of global constraints on similarity measures for time-series databases," *Knowledge-Based Systems*, vol. 56, pp. 49–67, 2014.

[29] Z. Geler, V. Kurbalija, M. Radovanović, and M. Ivanović, "Comparison of different weighting schemes for the knn classifier on time-series data," *Knowledge and Information Systems*, pp. 1–48, 2015.