Received: 11 December 2015 | Revised: 10 April 2016 | Accepted: 8 July 2016

DOI 10.1111/exsy.12171

WILEY Expert Systems

INVITED REVIEW

Aggressive pruning strategy for time series retrieval using a multi-resolution representation based on vector quantization coupled with discrete wavelet transform

Muhammad Marwan Muhammad Fuad

Aarhus University, MOMA, Palle Juul-Jensens Boulevard 99, 8200 Aarhus, Denmark

Correspondence
Muhammad Marwan Muhammad Fuad, Aarhus University, Aarhus, Denmark.
Email: marwan.fuad@clin.au.dk

Abstract

Time series representation methods are widely used to handle time series data by projecting them onto low-dimensional spaces where queries are processed. Multi-resolution representation methods speed up the similarity search process by using pre-computed distances, which are calculated and stored at the indexing stage and then used at the query stage, together with filters in the form of exclusion conditions. In this paper, we present a new multi-resolution representation method that combines the Haar wavelet-based multi-resolution method with vector quantization to maximize the pruning power of the similarity search algorithm. The new method is validated through extensive experiments on different datasets from several time series repositories. The results obtained prove the efficiency of the new method.

KEYWORDS

dimensionality reduction, Haar wavelets, multi-resolution, time series, vector quantization

1 | INTRODUCTION

A time series $S = \langle s_1 = \langle v_1, t_1 \rangle, s_2 = \langle v_2, t_2 \rangle, \ldots, s_n = \langle v_n, t_n \rangle \rangle$ of length n is a chronological collection of observations v_n measured at timestamps t_n.

Because of their particular nature, time series are usually treated as whole entities rather than individual numeric values.

Owing to its numerous applications, time series data mining received much attention in the last two decades. Research in time series data mining has focused on different aspects such as finding similar time series, subsequence searching in time series, dimensionality reduction, and segmentation (Fu, 2011).

The fundamental problem of time series data mining is to represent the data so that different data mining tasks can be handled effectively and efficiently.

The *generic multimedia indexing* (GEMINI) algorithm has been proposed to tackle the similarity search problem (Faloutsos, Ranganathan, & Manolopoulos, 1994). GEMINI reduces the dimensionality of time series by converting them from a point in an n-dimensional space into a point in an N-dimensional space, where $N << n$. A similarity measure is defined on the reduced space, which is a lower bound of the original similarity measure. Under this condition, the similarity search returns no false dismissals. A post-processing sequential scan on the candidate response set is performed to filter out all false alarms and return the final response set. Figure 1 illustrates the GEMINI algorithm.

Dimensionality reduction techniques, also known as representation methods, map the time series onto low-dimension spaces, thus reduce their dimensionality. The different data mining tasks can then be performed in these spaces under certain conditions, mainly a lower bounding distance.

Instead of mapping the data into a single space, *multi-representation* approaches store data at different levels. These levels are called *resolution levels*. The principle of this representation is that a representation with a maximum resolution contains all data of the lower resolutions (Sun & Zhou, 2005).

Traditional time series dimensionality reduction techniques use a common approach; that is, they predefine the dimensionality of the reduced space at indexing time (based on experiments, for instance), and the performance of the method at query time completely depends on the choice made at indexing time. But in practice, we do not necessarily know *a priori* the optimal dimension of the reduced space.

This was the motivation behind our multi-resolution approaches that offer more control on the parameters that determine the effectiveness and efficiency of the dimensionality reduction methods. The basis of these multi-resolution methods is to map the time series to multiple spaces instead of one.

Algorithm Range Query

Require q (query), ε (threshold)

1. Transform the time series in database U from the original n-dimensional space into a lower dimensional space of N dimensions
2. Define a lower bounding distance on the reduced space:

$$d^N(s_i, s_j) \leq d^n(s_i, s_j) \quad \forall s_i, s_j \in U$$

3. Eliminate all the time series for which we have $d^N(q, s) > \varepsilon$ to obtain a candidate answer set
4. Apply d^n to the candidate answer set and eliminate all the time series that are farther than ε from q to get the final answer set.

FIGURE 1 The generic multimedia indexing algorithm for range queries

In this paper, we enhance the pruning power of the multi-resolution method based on Haar wavelets by combining it with vector quantization (VQ) to obtain the maximum pruning power.

The rest of this paper is organized as follows: in Section 2, we introduce the related background. The new method is presented in Section 3 and validated experimentally in Section 4. We concluded this paper with Section 5.

2 | RELATED WORK

2.1 | Multi-resolution representation of time series

In Muhammad Fuad and Marteau (2010b), we presented weak-MIR, which is a standalone, multi-resolution algorithm for indexing and retrieval of time series. Weak-MIR consists of two stages: the indexing stage and the query stage. In the former stage, the algorithm computes and stores distances corresponding to a number of resolution levels, with lower resolution levels having lower dimensions. In the later stage, the algorithm takes advantage of these precomputed distances to speed up the similarity search by applying two filters in the form of pruning conditions.

Let U be the original n-dimension space and R be a $2m$-dimension space, where $2m \leq n$. In the indexing stage, each time series $u \in U$ is divided into m segments each of which is approximated by a polynomial so that the approximation error between each segment and the corresponding polynomial is minimal. By definition, the n-dimension vector whose components are the images of all the points of all the segments of a time series on that approximating function is called the *image vector* and denoted by $\overline{u}$. The images of the two end points of a segment are called the *main image* of that segment. The $2m$ main images of each time series are called the *projection vector* u^R.

Weak-MIR defines two distances; the first is d, which is a distance in an n-dimension space, so it is the distance between two time series in the original space, that is, $d(u_i, u_j)$, or the distance between the original time series and its image vector, that is, $d(u_i, \overline{u}_i)$. The second distance is d^R, which is defined on a $2m$-dimension space, so it is the distance between two projection vectors, that is, $d^R\left(u_i^R, u_j^R\right)$. We proved in Muhammad Fuad and Marteau (2010b) that d^R is a lower bound of d when the Minkowski distance is used.

The resolution level k is an integer related to the dimension of the reduced space R. So the aforementioned definitions of the projection vector and the image vector can be extended to further segmentation of the time series using different values $m \leq m_k$. The image vector and the projection vector at level k are denoted by $\overline{u}^{(k)}$ and $u^{R(k)}$, respectively.

Given a query (q, ε), let $\overline{u}^{(k)}$ and $\overline{q}^{(k)}$ be the projection vectors of u and q, respectively, on their approximating functions at resolution level k , where $u \in U$. By applying the triangle inequality, we get:

$$\left|d\left(u, \overline{u}^{(k)}\right) - d\left(q, \overline{q}^{(k)}\right)\right| > \varepsilon. \tag{1}$$

This equation represents a pruning condition, which is the first filter of weak-MIR.

Weak-MIR also uses another filter that is also obtained by applying the triangle inequality, but it is not related to the topic of this paper.

In Muhammad Fuad and Marteau (2010a), we introduced another multi-resolution method for time series indexing and retrieval, MIR-X. MIR-X uses one of the two filters that weak-MIR uses (Equation 1) together with the low-dimension distance of a time series dimensionality reduction technique. We also showed how MIR-X is faster than weak-MIR.

Whereas weak-MIR has the advantage of being a standalone indexing method, MIR-X, on the other hand, has a better performance in terms of speed.

In Muhammad Fuad and Marteau (2010c), we presented tight-MIR, which has the advantages of both weak-MIR and MIR-X in that it is a standalone method, like weak-MIR, yet it has the same competitive performance of MIR-X. In order to calculate d^R, tight-MIR computes the second filter of weak-MIR differently. This makes d^R tighter than in the way it is as computed in weak-MIR.

In Muhammad Fuad (2015), we presented H-MIR, which is a multi-resolution time series representation and indexing method based on Haar wavelets.

Wavelets are mathematical tools for decomposing functions hierarchically. Regardless of whether the function of interest is an image, a curve, or a surface, wavelets offer an elegant technique for representing the levels of details present (Stollnitz, DeRose, & Salesin, 1995). Wavelets have successfully been used in many fields of computer science such as image compression (DeVore, Jawerth, & Lucier, 1992), image querying (Jacobs, Finkelstein, & Salesin, 1995), and many others. Discrete wavelet transform (DWT) has also been used in time

series information retrieval as a dimensionality reduction technique (Chan & Wai-chee Fu, 1999), (Popivanov & Miller, 2002), (Wu, Agrawal, & Abbadi, 2000). The advantage that DWT has over other methods in indexing time series data is that DWT is a multi-resolution representation method and it can represent local information in addition to global information.

Haar wavelets are the simplest form of wavelets. Haar wavelet transform is a series of averaging and differentiating operations. The basic idea of using DWT as a dimensionality reduction technique is that a time series can uniquely be represented by a wavelet transform, but by keeping only the first *N* coefficients, we can reduce the dimensionality and keep much of the information that exists in the original time series. For instance, Figure 2 shows the DWT decomposition at level 7 of a 128-dimension time series.

A lower bounding distance to the Euclidean distance was presented in Chan and Fu (1999), and it was proven that this lower bound guarantees no false dismissals. It is important to mention that DWT requires that the length of the time series be a power of 2.

Wavelets are particularly adapted to our approach because they are multi-resolution by nature. Haar wavelets are the simplest form of wavelets. A Haar wavelet transform is a series of averaging and differentiating operations.

H-MIR is based on Haar wavelets. In the following, we present a summary of H-MIR. Let U be an n-dimension space where the data are embedded. Each time series is represented by a DWT (Haar wavelet) at each resolution level k, keeping only the first 2^kcoefficients. This reduced space is referred to by $R^{(k)}$. This representation is the optimal approximation at level k. The image of all the points of the time series on DWT is an n-dimension vector called the *image vector* and denoted by $\overline{u}^{(k)}$, and the DWT representation at every resolution level is denoted by $u^{R(k)}$. H-MIR applies two distances; the first is d^n, which is an n-dimension distance metric. The second is $d^{R(k)}$, which is the distance between two DWT representations of two time series at level k. This last distance is a lower bound of the Euclidean distance.

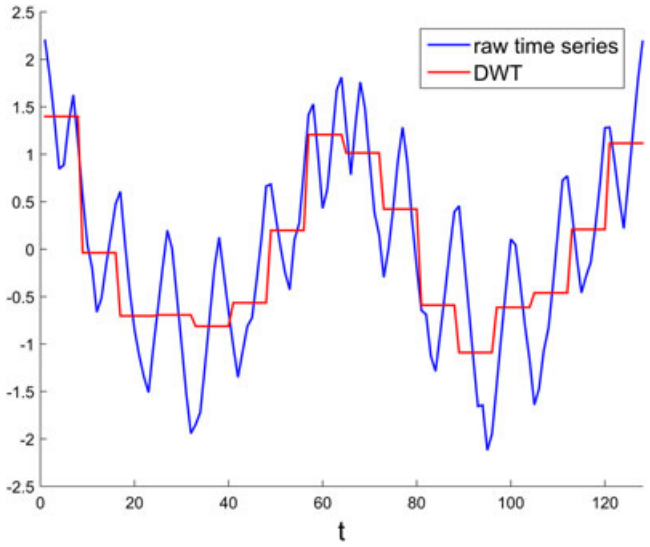

FIGURE 2 Discrete wavelet transform (DWT) representation of a time series

H-MIR applies two exclusion conditions; the first is identical to the one given in Equation 1 (but $\overline{u}^{(k)}$ and $\overline{q}^{(k)}$ in this case are the projection vectors of u and q on the corresponding Haar representation at level k), and the second is as follows:

$$d^R\left(u^{R(k)}, q^{R(k)}\right) > \varepsilon. \tag{2}$$

H-MIR consists of two stages: the indexing stage and the query stage.

Indexing stage: In this stage, each time series, and for each resolution level k, is mapped onto a 2^k-dimension space. This is achieved by using a DWT transform and keeping the first 2^k coefficients. The distances $d^n\left(u, \overline{u}^{(k)}\right), \forall u \in U$ are computed and stored.

Query stage: The query is also mapped onto a 2^k-dimension space using a DWT transform and keeping the first 2^k coefficients.

2.2 | Piecewise vector-quantized approximation

Vector quantization is a data compression technique that has been used in different domains such as signal compression and coding. The principle of VQ is that coding systems can perform better if they operate on vectors or groups of symbols rather than on individual symbols (Wang, Megalooikonomou, & Faloutsos, 2008).

Designing a VQ-based system consists of three steps (Wang et al., 2008):

1. Constructing a codebook from a set of training samples.
2. Encoding the time series with the indices of the nearest code vectors in the codebook.
3. Using an index representation to reconstruct the time series by looking up in the codebook.

Wang, Megalooikonomou, and Faloutsos (2008) introduce a time series representation method that reduces dimensionality from n to m using a codebook of size s by applying VQ to the segmented time series. They call their method the *piecewise vector quantized approximation* (PVQA). In PVQA, each time series is divided into m segments, each with a length l. For each segment, the closest code word in the codebook is found, and the corresponding index is stored. As a result, each time series is represented by a vector of indices to code words.

The codebook is created by forming a training set of segmented time series into m segments each of length l. After training, each code word in the codebook represents a key sequence, which is an approximation for a certain segment of the same length. The key sequences are obtained by applying the generalized Lloyd algorithm (Lloyd, 1982).

After training, all the time series in the database are segmented and encoded using the codebook. During the query stage, the query is first encoded using the codebook and processed by computing the similarity between its encoded representation and the encoded representations of the time series in the database.

The distance between two time series in Wang, Megalooikonomou, and Faloutsos (2008) is a rough distance, which is defined as the distance between their encoded representations. This distance is approximate and thus not appropriate for our filters defined in

Equations 1 and 2. However, Wang et al. (2008) do present in the Appendix section of their paper a lower bounding distance, which guarantees no false dismissals and returns exact results. In order to calculate this distance, which is called the *lower bounding rough distance*, the minimum distance between partitions of two code words is calculated and stored:

$$d'_{ij} = d_{ij} - r_i - r_j, \tag{3}$$

where r_i and r_j are the radii of the two partitions and d_{ij} is the Euclidean distance between the two code words c_i and c_j. The lower bounding rough distance between two time series S and T, whose encoded representations are $S' = s'_1, s'_2, \ldots, s'_w$ and $T' = t'_1, t'_2, \ldots, t'_w$, respectively, is defined as follows:

$$LBRD(S,T) = \sqrt{\sum_{i=1}^{w} \left(d'_{s'_i t'_i}\right)^2}. \tag{4}$$

3 | AGGRESSIVE PRUNING STRATEGY

In many similarity search problems, distance computations can be a very time-consuming task to a point that other tasks such as CPU time or even I/O time can be neglected. The basis of the method we present in this work, which we denote by A-MIR, is to speed up the similarity search process as much as possible by reducing query-time distance evaluations to the least degree possible. This is achieved by using, for each resolution level, two representations of each time series: The first is for the whole time series, and the other is for a segmentation of this time series. Global representation is obtained by using Haar wavelets. This representation is particularly adapted to our multi-resolution approach as DWTs have the ability of reflecting global, and also local, information.

Representing local information faithfully requires, however, a more refined representation, which we obtain through the PVQA on a segmentation of each time series; that is, each time series is segmented, and each segment is represented by the closest code word of the codebook that is constructed at an earlier stage.

In order to generate the codebook, we use the same algorithm that was used in Wang, Megalooikonomou, and Faloutsos (2008), that is, a modification of the generalized Lloyd algorithm (Lloyd, 1982). The algorithm starts with a general population, which represents the codebook, to which we iteratively apply the two optimality conditions. Those are the nearest neighbor condition and the centroid condition. These two conditions constitute the distortion function between an input vector and a code word. In other words, at each iteration, the optimal partition is found by applying the nearest neighbor condition, and the codebook is updated by applying the centroid condition. This process repeats until the distortion function between each vector and its nearest code word drops under a predefined threshold ε.

The algorithm starts with a codebook containing only one code word that is the centroid of the whole dataset. In each repetition and before applying the Lloyd iteration, it splits all cells and doubles the number of code words from the previous iteration (Wang et al., 2008).

After constructing the codebook, each time series in the database is segmented into m segments and represented by the closest code word from the codebook.

The powerful pruning performance of A-MIR is the result of its using four filters at each resolution level. These four filters are the following: The first filter is Equation 1 applied to the whole time series represented by Haar wavelets, which we will refer to as *condition*-1 hereafter. The second filter is Equation 2 applied to the whole time series represented by Haar wavelets, which we will refer to as *condition*-2 hereafter; *condition*-3 is the application of Equation 1 to the segmented time series and represented by a code vector, and finally, *condition*-4 is the application of Equation 2 to the segmented time series and represented by a code vector. Notice that in this last case, $d^R(u^{R(k)}, q^{R(k)})$ in Equation 2 is actually equal to *LBRD* given by Equation 4.

It is important to mention that applying *condition*-1 and *condition*-3 implies applying Equation 1, which has a much lower computational cost than applying Equation 2 (which is the case when using *condition*-2 and *condition*-4) because Equation 2 includes distance evaluations, which, as we mentioned earlier, has a much higher computational cost than the I/O task required for Equation 1.

The computational cost of applying Equation 2 (*condition*-2 and *condition*-4) increases as the value of k —the resolution level—gets

Algorithm A-MIR_indexing

Require codebook, m (number of segments), k (number of resolution levels), U (database of size s)

1. **for** i=1 to k **do**
2. Divide each time series in U into $m(i)$ segments
3. Represent each segmented time series with a vector from codebook
4. Compute and store the Euclidean distance between each time series and its PVQA representation at resolution level i
5. Represent each time series with a Haar wavelet$i$
6. Compute and store the Euclidean distance between each time series and its Haar wavelet representation at resolution level i
7. **end for**

FIGURE 3 The A-MIR indexing algorithm

higher, because higher resolution levels correspond to smaller segments thus more data points.

As for the computational cost of *condition*-2 compared to that of *condition*-4 for the same resolution level, they are not actually comparable because the computational cost of *condition*-2 depends on the number of coefficients N we use in the Haar wavelet representation, whereas the computational cost of applying *condition*-4 is determined by the compression ratio (the number of segments—m) we use in the PVQA representation.

Figure 3 shows the A-MIR indexing algorithm, and Figure 4 shows the A-MIR querying algorithm.

4 | EXPERIMENTAL EVALUATION

We validated our new algorithm A-MIR by conducting extensive experiments on time series datasets of various sizes and dimensions and from different repositories (Povinelli; SISTA's Identification Database; StatLib – Datasets Archive; Chen et al., 2015) using different threshold values. This last repository is particularly important as it makes up between 90% and 100% of all publicly available, labeled time series datasets in the world, and it represents the interest of the data mining/database community, and not just one group (Ding, Trajcevski, Scheuermann, Wang, & Keogh, 2008).

Algorithm A-MIR_querying

Require codebook, m (number of segments), k (number of resolution levels), U (database of size s), q(query), ε (threshold)

1. $Ans = U$ //Ans is the answer set
2. **for** i=1 to k **do** //query pre-processing
3. Divide q into $m(i)$ segments
4. Represent segmented q with a vector from codebook
5. Compute and store the Euclidean distance between q and its PVQA representation at resolution level i
6. Represent q with a Haar wavelet
7. Compute and store the Euclidean distance between q and its Haar wavelet representation at resolution level i
8. **end for**
9. **for** j=1 to k **do** //Applying condition 1
10. **for** l=1 to s **do**
11. **if** $\left|d\left(u(l),\bar{u}^{(j)}(l)\right)-d\left(q(l),\bar{q}^{(j)}(l)\right)\right|>\varepsilon$
12. $Ans(l)\leftarrow$ ()//remove $u^{(j)}(l)$ from Ans
13. **end for**
14. **end for**
15. **for** p=1 to k **do** //Applying condition 3
16. **for** n=1 to s **do**
17. **if** $\left|d\left(u(n),u'^{(p)}(n)\right)-\left(q(n),q'^{(p)}(n)\right)\right|>\varepsilon$
18. $Ans(n)\leftarrow$ ()//remove $u^{(p)}(n)$ from Ans
19. **end for**
20. **end for**
21. **for** r=1 to k **do** //Applying condition 2
22. **for** t=1 to s **do**
23. **if** $d^{R}\left(u^{R(r)}(t),q^{R(r)}\right)>\varepsilon$
24. $Ans(t)\leftarrow$ ()//remove $u^{(r)}(t)$ from Ans
25. **end for**
26. **end for**
27. **for** v=1 to k **do** //Applying condition 4
28. **for** z=1 to s **do**
29. **if** $\sqrt{\sum_{ik=1}^{wk}\left(d'_{u'^{(z)}_{ik}(v)\,q'^{(z)}_{ik}(v)}\right)^{2}}>\varepsilon$ //Eq. 4
30. $Ans(z)\leftarrow$ ()//remove $u^{(v)}(z)$ from Ans
31. **end for**
32. **end for**
33. Scan Ans sequentially in the original space to eliminate false alarms and get the final answer set

FIGURE 4 The A-MIR querying algorithm

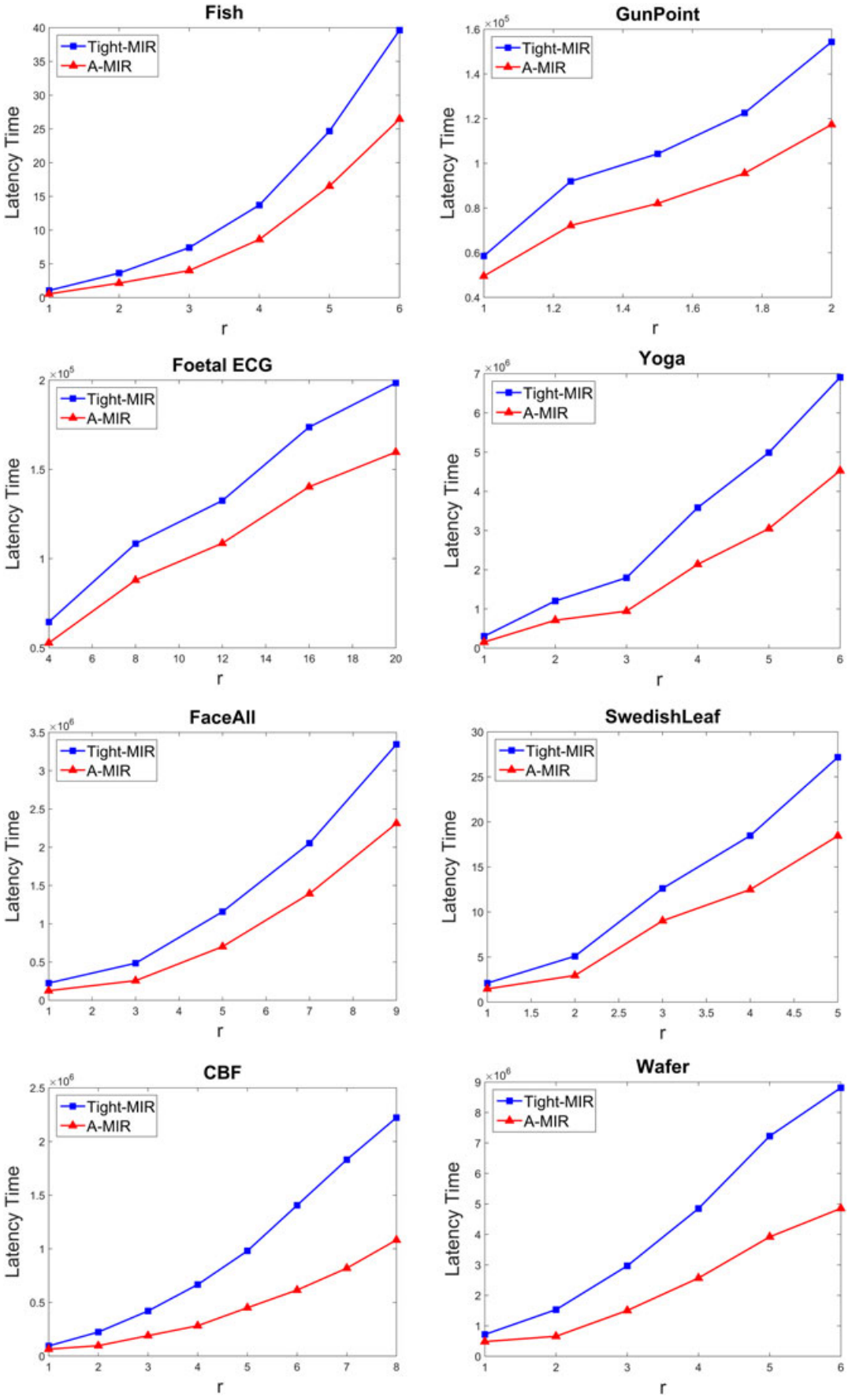

FIGURE 5 Comparison of the latency time between A-MIR and tight-MIR

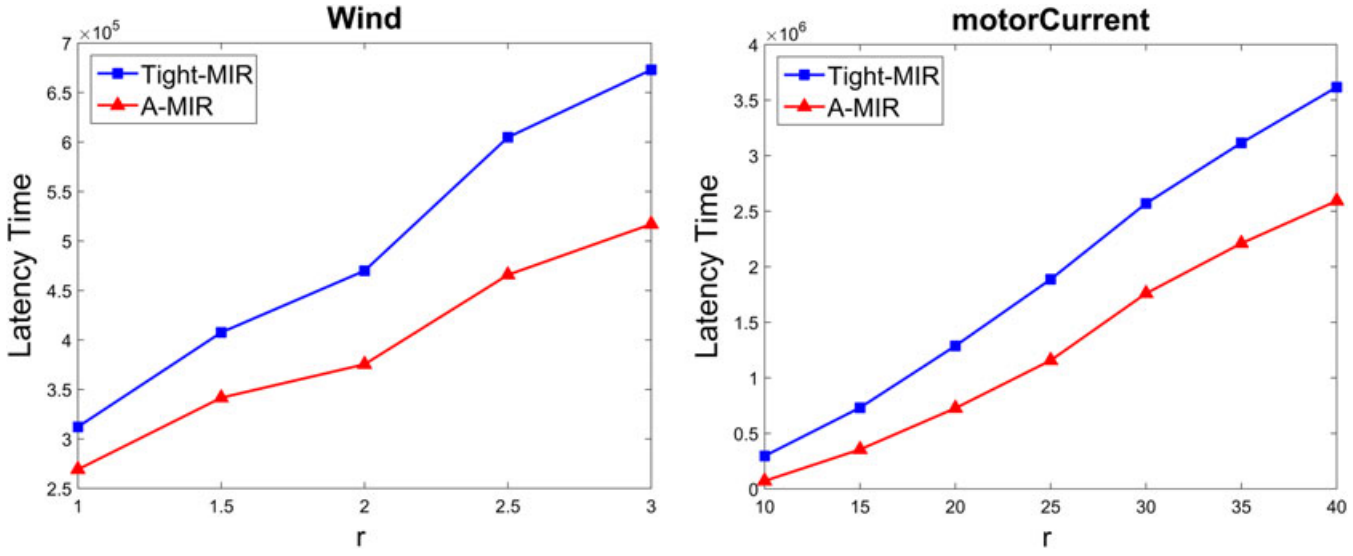

FIGURE 6 Comparison between A-MIR and tight-MIR on (wind) and (motor current)

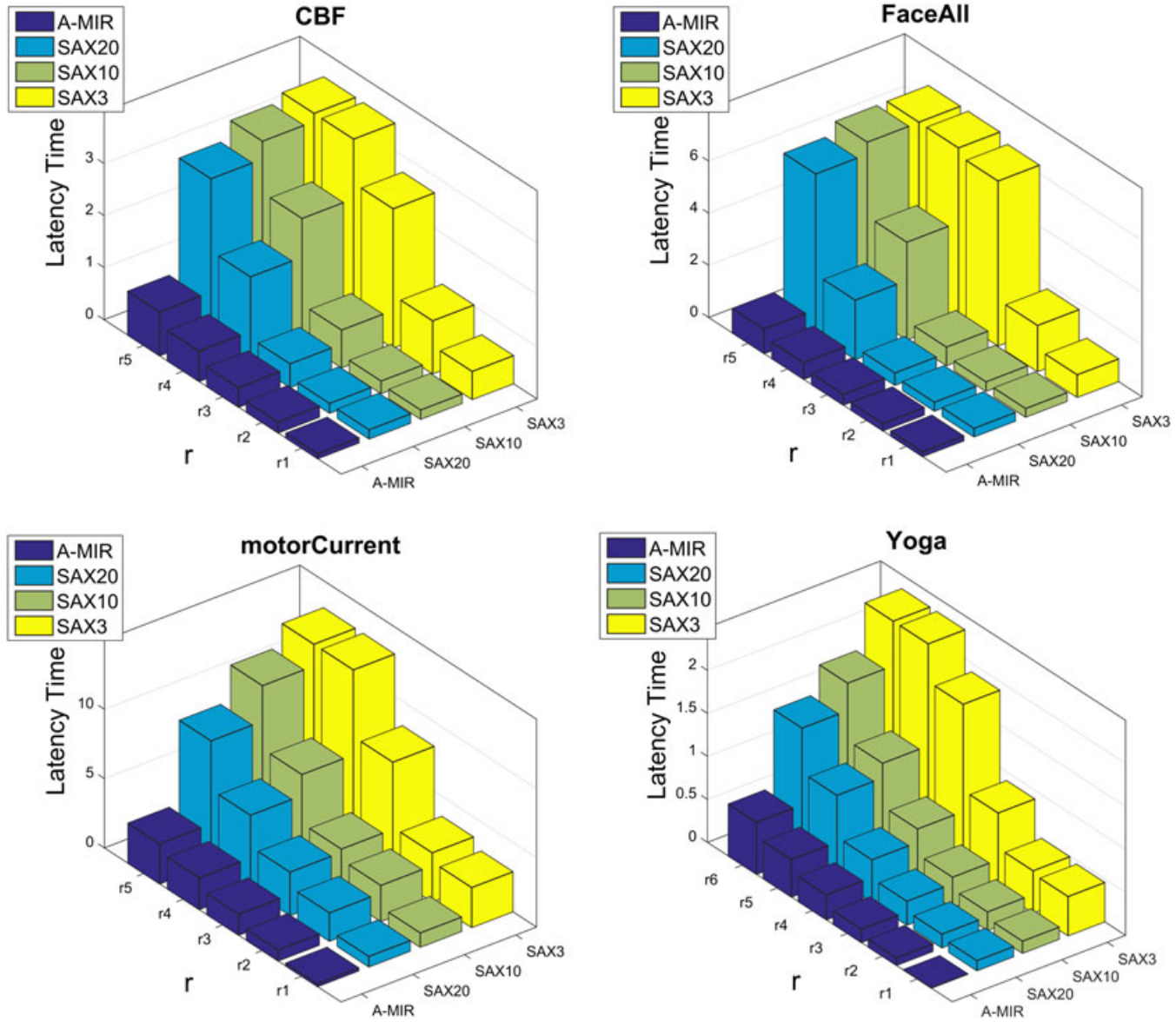

FIGURE 7 Comparison of the latency time between A-MIR and symbolic aggregate approximation (SAX)

The size of the tested datasets varies between 150 instances (gunpoint) and 6,174 instances (wafer). The dimensionality of the datasets (the length of the time series) varies between 12 (wind) and 1,500 (motorCurrent).

As mentioned in Section 2.1, DWT is applicable to time series whose lengths are a power of 2; when this is not the case for the dataset tested, the time series of that dataset are padded with 0s.

Although several papers present experiments based on wall clock time, this criterion is a poor choice and subject to bias (Keogh, Chakrabarti, Pazzani, & Mehrotra, 2000; Ding et al., 2008), so we preferred to use a platform-independent criterion in our experiments that is the *latency time* (Schulte, Lindberg, & Laxminarain, 2005). The latency time is based on the number of cycles the processor takes to perform different arithmetic operations, so we added a counter to compute the number of different operations (>, +, −, *, *abs*, *sqrt*) that the algorithm takes to perform the similarity search process. Then the number of each operation is multiplied by the latency time of that operation to get the total latency time for our method (or the other methods it is compared with). The latency time is 5 cycles for (>, +, −), 1 cycle for (*abs*), 24 cycles for (*), and 209 cycles for (*sqrt*).

In the first set of experiments, we compared A-MIR with tight-MIR because we showed in Muhammad Fuad and Marteau (2010a) that tight-MIR outperforms both weak-MIR and MIR-X.

Figure 5 shows some of the results we obtained. As we can see from the results, A-MIR outperforms tight-MIR for all the datasets tested and for all values of threshold *r*. The results also show that for both A-MIR and tight-MIR, the latency time gets longer as the value of *r* gets higher, which is expected because the query answer set includes more elements (time series) in this case. Yet it seems that A-MIR handles the increase in the value of *r* better than tight-MR does.

In the second set of experiments, we compared the performance of the two methods in extreme situations, that is, on the two datasets in the archive with the lowest dimensionality (wind, 12 dimensions) and the highest dimensionality (motorCurrent, 1,500 dimensions). We report the results in Figure 6. As we can see, A-MIR also outperforms tight-MIR for the two datasets.

In the last set of experiments, we compared the performance of A-MIR with one of the fastest, "single resolution", methods in time series similarity search that is the *symbolic aggregate approximation* method (SAX; Lin, Keogh, Lonardi, & Chiu, 2003). The particularity of SAX is that computing its similarity measure, which is called MINDIST, is based on using statistical lookup tables, which makes it easy to compute with an overall complexity of $O(N)$. SAX is based on the assumption that normalized time series have Gaussian distribution, so by determining the breakpoints that correspond to a particular alphabet size, one can obtain equal-sized areas under the Gaussian curve. SAX is applied as follows:

1. The time series are normalized.
2. The dimensionality of the time series is reduced using Piecewise Aggregate Approximation (PAA) (Keogh *et al.*, 2000; Yi & Faloutsos, 2000).
3. The PAA representation of the time series is discretized by determining the number and location of the breakpoints.

We compared A-MIR with SAX on different datasets, and for different values of the alphabet size. We report in Figure 7 the results of several datasets and for different values of *r*. The results shown here are for alphabet sizes 3 (the smallest alphabet size possible for SAX), 10 (the largest alphabet size in the first version of SAX), and 20 (the largest alphabet size in the second version of SAX).

We have to mention that the datasets used in these last experiments were normalized because SAX can only be applied to normalized time series.

The results obtained show that A-MIR clearly outperforms SAX for the different values of *r* and for the different values of the alphabet size.

It is worth mentioning that the results of SAX as shown in Figure 7 may give the false impression that for some datasets, the number of operations seems to be stable after a certain value for *r*. This phenomenon does not indicate stability of performance. It only indicates that SAX exploited all the indexed time series using the lower bounding condition without being able to exclude any time series, so the search process moved to sequential scanning, so this phenomenon is in fact the worst scenario possible because the number of operations exceeds even that of sequential scanning and reaches the maximum number possible of operations, that is, the maximum number of distance evaluations.

5 | CONCLUSION

In this paper, we presented a new algorithm for time series indexing and retrieval. This method is based on a multi-resolution representation of time series based simultaneously on the Haar wavelets, with their multi-resolution nature, and on VQ. This multi-resolution representation is equipped with four filters in the form of pruning conditions that speed up the similarity search by taking advantage of stored precomputed distances. This representation and indexing scheme reduce query time distance calculations. We conducted experiments that compare our new method with another multi-resolution method and also with a very fast, single-resolution representation method, on different datasets obtained from several time series archives. The results obtained show the superior performance of our new method compared with that of the two aforementioned methods, and for different values of the threshold.

Although the method we presented in this paper was applied to time series data in this work, we see no reason why it cannot be extended to other data types, which is what we plan to study in the future. We believe that, under certain conditions, this method can be applied to images, sequential data, texts, and other data types. We think the two main challenges in this case are (a) to find appropriate lower bounding conditions and (2) to handle issues related to required storage space.

REFERENCES

Chan, K., & Wai-chee Fu, A. (1999). Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering*.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., A. Mueen, & G. Batista. (2015). The UCR time series classification archive. Retrieved from: www.cs.ucr.edu/~eamonn/time_series_data

DeVore, R., Jawerth, B., & Lucier, B.. (1992). Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: experimental comparison of representations and distance measures. In *Proceedings of the 34th VLDB*.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings ACM SIGMOD Conference, Minneapolis*.

Fu, T.-C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*, 164–181.

Jacobs, C. E., Finkelstein, A., & Salesin, D. H. (1995). Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95, ACM, New York*.

Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2000). Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information System*, *3*, 263–286.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. Y. (2003). A symbolic representation of time series, with implications for streaming algorithms. *DMKD*, *2003*, 2–11.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*, 127–135.

Muhammad Fuad, M.M. (2015). A Haar wavelet-based multi-resolution representation method of time series data. *The 7th International Conference on Agents and Artificial Intelligence - ICAART 2015*, January 10–12, 2015. Lisbon, Portugal. *SCITEPRESS Digital Library*.

Muhammad Fuad, M. M., & Marteau, P. F. (2010a). Fast retrieval of time series using a multi-resolution filter with multiple reduced spaces. In *The International Conference on Advanced Data Mining and Applications–ADMA2010* (Vol. 6440). (pp. 137–148). ChongQing, China: Springer-Verlag in Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence.

Muhammad Fuad, M. M., & Marteau, P. F. (2010b). Multi-resolution approach to time series retrieval. In *Fourteenth International Database Engineering & Applications Symposium—IDEAS 2010*. Montreal, QC, Canada.

Muhammad Fuad, M. M., & Marteau, P. F. (2010c). Speeding-up the similarity search in time series databases by coupling dimensionality reduction techniques with a fast-and-dirty filter. In *Fourth IEEE International Conference on Semantic Computing—ICSC 2010*. Pittsburgh, PA, USA: Carnegie Mellon University.

Popivanov, I., & Miller, R. J. (2002). Similarity search over time series data using wavelets. *ICDE*.

Povinelli, R. Retrieved from: http://povinelli.eece.mu.edu/

Schulte, M. J., Lindberg, M., & Laxminarain, A. (2005). Performance evaluation of decimal floating-point arithmetic. In *IBM Austin Center for Advanced Studies Conference*.

SISTA's Identification Database. http://www.esat.kuleuven.ac.be/~tokka/daisydata.html

StatLib - Datasets Archive. http://lib.stat.cmu.edu/datasets/.

Stollnitz, E., DeRose, T., & Salesin, D. (1995). Wavelets for computer graphics: a primer, part 1. *IEEE Computer Graphics and Applications*.

Sun, S., & Zhou, X. (2005). Semantic caching for web-based spatial applications. In *Proceeding of APWeb 2005*, Shanghai, China.

Wang, Q., Megalooikonomou, V., & Faloutsos, C. (2008). Time series analysis with multiple resolutions. *Information Systems*, *35*, 1.

Wu, Y. L., Agrawal, D., & Abbadi, A. E. (2000). A comparison of DFT and DWT based similarity search in time-series databases. In *Proceedings 9th International Conference on Information and Knowledge Management*.

Yi, B. K., & Faloutsos, C. (2000). Fast time sequence indexing for arbitrary Lp norms. In *Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt*.

How to cite this article: Muhammad Fuad MM. Aggressive pruning strategy for time series retrieval using a multi-resolution representation based on vector quantization coupled with discrete wavelet. *Expert Systems*. 2017;34:e12171. https://doi.org/10.1111/exsy.12171

AUTHOR BIOGRAPHY

M. M. Muhammad Fuad has received a PhD in Information Technology from the University of Bretagne-Sud, a DU in Facial Surgery from the Faculty of Medicine, University of Lorraine, and a CES in maxillofacial prosthetics from the Faculty of Dentistry, University of Nantes. He is currently a postdoctoral fellow at the department of clinical medicine at Aarhus University. His research interests include data mining, machine learning, bio-inspired optimization, and bioinformatics.