International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 98 - 114

Forecasting Through Motif Discovered by Particle Swarm Optimization Algorithm

Dr. R. Kumar[1] and B. Ayshwarya[2]
[1]Associate Professor, [2]Assistant Professor

Department of Computer Science
Kristu Jayanti College, Bangalore, India

Abstract

Forecasting is an operational research technique used as a basis for management planning and decision making. Forecasting provides clues and reduce risk and uncertainties. This technique has its own indispensable importance in a wide range of zones including meteorology, energy economics and the study stock market etc. The main goal of this work is to predict the occurrence of outliers in time series, based on the discovery of motifs by using Particle Swarm Optimization (PSO) algorithm. Forecasting the future values of the time series are carried out by the discovered motifs. Such forecasted values are compared with its counterparts obtained by different kind of methods like Vector Auto Regressive Prediction model, native modal and motif based forecasting using Genetic Algorithms. The achieved results prove the potency of the proposed PSO algorithm.

Keywords---Forecasting, Genetic Algorithms, Motif Discovery, Operational Research, Particle Swarm Optimization.

1. Introduction

Time series are sequences of real numbers measured at successive, usually regular time intervals. Data in the form of time series pervade science, business, and society. Examples range from economics to medicine, from biology to physics, and from social to computer sciences. Repetitions or recurrences of similar phenomena are a fundamental characteristic of non-random natural and artificial systems and, as a measurement of the activity of such systems, time series often include pairs of segments of strikingly high similarity. These segment pairs are commonly called motifs [1], and their existence is unlikely to be due to chance alone. In fact, they usually carry important information about the underlying system. Thus, motif discovery is fundamental for understanding, characterizing, modeling, and predicting the system behind the time series [2].

Motif discovery is a core part of several higher-level algorithms dealing with time series, [2], [3]. Identifying similar segment pairs or motifs implies examining all pairwise comparisons between all possible segments in a time series. This, especially when dealing with long time series streams, results in prohibitive time and space complexities. It is for this reason that the majority of motif discovery algorithms resort to some kind of data discretization or approximation that allows them to hash and retrieve segments efficiently. Following the works by Lin et al. [1] and Chiu et al. [4], many of such approaches employ the SAX representation [5] and/or a sparse collision matrix [6]. These allow them to achieve a theoretically low

ISSN: 2233-7857 IJFGCN

computational complexity, but sometimes at the expense of very high constant factors.

A few recent approaches overcome some of these limitations. For instance, [7] propose an amplitude multi-resolution approach to detect frequent segments, [8] use a grammar inference algorithm for exploring motifs with lengths above a certain threshold, [9] use concepts from immune memory to deal with different lengths, and [10] combine suffix trees with segment models to find motifs of any length. Nevertheless, in general, these approaches still suffer from other data-dependent parameters whose correct tuning can require considerable time. Finally, to the best of our knowledge, only [12]–[14] consider the identification of motif pairs containing segments of different lengths. This can be considered a relevant feature, as it produces better results in a number of different domains [13]. In contrast to approximate approaches, algorithms that do not discretize the data have been comparatively much less popular, with low efficiency generally. Exceptions to this statement achieved efficiency by sampling the data stream [15] or by identifying extreme points that constrained the search [16].

In fact, until the work in [17], the exact identification of time series motifs was thought to be intractable for even time series of moderate length. In said work, a clever segment ordering was combined with a lower bound based on the triangular inequality to yield the true, exact, most similar motif. According to the authors, the proposed algorithm was more efficient than existing approaches, including all exact and many approximate ones [17]. After this work, a number of improvements have been proposed, the majority focusing on eliminating the need to set a fixed segment length [18]–[20]. This algorithm, called MOEN [3], is essentially parameter-free, and is believed to be one of the most efficient motif discovery algorithms available nowadays. However, its complexity is still quadratic in the length of the time series [3], and hence its applicability to large-scale time series streams remains problematic. In general, exact motif discovery algorithms have important restrictions with regard to the dissimilarity measure, and many of them still suffer from being non-intuitive and tedious to tune parameters. Moreover, few of them allow for anytime versions and, to the best of our knowledge, not one of them is able to identify motif pairs containing segments of different lengths.

In this article, Particle Swarm Optimization Algorithm (PSO) has been used for finding motifs through a heuristic algorithms complete search is very time-consuming. Also, due to rapid advancement in parallel computing, PSO would become much faster and more applicable in the future. Joan [20,21] used PSO for motif discovery. He showed the effectiveness of PSO in discovering motifs in discrete time-series data of genomes. The present paper takes inspiration from Joan's work to find multi-dimensional motifs in multi-dimensional time-series data. Then, these motifs are used for forecasting of multi-dimensional time-series data in a way similar to [22]. The main contribution of this paper is the development of PSO for multi-dimensional motif discovery.

ISSN: 2233-7857 IJFGCN

This paper is organized as follows. Section 2 discusses the related works. Section 3 gives various definitions, anytime solution and Particle Swarm optimization related to time series motif discovery and pseudo code for the algorithm proposed in the present paper. Section 4 presents implementation details and forecasting using discovered motifs. Section 5 gives the comparison results between the proposed forecasting algorithm and other models and GA based motif discovery. Section 6 contains the conclusion.

2. Related Works

Lin et al., (2002), proposed an efficient algorithm to discover motifs, and demonstrated the utility and efficiency of the proposed approach on several real world datasets. In their work they carefully motivate, then introduce a non-trivial definition of time series motifs. They have formalized the problem of finding repeated patterns in time series, and introduced an algorithm to efficiently locate them. In addition, a minor contribution of this paper is to introduce the first discrete representation of time series that allows a lower bounding approximation of the Euclidean distance. This representation may be of independent interest to researchers who use symbolic representations for similarity search, change point detection, and extracting rules from time series.

Mueen (2014), described a set of applications of time series motif in various domains and elaborate on a certain application in entomology to analyze insect behavior. He described definition of motifs, domain based processing and the properties of algorithms in detail to picture different approaches and provide recommendation on adopting them where appropriate. He shows a comparison between motif discovery and similarity search and argue that simple representation and similarity measure are sufficient to find good motifs in large datasets. Finally, he briefly lists a set of interdisciplinary applications of motif discovery.

Chiu et al., (2003), generalized the definition of time series motifs to allow for don't-care subsections, and they introduce a novel time- and space-efficient algorithm to discover motifs. Their method is based on an algorithm for pattern discovery in DNA sequences. The intuition behind the algorithm is to project the data objects (in our case, time series), onto lower dimensional subspaces, based on a randomly chosen subset of the objects features. The lower dimensional space can be quickly post-processed to discover likely candidates for motifs, while the candidates can be quickly checked against the original data.

Castro and Azevedo (2010), tackled the motif discovery problem as an approximate Top-K frequent subsequence discovery problem. They fully exploited state of the art iSAX representation multiresolution capability to obtain motifs at different resolutions. This property yields interactivity, allowing the user to navigate along the Top-K motifs structure. This permits a deeper understanding of the time series database. MrMotif is scalable and can have a strong impact on different application areas due to the good performance and robustness to noise. Further, they applied the Top-K space saving algorithm to our frequent subsequences approach.

ISSN: 2233-7857 IJFGCN

A scalable algorithm was obtained that is suitable for data stream like applications where small memory devices such as sensors are used.

Li and Len (2010), proposed a novel approach, based on grammar induction, for approximate variable-length time series motif discovery. Their algorithm offers the advantage of discovering hierarchical structure, regularity and grammar from the data. The preliminary results are promising. They show that the grammar-based approach is able to find some important motifs, and suggest that the new direction of using grammar-based algorithms for time series pattern discovery might be worth exploring. Several algorithms were proposed to discover motifs of variable lengths; however, they either do so via post-processing, scale poorly, or quantize the whole data rather than considering overlapping subsequences, resulting in inaccurate and incomplete patterns found.

3. Proposed Methodology and Discovery of Time Series Motif

3.1 Definition and Task Entanglement

We can derive a formal, generic similarity-based definition [2] of time series motifs. Given a time series z of length n, $z = [z_1, \ldots z_n]$, a normalized segment dissimilarity measure D, and a temporal window of interest between w_{min} and w_{max} samples, the two p_k time series motifs $M = \{m_1, \ldots m_k\}$ correspond to the k most similar segment pairs,

$$z_a^{wa} = [z_a, \ldots z_{a+w_{a-1}}] \text{ and } z_b^{wb} = [z_b, \ldots z_{b+w_{b-1}}], \text{ for } w_a, w_b \in [w_{min}, w_{max}], a \in [1, n-w_{a+1}], \text{ and } b \in [1, n - w_{b+1}]$$

where, in order to avoid repeated and trivial matches [1], $a + w_a < b$. Thus, the i^{th} motif can be fully described by the tuple, $m_i = \{a, w_a, b, w_b\}$.

The motifs in M are non-overlapping1 and ordered from lowest to highest dissimilarity such that, $D(m_1) \leq D(m_2) \leq \cdots \leq D(m_k)$ where $D(m_i) = D(\{a, w_a, b, w_b\}) = D(z_a^{wa}, z_b^{wb})$. It is important to stress that D needs to normalize with respect to the lengths of the considered segments. Otherwise, we would not be able to compare motifs of different lengths.

From the definitions above, we can see that a brute-force search in the motif space for the most similar motifs is of $O(n^2 w_\Delta^2)$, where $w_\Delta = w_{max} - w_{min} + 1$ (for the final time complexity one needs to further multiply by the cost of calculating D). Hence, for instance, in a perfectly feasible case where n = 107 and $w_\Delta = 103$, we have 1020 possibilities. Magnitudes like this challenge the memory and speed of any optimization algorithm, especially if we have no clue to guide the search [24]. However, it is one of our main objectives to show here that time series generally provide some continuity to this search space, and that this continuity can be exploited by optimization algorithms.

3.2 Continuity

A fundamental property of time series is autocorrelation, implying that consecutive samples in a time series have some degree of resemblance and that,

most of the time, we do not observe extremal differences between them. This property, together with the established ways of computing similarity between time series [23], is what gives continuity to our search space. Consider a typical dissimilarity measure like dynamic time warping between z-normalized segments and the time series. If we fix the motif starting points a and b to some random values, we can compute D (z^i_a, z^j_b) for i, j = w_{min}, . . ., w_{max}.

3.3 Anytime Solutions

Finding an optimization algorithm that can locate the global minima of the previous search spaces faster than existing motif discovery algorithms can be a difficult task. However, we have robust and established algorithms for efficiently locating prominent local minima in complex search spaces [25]–[27]. Hence, we can intuitively devise a simple strategy: if we keep the best found minima and randomly reinitialize the optimization algorithm every time it stagnates, we should, sooner or later, start locating the global minima. In the meantime, we could have obtained relatively good candidates. This corresponds to the basic paradigm of anytime algorithms [11]. Anytime algorithms have recently been highlighted as "very beneficial for motif discovery in massive [time series] datasets" [19]. In an anytime algorithm for motif discovery, D(mi) improves over time, until it reaches the top-k dissimilarity values D(mi)* obtained by a brute-force search approach. Thus, we gradually improve M until we reach the true exact solution M*.

A good anytime algorithm will quickly find low D(mi), ideally reaching D(mi)* earlier than its non-anytime competitors. Note that a sufficiently good M may suffice in most situations, without the need that M = M*. This is particularly true for more exploratory tasks, where one is typically interested in data understanding and visual inspection (see [2]), and can also hold for other tasks, as top-k motifs can be very similar among themselves. In the latter situation, given a seed within M*, we can easily and efficiently retrieve further repetitions via common established approaches [28], [29]. Thus, only non-frequent or singular motifs may be missed. These can be valuable too, as the fact that they are non-frequent does not imply that they cannot carry important information (think for example of extreme events of interest that perhaps only happen twice in a measurement). For those singular motifs, we can wait longer if using an anytime algorithm, or we can resort to the state-of-the-art if that is able to provide its output within an affordable time limit.

3.4 Particle Swarm Optimization

The continuity and anytime observations above (Secs. II-B and II-C) relax the requirements for the optimization algorithm to be employed in the considered motif spaces. In fact, if we do not have to assess the global optimality of a solution, we have a number of approaches that can deal with large, multimodal, continuous but noisy search spaces [25]–[27]. Among them, we choose PSO [30]–[34]. PSO is a population based stochastic approach for solving continuous and discrete optimization problems [33] which has been applied to multimodal problems [35]. It is a metaheuristic [27], meaning that it cannot guarantee whether the found solution

ISSN: 2233-7857 IJFGCN

corresponds to a global optimum. The original PSO algorithm cannot even guarantee the convergence to a local optimum, but adapted versions of it have been proven to solve this issue [36]. Other versions guarantee the convergence to the global optimum, but only with the number of iterations approaching infinity [36].

PSO has gained increasing popularity among researchers and practitioners as a robust and efficient technique for solving difficult optimization problems. It makes few or no assumptions about the problem being optimized, does not require it to be differentiable, can search very large spaces of candidate solutions, and can be applied to problems that are irregular, incomplete, noisy, dynamic, etc. (see [30]–[35] and references therein). PSO iteratively tries to improve a candidate solution with regard to a given measure of quality or fitness function. Hence, furthermore, it can be considered an anytime algorithm.

3.5 Advantages of an Optimization-Based Solution Using Particle Swarms

Notice that treating time series motif discovery as an optimization problem naturally yields several advantages:

1. We do not require much memory, as we can basically store only the stream time series and preprocess the required segments at every fitness evaluation.
2. We are able to achieve a certain efficiency, as optimization algorithms do not usually explore the full solution space and perform few fitness evaluations [24].
3. We can employ any dissimilarity measure D as our fitness function. Its only requirements are segment length independence and a minimal search space continuity. Intuitively, this holds for the high majority of time series dissimilarity measures that are currently used. Additionally, we can straightforwardly incorporate notions of 'interestingness', hubness, or complexity (see references in [23]). This flexibility is very uncommon in current time series motif discovery algorithms.
4. We do not need to force the two segments of the motif to be of the same length. The dissimilarity function D can expressly handle segments of different lengths or we can simply up sample to the largest length (see [22]). Although considering different segment lengths has been highlighted as an objectively better approach, practically none of the current time series motif discovery algorithms contemplates this option.
5. Since we search for the optimal wa and wb, together with a and b, we do not need to set the exact segment lengths as a parameter. Instead, we can use a more intuitive and easier to set range of lengths wa, wb $\in [w_{min}, w_{max}]$.
6. We can easily modify our fitness criterion to work with different task settings. Thus, just by replacing D, we are able to work with multi-dimensional time series [37], detect sub-dimensional motifs [38], perform a constrained motif discovery task [16], etc.

ISSN: 2233-7857 IJFGCN

7. We can incorporate notions of motif frequency to our fitness function and hence expand our similarity-based definition of motif to incorporate both notions [2].

3.6. Proposed Particle Swarm Algorithm

This innovative strategy to time series motif identification is based on the combination of two well-known extensions to the canonical PSO [31]. On one hand, we employ multiple initializations of the swarm on stagnation [39]. On the other hand, we exploit the particles' "local memories" with the intention of forming stable niches across different local minima [40]. The former emulates a parallel multi-swarm approach [35] without the need of having to define the number of swarms and their communication. The latter, when combined with the former, results in a low-complexity niching strategy [35] that does not require niching parameters (see the related discussion in [41], [42]). the implementation of the two extensions, is detailed in Algorithm.

Implementation takes a time series z of length n as input, together with a segment dissimilarity measure D, and the range of segment lengths of interest, limited by wmin and wmax. The user also needs to specify k, the desired number of motifs, and t_{max}, the maximum time spent by the algorithm (in iterations5). The implementation outputs a set of k non overlapping motifs M. We implement M as a priority queue, which typically stores more than k elements to ensure that it contains k non-overlapping motifs. This way, by sorting the motif candidates as soon as they are found, we allow potential queries to M at any time during the algorithm's execution. In that case, we only need to dynamically check the candidates' overlap. Notice that n, D, w_{min}, w_{max}, k, and t_{max} are not parameters of the algorithm, but requirements of the task (they depend on the data, the problem, and the available time). The only parameters to be set, as specified in Algorithm 1's requirements, are the number of particles κ, the topology θ, the constriction constant φ, and the maximum amount of iterations at stagnation τ. Nevertheless, we will show that practically none of the possible parameter choices introduces a significant variation in the reported performance. Having clarified the implementation's input, output, and requirements, we now elaborate on its procedures.

Input: Time series z of length n, dissimilarity measure D, minimum and maximum segment length wmin and wmax, number of motifs k, and maximum amount of time (number of iterations) tmax.

Require: Number of particles κ, topology θ, constriction constant φ, and maximum amount of time at stagnation (number of iterations) τ. Output: A set of motifs M.

1: c0, c1, c2 ← GET CONSTANTS (φ)
2: X, V, S, P ← INITIALIZE SWARM (n, w_{min}, w_{max}, κ)
3: Θ ← INITIALIZE TOPOLOGY (θ, κ)
4: s ∗ ← ∞
5: M ← EMPTY PRIORITY QUEUE ()

6: for t = 1, . . . tmax
7: for i = 1, . . . κ
8: if VALIDPOSITION (xi)
9: d ← D(xi) 10: if d < si
11: si ← d
12: pi ← xi
13: M. PUSH (d, xi)
14: if d < s∗
15: s ∗ ← d
16: t_{update} ← t
17: for i = 1, . . . κ
18: g ← i
19: for j in Θi
20: if sj ≤ sg
21: g ← j
22: vi ← c0vi + c1u1 ⊗ (pi − xi) + c2u2 ⊗ (pg − xi)
23: xi ← xi + vi
24: if t − t_{update} = τ
25: X, V, S, P ← INITIALIZE SWARM (n, w_{min}, w_{max}, κ)
26: s ∗ ← ∞
27: return NON OVERLAPPING (M, k)

Algorithm starts by computing the velocity update constants (line 1) following Clark's constriction method [43].

$$C_0 = \frac{2}{\left|2-\emptyset-\sqrt{\emptyset^2-4\emptyset}\right|} \quad \text{and} \quad C_1 = C_2 = C_0\,\emptyset/2 \qquad (1)$$

Next, a swarm with κ particles is initialized (line 2). The swarm is formed by four data structures: a set of particle positions X = {x1, . . . xκ}, a set of particle velocities V = {v1, . . . vκ}, a set of particle best scores S = {s1, . . . sκ}, and a set of particle best positions P = {p1, . . . pκ} (the initialization of these four data structures is detailed in following algorithm.

Input: Time series length n, minimum and maximum segment length wmin and wmax, and number of particles κ.

Output: Particle positions X, velocities V, best scores S, and best positions P.

1: for i = 1, . . . κ
2: xi,2 ← wmin + (wmax + 1 − wmin) u
3: xi,4 ← wmin + (wmax + 1 − wmin) u
4: xi,1 ← 1 + (n − xi,2) (1 − √ u)
5: xi,3 ← 1 + (n − xi,4 − (xi,1 + xi,2)) u
6: x 0 ← As in lines 2–5
7: vi ← x 0 − xi
8: si ← ∞
9: pi ← xi
10: return X, V, S, P.

Particles' positions xi and pi completely determine a motif candidate, and have a direct correspondence with mi. A further data structure Θ indicates the indices of the neighbors of each particle according to a given social topology θ (line 3). Apart from the swarm, we also initialize a global best score s ∗ (line 4) and the priority queue M (line 5). We then enter the main loop (lines 6–26). In it, we perform three main actions. Firstly, we compute the particles' fitness and perform the necessary updates (lines 7–16). Secondly, we modify the particles' position and velocity using their personal and neighborhood best positions (lines 17–23). Thirdly, we control for stagnation and reinitialize the swarm if needed (lines 24–26). Finally, when we exit the loop, we return the first k non-overlapping motif candidates from M (line 27).

4. Implementation

Firstly, positions are floored component-wise inside VALIDPOSITION, D, and M (thus obtaining motif mi). Secondly, the motif priority queue M is implemented as an associative container (logarithmic insertion time) that sorts its elements according to d and stores m_i. Thirdly, the last visited positions are cached into a hash table (constant lookup time) in order to avoid some of the possible repeated dissimilarity computations. Fourthly, we incorporate the option to constrain the motif search by specifying a maximum segment stretch in Algorithm 2 and VALIDPOSITION. Finally, the function that returns the non-overlapping top-k motifs employs a Boolean array of size n in order to avoid O(k2) comparisons between members of the queue (cf. [3]). Notice that we have a memory efficient implementation, as we basically only need to store z and the Boolean array (both of O(n) space), M (of O(k) space, $k << n$), and X, V, S, P, and Θ (all of them of $O(\kappa)$ space, $\kappa << n$). The aforementioned hash table (optional) can be allocated in any predefined, available memory segment.

PSO codes were implemented using the Python Deap module [27]. The Deap module provides great flexibility to code different kinds of PSO. The experiments were conducted on real and synthetic time-series data-sets. The synthetic data-set consisted of time-series data which was first filled with random values between 0 and 1. Then, motifs were implanted in each time series at random locations. The real data was collected from Thomas Reuters Eikon tool. It consisted of the End-of-Day (EOD) prices of some companies listed in S&P 500 NYSE, all belonging to the same subsector. The companies belonging to the same experiment were selected in such a way that the prices of each of those companies were useful in predicting the prices of other companies. Also, to verify this, Granger causality test was done on random pairs of these companies, and it was found that the past prices of one company are useful in predicting the future prices of another company.

The algorithm was used for forecasting in the data mentioned above. Forecasts were obtained from the proposed algorithm for some of the future timestamps and dimension. Then, forecasting was done for these timestamps and dimensions using GA based motif discovering, Vector Auto-regressive (VAR) model and random

ISSN: 2233-7857 IJFGCN

walk model. All these forecasts were compared to the actual price values. The Mean Absolute Deviation (MAD) error was calculated for each of these models which is given by

$$MAD = \frac{\sum_{i=0}^{N} |pi - ti|}{N}$$

where pi, ti, and N denote the i^{th} prediction, i^{th} target value, and the total number of predictions, respectively. Each set of experiments is characterized by the following parameters: train length, population length, number of generations, test length, distance threshold, variance, and evaluation measures. Here, train length refers to the length of multi-dimensional time series used for finding potential motifs. Population length refers to the number of individuals in the initial population supplied to the PSO. Then, the number of individuals in the population remains the same, at the end of each generation.

The number of iterations done by the PSO is equivalent to the "number of generations." In each generation/iteration, various operators like mate operator, mutation operator, selection operator are applied to the population members. In the experiments conducted, population length, train length, and the number of generations were taken to be 1000, 1000, and 1500, respectively. Test length is a parameter which is strictly greater than train length. The time-series timestamps whose value is greater than train length and less than or equal to test length are given to the proposed algorithm to predict their values. Then, the proposed algorithm predicts values only for some of the timestamps and along certain dimensions. "Distance threshold" refers to the distance value such that all motif instances of a motif are less than this distance apart from each other. Also, the prediction is only made by a motif if the distance between the initial subpart of a motif and the corresponding values in the test data-set is less than this distance threshold. If more than one motifs are within this distance threshold, then all those motifs are considered while doing forecasts. "Variance" column refers to the limit on the variance. The predicted value at any point has a variance attached to it as explained earlier. Then, the prediction is only considered for final evaluation if that variance is less than this variance value. Intuitively, "variance" limits the risk in all the predicted values by placing a threshold on the variance of the predicted value. The value of "variance" is taken to be infinite except otherwise stated. "MAD motif" refers to the MAD error between the predicted values by the proposed algorithm and the actual values. "MAD VAR" refers to the MAD error between the predicted values by the VAR prediction model and the actual values. Similarly, "MAD halves" correspond to the natıve random walk model.

In experiments with the "number of generations" equal to 1500, around 2 hours of clock time was utilized by the Python codes while running on a single core of 8 GB RAM 3.4 GHz machine. The good results of these experiments may be attributed to the choice of initial parameters which were not chosen randomly, but extensive experimentation was also not done for fixing the parameters for each

experiment. Predicting financial stock prices is a rather difficult problem, and even small success in the following experiments indicates that the proposed algorithm may become much more applicable in other domains. Also, the underlying assumption in all these experiments is that motifs are present in the datasets, which can be utilized for making predictions.

Finally, as execution time t progresses, we see that the algorithm consistently retrieves lower dissimilarities, up to the point that M ' M∗. Following the condition, we specify, this means that the distances in the motif set obtained by algorithm are not statistically worse than the ones of the true exact motif set. Overall, we believe this is an extremely competitive performance for an anytime motif discovery algorithm.

4.1. Forecasting Through Discovered Motifs

The above discussion was about the use of PSO for discovering potential motifs. Still, these motifs need to be processed and then used for forecasting. The individuals obtained through the PSO are processed mainly to identify individuals which represent the same motif. The motif instances in these individuals are combined to form a single motif. The motifs thus obtained are used for forecasting. The pseudo codes for the two different parts of the algorithm are briefly explained in the following.

This part of the code is used for identifying relevant motifs, which are then used for making forecasts.

(1) Remove individuals with the distance between motif instances greater than the “distance threshold.”
(2) From all the pool of individuals, group individuals with the same structure.
(3) From the groups obtained above, separate the two motif instances from each individual and put them all in a basket corresponding to each group.
(4) From the separate baskets obtained above, cluster motif instances which are close to each other in terms of the distance measure.
(5) Finally, all the different clusters obtained in Step 4 from all the baskets are identified and stored separately.
(6) Here, each cluster would have a minimum of two motif instances. Each of these clusters represents a motif and is used for forecasting purposes.

The motifs obtained above are used for making forecasts using the algorithm described as follows:

(1) Choose the location/timestamp in the motif structure of each motif, which would be mapped to the first predicted timestamp by this corresponding motif. Let this timestamp be called “break-point.” In the present paper, “break-point” is always chosen to be the last timestamp amongst all dimensions.
(2) Compare the past values before the break-point from each motif to the corresponding latest values in the test data-set.

ISSN: 2233-7857 IJFGCN

(3) Single-out the motifs which match best according to the distance measure chosen.
(4) Use these motifs to predict the time-series values in certain dimensions in the next timestamp. The predicted value is the mean of the specific "motif structure timestamp" values in all motif instances of all different motifs. The prediction also comes attached with a variance of prediction which is the variance of the timestamp value amongst all motif instances of all different motifs used for prediction.

5. Experimental Results and Discussions

Three experiments were conducted on simulated data-sets using proposed system and documented. This data-set consisted of two-dimensional time-series data with 2000 timestamp values, initially filled with random values between 0 and 1. Then, two multidimensional motifs were planted into the time-series data-sets. Each motif was planted at eight random locations in the time series. Four locations were chosen from training timestamps (1–1000) and four locations were chosen from test timestamps (1001–2000). Table 1 give the results of the experiments.

TABLE 1
THE TWO MOTIFS INSERTED INTO SIMULATED DATA 2

Timestamp	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Dimension1	0.96	0.55	0.68	0.83	0.52	0.54	0.19	0.15	0.80	0.90	0.28	0.69	0.62	0.72
Dimension2	0.09	0.79	0.78	0.31	0.83	0.38	0.73	0.70	0.36	0.96	0.68	0.79	0.72	0.02
Dimension3	0.16	0.70	0.78	0.98	0.58	0.72	0.04	0.14	0.47	0.41	0.79	0.96	0.33	0.61
Dimension4	0.93	0.71	0.35	0.17	0.92	0.77	0.64	0.10	0.47	0.11	0.46	0.04	0.53	0.86
Dimension5	0.49	0.34	0.06	0.30	0.17	0.55	0.99	0.12	0.88	0.38	0.08	0.83	0.12	0.69

Here, "MAD halves" give the MAD value for the natıve model which predicts each value as 0.5. Here, the average MAD value of all predictions for the proposed, GA based, VAR, and natıve models comes out to be0.0468, 0.0474, 0.2672, and 0.2725, respectively. Lesser MAD value of the proposed model shows its superiority over other models.

TABLE 2
RESULTS ON SIMULATED DATA 1

S.NO	Distance Threshold	Test End	MAD Motif (PSO)	MAD Motif (GA)	MAD VAR	MAD Halves	Number of Prediction	Mean Gain	Total Gain
1	0.06	1100	0.0750	0.0757	0.3295	0.4181	1	0.3425	0.3425
2	0.065	1100	0.0750	0.0757	0.3295	0.4181	1	0.3425	0.3425
3	0.06	1500	0.0520	0.0528	0.2597	0.2851	9	0.2322	2.09

ISSN: 2233-7857 IJFGCN

4	0.065	1500	0.0489	0.0496	0.2447	0.26	10	0.2104	2.1037
5	0.06	2000	0.0615	0.0625	0.2771	0.2692	14	0.2067	2.8936
6	0.065	2000	0.0497	0.0504	0.2197	0.2138	19	0.1633	3.1034
7	0.06	1100	0.0749	0.0757	0.3295	0.4181	1	0.3425	0.3425
8	0.065	1100	0.0749	0.0757	0.3295	0.4181	1	0.3425	0.3425
9	0.06	1500	0.0375	0.0384	0.3037	0.3456	7	0.3071	2.1497
10	0.065	1500	0.0350	0.0362	0.2794	0.3067	8	0.2704	2.1634
11	0.06	2000	0.0389	0.0398	0.3172	0.3018	13	0.262	3.4065
12	0.065	2000	0.0344	0.0353	0.2555	0.2389	17	0.2036	3.4065

5.1. Simulated Data 2

This data-set consisted of five-dimensional time series. Two multi-dimensional motifs were planted into the time-series data-set in the same way as done in synthetic data 1. Almost no predictions were made by the algorithm when the number of generations was equal to 1500. Thus, the experiments were repeated with the number of generations increased to 15,000. Then, the algorithm was able to identify the planted motifs and made relevant predictions at future motif occurrences. Table 1 gives the results of the experiments. Here, the average MAD value of all predictions for the proposed, GA based, VAR, and natıve models comes out to be 0.1458,0.1461, 0.3168, and 0.3311, respectively.

TABLE.3
RESULT ON SIMULATED DATA 2

S.NO	Distance Threshold	Test End	MAD Motif (PSO)	MAD Motif (GA)	MAD VAR	MAD halves	Number of Prediction	Mean Gain	Total Gain
1	0.06	1100	0.0000	0.0000	0.2879	0.1242	1	0.1242	0.1242
2	0.065	1100	0.0000	0.0000	0.2879	0.1242	1	0.1242	0.1242
3	0.06	1500	0.0000	0.0000	0.2879	0.1242	1	0.1242	0.1242
4	0.065	1500	0.1010	0.1018	0.3836	0.2260	2	0.1242	0.2485
5	0.06	2000	0.0000	0.0000	0.2879	0.1242	1	0.1242	0.1242
6	0.065	1100	0.1011	0.1018	0.3836	0.2260	2	0.1242	0.2485
7	0.06	1100	0.1720	0.1730	0.4446	0.3933	2	0.2202	0.4405
8	0.065	1500	0.1725	0.1730	0.4446	0.3511	2	0.2202	0.4405
9	0.06	1500	0.1970	0.1976	0.2942	0.3511	6	0.1535	0.9209
10	0.065	2000	0.1972	0.1976	0.2942	0.3785	6	0.1535	0.9209
11	0.06	2000	0.1475	0.1482	0.2924	0.3785	8	0.2302	1.8419
12	0.065	2000	0.1475	0.1482	0.2924	0.3785	8	0.2302	1.8419

It can be noted that positive “Mean gain” value for the data sets denote better performance of the proposed algorithm. Three different test ends are selected and

the corresponding MAD values against randomly selected test ends are plotted in bar graph to compare the performance of our proposed PSO based Motif discovery methodology as given below.

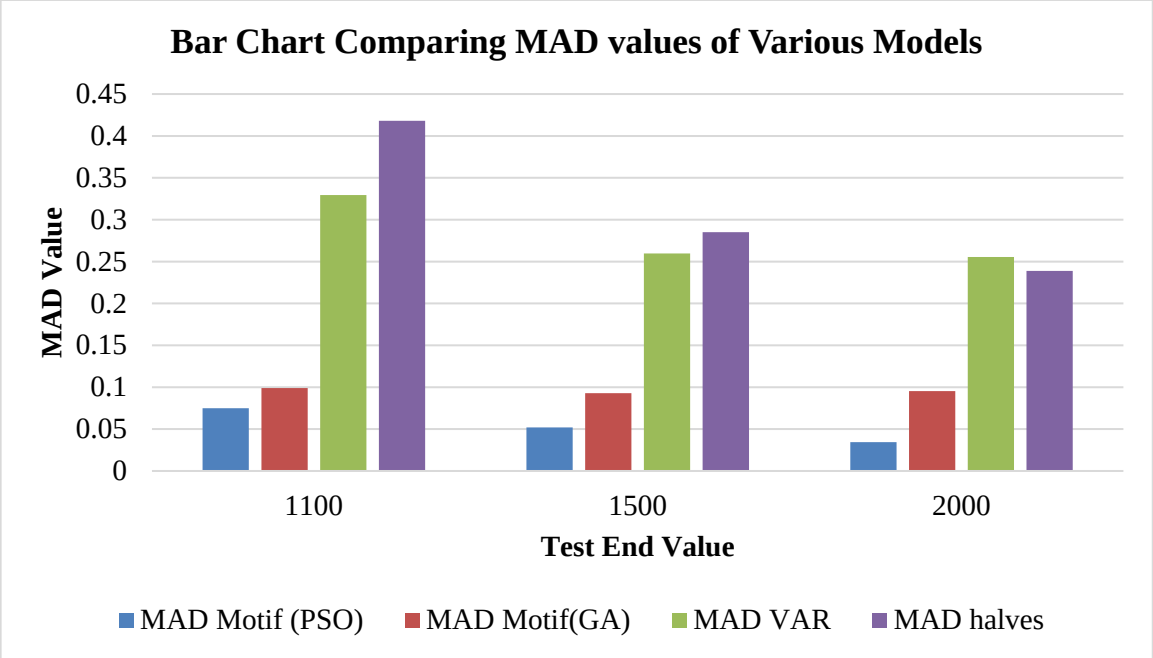

Fig. 1 Comparison of Performance through MAD of Proposed System with Various Existing Models

From Fig, 1 it is clearly noted that the lower MAD values of our proposed PSO based Motif discovery strategy depicts the higher performance when compared with various existing models like GA based Motif discovery, VAR model and the native model.

5.2. Simulated Data 3

This data-set consisted of nine-dimensional time series. As in the last two cases, two multi-dimensional motifs were planted into the time-series data-sets. Motifs were discovered by the algorithm when the number of generations was equal to 1500. Then, similarly as in the past experiments, the number of generations was increased to 15,000. A strategy to find optimal parameters is achieved. Other distance measures are tried, further improves the results. The proposed PSO motif discovery algorithm was compared with GA based motif discovery algorithms. Moreover, the proposed algorithm was tested on time-series data of other domains.

Overall, the result of our pre-analysis suggests a high degree of robustness with respect to the possible configurations. We believe that the reported stability of proposed PSO based motif against the tested configurations and data sets justifies the use of our setting for finding motifs in diverse time series coming from further application domains.

ISSN: 2233-7857 IJFGCN

6. Conclusion

In this article, we proposed an innovative standpoint to the task of time series motif discovery by formulating it as an anytime multimodal optimization problem using a new forecasting algorithm based on motifs discovered through a PSO. The forecast values were compared with forecasts from VAR predicting model, a natıve model and GA based Motif discovering. The proposed algorithm was tested on synthetic datasets. The results demonstrate the effectiveness of the proposed algorithm. Also, the proposed algorithm is very flexible and can be applied to diverse domains by modification in the definition of a motif. The proposed algorithm can be used to do multi-dimensional time-series forecasting in domains like weather forecasting, stock market prediction, economics, and energy forecasting.

References

[1] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in Proc. of the Workshop on Temporal Data Mining, 2002, pp. 53–56.
[2] Mueen, "Time series motif discovery: dimensions and applications," WIREs Data Mining and Knowledge Discovery, vol. 4, no. 2, pp. 152– 159, 2014.
[3] Mueen, A., & Chavoshi, N. (2015). Enumeration of time series motifs of all lengths. Knowledge and Information Systems, 45(1), 105-132. B. Chiu, E. Keogh, and S. Lonardi, "Probabilistic discovery of time series motifs," in Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2003, pp. 493–498.
[4] 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.
[5] J. Buhler and M. Tompa, "Finding motifs using random projections," Journal of Computational Biology, vol. 9, no. 2, pp. 225–242, 2002.
[6] N. Castro and P. Azevedo, "Multiresolution motif discovery in time series," in Proc. of the SIAM Int. Conf. on Data Mining (SDM), 2010, pp. 665–676.
[7] Y. Li and J. Lin, "Approximate variable-length time series motif discovery using grammar inference," in Proc. of the Int. Workshop on Multimedia Data Mining (MDM), 2010, p. 10.
[8] W. Wilson, P. Birkin, and U. Aickelin, "The motif tracking algorithm," International Journal of Automation and Computing, vol. 5, no. 1, pp. 32–44, 2008.
[9] Floratou, S. Tata, and J. M. Patel, "Efficient and accurate discovery of patterns in sequence data sets," IEEE Trans. on Knowledge and Data Engineering, vol. 23, no. 8, pp. 1154–1168, 2011.
[10] S. Zilberstein, "Using anytime algorithms in intelligent systems," AI Magazine, vol. 17, no. 3, pp. 73–83, 1996.
[11] Y. Tanaka, K. Iwamoto, and K. Uehara, "Discovery of time-series motif from multi-dimensional data based on MDL principle," Machine Learning, vol. 58, pp. 269–300, 2005.
[12] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan, "Detecting time series motifs under uniform scaling," in Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2007, pp. 844–853.
[13] H. Tang and S. S. Liao, "Discovering original motifs with different lengths from time series," Knowledge-Based Systems, vol. 21, pp. 666– 671, 2008.
[14] J. Catalano, T. Armstrong, and T. Oates, "Discovering patterns in realvalued time series," in Knowledge Discovery in Databases: PKDD 2006, ser. Lecture Notes on Artificial Intelligence, J. Furnkranz, T. Scheffer, ¨ and M. Spiliopoulou, Eds. Berlin, Germany: Springer, 2006, vol. 4213, pp. 462–469.
[15] Y. Mohammad and T. Nishida, "Constrained motif discovery in time series," New Generation Computing, vol. 27, no. 4, pp. 319–346, 2009.
[16] Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, "Exact discovery of time series motifs," in Proc. of the SIAM Int. Conf. on Data Mining (SDM), 2009, pp. 473–484.

[17] P. Nunthanid, V. Niennattrakul, and C. A. Ratanamahatana, “Discovery of variable length time series motif,” in Proc. of the Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011, pp. 472–475.
[18] S. Yingchareonthawornchai, H. Sivaraks, T. Rakthanmanon, and C. A. Ratanamahatana, “Efficient proper length time series motif discovery,” in Proc. of the IEEE Int. Conf. on Data Mining (ICDM), 2013, pp. 1265–1270.
[19] Y. Mohammad and T. Nishida, “Exact discovery of length-range motifs,” in Intelligent Information and Database Systems, ser. Lecture Notes in Artificial Intelligence, N. T. Nguyen, B. Attachoo, B. Trawiski, and K. Somboonviwat, Eds. Cham, Switzerland: Springer Int. Publishing, 2014, vol. 8398, pp. 23–32.
[20] Mueen and E. Keogh, “Online discovery and maintenance of time series motifs,” in Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), 2010, pp. 1089–1098.
[21] A. Ratanamahatana and E. Keogh, “Everything you know about dynamic time warping is wrong,” in ACM SIGKDD Workshop on Mining Temporal and Sequential Data, 2004, pp. 22–25.
[22] J. Serra and J. L. Arcos, “An empirical evaluation of similarity measures ` for time series classification,” Knowledge-Based Systems, vol. 67, pp. 305–314, 2014.
[23] F. S. Hillier and G. J. Lieberman, Introduction to operations research, 9th ed. New York, USA: McGraw-Hill, 2010.
[24] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: overview and conceptual comparison,” ACM Computing Surveys, vol. 35, no. 3, pp. 268–308, 2003.
[25] M. Oprisan and S. Pneumaticos, “Potential for electricity generation from emerging renewable sources in Canada,” in Proc. IEEE EIC. Climate Change Technology conf., pp. 1-9, May 2006.
[26] Y. Bo, L.Wuhua, Z. Yi, and H. Xiangning, “Design and analysis of a grid connected photovoltaic power system,” IEEE Trans. Power Electron., vol. 25, no. 4, pp. 992–1000, Apr. 2010.
[27] Femia, N.; Granozio, D.; Petrone, G.; Spagnuolo, G.; Vitelli, M., "Optimized one-cycle control in photovoltaic grid connected applications," Aerospace and Electronic Systems, IEEE Transactions on , vol.42, no.3, pp.954,972, Jul. 2006.
[28] G. Petrone, G. Spagnuolo, R. Teodorescu, M. Veerachary, and M. Vitelli, “Reliability issues in photovoltaic power processing systems,” IEEE Trans. Industrial Electronics, vol. 55, no. 7, pp. 2569-2580, Jul. 2008.
[29] F. Katiraei and J. Aguero, “Solar pv integration challenges,” IEEE Power Energy Mag., vol. 9, no. 3, pp. 62–71, May-June 2011.
[30] R. A. Mastromauro, M. Liserre and A. Dell'Aquila, “Control issues in single-stage photovoltaic systems: MPPT, current and voltage control,” IEEE Trans. Industrial Informatics, vol. 8, no. 2, pp. 241-254, May 2012.
[31] S. Jain and V. Agarwal, “A single-stage grid connected inverter topology for solar PV systems with maximum power point tracking,” IEEE Trans. Power Electron., vol. 22, no. 5, pp. 1928–1940, Sep. 2007.
[32] T. Esram and P. Chapman, “Comparison of photovoltaic array maximum power point tracking techniques,” IEEE Trans. Energy Convers., vol. 22, no. 2, pp. 439–449, Jun. 2007.
[33] Al Nabulsi, A.; Dhaouadi, R., "Efficiency Optimization of a DSP-Based Standalone PV System Using Fuzzy Logic and Dual-MPPT Control," IEEE Trans. Industrial Informatics, vol.8, no.3, pp.573,584, Aug. 2012.
[34] L. Zhang, Y. Bai, and A. Al-Amoudi, “GA-RBF neural network based maximum power point tracking for grid-connected photovoltaic systems,” in Proc. Int.Conf. Power Electron.,Machines and Drives, 2002, pp. 18–23.
[35] V. Phimmasone, Y. Kondo, T. Kamejima, and M. Miyatake, “Evaluation of extracted energy from PV with PSO-based MPPT against various types of extracted energy from PV with PSO-based MPPT against various types of solar irradiation changes,” in Int. Conf. Electrical Machines and Systems, Incheon, Korea, 2010.
[36] Shell® SP150-P photovoltaic solar module characteristics sheet. http://www.solarelectricsupply.com/Solar_Panels/Shell/SP-150.html; http://www.meet-egypt.com/Downloads/SOLARM/MONO/ SHELLsp150p
[37] A. Rico, E. R. Cadaval and M. I. M. Montero, “Power Injection Control System and Experimental Model Based on Manufacturer Characteristic Curves for a Photovoltaic Generation System,” Electrical Power Quality and Utilization, Journal, Vol. XIII, No. 2, 2007.
[38] H. Guan-Chyun and J. C. Hung, “Phase-locked loop techniques. A survey,” IEEE Trans. Industrial Electronics, vol. 43, no. 6, pp 609-615, Dec. 1996.

[39] Kennedy, J. and Eberhart, R., "Particle swarm optimization", In Proceedings of IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, 1995, 1942-1948.
[40] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Proc. 6th Int. Symp. Micro Mach. Human Sci., ., 1995, pp. 39–43.
[41] K. E. Parsopoulos and M. N. Vrahatis, Particle Swarm Optimization and Intelligence: Advances and Applications. Hershey, PA: IGI Global, 2010.
[42] V. N. Lal, M. Siddhardha, S. N. Singh, "Control of a Large Scale SingleStage Grid-Connected PV System Utilizing MPPT and Reactive Power Capability," in IEEE PES General Meeting, July 2013.
[43] A. Yazdani et al., "Modeling guidelines and a benchmark for power system simulation studies of three-phase single-stage photovoltaic systems," IEEE Trans. Power Delivery, vol. 26, no. 2, pp. 1247-1264, Apr. 2011.
[44] Y. Mohammad and T. Nishida, "Exact multi-length scale and mean invariant motif discovery," Appl. Intell., Vol. 44, no. 2, pp. 322–39, Jul. 2015.
[45] A. Mueen and N. Chavoshi, "Enumeration of time series motifs of all lengths," Knowl. Inf. Syst., Vol. 45, no. 1, pp. 105–32, Oct. 2015.
[46] Chin-Chia M. Yeh, et al., "Matrix profile VI: Meaningful multidimensional motif discovery," in Proceedings of the IEEE International Conference on Data Mining, New Orleans, LA, 2017, pp. 565–74.

ISSN: 2233-7857 IJFGCN