A Novel Time Series Representation Approach for Dimensionality Reduction

Mohammad Bawaneh and Vilmos Simon

***Abstract*—With the growth of streaming data from many domains such as transportation, finance, weather, etc, there has been a surge in interest in time series data mining. With this growth and massive amounts of time series data, time series representation has become essential for reducing dimensionality to overcome the available memory constraints. Moreover, time series data mining processes include similarity search and learning of historical data tasks. These tasks require high computation time, which can be reduced by reducing the data dimensionality. This paper proposes a novel time series representation called Adaptive Simulated Annealing Representation (ASAR). ASAR considers the time series representation as an optimization problem with the objective of preserving the time series shape and reducing the dimensionality. ASAR looks for the instances in the raw time series that can represent the local trends and neglect the rest. The Simulated Annealing optimization algorithm is adapted in this paper to fulfill the objective mentioned above. We compare ASAR to three well-known representation approaches from the literature. The experimental results have shown that ASAR achieved the highest reduction in the dimensions. Moreover, it has been shown that using the ASAR representation, the data mining process is accelerated the most. The ASAR has also been tested in terms of preserving the shape and the information of the time series by performing One Nearest Neighbor (1-NN) classification and K-means clustering, which assures its ability to preserve them by outperforming the competing approaches in the K-means task and achieving close accuracy in the 1-NN classification task.**

***Index Terms*—Time Series Representation. Time Series Segmentation, Big Data, Dimensionality Reduction, Time Series Analysis.**

I. INTRODUCTION

Nowadays, owing to the rapid advancement of the core technologies of data acquisition including the cloud data centers, cell towers, and personal computers and smartphones, notably with the emerging of the Internet of Things (IoT) technology which automates the process of data collecting and storing, massive amounts of data are being stored continuously for future data mining tasks,which could contribute to the sustainable development goals (including Good Health, Sustainable Cities, and Economic Growth) [1, 2]. The amount of available data, either created, consumed, or stored, was estimated at 4.4 zettabytes in 2013, reaching 64.2 zettabytes in 2020, and is expected to reach more than 180 zettabytes in 2025 [3, 4, 5]. Recently, Wu et al. [6] have studied the relation between greening and big data by introducing the issues of big data from the greening point of view. They have identified three main domains which require greening. First, big data acquisitions necessitate significant energy consumption for data collecting as well as data transfer through networks. Second, storing massive data has called for more advanced technologies that are inefficient in terms of energy and resources. Third, the process of analytics of big data is usually computationally expensive, consuming time, energy, and resources. As a result, a dimensionality reduction technique can contribute to greening big data storage and analytics by conserving storage space while also reducing the computational complexity of the data analytics process.

A significant amount of the generated data are streaming data which is also known as time series data. Time series is a sequence of observations, where each observation is recorded sequentially with time [7]. Time Series data are used in various domains including finance and stock market [8, 9], voice recognition [10], online signature verification [11], failure prediction in high performance computing and cloud systems [12], earthquake forecasting[13], weather prediction [14], and intelligent transportation systems [15]. Consequently, an enormous amount of data are generated daily and requires special memory management. As previously stated, such massive data has two major consequences. First, a significant quantity of memory must be provisioned, consuming energy and resources. Second, because of the inherited high computation complexity, processing and analyzing high-dimensional data is challenging, making it difficult to analyze the time series in its raw form. To achieve that, many researchers have investigated time series representation approaches, with various ways offered to minimize time series high dimensionality by expressing the time series in a new representation form in a lower dimension space [16]. However, a common key concept for applying valuable time series representation is that the new representation of the time series must include the original characteristic features in order to preserve the important information of the raw time series (such as local trends information and basic data distribution). Furthermore, these features must be acquired while keeping the new representation as simple as possible. Moreover, because time series data comes from various domains and represents distinct behaviors, the representation approach should be applicable to numerous types of time series datasets. Therefore, the time series representation approach should be general and applicable to any dataset to be used as a preprocessing step. As a result of these transformation criteria, storage space will be saved and further processing and analysis of data will be accelerated. In this paper, we adopt these criteria to propose an effective offline time series representation approach termed

Mohammad Bawaneh is with the Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary, e-mail: mbawaneh@hit.bme.hu.

Vilmos Simon is with the Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest Uni- versity of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary, e-mail: svilmos@hit.bme.hu.

DOI: 10.36244/ICJ.2022.2.5

Adaptive Simulated Annealing Representation (ASAR). The proposed approach treats the time series representation as an optimization problem, with the aim of retaining the time series shape while lowering dimensionality. Moreover, because it is focused on tracking the local trends of the time series, the proposed approach is able to transform any sort of time series data with diverse characteristics and behavior.

Transforming the time series into a new representation has several advantages. When it comes to extracting information from time series data, several data mining tasks, such as classification and clustering [17, 18], may be used to analyze the time series data. As a consequence, with the requirement to measure similarity and examine historical time series data in order to apply effective classification or clustering tasks, transforming the time series as a preprocessing step would give minimal computing complexity and hence speedier results. Furthermore, certain similarity metrics may get skewed due to the distortion in the raw time series. As a result, changing the time series while retaining its fundamental characteristic features overcome this issue as well [19].

Time series representation methods that have been proposed in the literature have several flaws; we will discuss them in detail in the next section. Some of these methods transform the time series into symbolic form and by this lose the original structure, which makes it impossible to restore the shape of the time series. In addition, some methods lose the local trend information, which is crucial information for similarity measuring of time series data. Some variants of these have been proposed to include the trend information; however, this comes with a cost of insufficient compression ratio, which is considered one of the main objectives when representing time series in a new form. In this paper, ASAR is proposed to overcome the shortcomings of these methods by introducing a shape-based representation of time series. ASAR keeps the new form of the time series as simple as possible by transforming the data into a lower dimension but with the same shape as the raw time series together with the same data distribution. This way, it addresses the issue of keeping the original structure while compressing the data. Moreover, by preserving the shape of the time series, the local trend information is preserved, with no cost of including additional information in the new representation. The proposed approach ASAR is assessed and compared with some approaches from the literature in this paper by measuring the Compression Ratio (CN) to determine which approach saves the most memory. Furthermore, classification and clustering tasks are used to assess ASAR's capacity to maintain time series information (i.e., such as the local trends) and to demonstrate the process acceleration feature. The following are the key contributions of this paper:

- The new representation of time series ASAR can significantly reduce dimensionality while retaining the shape of the time series. This conserves storage space without losing the information required for future data mining operations. Moreover, the high compression ratio that can be achieved by ASAR accelerates future data mining operations.
- The ASAR approach views time series representation as an optimization issue with the objective of maintaining the raw time series shape. This is achieved by tracking local trends in the raw time series and expressing these trends by the least number of segments. As a result, ASAR has no restrictions on the type, shape, distribution, or source domain of time series data.

This paper is organized as follows. Section 2 includes a review of prior similar studies from the literature. The proposed approach is explained in detail in Section 3. Section 4 presents the findings of the experimental analysis. Finally, in section 5, the paper's conclusion is provided.

II. Related Works

In the last two decades, the applications domains that apply time series analysis have grown tremendously. In addition, the rapid advancement in data acquisition technologies offered an enormous amount of data which in turn could be mined to form significant knowledge. As a consequence, numerous time series representation methods were developed to overcome the challenges of the data's high dimensionality [20, 16]. Aghabozorgi et al. [19] classified the time series representation methods into four main categories: data adaptive, non-data adaptive, model-based, and data dictated representation methods. This section provides a brief overview of these categories and the most significant approaches presented in the previous two decades.

In data adaptive methods, the segmentation of the time series is done with varied length segments. Singular Value Decomposition (SVD) was one of the earliest methods proposed in time series dimensionality reduction [21]. It can be used to represent multivariate time series data. SVD deals with the multivariate time series as an $(m*n)$ matrix. It applies a space rotation process to the best least-squares fit direction by factorizing the matrix into three other matrices ($A = U\Sigma V^T$). U is $m*m$ unitary matrix, Σ is $m*n$ rectangular diagonal matrix with diagonal non-negative elements called singular values, and V^T is $n*n$ unitary matrix. The dimension of the matrix is reduced by removing the least significant singular values in Σ and the corresponding entries in U and V^T. The disadvantage of SVD is that it has high computation complexity $O(mn^2)$. Years later, the Adaptive Piecewise Constant Approximation (APCA) was proposed [22]. APCA segments the time series into constant segments but with varying lengths. The new representation is simply the records of the endpoints for each segment with the mean value of the segment in the original raw time series. With a computation complexity of $O(n)$, APCA is a faster method than SVD. However, a significant disadvantage of APCA is that it loses the trend information since two segments with two different trends may have the same mean values. Gullo et al. [23] have proposed the Derivative time series Segment Approximation (DSA) representation model. DSA model transforms the raw time series into the derivative estimation by computing the first derivative of each sample. Then it segments the derivative estimation into variable-length segments, where the breaking criterion is that the points that have close slopes (close first derivative values) are in the same segment. In other words, the segment keeps expanding while the absolute difference between the new sample and the mean value of the previous samples within the segment is less than a certain threshold. Finally, the new representation is formed by pairs representing the segments. Each pair consists of the timestamp of the last point in the segment, and an angle demonstrates the average slope of this segment.

Non-data adaptive methods segment the time series with fixed-length segments. One of the widely used time series representation methods under this category is the symbolic representation called the Symbolic Aggregate approXimation (SAX) [24, 25]. SAX normalizes the time series to a zero mean distribution and standard deviation of 1, keeping the different time series within the same offset. Then the time series is transformed into the Piecewise Aggregate Approximation (PAA) representation [26], which in turn reduces the dimensionality. PAA divides the time series into a number of equal-sized frames. Then for each frame, the mean value of the points within the frame is calculated, and finally, the sequence of the mean values of all frames will be the new PAA representation. As a result of the normalizing process, the time series follows a Gaussian distribution. In the next step, the authors divide the time series into equal-sized areas under the curve of the Gaussian distribution (the same size as the PAA representation's frame). Finally, they assign a symbol for each area which will be later assigned for all samples within this area. Based on the sequence values obtained by the PAA representation, the time series is transformed into a sequence of symbols called a word. Similar to APCA, SAX has a drawback of losing the trend information since segments have different trends but similar slope values will be assigned by similar symbols. There are several variants that have been proposed as SAX extensions. Lkhagva et al. [27] have proposed to use the minimum and maximum values within the segment in addition to the mean value to overcome the drawback of SAX. However, this will triple the dimension reduced by SAX. Another Variation is proposed by Sun et al. [28] in their SAX-TD method. SAX-TD adds the trend information of each segment to the SAX representation by calculating the distance between the segment's ending points which they called the trend distance. Consequently, the dimension is double that reduced by SAX. Another extension, SAX with Standard Deviation (SAX_SD), has been proposed [29]. The authors improved SAX by adding the standard deviation feature in addition to the mean value in order to study the spread of the values within the segment and to improve the similarity measure. In [30], Multivariate Symbolic Aggregate Approximation (MSAX) was proposed to represent multivariate time series data. Some applications contain more than one variable explaining the same behavior. Therefore, MSAX integrates the information of the different time series in one symbolic representation. MSAX first checks the dependency between the variables. If they are independent of each other, the data are normalized. However, in the case of dependent variables, a linear transformation must be applied. Then, all the time series in the matrix are represented using the PAA method. Last, discretization is applied resulting in a symbol matrix. As a final step, the symbols in the matrix are transformed into a sequence of symbols with a length equal to the columns, where each entry is represented by compressing the symbols in all rows (all the time series) in the corresponding column.

Model-based representation methods transform the time series stochastically. Time Series Bitmaps belongs to this category [31]. Time Series Bitmaps uses the time series extracted features and their frequencies to color a Bitmap. This visualization of the similarities between time series offers the users a fast discovery of the clusters, classes, anomalies, and other shape-based tasks. This is done by first transforming the continuous time series into discrete time series by applying SAX. Then, the frequencies of the sub-words in the SAX representation are counted, where the desired level of recursion defines the length of the sub-word. These frequencies are mapped into the corresponding pixel of the grid, where the grid contains pixels that represent all possible sub-words based on the desired level. The frequencies are normalized by dividing them by the largest value to handle the length variety between the time series. The final step is the color mapping of these frequencies into the grid, which offers the ability to compare the time series. It is not recommended to use bitmaps representation for a single time series as it does not offer any information. Another drawback of Bitmaps is that the structure of the raw time series is hidden and cannot be captured.

In data dictated methods, the compression ratio is not defined in advance where it is dependent on the raw time series behavior. The Clipped representation is an example of this category [32, 33]. Clipped represents the time series as binary values. The raw time series' samples above the population's mean will be represented by 1, whereas those below the mean will be represented by 0. The new binary representation is compressed to a new sequence that contains the lengths of the subsequences with the same value. It is unnecessary to mention the sample value in addition to the length as a pair because it is a binary representation. Hence, including the first value is enough where the rest of the values will be only toggling between 0 and 1. Zhan et al. [34] proposed the Feature-based Clipped Representation (FCR). FCR divides the time series into equal-length segments. Then it finds the trends' turning points within each segment and their corresponding importance indices using the method presented in [35]. The turning points are then chosen based on their importance and converted into binary values using the clipped representation, which will be compressed to a new sequence that contains the lengths of the subsequences with the same value. The clipped representation here compares the values to the segment's mean instead of the population's mean. Another example of this category is the symbolic representation of the Fragment Alignment Distance (FAD) method [36]. FAD estimates the derivative of time series using the DSA method [23]. This derivative estimation contains the trend information. After that, FAD converts this derivative sequence into a symbolic sequence R by setting a threshold and comparing it with the derivative estimation value of each sample. If the value is less than the threshold, the point has a small change compared to the previous point, and they will be assigned with the same symbol. However, if the value is bigger, the point has a big change compared to the previous point, and so a different symbol will be assigned for this point. Finally, FAD transforms the resulted symbolic representation series R into feature series consisting of pairs of values. Each pair represents the symbol of a similar subsequence and the length of this subsequence. Another method that belongs to this category was proposed in the paper [37] which is called Adaptive Particle Swarm Optimization Segmentation (APSOS). APSOS deals with the time series segmentation as an optimization problem. The goal of the optimization is to minimize the error function between the raw time series and the segmented time series. To find the samples that best segment the series, they have adapted the particle swarm optimization algorithm to find the best segments' endpoints. APSOS is able to capture the trend information of the time series; however, it has high computation complexity $O(n^2)$, which makes it difficult to use with the high daily acquired streaming data.

The proposed approach in this paper is part of the data dictated methods since it is based on tracking the local trends in the raw-time series, and consequently, the compression ratio is dependent on the time series behavior. It is inspired by the APSOS approach by dealing with the time series segmentation as an optimization problem. The following section introduces the proposed approach in detail.

III. Adaptive Simulated Annealing Representation (ASAR)

A brief summary of the significant approaches proposed in the literature for representing time series was introduced in the previous section. These approaches suffer from different drawbacks. Some approaches are time-consuming due to the high computational complexity required to create the representation of the raw time series. On the other hand, some of those solutions with low computational complexity failed to capture the local trends information. Furthermore, some approaches do not offer a high enough compression ratio, where the high compression ratio is one of the crucial features of a time series representation; therefore, it became the key objective of our work. The Adaptive Simulated Annealing Representation (ASAR) is introduced in this paper to overcome these issues. ASAR's objective is to represent the time series in a new form to achieve a high compression ratio, this way saving the storage space and at the same time preserving the shape of the time series, which will keep the essential features and prevent information loss. Inspiring by the APSOS approach [37], ASAR deals with the time series representation as an optimization problem. This optimization aims to find the instances in the raw time series that can describe the shape in the possible best way, ignoring the rest of the instances. In the following subsection, we define the time series representation as an optimization problem.

A. *Formulating Time Series Segmentation as an Optimization Problem*

Each time series contains several local trends, forming a time series shape. For example, two time series may have the same shape, which means that they follow the same local trends. However, the time of occurrence of the local trends does not have to be the same. As mentioned earlier, ASAR is proposed to reduce the time series dimensions while maintaining the time series shape. For this purpose, a heuristic algorithm can be utilized. Heuristic algorithms are optimization algorithms that can find an approximated optimum global value for a particular function. Accordingly, in order to use a heuristic algorithm to apply time series representation, the time series representation must be formulated first as an optimization problem with the objective of reducing the time series dimensions while preserving the shape. Let us assume that X is a time series of length n and is defined as:

$$X = \{X_1, X_2, ..., X_n\}$$

Our target is to find a new time series R, representing X time series shape with a reduced dimensionality. The new representation R can be defined as follows:

$$R = \{R_1, R_2, ..., R_k\} \tag{1}$$

where $k \ll n$. To illustrate, Figure 1 shows the objective of the proposed approach using a synthetic time series. The length of the raw time series (depicted by the blue line) is 1000, whereas it can be reduced to 22 samples while preserving the shape of the raw time series (the orange line). It must be noted that this is just an illustrative example of the approach's objective, not the result of the ASAR's transformation. The segment from the time series R is defined

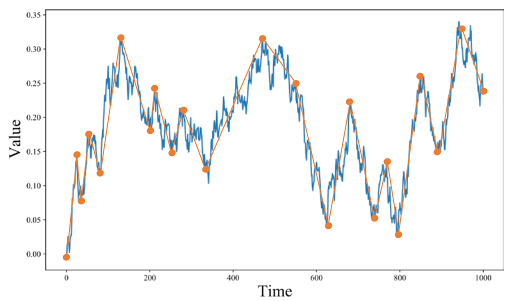

Fig. (1) An illustration example of the time series segmentation result, note that the blue line represents the raw time series, while the orange one represents the new time series representation.

as the line connecting two consecutive points in the new representation. Hence, R will contain $k-1$ segments. This segment is obtained by recording two timestamps from the raw time series as endpoints and neglecting the timestamps between them. However, the segment may still be used to estimate the value for each timestamp of the raw time series (even the neglected ones). This estimation can be specified by the line equation (the line connecting the two endpoints). Let us assume that RX represents a time series for the estimated values of the raw time series from the point of view of the new representation R, Then, the RX_i's approximate corresponding value of X_i can be computed as follows:

$$RX_i = \frac{1}{(e-s)}[(i-s)X_e + (e-i)X_s] \tag{2}$$

where i is the index of the point that will be estimated, s is the starting point of the segment where i locate, and e is the endpoint of this segment (s,i, and e are timestamps from the raw time series). The proof is as follows:

Based on the straight line equation:

$$RX_i = m \times i + b \tag{3}$$

where m is the slope of the line, and b is the y-intercept. The slope m is calculated as follow:

$$m = \frac{(X_e - X_s)}{(e - s)} \tag{4}$$

now substitute 4 in 3:

$$RX_i = \left(\frac{(X_e - X_s)}{(e - s)}\right) \times i + b \tag{5}$$

Hence, the y-intercept b can be defined as:

$$b = RX_i - \left(\frac{(X_e - X_s)}{(e - s)}\right) \times i \tag{6}$$

by using the point e:

$$b = X_e - \left(\frac{(X_e - X_s)}{(e - s)}\right) \times e \tag{7}$$

now substitute 7 in 5:

$$RX_i = \left(\frac{(X_e - X_s)}{(e - s)}\right) \times i + X_e - \left(\frac{(X_e - X_s)}{(e - s)}\right) \times e \tag{8}$$

by expanding out the brackets we get equation 2:

$$RX_i = \frac{1}{(e - s)}[(i - s) \times X_e + (e - i) \times X_s]$$

The approximating values ($R_i \quad \forall\ s < i < e$) will be used to calculate the Mean of Squared Errors (MSE) for each segment between the raw time series and the corresponding approximate values in the new representation. MSE is the average of the squared errors and is defined as:

$$MSE(s, e) = \frac{1}{e - s + 1} \times \sum_{i=s}^{e} (X_i - RX_i)^2 \tag{9}$$

MSE serves as an indicator of how much the segment that has a starting point s and an endpoint e aligns the samples in the range (s, e) from the raw time series. The MSE values then will be compared to indicate the superiority between different possible segments to represent the data. In other words, ASAR computes the MSE for different segments (same starting point but different endpoints) to find the best segment that has the minimum MSE among them, which points to the best alignment between the new segment and the corresponding samples from the raw time series. For simplifying the implementation and for the purpose of direct calculations, equation 2 is substituted to equation 9, which gives us the following formula:

$$MSE(s, e) = \frac{1}{e - s + 1} \times \sum_{i=s}^{e} \left(X_i - \frac{(i - s)X_e}{(e - s)} - \frac{(e - i)X_s}{(e - s)}\right)^2 \tag{10}$$

The proposed ASAR algorithm aims to segment the raw time series based on the local trends so that the points that follow the same trend will be covered by one segment. However, once the trend changes significantly, a new segment should be used to cover the next points in the raw time series. Moreover, since different segments can represent a trend, the selected segment should be the best-aligned one with the points that follows the same trend. From this point of view, we have defined our objective function to find the best segment's endpoint as follows:

$$e = \underset{state}{\operatorname{argmin}}\ MSE(s, state) \tag{11}$$

In the first segment, the starting point will be the first point in the raw time series. So basically, this objective function searches for the endpoint of the segment. Once the optimal endpoint is found and recorded, our method starts to search for the next endpoint, considering the previous endpoint as the new segment's starting point.

B. Simulated Annealing for ASAR

The Simulated Annealing (SA) heuristic algorithm [38] is utilized in our method to find the new representation of the time series which can maintain its shape. SA is chosen as the heuristic algorithm in this paper due to its ability to overcome the issue of being stuck at some local optima during the global optimum solution search process. In addition, SA is a robust and general algorithm as it makes no constraints on the type of data. This allows for proposing a general time series representation algorithm that can be applied for streaming data coming from various domains. Moreover, in case the time for the searching process is due to end, SA returns the best-known solution. In other words, there is always a best-known solution even if the time was not enough to complete the search process.

The general principle of SA and its use in this paper can be clarified as follows: the computational optimization by SA is a probabilistic technique with the objective of finding the global optimum value for a particular function within a solution search space. The algorithm starts searching for the optimum value by moving between the possible solutions randomly. However, each transition is evaluated, and only those who have a high transition probability will be considered. The probability is controlled by two parameters, a change in the system Energy ΔE, and a system temperature T. The solutions resulting in smaller system energy are better than those solutions with greater energy. Therefore, SA accepts the solutions that result in smaller system energy. However, worse solutions may still get accepted with a certain probability controlled by the change in the system energy and temperature. The system temperature is used to reduce the likelihood of accepting worse solutions as the solution search space is investigated. Accepting a worse solution provides a more thorough search for the ideal global solution. Simulated Annealing operates in the following manner. The temperature gradually declines from a positive starting point to zero. At each time step, the Simulated Annealing randomly picks a solution similar to the current one, evaluates its quality, and progresses to it based on the probability of picking better or worse solutions.

The time series representation has been formulated as an optimization problem in order to apply the Simulated Annealing algorithm (see equation 11). The solution search space has been adapted in this study to cover only a part of the raw time series rather than the entire space. In other words, because the time series contains several local trends, only a window of the raw time series is inspected each time in order to locate the best segment. As a result, the window of the raw time series is investigated each time to locate the optimal endpoint that depicts a local trend. The discovered endpoint is then utilized as the starting point for the next segment search, with the same window size serving as the search space. As the entire solution space (the entire raw time series) is not required in each segment inquiry, this reduces the computing complexity of ASAR. In addition, for the reason that any local trend consists of consecutive samples, the transition between states in the Simulated Annealing is made incremental rather than random in this paper. It should be emphasized that the window size does not determine the size of the segments but rather the size of the solution search space (i.e., the search space for the segment endpoint). As a result, ASAR is not sensitive to this window size setting because it is utilized to reduce calculation complexity. We recommend setting a large enough window size that can suit the majority of the local trends (i.e., the window size is larger than the length of the majority of the local trends). Nonetheless, not having a large enough window size does not cause a problem in preserving the shape of the time series; however, local trends with lengths greater than the window size will be represented by more than one segment.

1) Energy Function: The Mean of Squared Errors defined in equation 10 is used as the energy function in this paper. The change in the energy function when moving from a state ($state$) to a state ($state + 1$) is defined as follows:

$$\Delta E_{state} = MSE(s, state - 1) - MSE(s, state) \tag{12}$$

2) Cooling Schedule: The other parameter which controls the acceptance probability of the new $state$ is the temperature. The temperature should decline gradually, which is controlled by the cooling schedule. In this paper, the cooling schedule is defined linearly as follows:

$$T_{state} = \alpha \times T_{state-1} \tag{13}$$

where T is the system temperature, and α is cooling parameter. In this paper, we adopt the definition provided in the paper [39] to calculate the α value.

$$\alpha = \left(\frac{T_{initial}}{T_{final}}\right)^{\left(\frac{1}{w}\right)} \tag{14}$$

where w is the window size used for the search space. The final temperature is typically set close to 0. However, the initial temperature is problem-dependent and has to be studied and set based on the use case. This definition of α ensures that the temperature starts with a high value and decreases gradually towards the low final value. In other words, at the beginning of the process, the temperature is set to be high, resulting in a high probability of accepting states. However, as the process gets through, the temperature value decreases, resulting in a lower probability of accepting solutions.

3) Acceptance Probability: The probability function used in this paper is defined as:

$$Pr_{state} = \begin{cases} 1 & \Delta E_{state} > 0 \\ e^{\frac{\Delta E_{state}}{T_{state}}} & \Delta E_{state} < 0 \end{cases} \tag{15}$$

Pr_{state} contains two different factors; the system temperature T_{state}, and the change in energy function ΔE_{state}. The probability is directly proportional to the system temperature and the change in energy. Table I explain the effects of these factors.

TABLE (I)
THE EFFECTS THAT EACH FACTOR CAN CAUSE ON THE PROBABILITY VALUE.

Factor	Magnitude	Effect on probability
T	High	High
	Low	Low
ΔE	High	High
	Low	Low

4) Acceptance Criteria: SA searches for the segment that will keep the change in the energy positive as it indicates that the new MSE is better than the previous one, meaning that it is a good move. However, in case of a bad move, i.e., the energy is negative, the SA may still make the move but with a certain probability to ensure finding the global minima where the segment's MSE reflects the best segment alignment with the raw data. Once the state is rejected, the previous state will be used to record a new value in the new representation R as it indicates the best endpoint that makes the best segment alignment. This is expressed as follows:

$$state = \begin{cases} Accepted & \Delta E_{state} > 0 \\ Accepted & Pr_{state} > rand(0,1) \\ Rejected\,(R_c = X_{state-1}) & Pr_{state} < rand(0,1) \end{cases} \tag{16}$$

where c is the timestamp of the new record in the new representation, starting from 1 and increasing by 1 with each new record until it reaches the new representation length k. To summarize, the ASAR approach searches for the points that can explain the local trends and neglects the rest to reduce the data dimensions while preserving the shape of the time series. Table II shows the ASAR algorithm's pseudocode.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

In this section, the proposed approach is compared to three significant approaches from the literature, the PAA, the SAX, and the FAD approaches. Since representation approaches are not direct methods to extract information, a validation experiment is designed in this paper to test the ability of ASAR to preserve the information of the raw time series by applying similarity search and detection tasks using time series classification and time series clustering methods. In addition, to test the effectiveness of ASAR, it is compared with the competing approaches in terms of the compression ratio in order to test the storage space saving supremacy. Moreover, the time performance of each approach for applying classification or clustering tasks is compared to demonstrate which approach accelerates the data mining process the most.

TABLE (II)
THE PSEUDO CODE IMPLEMENTATION OF THE PRO-POSED ASAR ALGORITHM.

```
Algorithm: Adaptive Simulated Annealing Representation (ASAR)
Input: Time Series X = (X_1, X_2, ..., X_n)
Output: Time Series R = (R_1, R_2, ..., R_k)

1. k = 1, s = 1, state = s + 3, R_k = X_1
2. For T = T_max:T_min do:
        If (ΔE_state > 0):
            state = state + 1
        Else if (Pr_state > rand(0,1))
            state = state + 1
        Else
            Break
        End if
   End for
3. k = k + 1
   R_k = X_{state-1}
   s = state - 1
   Go to 2
```

A. Assessment algorithms

In order to evaluate the ability of ASAR to maintain the time series information, it is tested and compared with other approaches based on similarity search tasks. Standard classification and clustering algorithms are applied as the paper's objective is not the similarity search tasks themselves but the dimensionality reduction. Therefore, we chose to apply the well-known algorithms, One Nearest Neighbor (1-NN) for the classification and K-means for the clustering.

1) One Nearest Neighbor classification (1-NN): In K Nearest Neighbor (K-NN) classification [40], the tested instance is classified based on the classes of the closest k instances. In other words, the algorithm checks the class of closest k instances (using a similarity metric) and applies majority voting to predict the tested instance class. Since the classification task is not the main objective of this paper, the One Nearest Neighbor (1-NN) has been chosen in this paper as the most basic, straightforward, and standard method to check the similarity between time series. Moreover, it provides a fairer comparison as it does not require parameter tuning, leading to unbiased results. We use 50% of the dataset under study as a training dataset and 50% as a testing dataset.

2) K-means Clustering: One of the well-known and most used clustering algorithms is the K-means algorithm [19]. It is a partitioning clustering approach that separates the data into k clusters, intending to minimize the distance between the instances and the cluster center and maximize the distance between the instances from different clusters. In this paper, K-means is used to provide a more thorough analysis of the ability of ASAR to preserve the shape of the time series by testing its accuracy of clustering the data based on the time series shape. Since K-means requires setting the number of clusters k in advance, we use the actual number of the classes of the labeled dataset under study.

B. Assessment Criteria

To assess the classification and clustering results of the competing approaches, the F-measure (or F-score) [41] is used in this paper. It is an accuracy measure determined using the test's precision and recall. The precision is the number of the true predicted positives divided by the number of all predicted positives. The recall is the number of the true predicted positives divided by the number of actual positives. The F-measure is defined as follows:

$$F\text{-}measure = \frac{2\times Precision \times Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP+FN)} \quad (17)$$

The compression ratio (CR) in the new representation of the time series is computed to assess the degree of dimensionality reduction achieved by the competing approaches and demonstrate the superiority in storage space savings. The compression ratio explains the proportion of the reduction in the length of the time series to the original length and can be calculated as follows:

$$CR = \frac{n-k}{n} \times 100\% \quad (18)$$

where n is the raw time series length, and k is the length of the time series in the new representation form.

C. Dataset Description

The UCR Time Series Classification Archive [42] is employed in this paper to evaluate the proposed approach. The data in the archive is z-normalized and originates from several domains, providing variety for the time series form. The experiment in this study has been applied to eight datasets. The datasets are chosen to have varying lengths and to exhibit a variety of characteristics to evaluate ASAR's ability to handle diverse types of time series. The data in the UCR repository is divided into two sets: training and testing. In this experiment, we choose the testing set since its bigger size provides for more robust results. Table III shows the dataset information used in the experiment.

TABLE (III)
THE INFORMATION OF THE SELECTED DATASETS FROM THE UCR ARCHIVE.

Name	Dataset Size	Time Series Length	Number of Classes
HandOutlines	1000	2709	2
StarLightCurves	1000	1024	3
Lightning-2	60	637	2
OSU Leaf	200	427	6
ShapeletSim	180	500	2
WormsTwoClass	180	900	2
Yoga	1000	426	2
Trace	100	275	4

D. Parameters Tuning

One or more parameters must be tuned for the competing solutions. We used the parameter sweeping technique to tune the parameters as in the original articles [36, 26, 25]. FAD requires adjusting two parameters: the threshold used to determine symbol changes and the number of symbols. The threshold is changed by 0.01 increments between 0 and 0.2, whereas the number of symbols is adjusted by one increment between 3 and 7. A threshold of 0.02 with a number of symbols of 5 has shown the best performance for the classification and the clustering tasks.

Since the time series is divided into equal-sized frames, PAA requires tuning the frame size. The frame size is adjusted by one increment between 2 and 100. A frame size of 8 has shown the best classification and clustering results.

In SAX, two parameters must be tuned: the frame size and the alphabet size (the number of symbols to represent the data). As in PAA, the frame size is adjusted by one increment between 2 and 100, while the alphabet size is adjusted by one increment between 2 and 10. The best performance has been shown using a frame size of 4 with an alphabet size of 8.

For the proposed approach ASAR, two parameters need to be tuned. The first parameter is w, the window size used for the search space. The second parameter is the initial temperature (we use 0.1 for the final temperature as it must be close to 0). We adjust w by one increment between 10 and 150. We must note that this window size does not generate equal-sized frames but rather a search space for the segment's endpoint. Regarding the initial temperature, it is adjusted by 0.5 increments between 0.5 and 5. A window size of 25 with an initial temperature of 2.5 has shown the best performance in the experiment.

E. Time Series Representation Using ASAR

As previously stated, ASAR is intended to track time series local trends. Its objective is to reduce the time series' dimensions while preserving its shape. This was accomplished by retaining the data points that best explain the time series and its trends and eliminating the others. We show an example from each dataset to demonstrate the ASAR's ability to represent time series with this objective. Figure 2 shows the raw time series of each sample on the left and its ASAR representation on the right. It can be seen that ASAR representations are totally following the raw time series shapes with much lower lengths, which confirms the fulfillment of the paper's objective.

F. Effectiveness Evaluation

The proposed approach is compared to FAD, PAA, and SAX. In this research, four distinct assessments are used. To assess ASAR's efficiency in preserving the shape and time series information, we use 1-NN classification and k-means clustering. The compression ratio (CR) is used to measure the efficiency of decreasing dimensions to conserve storage space (see equation 18). Furthermore, the efficacy of ASAR in facilitating accelerated data mining processes is presented by comparing the time required to execute classification and clustering tasks using raw time series and the competing representation techniques. To guarantee a robust outcome, the F-measure (equation 17) for 1-NN classification and k-means clustering is calculated by averaging 50 experimental runs. Likewise, the time necessary to complete these activities is averaged across 50 attempts. As mentioned in section II, ASAR belongs to the data dictated methods, which means that the compression ratio is not previously defined and fixed for all time series but is dependent on the time series behavior. Therefore, CR here indicates the dataset's average compression ratio (the average of the compression ratio for all time series in the dataset). Since the classification and the clustering tasks are not the main objectives of this paper, we calculate the relative results of the F-measure for the representation approaches and the raw time series. This allows us to see how effective is a representation method compared to the case when the time series is used in its raw form. Table IV shows the F-measure relative results for the 1-NN classification. The table shows that ASAR comes in third place after PAA and SAX and performs better than FAD.

On the other hand, Table V shows the F-measure relative results for the K-means clustering task. ASAR achieved the best performance among the competing methods. It has increased the clustering accuracy by 8%.

When it comes to the main objective of this paper, ASAR has achieved the maximum compression ratio (equation 18) among the competing representation approaches. Table VI shows the compression ratios and the average compression ratio for each representation approach as a percentage (%).

This high compression ratio accelerates the data mining process since the training dataset dimension is reduced. To show and compare the efficiency of the dimensionality reduction, we have calculated the time needed to perform the 1-NN classification and the K-means clustering tasks using the data representation provided by the various methods. The experiments in this paper have been conducted on a platform with an Intel(R) Xeon(R) Silver 4215 CPU with clock speed of 2.50GHz and 2.49 GHz (2 processors) with 8 GB RAM, running Windows 10 (64-bit). Python programming language was used to implement all the approaches. Table VII shows the performance time in seconds. It can be seen that by using ASAR, the data mining process can be accelerated the most, which proves its advantage in fast information extraction besides the storage space saving.

G. Results Discussion

According to Figure 2, it is pretty clear that ASAR is able to fulfill the objective of this paper by significantly reducing the dimensionality while preserving the time series shape. This confirms that ASAR is an applicable and reliable dimensionality reduction approach for the data mining tasks (such as time series classification, clustering, and anomaly detection) that considers the original shape of the time series as a crucial feature . All examples show clearly that the ASAR representation follows the local trends of the raw time series except for the ShapeletSim dataset due to the high frequency of trends reversal. However, tables IV and V show that using ASAR has resulted in no significant information loss in the ShapeletSim dataset by achieving an accuracy of 94% and 100% in the classification and clustering tasks, respectively.

To test the information preservation capability of the representation methods, the 1-NN classification and the K-means clustering were employed in this paper. The results for the 1-NN classification (table IV) locate ASAR in the third place among the competing approaches. The average F-measure of ASAR is 96% of the accuracy obtained by using the raw time series. This shows that ASAR preserves the time series information. On the other hand, Table V shows that ASAR has achieved the highest accuracy among the competing approaches. It outperformed the results by 8% when compared to the raw time series used for K-means clustering. While the main objective of this paper is dimensionality reduction, Table VI shows the compression ratio obtained using each

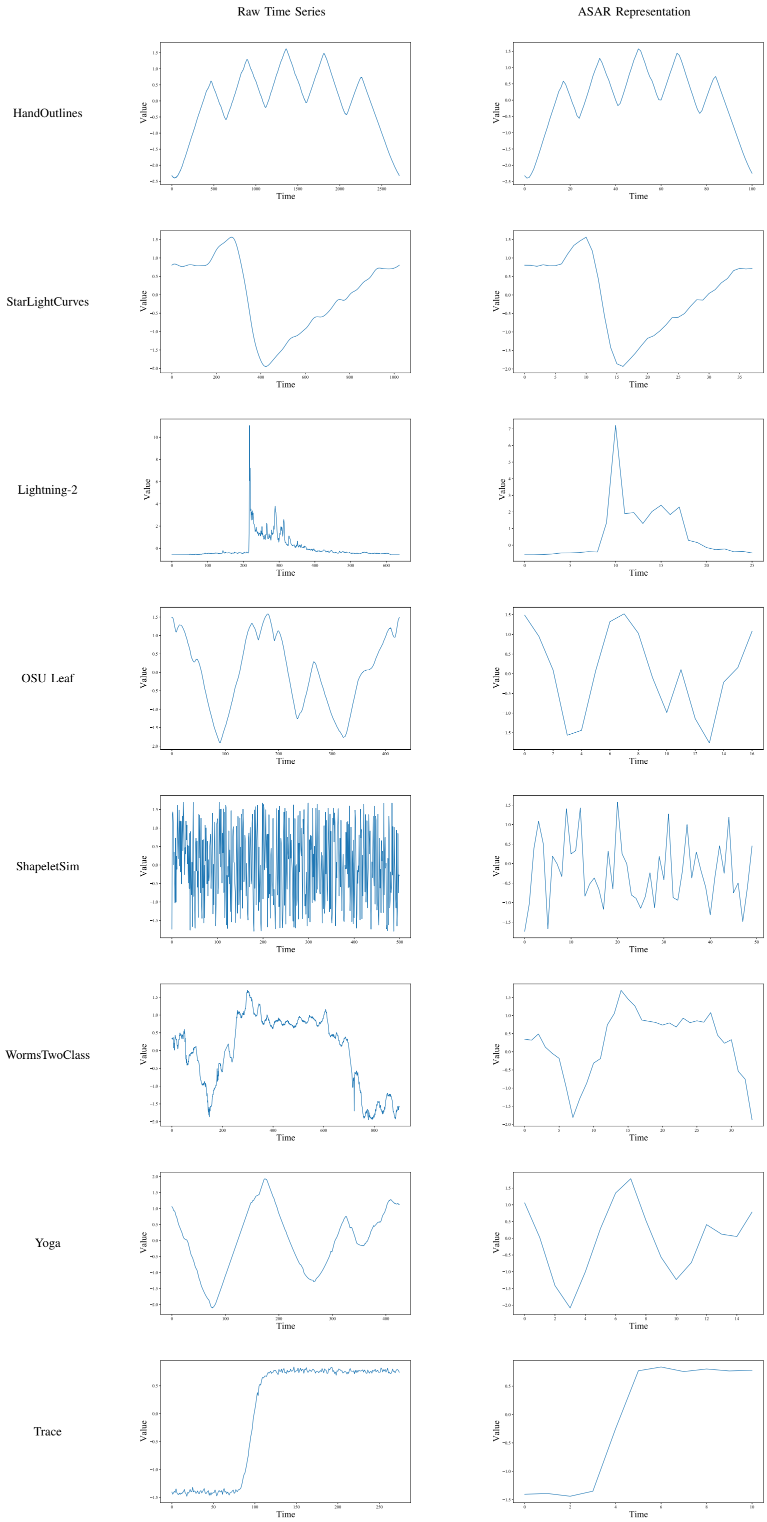

Fig. 2.: ASAR representation examples compared to the raw ones.

TABLE (IV)
F-MEASURE RELATIVE RESULTS FOR 1-NN CLASSIFICATION.

Method	Raw Time Series	FAD	PAA	SAX	ASAR
HandOutlines	1.00	0.64	1.00	1.00	0.99
StarLightCurves	1.00	0.97	1.00	1.00	1.00
Lightning-2	1.00	0.74	1.15	0.97	0.99
OSU Leaf	1.00	0.75	1.00	0.98	0.80
ShapeletSim	1.00	1.12	1.16	1.20	0.94
WormsTwoClass	1.00	0.92	1.00	1.02	0.98
Yoga	1.00	0.97	1.00	0.99	0.82
Trace	1.00	0.83	1.06	1.02	1.19
Average	1.00	0.87	1.05	1.02	0.96

TABLE (V)
F-MEASURE RELATIVE RESULTS FOR K-MEANS CLUSTERING.

Method	Raw Time Series	FAD	PAA	SAX	ASAR
HandOutlines	1.00	0.86	1.14	1.29	1.21
StarLightCurves	1.00	1.33	1.08	1.25	1.33
Lightning-2	1.00	0.93	0.96	0.85	0.89
OSU Leaf	1.00	0.94	0.88	0.94	1.00
ShapeletSim	1.00	1.02	1.02	0.98	1.00
WormsTwoClass	1.00	0.91	1.00	1.05	1.00
Yoga	1.00	0.93	0.82	0.93	1.04
Trace	1.00	0.90	1.15	0.90	1.15
Average	1.00	0.98	1.01	1.02	1.08

TABLE (VI)
THE COMPRESSION RATIO AS A PERCENTAGE (%).

Method	FAD	PAA	SAX	ASAR
HandOutlines	99.8	87.5	75	96.3
StarLightCurves	98.1	87.5	75	96.2
Lightning-2	4.9	87.4	74.9	94.8
OSU Leaf	56	87.4	74.9	95.5
ShapeletSim	0	87.4	75	88.2
WormsTwoClass	17.5	87.5	75	95.5
Yoga	52.7	87.3	74.9	95.8
Trace	10	87.3	74.9	95.4
Average	42.4	87.4	75	94.7

TABLE (VII)
THE TIME NEEDED (IN SECONDS) TO PERFORM 1-NN CLASSIFICATION AND K-MEANS CLUSTERING USING THE DATA REPRESENTED BY FAD, PAA, SAX, AND ASAR.

Method	Raw Time Series	FAD	PAA	SAX	ASAR
1-NN Classification	11.7	11.5	3.3	4.3	2.3
K-means Clustering	11.9	11.2	3.5	4.4	2.5

representation approach. With an average compression ratio of 94.7%, ASAR surpasses the competing approaches in reducing the dimensionality of the time series. PAA and SAX come in second and third places, respectively. However, FAD shows an unstable compression ratio among the datasets due to its high reliance on the data behavior, resulting in not guaranteed dimensionality reduction. Furthermore, the data mining tasks acceleration feature was demonstrated by measuring the time needed to apply the 1-NN classification and the K-means clustering. Table VII shows that these tasks have been performed the fastest by using the ASAR representation. ASAR took around 70% of the time needed to perform the tasks when compared to the PAA representation, which comes in second place. It is even less for others. Furthermore, it took around 20% of the time needed to perform these tasks using the time series in its raw form.

In summary, ASAR has achieved the best results in terms of the compression ratio. In addition, it has achieved the best results in terms of the data mining process acceleration and the K-means clustering. Moreover, even though it comes in third place in terms of the 1-NN classification, it has achieved a pretty good result, which shows the ability to preserve the time series information. This is also proved by ASAR's superiority in the K-means clustering results. These results were accomplished independently of the data behavior and domain, which provides extra advantage for ASAR as it can be used without any constraints on the type of data. The latter feature was proved by employing data from different domains with a diversity of behaviors and time series lengths (see Table III and Figure 2).

V. CONCLUSION AND FUTURE WORKS

In this paper, a novel time series representation approach has been proposed. The Adaptive Simulated Annealing Representation (ASAR) approach treats the time series representation as an optimization problem. Its objective is to segment the time series based on the tendencies by recording the instances that best explain the tendencies and neglecting the rest. By tracking the tendencies in the time series, ASAR was able to transform the time series into new dimensions while preserving the shape and the information. An experiment was designed to test its ability of maintaining the information, reducing dimensionality, and accelerating the data mining process. The experimental results have shown that ASAR outperforms FAD, PAA, and SAX approaches in terms of the compression ratio, the time needed to perform 1-NN classification and K-means clustering, and in the K-means clustering accuracy. It has also achieved high accuracy results in the 1-NN classification. These results assure that ASAR is able to conserve storage space and accelerate the data mining process while preserving the shape and the information of the time series. In addition, the experimental results have shown that ASAR is independent of the data type, behavior, domain, or length.

Some domains provide multivariate time series data (such as speed, flow, and occupancy in the intelligent transportation systems domain), which usually describe the same process. A possible future work is to extend ASAR to represent these multivariate time series data in a unified representation. Another possible future work is to define a new similarity measure based on the ASAR representation by utilizing the shape-preserving feature. Since this similarity measure will be tailored to the ASAR representation, this extension could achieve better results for the data mining tasks than using other similarity measures such as Euclidean distance or Dynamic Time Warping.

ACKNOWLEDGEMENT

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

REFERENCES

[1] Q.-T. Doan, A. Kayes, W. Rahayu, and K. Nguyen, "Integration of iot streaming data with efficient indexing andstorage optimization," *IEEE Access*, vol. 8, pp. 47 456–47 467, 2020, **DOI**: 10.1109/ACCESS.2020.2980006.

[2] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2389–2406, 2018, **DOI**: 10.1109/COMST.2018.2812301.

[3] S. Erevelles, N. Fukawa, and L. Swayne, "Big data consumer analytics and the transformation of marketing," *Journal of business research*, vol. 69, no. 2, pp. 897–904, 2016, **DOI**: 10.1016/j.jbusres.2015.07.001.

[4] D. R.-J. G.-J. Rydning, "The digitization of the world from edge to core," *Framingham: International Data Corporation*, p. 16, 2018.

[5] A. Holst, "Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025," https://www.statista.com/statistics/871513/worldwide-data-created/, 2021, accessed: 2021-06-30.

[6] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Greening big data," *IEEE Systems Journal,* vol. 10, no. 3, pp. 873–887, 2016, **DOI**: 10.1109/JSYST.2016.2550538.

[7] C. Chatfield, *Time-series forecasting*. CRC press, 2000.

[8] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied Soft Computing*, p. 106384, 2020, **DOI**: 10.1016/j.asoc.2020.106384.

[9] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17 287–17 298, 2019, **DOI**: 10.1109/ACCESS.2019.2895252.

[10] M. G. Tulics and K. Vicsi, "Automatic classification possibilities of the voices of children with dysphonia," *Infocommunications Journal,* vol. 10, no. 3, pp. 30–36, 2018, **DOI**: 10.36244/ICJ.2018.3.5.

[11] M. Saleem and B. Kovari, "Online signature verification using signature down-sampling and signer-dependent sampling frequency," *Neural Computing and Applications*, pp. 1–13, 2021, doi: **DOI**: 10.1007/s00521-021-06536-z.

[12] B. Mohammed, I. Awan, H. Ugail, and M. Younas, "Failure prediction using machine learning in a virtualised hpc system and application," *Cluster Computing*, vol. 22, no. 2, pp. 471–485, 2019, **DOI**: 10.1007/s10586-019-02917-1.

[13] A. Bhatia, S. Pasari, and A. Mehta, "Earthquake forecasting using artificial neural networks," *Int. Arch. Photogram. Rem. Sens. Spatial. Inform. Sci*., pp. 823–827, 2018, **DOI**: 10.5194/isprs-archives-XLII-5-823-2018.

[14] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018, **DOI**: 10.1016/j.asoc.2017.12.032.

[15] A. M. Nagy, B. Wiandt, and V. Simon, "Transient- based automatic incident detection method for intelligent transport systems," *Infocommunications Journal*, vol. 13, no. 3, pp. 2–13, 2021, **DOI**: 10.36244/ICJ.2021.3.1.

[16] S. J. Wilson, "Data representation for time series data mining: time domain approaches," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 9, no. 1, p. e1392, 2017, **DOI**: 10.1002/wics.1392.

[17] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.- A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery,* vol. 33, no. 4, pp. 917–963, 2019, **DOI**: 10.1007/s10618-019-00619-1.

[18] A. Belhadi, Y. Djenouri, K. Nørvåg, H. Ramampiaro, F. Masseglia, and J. C.-W. Lin, "Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103857, 2020, **DOI**: 10.1016/j.engappai.2020.103857.

[19] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015, **DOI**: 10.1016/j.is.2015.04.007.

[20] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery,* vol. 26, no. 2, pp. 275–309, 2013, **DOI**: 10.1007/s10618-012-0250-5.

[21] D. Wu, A. Singh, D. Agrawal, A. El Abbadi, and T. R. Smith, "Efficient retrieval for browsing large image databases," in *Proceedings of the fifth international conference on Information and knowledge management,* 1996, pp. 11–18.

[22] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," in *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, 2001, pp. 151–162, **DOI**: 10.1145/375663.375680.

[23] F. Gullo, G. Ponti, A. Tagarelli, and S. Greco, "A time series representation model for accurate and fast similarity detection," *Pattern Recognition*, vol. 42, no. 11, pp. 2998–3014, 2009, **DOI**: 10.1016/j.patcog.2009.03.030.

[24] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11, **DOI**: 10.1145/882082.882086.

[25] 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, **DOI**: 10.1007/s10618-007-0064-z.

[26] 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.

[27] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "New time series data representation esax for financial applications," in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. IEEE, 2006, pp. x115–x115, **DOI**: 10.1109/ICDEW.2006.99.

[28] Y. Sun, J. Li, J. Liu, B. Sun, and C. Chow, "An improvement of symbolic aggregate approximation distance measure for time series," *Neurocomputing*, vol. 138, pp. 189–198, 2014, **DOI**: 10.1016/j.neucom.2014.01.045.

[29] C. T. Zan and H. Yamana, "An improved symbolic aggre- gate approximation distance measure based on its statistical features," in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, 2016, pp. 72–80, **DOI**: 10.1145/3011141.3011146.

[30] A. M. Nagy and V. Simon, "A novel data representation method for smart cities' big data," in *Artificial Intelligence, Machine Learning, and Optimization Tools for Smart Cities: Designing for Sustainability.* Springer International Publishing, 2022, pp. 97–122, **DOI**: 10.1007/978-3-030-84459-2_6.

[31] N. Kumar, V. N. Lolla, E. Keogh, S. Lonardi, C. A. Ratanamahatana, and L. Wei, "Time-series bitmaps: a practical visualization tool for working with large time series databases," in *Proceedings of the 2005 SIAM international conference on data mining*. *SIAM*, 2005, pp. 531–535, **DOI**: 10.1137/1.9781611972757.55.

[32] C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi, "A novel bit level time series representation with implication of similarity search and clustering," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2005, pp. 771–777.

[33] A. Bagnall, E. Keogh, S. Lonardi, G. Janacek et al., "A bit level representation for time series data mining with shape based similarity," *Data Mining and Knowledge Discovery,* vol. 13, no. 1, pp. 11–40, 2006, **DOI**: 10.1007/s10618-005-0028-0.

[34] P. Zhan, H. Xu, and L. Chen, "Fcad: Feature-based clipped representation for time series anomaly detection," in *2020 IEEE 3rd International Conference on Information Systems and Computer Aided Education (ICISCAE)*. IEEE, 2020, pp. 206–210, **DOI**: 10.1109/ICISCAE51034.2020.9236862.

[35] P. Zhan, C. Sun, Y. Hu, W. Luo, J. Zheng, and X. Li, "Feature-based online representation algorithm for streaming time series similarity search," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 34, no. 05, p. 2050010, 2020,
DOI: 10.1142/S021800142050010X.

[36] M. Zhang and D. Pi, "A new time series representation model and corresponding similarity measure for fast and accurate similarity detection," *IEEE Access*, vol. 5, pp. 24503–24519, 2017,
DOI: 10.1109/ACCESS.2017.2764633.

[37] H. Kamalzadeh, A. Ahmadi, and S. Mansour, "A shapebased adaptive segmentation of time-series using particle swarm optimization," *Information Systems,* vol. 67, pp. 1–18, 2017,
DOI: 10.1016/j.is.2017.03.004.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983,
DOI: 10.1126/science.220.4598.671.

[39] S. Sakamoto, K. Ozera, A. Barolli, M. Ikeda, L. Barolli, and M. Takizawa, "Implementation of an intelligent hybrid simulation systems for wmns based on particle swarm optimization and simulated annealing: performance evaluation for different replacement methods," *Soft Computing*, vol. 23, no. 9, pp. 3029–3035, 2019,
DOI: 10.1007/s00500-017-2948-1.

[40] Y.-H. Lee, C.-P. Wei, T.-H. Cheng, and C.-T. Yang, "Nearest-neighbor-based approach to time-series classification," *Decision Support Systems*, vol. 53, no. 1, pp. 207–217, 2012,
DOI: 10.1016/j.dss.2011.12.014.

[41] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006,
DOI: 10.1016/j.patrec.2005.10.010.

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

Mohammad Bawaneh received his M.Sc. degree in Computer engineering from Yarmouk University (YU)-Jordan in 2017. He is currently a Ph.D. candidate at Budapest University of Technology and Economics – Department of Networked Systems and Services (BME-HIT) in the Multimedia Networks and Services Laboratory (MEDIANETS). His research interests include time series data mining and analysis, and machine learning and data analytics for smart cities and intelligent transportation management systems.

Vilmos Simon received his PhD from the Budapest University of Technology and Economics (BME) in 2009. Currently he is an Associate Professor at the Department of Networked Systems and Services and Head of the Multimedia Networks and Services Laboratory. He has done research on mobility management and energy efficiency in mobile cellular systems and self-organized mobile networks, recently his research interests include machine learning and data analytics for smart cities and intelligent transportation management systems. He published 50+ papers in international journals and conferences, and acts as a reviewer or organizer for numerous scientific conferences. He serves currently as the Corporate liaison vice president for the Connected and Automated Mobility Cluster of Zala.