Slopewise Aggregate Approximation SAX: keeping the trend of a time series

Lamprini Pappa
Department of Informatics and Telecommunications, University of Ioannina
Arta, Greece
pappa.l@kic.uoi.gr

Petros Karvelis
Department of Informatics and Telecommunications, University of Ioannina
Arta, Greece
pkarvelis@uoi.gr

George Georgoulas
Department of Informatics and Telecommunications, University of Ioannina
Arta, Greece
georgoul@gmail.com

Chrysostomos Stylios
Department of Informatics and Telecommunications, University of Ioannina
Arta, Greece
stylios@uoi.gr

Industrial Systems Institute, Athena Research Center
Patras, Greece
stylios@isi.gr

2021 IEEE Symposium Series on Computational Intelligence (SSCI) | 978-1-7281-9048-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSCI50451.2021.9660130

***Abstract*— In this work, we introduce the Slopewise Aggregate Approximation (SAA), an innovative variation of the Piecewise Aggregate Approximation. The Slopewise Aggregate Approximation (SAA) is used as a novel core step for the Symbolic Aggregate Approximation method. SAA efficiently describes the trend at a time series signal since it incorporates information regarding the shape and fluctuation of the time series while simultaneously achieving the problem's dimensionality reduction. Then, by applying the discretization technique, the problem is transformed into a symbolic space problem, and the Intelligent Icons are the features that come out and feed a Near Neighbour classifier for a Human Activity Recognition problem. The results achieved by the proposed method are directly compared to two relevant past implementations and exhibit a considerable increase in classification metrics.**

Keywords—Piecewise Aggregate Approximation, Symbolic Aggregate Approximation, Slopewise Aggregate Approximation, Intelligent Icons, Classification

I. Introduction

The Symbolic Aggregate Approximation (SAX) method is mainly an approach that transforms time series into symbols [1] and offers various advantages, such as the dimensionality reduction of the problem, which is achieved by implementing the Piecewise Aggregate Approximation technique. The SAX method [2] has met with wide approval due to its comparative advantages and intuitive nature; therefore, many researchers have dealt with it over time. As a result, various interesting variations have been developed that aim at optimizing the method. It is reasonable for the SAX method to have certain flaws. For example, on the one hand, there is a need for defining several parameters that affect the final result; on the other hand, the method bears an innate defect, namely the ability to provide information regarding abrupt changes in the data values of the initial time series. The latter is of particular importance, especially in cases where these changes correspond to patterns of behaviour that describe critical information about the under examination time series.

Time series appear in many fields, economic data are characterised by sharp fluctuations that are of significant importance for analyzing these data. A solution to this problem was proposed by introducing the ESAX [3], [4]. In addition to the symbol for the mean value, two more symbols were added that describe the maximum and the minimum value of a segment, respectively. Li et al. [5] suggested the TSX, which introduces three more symbols to represent a segment, to address the same issue. The first represents the slope of the line that joins the initial to the endpoint. The other two symbols describe the maximum distance of the points above and below this line. The TFSAX method [6] adds just one additional symbol representing the slope of the line joining the initial to the endpoint.

The iSAX [7] is not exactly a variant but rather a superset of the SAX method, as it is a representation that supports the indexing of large data sets and can index up to one hundred million time series. In a later study [1], the authors showed that up to one billion time series could be indexed. Pham et al. [8] suggested *α*SAX and *iα*SAX. The first is a combination of the SAX algorithm and the k-means algorithm and addresses SAX's high dependence on the Gaussian distribution of time series. On the other hand, the *iα*SAX is nothing more than the iSAX-like indexing algorithm following the *α*SAX method. The SFA [9] relies on discretization through the discrete Fourier transform as it provides a statistically significant tighter lower bound and the ability to describe the signal more comprehensively due to the transition in the frequency domain.

The GASAX [10] was proposed for the determination of boundary points using a genetic algorithm. GASAX's goal is to find the nearly optimal distribution of breakpoints without assuming any particular distribution of the time series. Although the GASAX works well on both normalised and non-normalised time series data, appropriate control parameters must be specified, and it fails to include information about the current time series trend. Next, we find the 1d-SAX [11]. It essentially calculates the slope of the line of least squares (linear regression) in each segment and combines this value with the mean value of the segment so that, in the end, a single symbol emerges. In this way, the increase of complexity is avoided, as happens with other SAX variants, which add additional symbols. The TrSAX [12] is based on calculating the slope of the line of least squares in each segment; however, it assigns this value to an additional symbol.

978-1-7281-9048-8/21/$31.00 ©2021 IEEE

The SAX-EFG algorithm [13] combines the SAX with a technique for producing features based on an evolutionary algorithm named EFG, which was used to generate particular attributes for classifying DNA sequences. Accordingly, the SAX-EFG uses patterns derived from SAX as building blocks to construct more complex features. The SAX-TD [14] also attempts to address the problem of losing the time series trend by calculating the deviations of the distances of the initial and the endpoints from the mean value between successive segments of the time series. The TSAX also deals with maintaining the trend of the time series. The TFSA [15] focuses on preserving most of the trend characteristics and patterns of the original time series.

Zan & Yamana [16] proposed the SAX_SD, which adds an additional feature: a statistical measure, the standard deviation, and displays the distribution of points in each part of the time series. The APAA / ASAX method [17] uses a non-fixed segment size but an adaptable one to events of interest of the time series to avoid interrupting or prolonging them. The SAX-BD [18] combines the advantages of the ESAX and the SAX-TD using weighted boundary distance as a new distance measure to obtain a new time series representation. Finally, one of the most recent bibliographic references is the SAX-TM [19], which uses transition matrices to maintain information describing the time series trend.

The Multichannel SAX Intelligent Icons [20] is an extension of SAX that can be applied in multichannel signals. In particular, it creates a spatial correlation of the inherited information in all dimensions, and so it provides extra features for distinguishing the human activities, a fact that leads to increased accuracy and sensitivity of the model.

The remaining of the paper is structured as follows. Section II presents the main steps of the proposed methodology along with the existing one. Section III briefly presents the steps for the original implementation of the SAX method and the extraction of the Intelligent Icons. In section IV, we introduce our innovative Slopewise Aggregate Approximation (SAA). In section V we give all the necessary information regarding the dataset used and the signals utilized, while in Section VI, we refer to the essential preprocessing that the signals underwent to be prepared to apply our method on them. Section VII provides all the classification results that came out after implementing the method, and finally, Section VIII concludes the paper with directions for future work.

II. Overall Method

Fig. 1 presents the overall methodology. It illustrates where the proposed SAA is introduced to substitute the PAA. The overall method consists of the following steps, as shown in Fig. 1. In the first place, we use the raw signals, and the preprocessing stage takes place, which consists of applying noise removal filters for signal smoothing and the standardization technique. The data segmentation step follows by applying the sliding window technique that leads to an ensemble of windows. Afterwards, the process proceeds in parallel for comparison reasons. To elaborate, we apply the well-known Piecewise Aggregate Approximation to reduce the dimensions of the problem, and in parallel, we apply the proposed technique, the Slopewise Aggregate Approximation (SAA). The discretization step comes next for each of the two paths, where a string of symbols represents every window of the signal. The feature extraction step consists of combining the intelligent icons as they were produced from each of the two approaches. The last step is classification. A KNN classifier is employed to predict the class of the inputs and ultimately evaluate the performance of our model.

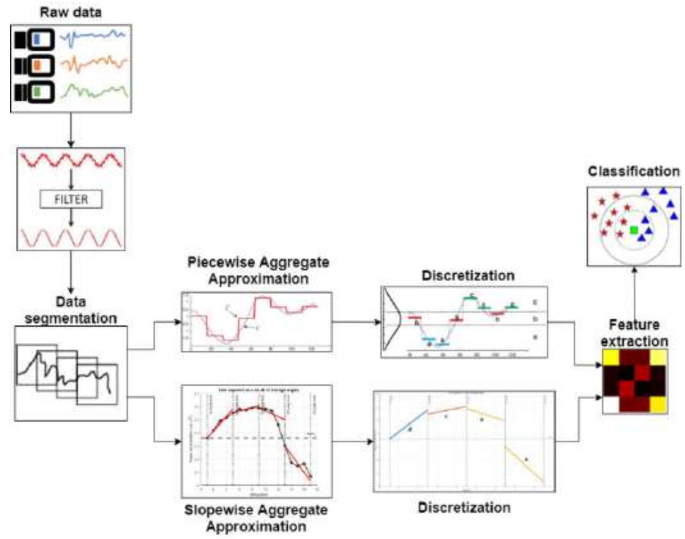

Fig. 1 The overall SAX method indicating the introduction of the proposed SAA

III. Classical Approach

A. The Symbolic Aggregate Approximation (SAX) method

The SAX method was first established by Lin et al. [2]. It constitutes an approach that allows the symbolic representation of time series. Utilizing this methodology, we achieve to reduce the dimensions of a problem, which is of paramount importance considering its effect on the speed and efficiency of the employed algorithms. Moreover, several distance measures can be used (e.g., Euclidean, Manhattan, Minkowski) to compare the symbolic series (in the symbolic space) related to the distance measures of the initial time series. Besides, the existence of lower bounding is guaranteed by applying these distance measures [2]. As a result, a reduction in complexity and time required to calculate the distance is achieved.

SAX method can be applied to time series of any length, while it can be implemented easily, with limited requirements for computational resources and memory capacity. Finally, it should be noted that its application does not require access to all database data, which makes it a viable option for processing and managing time series as streaming data [21].

The SAX method consists of the following steps:

- Piecewise Aggregate Approximation
- Discretization

1) Piecewise Aggregate Approximation (PAA)

The PAA is a technique that aims to reduce the dimensions of the problem [2], [22]. The main idea behind its application is to calculate the mean value of a set of points that make up a segment of the time series and finally replace that segment with the calculated average value. In this way, we achieve the dimensionality and noise reduction of the problem while maintaining the trend of the time series.

More specifically, with this technique, a time series X of length n is transformed into another time series X' of length m, where $m < n$.

$$X = \{x_1, x_2, \dots, x_n\} \rightarrow X' = \{x_1', x_2', \dots, x_m'\}, \ m < n,$$

where x_i' is calculated by the formula (1), which describes the calculation of mean value.

$$x_i' = \frac{m}{n} \sum_{j=\frac{n}{m}(i-1)+1}^{\frac{n}{m}i} x_j \tag{1}$$

In other words, to reduce the dimension from n to m, we first divide the time series into m equally-sized segments, and secondly, we calculate the mean value of each of the segments m, thus compressing the initial time series by a factor n/m. The sequence resulting from the calculated mean values (called PAA coefficients) is the Piecewise Aggregate Approximation of the initial time series.

2) Discretization

At the discretization stage, which is the core of the SAX method, we essentially assign a symbol to each of the m segments (i.e., the PAA coefficients).

$$X' = \{x_1', x_2', \dots, x_m'\} \rightarrow S = \{s_1, s_2, \dots, s_m\},$$

where X' is the time series of length m as it resulted after the application of PAA, and S is a string of symbols of length m that will occur after the application of the discretization step.

To do this, we first select the size of the alphabet, being symbolized as $\boldsymbol{\alpha}$. In other words, we choose how many symbols will be available. E.g., if $\boldsymbol{\alpha} = 3$ (i.e., the size of the alphabet is equal to 3), then each segment can be represented by one of the 'a', 'b', 'c' symbols. If we choose an alphabet size $\boldsymbol{\alpha} = p, p \in \mathbb{N}, p \geq 3$, then let us consider the alphabet A which will consist of the following symbols:

$$A = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_p\}.$$

Each symbol should have an equal probability of occurrence. Such a condition is ensured because we first normalize the time series; hence it follows the normal distribution [46]. Based on the alphabet size, we separate the space of real numbers in as many value areas as the size of the alphabet we chose. The breakpoints define the range of values of an area β_i, with $i = 1, 2, \dots, p-1$, as they are obtained from the lookup normal distribution table (see Fig. 2) so that each area satisfies the condition for a value to have the same probability of belonging to one of them [23], [24]. More specifically, the area below the normal distribution curve between the values β_i and β_{i+1} is equal to $1/\alpha$ [21].

β_i \ a	3	4	5	6	7	8
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67
β_3		0.67	0.25	0	-0.18	-0.32
β_4			0.84	0.43	0.18	0
β_5				0.97	0.57	0.32
β_6					1.07	0.67
β_7						1.15

Fig. 2 Lookup normal distribution table that contains the breakpoints β_i

At this point, we proceed to the transformation of real values to symbols consulting the formula (2). The overall process is depicted in Fig. 3 with application to a one-dimensional signal.

$$s_i = \begin{Bmatrix} a_1, \ if \ \{x_i' \in (-\infty, \beta_1] \\ a_2, \ if \ \{x_i' \in (\beta_1, \beta_2] \\ \vdots \\ a_p, \ if \ \{x_i' \in (\beta_{p-1}, +\infty) \end{Bmatrix}, \text{ where } i = 1, 2, \dots, m \tag{2}$$

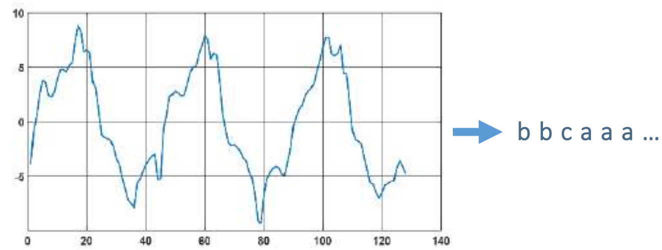

Fig. 3 Signal transformation from real numbers space to symbolic space

Intelligent icons

"Intelligent icons" is a method that displays the results after applying the discretization step [25]. In this way, we calculate the frequency of occurrence of a symbol or group of symbols (called words) within a window by creating approximations of the underlying probability mass functions [26]. The words can be of different lengths, and the choice of length (let us define it as $\boldsymbol{L}$) relies on the design, the resources availability, and the objectives of the researcher. In any case, the latter should pursue a balance between the model's prediction accuracy and the computational cost required to calculate the intelligent icons.

The steps followed for calculating the intelligent icons are:

- We compute all the possible combinations of alphabet symbols that form all the possible words. The number of all the combinations is $\boldsymbol{\alpha}^L$. Therefore, if we choose $\boldsymbol{\alpha} = 3$ and $\boldsymbol{L} = 2$, the number of all the possible words is $3^2 = 9$, which are presented in TABLE I.

TABLE I. The possible words that are formed after choosing $\boldsymbol{\alpha} = 3$ and $\boldsymbol{L} = 2$

aa	ba	ca
ab	bb	cb
ac	bc	cc

- Upon the completion of the discretization stage, a string of symbols is produced. For every word, we find the number of its appearances in the string.
- Create a table that depicts in the corresponding position of TABLE I the frequency of occurrence of every word, as shown in Fig. 4.

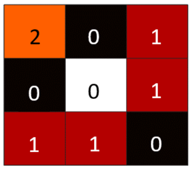

Fig. 4 The specific intelligent icon is calculated from the string ‘bbcaaacbbbb’. Each number shows the frequency of occurrence of the word that lies in the corresponding cell of TABLE I.

IV. Slopewise Aggregate Approximation

Here, we introduce the Slopewise Aggregate Approximation (SAA) being a variation of the PAA technique. Unlike PAA that takes into account just the mean values of data points in every segment, SAA represents the time series trend by incorporating information related to the slope of a line that describes a segment. The proposed procedure is explained in more detail:

Consider a data point as a point A in the cartesian coordinates system, which comprises an x-value (i.e., the value on the horizontal axis that in our case is time) and a y-value (i.e., the value on the vertical axis that in our case is the value that a quantity takes, e.g., linear acceleration or angular velocity). If we connect it with another point B in the above system, a vector $\overrightarrow{AB}$ is formed (see Fig. 5), which is described by its magnitude and its direction. To define the direction of the vector, we calculate its slope, from which we can easily calculate the angle θ that the vector forms with the horizontal axis.

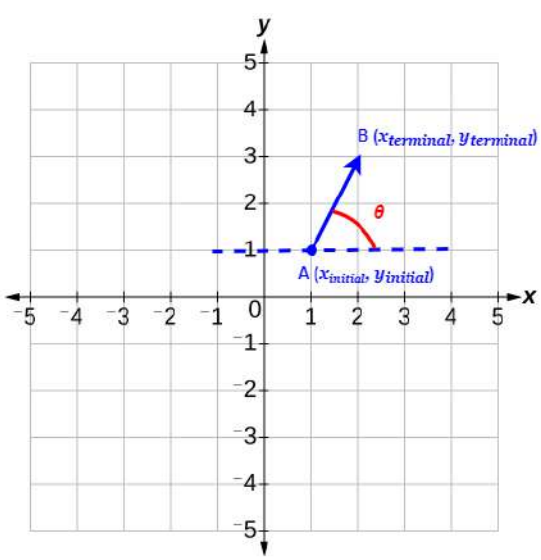

Fig. 5 A vector $\overrightarrow{AB}$ in the cartesian coordinates system. A and B are the data points of the under-study time series.

As it happens in PAA, the time series X of length n is divided into m segments and finally is transformed into another time series Θ' of length m, where $m < n$.

$$X = \{x_1, x_2, \dots, x_n\} \rightarrow \Theta' = \{\theta_1{}', \theta_2{}', \dots, \theta_m{}'\},\ m < n.$$

In short, with our technique, we transform the initial time series to angle values series. To do this, we connect the first point of each segment (this one will be the initial point of the vector) with all the other points (these will be the terminal points) of this segment to form vectors. Thus, we obtain vectors as follows:

1st point → 2nd point,

1st point → 3rd point,

1st point → ....

1st point → last point of the segment.

To avoid losing the continuity of the signal, we form a different vector where the initial point is the first point of the current segment, and the terminal point is the first point of the next segment. Every vector forms an angle with the horizontal axis. Fig. 6 illustrates the above-mentioned.

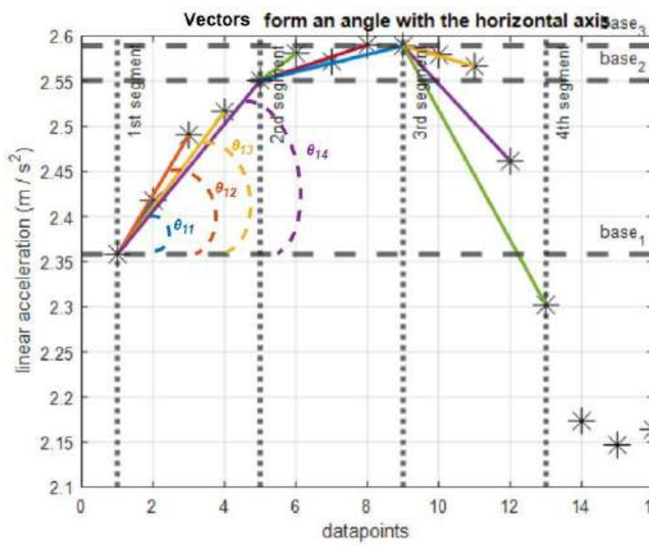

Fig. 6 By connecting every data point (asterisk *) in a segment with the first one within this segment, we form vectors indicated by the coloured lines. We calculate the angle θ_{ij} that each vector forms with the horizontal axis ($base_i$).

We calculate the angles of all the aforementioned vectors using the formulas that follow. We compute the following angles (in radians) examining the i – segment that comprises n/m data points.

$$\theta_{i1} = \tan^{-1}\left(\frac{y_{i2}-y_{i1}}{x_{i2}-x_{i1}}\right),$$

$$\theta_{i2} = \tan^{-1}\left(\frac{y_{i3}-y_{i1}}{x_{i3}-x_{i1}}\right),$$

.........,

$$\theta_{i\frac{n}{m}} = \tan^{-1}\left(\frac{y_{(i+1)1}-y_{i1}}{x_{(i+1)1}-x_{i1}}\right)$$

where:

θ_{i1} is the angle of the vector that is determined from the 1st point (x_{i1}, y_{i1}) to the 2nd one (x_{i2}, y_{i2}),

θ_{i2} is the angle of the vector that is determined from the 1st point (x_{i1}, y_{i1}) to the 3rd one (x_{i3}, y_{i3}),

......,

$\theta_{i\frac{n}{m}}$ is the angle of the vector that is determined from the 1st point (x_{i1}, y_{i1}) to the 1st one of the next segment $(x_{(i+1)1}, y_{(i+1)1})$.

Then we compute the mean value of the angles in every segment according to the formula (3).

$$\theta_i{}' = \frac{m}{n}\sum_{j=1}^{\frac{n}{m}} \vartheta_{ij} \quad (3)$$

Now we proceed to the final step of the proposed technique. We replace every segment of the time series with a vector that has its initial point at the first point of the segment and forms an angle with the horizontal axis equal to $\theta_i{}'$, as it is depicted in Fig. 7. Thus, with this process, not only do we achieve the dimensionality reduction of the problem, but also we succeed in maintaining the trend of the time series, something evident by comparing the time series (black line) to the ensemble of the vectors formed (Fig. 7).

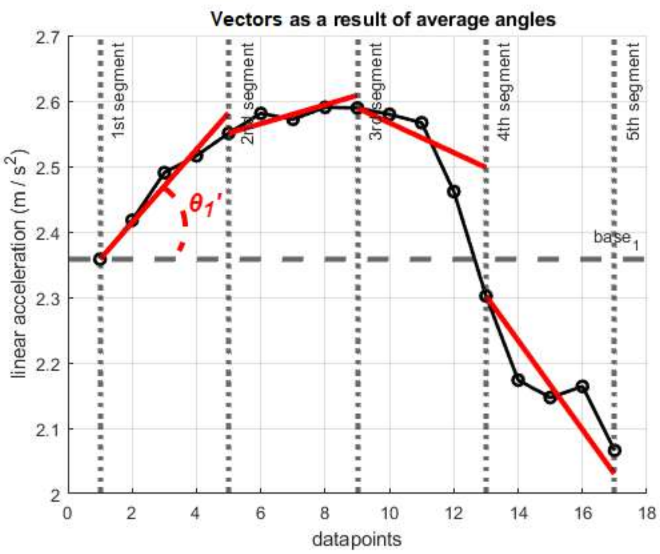

Fig. 7 Every segment of the time series is replaced with a vector that has its initial point at the first point of the segment and forms an angle with the horizontal axis equal to $\theta_i{}'$. Red lines are the slope approximations, and the continuous black line is the signal.

In a nutshell, the transformation stages that took place are:

$$X = \{x_1, x_2, \dots, x_n\} \rightarrow \Theta = \{\theta_1, \theta_2, \dots, \theta_n\} \rightarrow$$
$$\Theta' = \{\theta_1{}', \theta_2{}', \dots, \theta_m{}'\}, \quad m < n.$$

Then, the discretization step of the SAX method can be applied, which is followed by the intelligent icons extraction, which has been presented in detail in sections III.A.2 and III.B of the current document. The reader can find a comprehensive analysis in applying the steps of the SAX method in [20].

V. DATA SET USED

The proposed method is tested with signals (which are time series) from a publicly available web database, the RealWorld (HAR) [27], that contains signals recorded by fifteen subjects (age 31.9±12.4, height 173.1±6.9, weight 74.1±13.8, eight males and seven females), who performed eight different activities: climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking. Here, we only utilized the signals produced from the accelerometer and the gyroscope of the device (triaxial linear acceleration and angular velocity). The sampling rate was set at 50 Hz. Every subject performed each activity roughly for 10 minutes except for jumping due to the physical exertion (~1.7 minutes). Concerning gender, the amount of data is equally distributed.

VI. PREPROCESSING

Raw data should undergo a preprocessing procedure in order for us to be able to apply the proposed method. The first step in the preprocessing phase is synchronizing sensor data, as the instants of gyroscope recordings increasingly deviate from those from the accelerometer as time passes.

The filtering step is necessary to remove the unwanted noise and make the signal smoother without, of course, losing any critical points. In this direction, we applied a fifth-order median filter [28] and a fifth-order low-pass Butterworth filter with a 20 Hz cutoff frequency [29], [30].

We then applied z-score normalization to the signal with a mean value of zero and a standard deviation value of one, since it is meaningless to compare time series with different offsets and amplitudes [24]. By this, we achieve to remove the distortions, namely the offset translation and the amplitude scaling, that has a negative impact on the results of the activity recognition tasks [31].

A. Data segmentation

A common technique that is applied for signal segmentation in classification problems for activity recognition is the "sliding windows" technique [32], [33]. Therefore, we segmented the filtered and z-normalized signals into sliding windows of the same duration T (we chose 2.56 seconds). The windows are segmented successively with 50% overlapping percentage. Considering that the sampling rate of the signal is 50 Hz, we can easily calculate the number of data points that comprise a window, that is, 50 Hz * 2.56 sec = 128 data points. Fig. 8 illustrates the shape of the signal at one specific window and the corresponding data points.

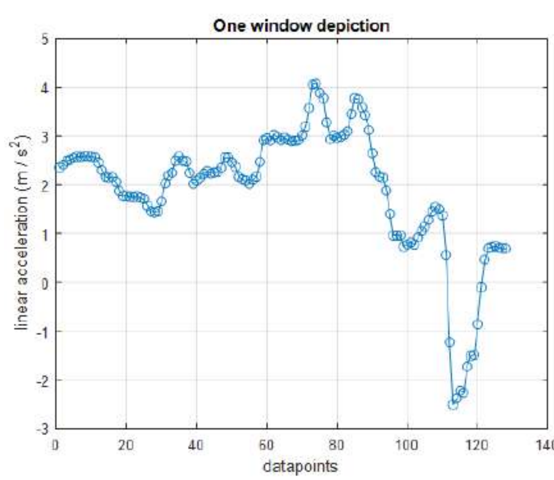

Fig. 8 One window of 128 data points (=2.56 sec) of the linear acceleration signal at x-dimension.

Then, we are able to implement our proposed method in every time window of each signal, namely the SAA technique, followed by the discretization stage, and finally, the extraction of the intelligent icons.

VII. Results

The table of features that functions as the input to a classifier is composed of the intelligent icons produced after the SAX implementation. We employed a 1-Nearest Neighbour classifier in order to examine the prediction accuracy and sensitivity of the model. We point out that our goal is not to achieve the best scores of classification metrics (so we do not focus on finding the best classifier) but rather to compare the proposed SAA method with existing ones. Thus, we conduct a two-direction comparison; that is, comparing the proposed method with:

- SAX [2]
- Multichannel SAX Intelligent Icons [20].

Both of them have been implemented in [20]. For that reason, to be absolutely fair, we used the exact same signals and parameters' values in all implementations, which are depicted in TABLE II.

TABLE II. The values of the parameters that are used in all the implementations.

Window duration	T	2.56 sec
Data points/window	n	128
Segments/window	m	32
Alphabet size	$\boldsymbol{a}$	4
Word length	$\boldsymbol{L}$	3

Just like in [20], we randomly separated the table of features to obtain a training dataset and a testing dataset. The training dataset consists of randomly extracted 80% of the features of every class, and the remaining 20% constitutes the testing dataset.

We repeated the execution of the Nearest Neighbour algorithm ten times, and TABLE III depicts the average classification accuracy of the three models along with the relevant standard deviation (STD). The results of the first two models are obtained from [20].

TABLE III The comparison table of the accuracy values that each one of the models under examination achieved.

ACCURACY (%)					
SAX		***Multichannel SAX***		***SAA SAX***	
MEAN	***STD***	***MEAN***	***STD***	***MEAN***	***STD***
90.13	0.20	92.39	0.24	96.00	0.19

Table II shows that the proposed SAA-SAX method surpasses both the classical SAX and the Multichannel SAX. SAA-SAX achieves better results in terms of accuracy while it exhibits the lowest standard deviation value.

Figure 9 (Fig. 9a, Fig. 9b, Fig. 9c) presents the confusion matrices that are extracted after the repetition of the KNN classifier ten times.

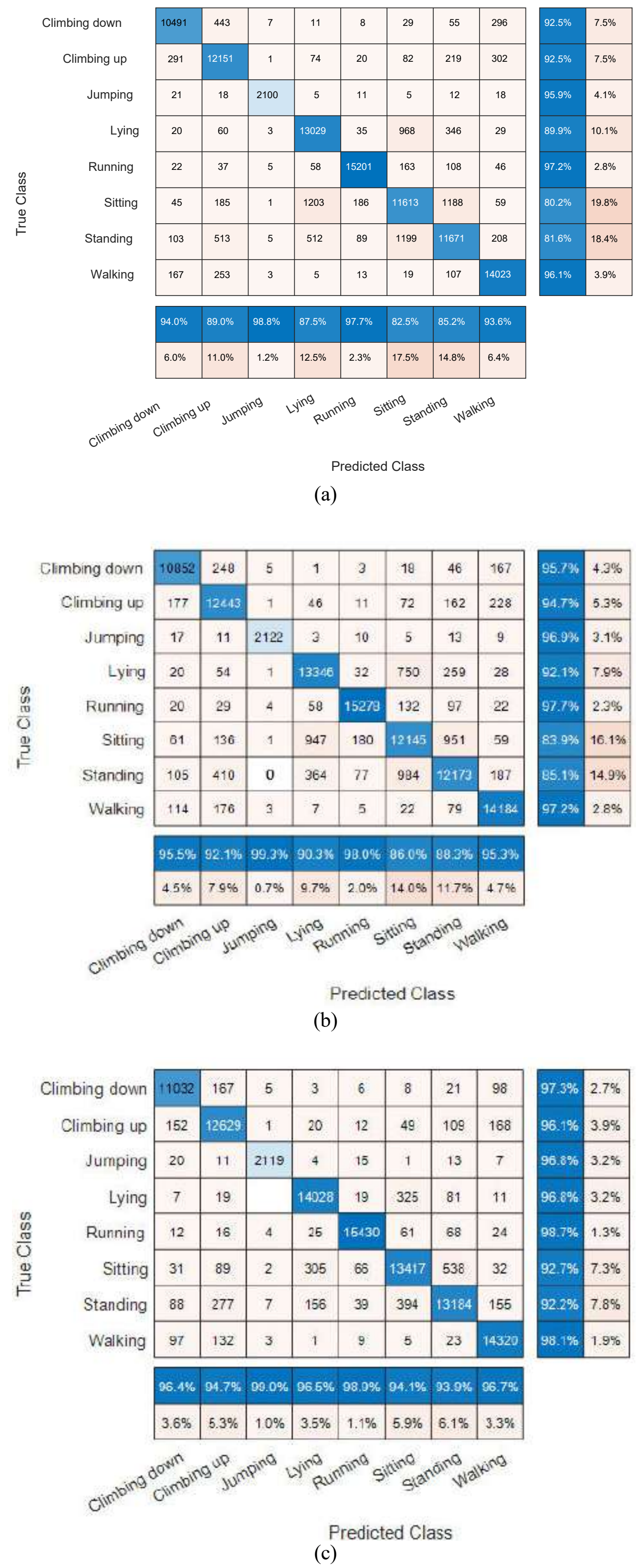

Fig. 9 Total confusion matrix of the three methods: SAX (a), Multichannel SAX (b), and SAA SAX (c). The last two columns display the percentage of correctly and incorrectly classified observations for each predicted class. The last two rows display the percentage of correctly and incorrectly classified observations for each true class.

TABLE IV depicts the average and standard deviation values of True Positive Rates (TPR) (or sensitivity) for the three approaches for each activity after the ten repetitions.

TABLE IV Total comparative table between our method and single-channel intelligent icons approach

	T P R (%) ± STD (%)		
	SAX	Multichannel SAX	SAA SAX
Climbing down	92.51 ± 0.88	95.70 ± 0.63	97.28 ± 0.26
Climbing up	92.47 ± 1.04	94.70 ± 0.98	96.11 ± 0.94
Jumping	95.89 ± 1.18	96.89 ± 0.91	96.76 ± 1.11
Lying	89.92 ± 0.74	92.10 ± 0.37	96.81 ± 0.48
Running	97.19 ± 0.44	97.69 ± 0.46	98.66 ± 0.20
Sitting	80.20 ± 0.75	83.87 ± 0.59	92.66 ± 0.81
Standing	81.62 ± 1.03	85.13 ± 0.91	92.20 ± 0.51
Walking	96.11 ± 0.50	97.22 ± 0.33	98.15 ± 0.43
MEAN	**90.74 ± 0.82**	**92.91 ± 0.65**	**96.08± 0.59**

Table IV proves that SAA SAX scores the best in terms of the model's sensitivity and each activity's sensitivity (only "Jumping" keeps the score at the same level), exhibiting a notable increase. In addition, the standard deviation of the observed values is kept low to acceptable levels, indicating the robustness of the model.

The most remarkable increase in the results is observed in the activities "Sitting" and "Standing", which are 92.66% and 92.20%, respectively. For these activities, SAA SAX improves the classification sensitivity by 10% in comparison to SAX and Multichannel SAX. This is very useful, considering that distinguishing these two activities is a common problem in Human Activity Recognition tasks, and they are often described as non-dynamic activities [29]. Nevertheless, there are leakages between the two above mentioned activities, as shown in Fig. 9c. However, they are limited to a great extent compared to the other two approaches. Another noteworthy observation concerns the activity "Lying", where the sensitivity increases by about 5%.

VIII. CONCLUSIONS

This study introduces the Slopeewise Aggregate Approximation (SAA) as a variation of the Piecewise Aggregate Approximation (PAA). Instead of extracting information related to the mean value of a set of data points, SAA achieves to capture the trend of the signal (time series) by, in a sense, calculating the rate of change of the signal. Then the results are discretized by assigning a symbol to each value, and finally, the Intelligent Icons are extracted.

The proposed approach SAA-SAX is successfully applied and tested to a Human Activity Recognition classification problem.

The results of our experiments, which are obtained after performing the Nearest Neighbour classification, indicate that the proposed approach achieves a substantial increase in accuracy and sensitivity rates compared to the classical SAX and Multichannel SAX demonstrated in [20]. Significantly, static activities, such as "sitting", "standing", and "lying" that exhibit difficulty in correctly classifying them, are benefited the most from the proposed SAA-SAX method.

An issue significant to mention concerns the pre-definition of the set of parameters whose values directly affect the final result. Here, we used the parameters' values of the previous work [20] to compare the proposed SAA-SAX.

To sum up, we proposed the SAA-SAX, a novel variation of SAX that achieves and integrates information regarding the trend and shape of the under examination signal. The algorithm is simple to implement, yet intuitive, which has a positive impact on computational and complexity costs.

ACKNOWLEDGMENT

This research work is funded by the Operational Programme "Epirus" 2014-2020, under the project "Integrated Support System for elderly people with health problems and lonely workers using Portable Devices and Machine learning Algorithms – TrackMyHealth", Co-financed by the European Regional Development Fund (ERDF).

REFERENCES

[1] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh, "iSAX 2.0: Indexing and Mining One Billion Time Series," in *2010 IEEE International Conference on Data Mining*, Dec. 2010, pp. 58–67. doi: 10.1109/ICDM.2010.124.

[2] 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*, New York, NY, USA, Jun. 2003, pp. 2–11. doi: 10.1145/882082.882086.

[3] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "Extended SAX: extension of symbolic aggregate approximation for financial time series data representation," *Proceeding of IEICE the 17th Data Engineering Workshop*, Jan. 2006.

[4] B. Lkhagva, Y. Suzuki, and K. Kawagoe, "New Time Series Data Representation ESAX for Financial Applications," Feb. 2006, pp. x115–x115. doi: 10.1109/ICDEW.2006.99.

[5] G. Li, L. Zhang, and L. Yang, "TSX: A Novel Symbolic Representation for Financial Time Series," in *PRICAI 2012: Trends in Artificial Intelligence*, Berlin, Heidelberg, 2012, pp. 262–273. doi: 10.1007/978-3-642-32695-0_25.

[6] Y. Yu, Y. Zhu, D. Wan, Q. Zhao, and H. Liu, "A Novel Trend Symbolic Aggregate Approximation for Time Series," *arXiv:1905.00421 [cs]*, May 2019, Accessed: Jun. 25, 2021. [Online]. Available: http://arxiv.org/abs/1905.00421

[7] J. Shieh and E. Keogh, "*i*SAX: indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, Aug. 2008, pp. 623–631. doi: 10.1145/1401890.1401966.

[8] N. D. Pham, Q. L. Le, and T. K. Dang, "Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases," in *2010 12th International Asia-Pacific Web Conference*, Apr. 2010, pp. 181–187. doi: 10.1109/APWeb.2010.23.

[9] P. Schäfer and M. Högqvist, "SFA: A symbolic fourier approximation and index for similarity search in high dimensional datasets," Mar. 2012, pp. 516–527. doi: 10.1145/2247596.2247656.

[10] M. M. Muhammad Fuad, "Genetic Algorithms-Based Symbolic Aggregate Approximation," in *Data Warehousing and Knowledge Discovery*, Berlin, Heidelberg, 2012, pp. 105–116. doi: 10.1007/978-3-642-32584-7_9.

[11] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard, "1d-SAX: A Novel Symbolic Representation for Time Series," in *Advances in Intelligent Data Analysis XII*, Berlin, Heidelberg, 2013, pp. 273–284. doi: 10.1007/978-3-642-41398-8_24.

[12] H. Ruan, X. Hu, J. Xiao, and G. Zhang, "TrSAX—An improved time series symbolic representation for classification," *ISA Transactions*, vol. 100, pp. 387–395, May 2020, doi: 10.1016/j.isatra.2019.11.018.

[13] U. Kamath, J. Lin, and K. De Jong, "SAX-EFG: An evolutionary feature generation framework for time series classification," *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, Jul. 2014, doi: 10.1145/2576768.2598321.

[14] 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, Aug. 2014, doi: 10.1016/j.neucom.2014.01.045.

[15] H. Yin, S. Yang, X. Zhu, S. Ma, and L. Zhang, "Symbolic representation based on trend features for knowledge discovery in long time series," *Frontiers Inf Technol Electronic Eng*, vol. 16, no. 9, pp. 744–758, Sep. 2015, doi: 10.1631/FITEE.1400376.

[16] C. T. Zan and H. Yamana, "An improved symbolic aggregate approximation distance measure based on its statistical features," in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, New York, NY, USA, Nov. 2016, pp. 72–80. doi: 10.1145/3011141.3011146.

[17] A. A. Ferreira *et al.*, "Adaptive Piecewise and Symbolic Aggregate Approximation as an Improved Representation Method for Heat Waves Detection," in *Intelligent Computing*, Cham, 2019, pp. 658–671. doi: 10.1007/978-3-030-01174-1_51.

[18] Z. He, S. Long, X. Ma, and H. Zhao, "A Boundary Distance-Based Symbolic Aggregate Approximation Method for Time Series Data," *Algorithms*, vol. 13, no. 11, Art. no. 11, Nov. 2020, doi: 10.3390/a13110284.

[19] K. Song, M. Ryu, and K. Lee, "Transitional SAX Representation for Knowledge Discovery for Time Series," *Applied Sciences*, vol. 10, no. 19, Art. no. 19, Jan. 2020, doi: 10.3390/app10196980.

[20] L. Pappa, P. Karvelis, G. Georgoulas, and C. Stylios, "Multichannel Symbolic Aggregate Approximation Intelligent Icons: Application for Activity Recognition," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2020, pp. 505–512. doi: 10.1109/SSCI47803.2020.9308497.

[21] "PapanikouMsc2008.pdf." Accessed: Jun. 25, 2021. [Online]. Available: https://dspace.lib.uom.gr/bitstream/2159/13370/2/PapanikouMsc2008.pdf

[22] 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, Aug. 2001, doi: 10.1007/PL00011669.

[23] E. Keogh, J. Lin, and A. Fu, "HOT SAX: efficiently finding the most unusual time series subsequence," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Nov. 2005, p. 8 pp.-. doi: 10.1109/ICDM.2005.79.

[24] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Min Knowl Disc*, vol. 15, no. 2, pp. 107–144, Oct. 2007, doi: 10.1007/s10618-007-0064-z.

[25] E. Keogh, L. Wei, X. Xi, S. Lonardi, J. Shieh, and S. Sirowy, "Intelligent Icons: Integrating Lite-Weight Data Mining and Visualization into GUI Operating Systems," in *Sixth International Conference on Data Mining (ICDM'06)*, Dec. 2006, pp. 912–916. doi: 10.1109/ICDM.2006.90.

[26] P. Karvelis, G. Georgoulas, I. P. Tsoumas, J. A. Antonino-Daviu, V. Climente-Alarcón, and C. D. Stylios, "A Symbolic Representation Approach for the Diagnosis of Broken Rotor Bars in Induction Motors," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1028–1037, Oct. 2015, doi: 10.1109/TII.2015.2463680.

[27] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Sydney, Australia, Mar. 2016, pp. 1–9. doi: 10.1109/PERCOM.2016.7456521.

[28] I. Pitas and A. N. Venetsanopoulos, "Median Filters," in *Nonlinear Digital Filters: Principles and Applications*, I. Pitas and A. N. Venetsanopoulos, Eds. Boston, MA: Springer US, 1990, pp. 63–116. doi: 10.1007/978-1-4757-6017-0_4.

[29] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition using Smartphones," 2013.

[30] M. Bansal, R. Sharma, and P. Grover, "Performance evaluation of Butterworth Filter for Signal Denoising," vol. 1, no. 1, p. 4, 2010.

[31] I. N. Junejo and Z. A. Aghbari, "Using SAX representation for human action recognition," *Journal of Visual Communication and Image Representation*, vol. 23, no. 6, pp. 853–861, Aug. 2012, doi: 10.1016/j.jvcir.2012.05.001.

[32] S. Kozina, M. Lusˇtrek, and M. Gams, "Dynamic signal segmentation for activity recognition," p. 6.

[33] J. Foster, M. Bevis, and S. Businger, "GPS Meteorology: Sliding-Window Analysis," *Journal of Atmospheric and Oceanic Technology*, vol. 22, no. 6, pp. 687–695, Jun. 2005, doi: 10.1175/JTECH1717.1.