Journal of Network Communications and Emerging Technologies (JNCET)
Volume 12, Issue 3, March (2022)

Adaptive Window Size Selection in Piecewise Aggregate Approximation on Univariate Time-Series Data

Dinesh R
Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Nagercoil, India
dineshmerein@gmail.com

Anitha A
Department of Computer Science and Engineering, Noorul Islam Centre for Higher Education, Nagercoil, India
anithadathi@gmail.com

ABSTRACT

Time series data need to represent to understand the data, reduce the dimensionality of the data and predict the future event. Many time series representation methods are available to present the time series. This research work took Piecewise Aggregate Approximation (PAA) for our research. PAA divides the time series into a number of the different and used aggregate function to reduce the dimensionality. The previous researchers used the window size of the PAA as a fixed value. It may lead to the wrong representation of the data. Our research suggested adaptive window size selection in piecewise aggregate approximation in time series representation. Our work is not with the PAA method but with adaptive window size selection of PAA. Our method first PAA and Simple Moving Average (SMA). Then do clustering analysis with Euclidean distance and cosine distance. Then choose adaptive PAA based on performance in similarity measures and processing time. The performance of the research work compared with different univariate time-series datasets.

Index Terms – Adaptive Piecewise Aggregate Approximation, Segmentation, Univariate Time Series, Time Series Analysis.

1. INTRODUCTION

The time series representation required to understand the trends or systemic pattern inside the organization over the period of time. Time series representations are used for the following purposes such as a considerable reduction in the dimensionality of the time series, a focus on the most basic (important) form qualities, treatment of noise implicitly. The memory needs and computational complexity of subsequent machine learning algorithms will be reduced when the dimension is reduced [1]. The time series representation approach was classified into four categories by Ratanamahatana et al. The following is a list of them. Data-adaptive, non-data-adaptive, model-based, and data-driven are all options [2].

The parameters of transformation in non-data adaptive representations are the same for all-time series, regardless of their type. The parameters of transformation in data adaptive representations vary based on the available data. The assumption behind a model-based representation is that the observed time series was constructed using a basic model. The goal is to determine the parameters of such a representational model. Two time series are deemed comparable if they were generated using the same set of fundamental model parameters. The compression ratio is automatically determined in data suggested techniques based on raw time series such as clipped [3].

PAA (Piecewise Aggregate Approximation) [4], DWT (Discrete Wavelet Transform) [5], DFT (Discrete Fourier Transform) [6], DCT (Discrete Cosine Transform) [7], and PIP (Piecewise Integral Approximation) are the most well-known (well-known) approaches for non-data adaptive representations (Perceptually Important Points). SAX (Symbolic Aggregate Approximation) [8], PLA (Piecewise Linear Approximation) [9], and SVD [10] are examples of data adaptive representations (Singular Value Decomposition). It's ARMA, mean profiles, or calculated regression coefficients from a statistical model for model-based representations (e.g. linear model). The data specified is a less well-known sort of representation, and clipping is the most well-known approach of this type [11][12].

2. RELATED WORK

Gonzalez-Vidal et al. [7] presented the BEAT model for time series segmentation, which splits time series data into variable number of blocks, merges them in square size matrices, and then computes and measures the Discrete Cosine Transform (DCT). BEAT is suitable for time series data from IoT-based devices and works with changing and multi-variable data variables. However, they are employing a segment with a set length and a sliding window size. It should, however, be dynamic in character and based on time series data. A data segmentation method that adjusts to unanticipated data alterations (i.e. data drifts). The optimization of the sliding window, on the other hand, was a work in progress. The segmentation length was fixed at 64 by the researchers. The sliding window was then set to 8. It might lead to erroneous predictions and representations of time series data.

Piecewise Linear Approximation (PLA) is a segmentation approach suggested by Duvignau, Romaric, et al. [13], which represents certain part of time series data by segments to reduce the volume of data transferred and stored by edge devices. It compresses time-series data and compares bounded accuracy loss to storage space savings. It reduces network transmission bandwidth while also resolving the aforementioned difficulties. PLA is used to reduce the amount of data in huge IoT-based streaming data sets. When compared to baseline PLA approaches, the techniques considerably reduce both the reconstructed stream's latency and individual mistakes. However, it does not address the algorithm's temporal complexity.

Havers et al. [14] proposed DRIVEN to encode the amounts of data to be collected using streaming-based error-bounded approximation via Piecewise Linear Approximation (PLA), which represents a sequence of time-stamped records by a sequence of line segments while maintaining the approximation error within a satisfactory error bound. By having autos send compact information via Piecewise Linear Approximation - PLA, the DRIVEN framework minimizes the amount of data gathered from them. Online clustering algorithms have addressed the constraints of batch-based clustering techniques. It makes advantage of the data streaming idea to achieve transparent distributed and parallel deployments. Data may be compressed to 10-35 percent of its original size, drastically reducing the time it takes to collect large volumes of data. It does, however, suffer from a loss of clustering precision.

Roonak Rezvani et al. [15] proposed a pattern representation method for aggregating and representing the original time-series data. First, they apply piecewise aggregate approximation to decrease the dimensionality in the time-series data. Second, they apply lagrangian multiplier gives vector representation used to analyze patterns and changes in the time-series data. They used Singular Spectrum Analysis (SSA) algorithm to give the lagrangian multiplier operation smoother. It gave better results against BEATS and SAX. But they considered the segmented slide window size was fixed. It may lead to wrong representation for some cases in time-series data.

H. Jiang et. al [16] proposed representation method TS-GLR (Time Series- Global trends and Local details Representation) to find tiny fluctuations in Adversarial Examples for time series data which extracts visual relationships of data points in global trends and local details converted into weighted complex networks and then created measurement called tensors to find Adversarial Examples with machine learning methods. This method can identify small fluctuations in adversarial time series examples and normal time-series samples but the complexity of the method is high.

Yongqiang Tang et al. [17] proposed time series clustering under a unified multiple kernels clustering (MKC) framework which first converts the raw time series space into multiple kernel spaces using elastic distance measure functions. Then apply tensor constraint-based self-representation clustering approach to cluster low and high dimensional structure of the data. The research work addresses several issues related to time series clustering such as high-dimension, warping, and the integration of multiple elastic measures. The adaptive learning technique for the critical attributes is not studied. The relationships between nearby timestamps have to elaborate detail.

3. PROPOSED METHOD

Our proposed system gives solutions to the existing system in time series representation method. Out work is focus on fixed window sliding problem in existing system and extending our work with univariate time series data. Our research work proposed Adaptive window size selection in piecewise aggregate Approximation in pattern representation Methods for time series data. Our proposed method allows an original times series of length n to be represented as unit vector w ($w<n$) dimensions. The proposed method first decreases the length of the time series using the adaptive aggregate approximation method. Then find piecewise aggregate approximation and simple moving average from original dataset. Then, apply the clustering method, the Euclidean and Cosine Distance are calculated against piecewise aggregate approximation and simple moving average. The processing time is calculated for piecewise aggregate approximation against the different window size for different dataset. After that, evaluate the efficiency of our proposed system.

3.1. Adaptive Window Size Selection

The adaptive window size selection technique is used to selection right window size to divide the times series in piecewise aggregated approximation.

Algorithm: adaptive window size technique

Input: time series data as numeric vector

Output: adaptive window size selection

1: ts ← times series

2: n ← length of time series

3: sws ← sliding window size

4: FOR (x:3; x<=$\log_2$ (n)-2; x++)

sws <-2^x

5: find PAA

6: find SMA

8: find Euclidean Distance

9: find Cosine Distance

10: find Processing Time

11: for end

12: choose window size (ws) corresponding to optimum value

13: end

The term "univariate time series" refers to a time series that consists of single observations recorded sequentially over equal time increments. Our research work is used the four univariate time series datasets that taken from the domain such as sales of products, weather, Physical Science and anthropology. These datasets are univariate time series dataset named as "Time Series Data Library" produced by Rob Hyndman, Professor of Statistics at Monash University, Australia. These data are very simple and easy to understand.

A simple moving average is a time series average that moves through the series by deleting the top items of the previous averaged group and adding the next in each consecutive average. The length of a line segment connecting two locations in Euclidean space is called the Euclidean distance. It is also referred to as the Pythagorean distance since it can be determined from the Cartesian coordinates of the locations using the Pythagorean Theorem. Cosine similarity is a statistic that may be used to determine how similar data items are regardless of their size. The cosine similarity is advantageous because, despite the fact that the two identical data items are separated by the Euclidean distance due to their size, they may have a lower angle between them. The greater the resemblance, the smaller the angle. The cosine similarity captures the orientation (angle) of the data items when displayed on a multi-dimensional space, not the magnitude.

4. IMPLEMENTATION

4.1. Experiment Setup

The paper work is used R programming language for implementation. The Integrated Development Environment used is R Studio (2022-03010), R version is 4.1.3, The package used are TSrepr for PAA, SMA calculation, time series for manage timeseries, microbenchmark for processing time calculation, philentropy for distance calculation and ggplot2 for visualizations. The windows 10 operating system with Intel(R) Core (TM) i7-1165G7 @ 2.80GHz processor and16 GB RAM are used.

4.2. Datasets

The term "univariate time series" refers to a time series that consists of single observations recorded sequentially over equal time increments.

4.2.1. Shampoo Sales Dataset (SSD)

Over a three-year period, this dataset depicts the monthly number of shampoo sales. There are 36 observations and the units represent a sales count. Makridakis, Wheelwright, and Hyndman are the authors of the original dataset (1998). The first six rows of data, including the header row, are shown below in Figure 1.

```
> head(shampoodata)
  Month Sales
1  1-01 266.0
2  1-02 145.9
3  1-03 183.1
4  1-04 119.3
5  1-05 180.3
6  1-06 168.5
```

Figure 1 Head of Shampoo Dataset

Figure 2 show that original Shampoo sale dataset in graph the x axis represents date and y axis represents sales of Shampoo values in numbers.

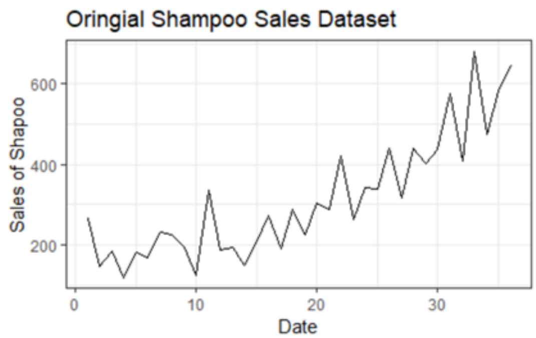

Figure 2 Original Shapoo Sales Dataset Graph

4.2.2. Minimum Daily Temperatures Dataset (MDTD)

This dataset depicts the lowest daily temperatures in Melbourne, Australia, for a ten-year period (1981-1990). There are 3650 observations and the units are in degrees Celsius. The Australian Bureau of Meteorology is acknowledged as the data's source. The first six rows of data, including the header row, are shown below in Figure 3.

```
> head(minTempdata)
        Date Temp
1 1981-01-01 20.7
2 1981-01-02 17.9
3 1981-01-03 18.8
4 1981-01-04 14.6
5 1981-01-05 15.8
6 1981-01-06 15.8
```

Figure 3 Display Shapoo Sales Dataset with Head ()

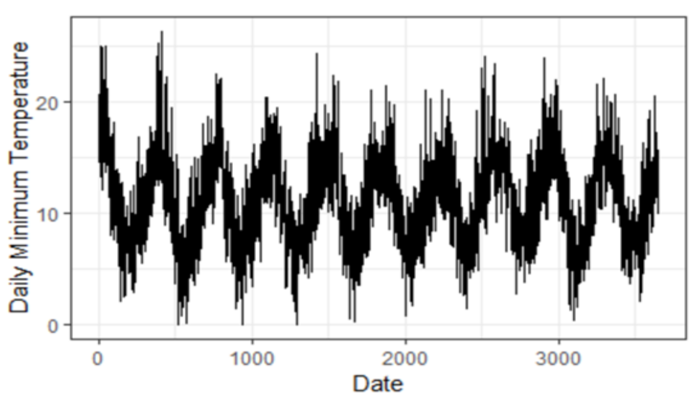

Figure 4 Daily Minimum Temperature Dataset Graph

4.2.3. Monthly Sunspot Dataset (MSD)

This dataset describes a monthly count of the number of observed sunspots for just over 230 years (1749-1983). The units are a count and there are 2,820 observations. The source of the dataset is credited to Andrews & Herzberg (1985). The first six rows of data, including the header row, are shown below in Figure 5.

```
> head(msspotdata)
    Month Sunspots
1 1749-01     58.0
2 1749-02     62.6
3 1749-03     70.0
4 1749-04     55.7
5 1749-05     85.0
6 1749-06     83.5
```

Figure 5 Monthly Sunspot Dataset with head()

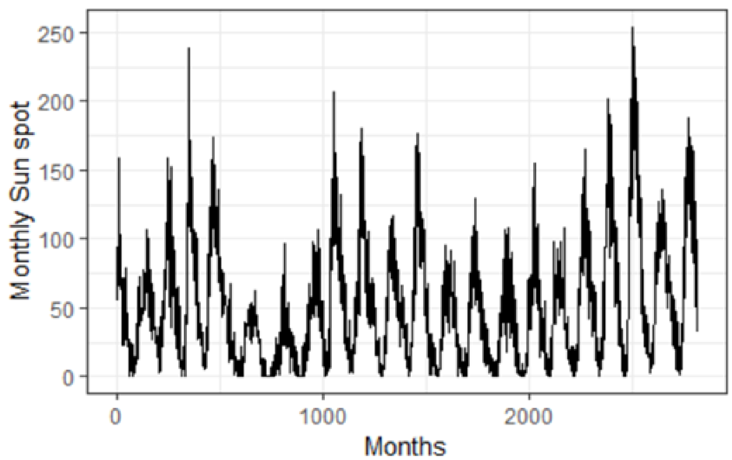

Figure 6 Monthly Sunspot Dataset Graph

4.2.4. Daily Female Births Dataset (DFBD)

In 1959, the number of daily female births in California was recorded in this dataset. There are 365 observations and the units are counts. Newton is acknowledged as the dataset's creator (1988). The first six rows of data, including the header row, are shown below in Figure.7 which have date and births variables.

```
> head(dtfbdata)
        Date Births
1 1959-01-01     35
2 1959-01-02     32
3 1959-01-03     30
4 1959-01-04     31
5 1959-01-05     44
6 1959-01-06     29
```

Figure 7 Daily Female Birth Dataset

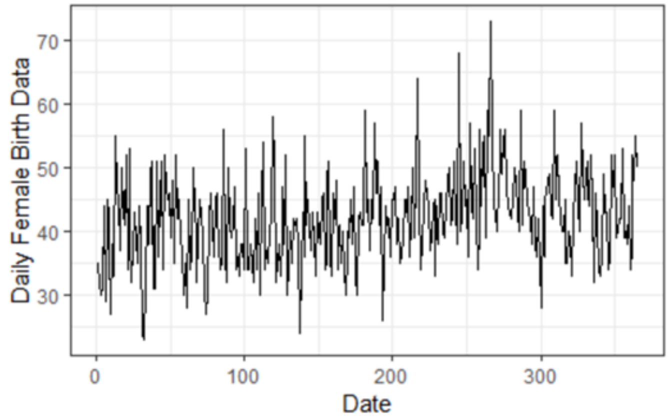

Figure 8 Daily Female Birth Dataset Graph

The adaptive piecewise aggregate approximation algorithm run datasets such as SSD, MDTD, MSD, DFBD as input. Piecewise Aggregated Approximation (PAA) and Simple Moving Average (SMA) is calculated using repr_paa and repr_sma function in TSrepr [1]. Let us see the implementation result of each dataset one by one in detail manner in next section.

5. RESULTS AND DISCUSSIONS

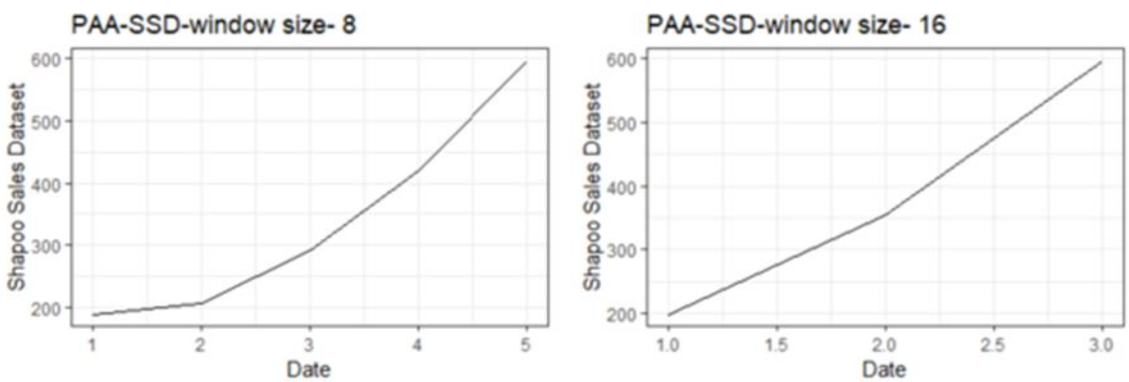

Figure 9 PAA-Shampoo Sales Dataset Segmenting with Window Size 8, 16

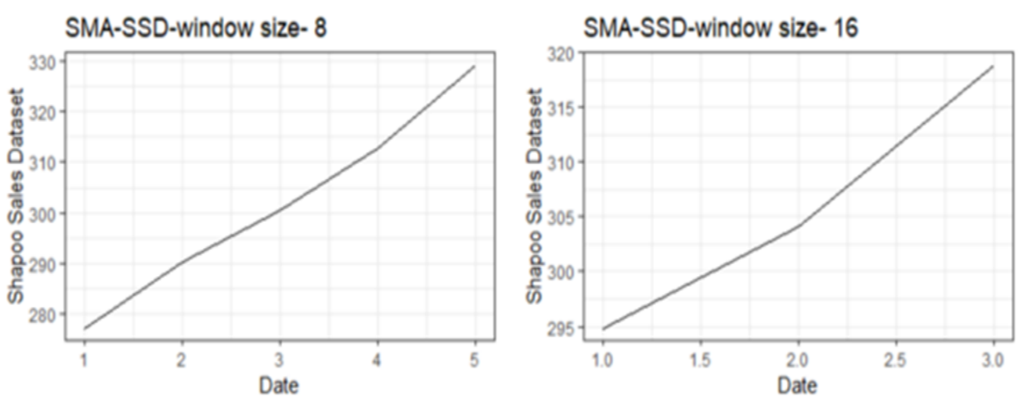

Figure 10 SMA-Shampoo Sales Dataset segmenting with Window Size 8, 16

Figure 9 shows the Piecewise Aggregate Approximation of the Shampoo Sales Dataset (SSD) segmentation graph by window size of 8 and window size of 16. The original SSD dataset graph is shown in figure 2 of size 36. The original dataset is segmented as number of pieces using the window size by 8, 16. The output of the calculation is plotted as a graph in Figure 9.

Figure 10 shows the Simple Moving Average of the Shampoo Sales Dataset (SSD) segmentation graph by window size of 8 and window size of 16. The original SSD dataset graph is shown in figure 2 of size 36. The original dataset is segmented as number of pieces using the window size by 8, 16. The output of the calculation is plotted as a graph in Figure 10.

Figure 11 shows the Piecewise Aggregate Approximation of the Minimum Daily Temperature Dataset (MDTD) segmentation graph by window size of 8, 16, 32, 64, 128, 256 and 512. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64, 128, 256 and 512. The output of the calculation is plotted as a graph in Figure 11.

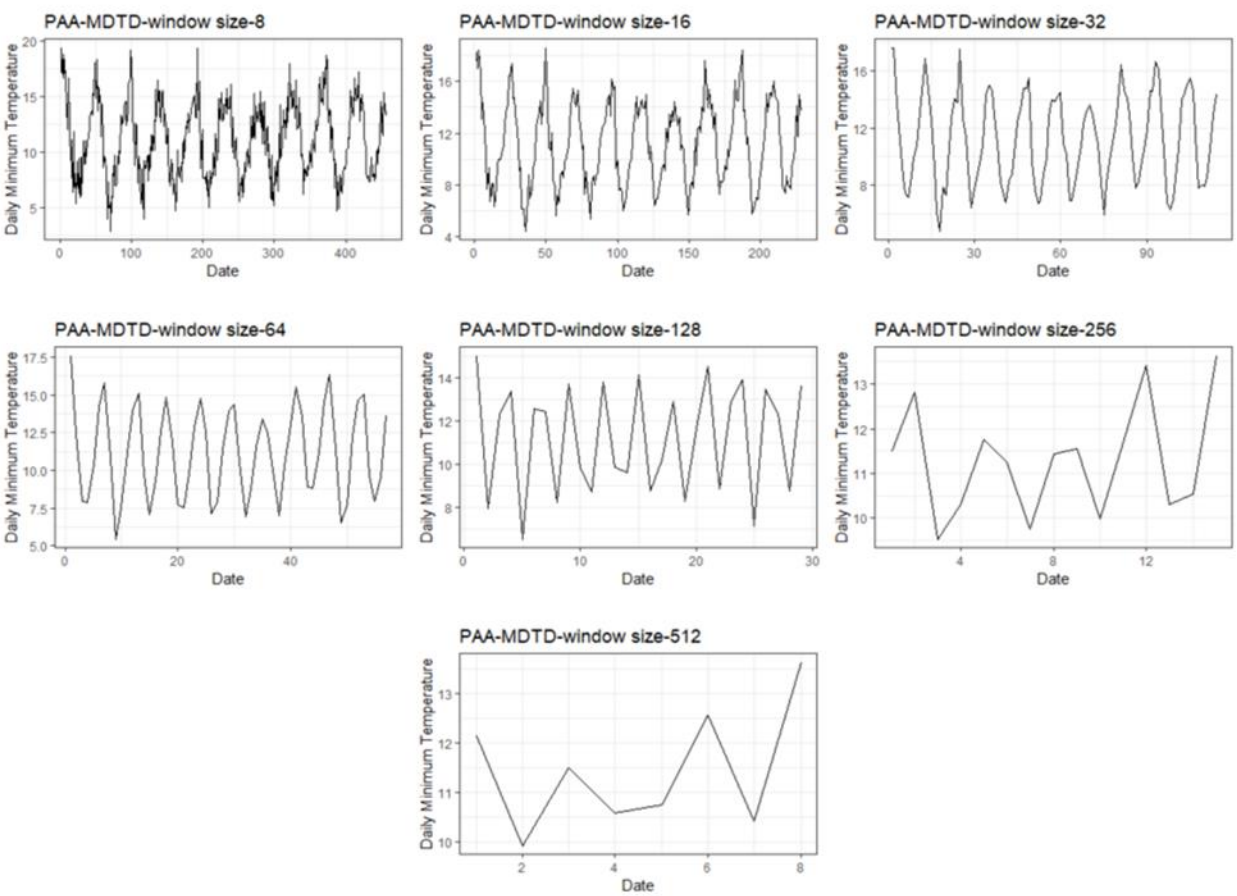

Figure 11 PAA-Minimum Daily Temperature Dataset-Window Size Range 8, 16, 32, 64, 128, 256, 512

Figure 12 shows the Simple Moving Average of the Minimum Daily Temperature Dataset (MDTD) segmentation graph by window size of 8, 16, 32, 64, 128, 256 and 512. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64, 128, 256 and 512. The output of the calculation is plotted as a graph in Figure 12.

Figure 13 shows the Piecewise Aggregate Approximation of the Monthly Sunspot Dataset (MSD) segmentation graph by window size of 8, 16, 32, 64, 128, 256, 512 and 1024. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64, 128, 256, 512 and 1024. The output of the calculation is plotted as a graph in Figure 13.

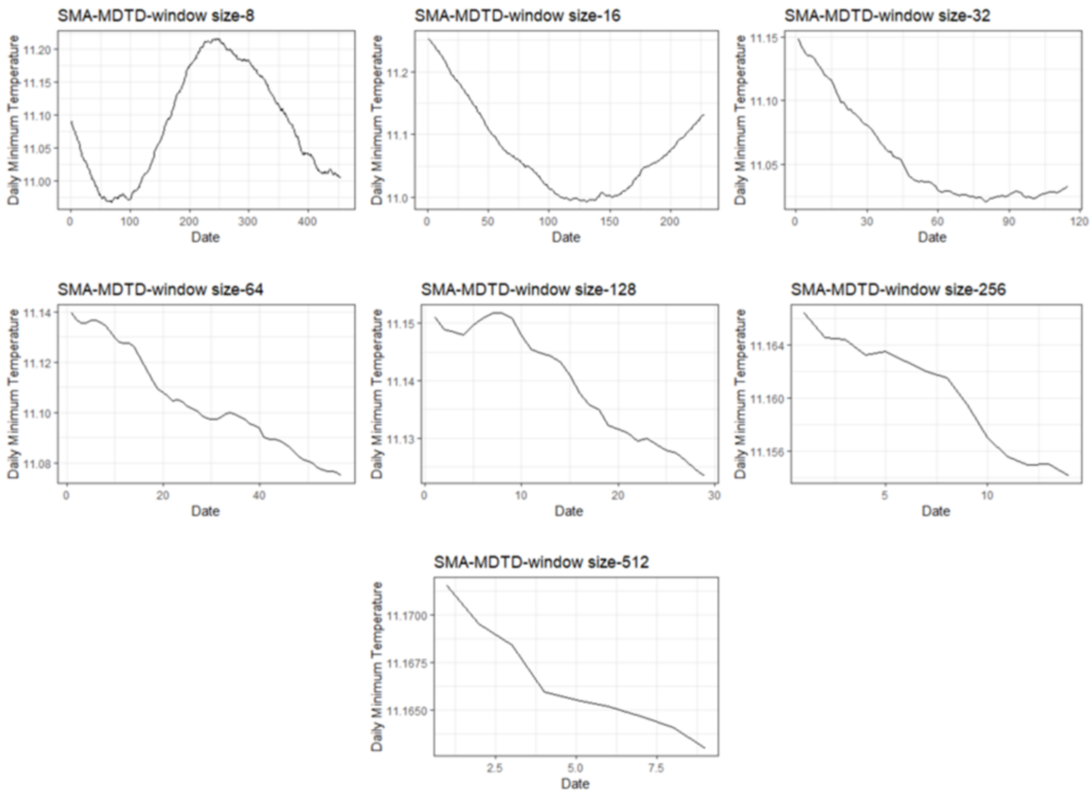

Figure 12 SMA-Minimum Daily Temperature Dataset-Window Size Range 8, 16, 32, 64, 128, 256, 512

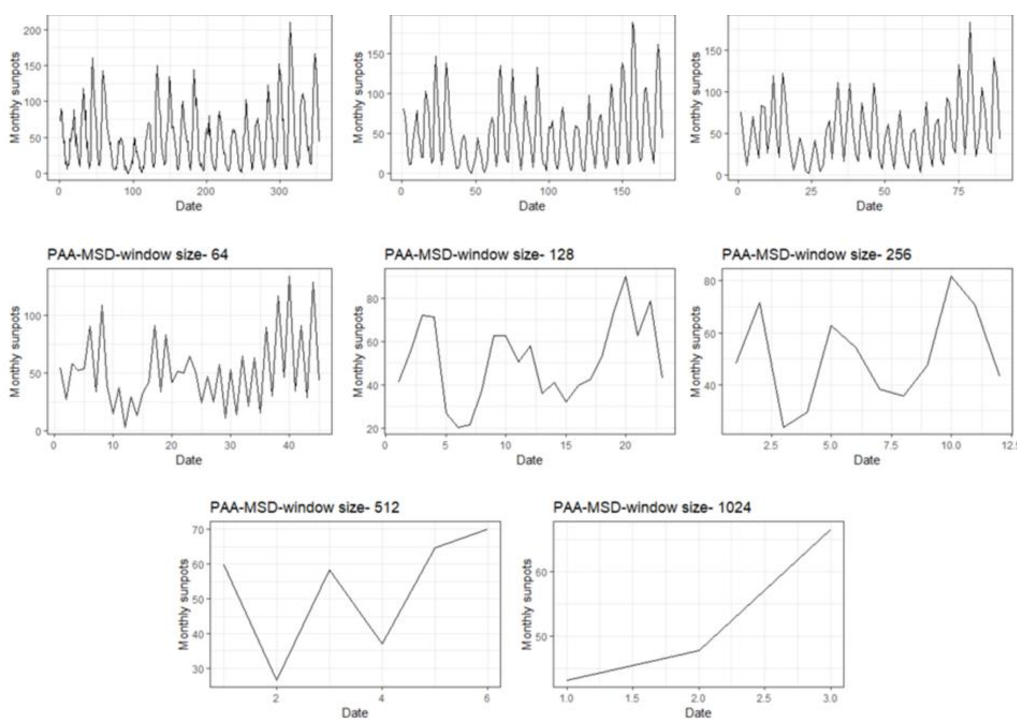

Figure 13 PAA-Monthly Sunspots Dataset Segmenting Window Size of 8, 16, 32, 64, 128, 256, 512, 1024

Figure 14 shows the Simple Moving Average of the Monthly Sunspot Dataset (MSD) segmentation graph by window size of 8, 16, 32, 64, 128, 256, 512 and 1024. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64, 128, 256, 512 and 1024. The output of the calculation is plotted as a graph in Figure 14.

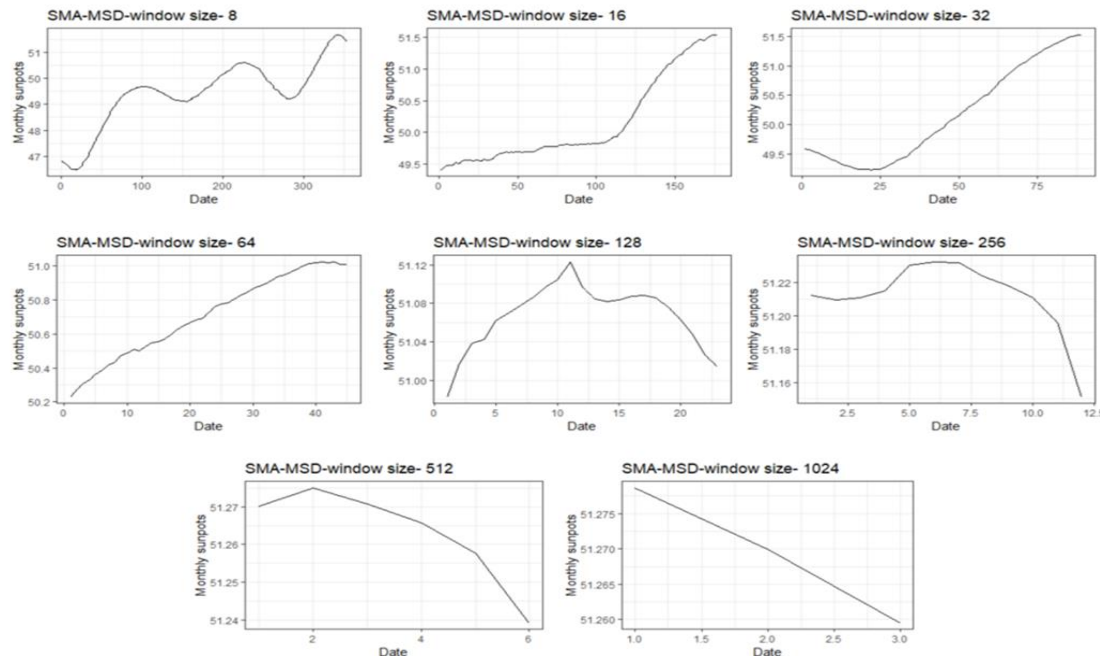

Figure14 SMA-Monthly Sunspots Dataset Segmenting Window Size of 8, 16, 32, 64, 128, 256, 512, 1024

Figure 15 shows the Piecewise Aggregate Approximation of the Daily Female Birth Dataset (DFBD) segmentation graph by window size of 8, 16, 32, 64 and 128. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64 and 128. The output of the calculation is plotted as a graph in Figure 15.

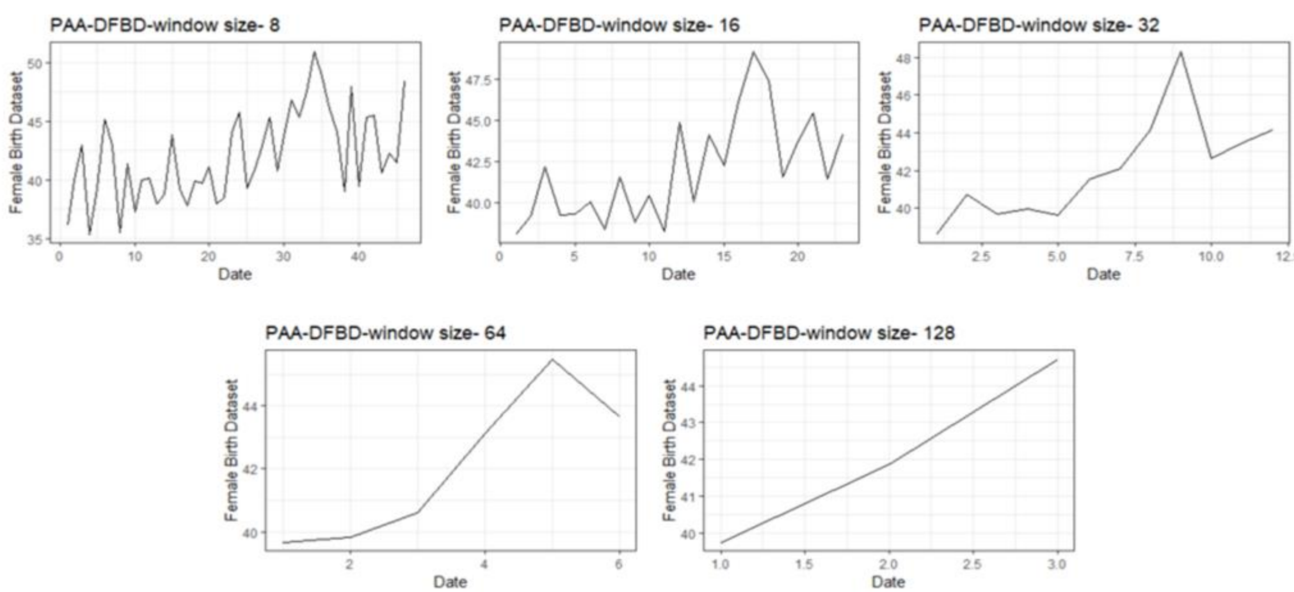

Figure15 PAA-Daily Female Birth Dataset Segmenting Window Size 8, 16, 32, 64, 128

Figure 16 shows the Simple Moving Average of the Daily Female Birth Dataset (DFBD) segmentation graph by window size of 8, 16, 32, 64 and 128. The original SSD dataset graph is shown in figure 2 of size 3650. The original dataset is segmented as number of pieces using the window size by 8, 16, 32, 64 and 128. The output of the calculation is plotted as a graph in Figure 16.

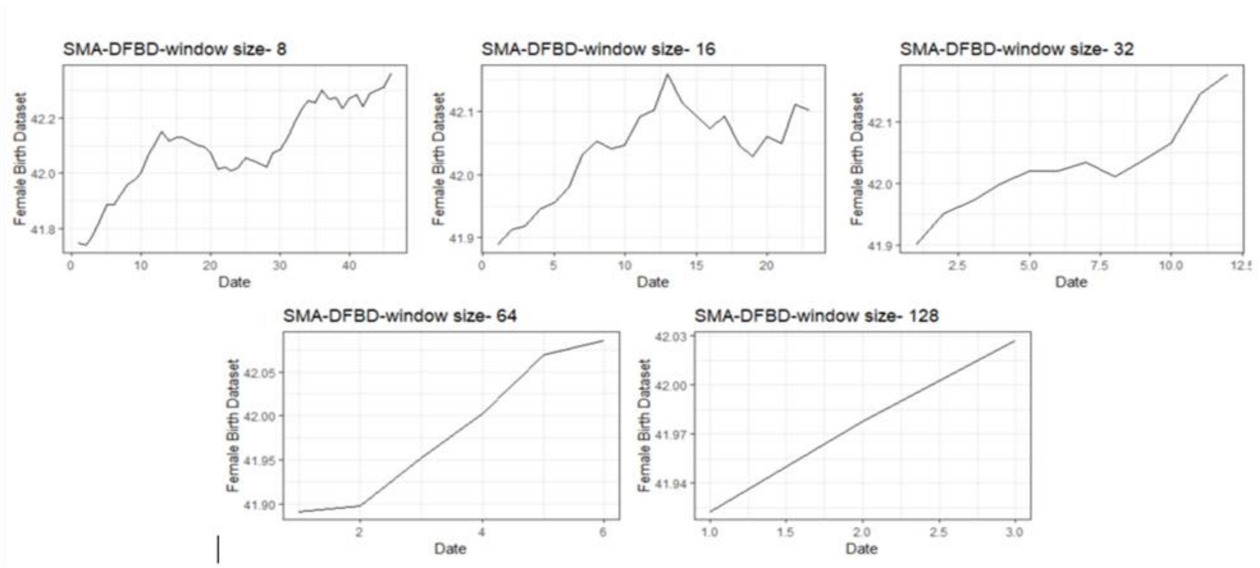

Figure16 SMA-Daily Female Birth Dataset Segmenting Window Size 8, 16, 32, 64, 128

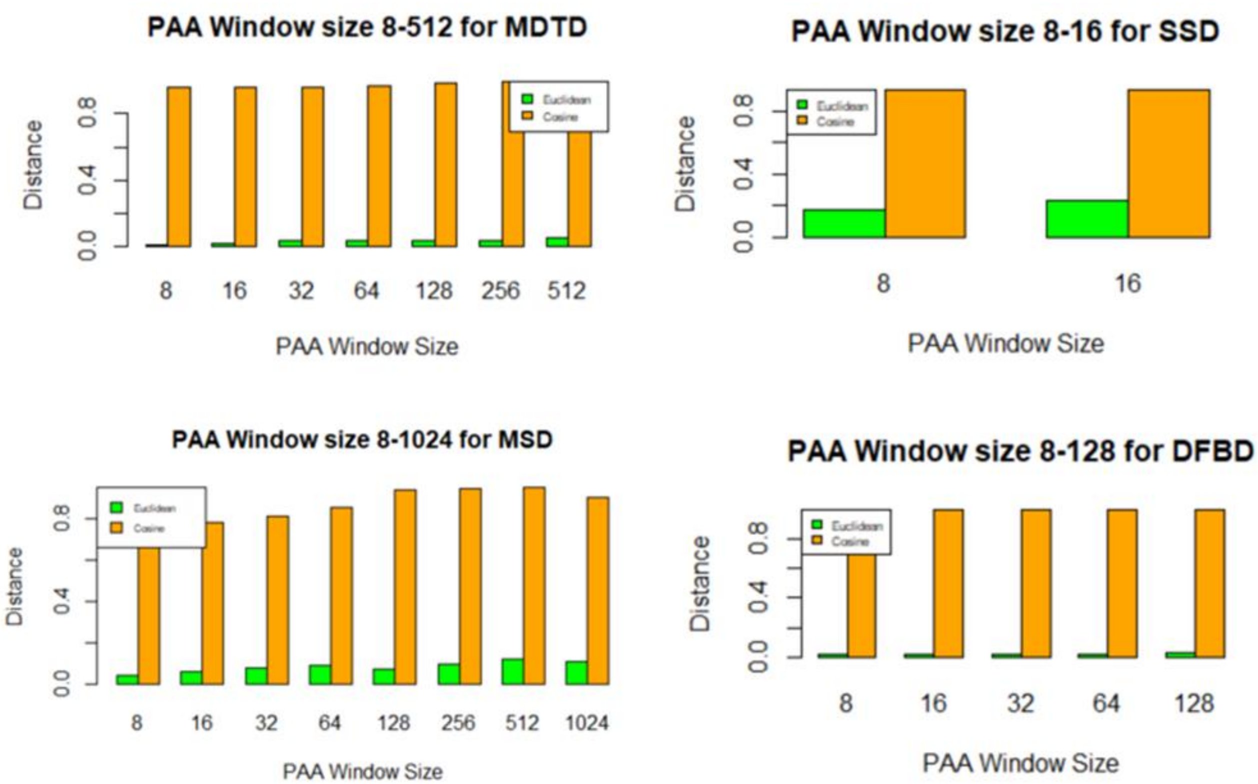

Figure17 PAA for Different Window Size with Euclidean and Cosine Distance

Figure 17 shows the Euclidean and Cosine Distance of different dataset with their own window size in the graph. The chart indicates the while increasing the window size the Euclidean distance and Cosine Distance also increases. The dataset MDTD divided into window size range from 8, 16, 32, 64, 128, 256, 512 compare similarity distance Euclidean, Cosine. The dataset SSD divided into window size of 8 and 16 compare similarity distance Euclidean, Cosine. The dataset MSD is divided into window size of 8, 16, 32, 64, 128, 256, 512, 1024 compare similarity distance Euclidean, Cosine. The dataset DFBD is divided into 8, 16, 32, 64 and128 compare similarity distance Euclidean, Cosine.

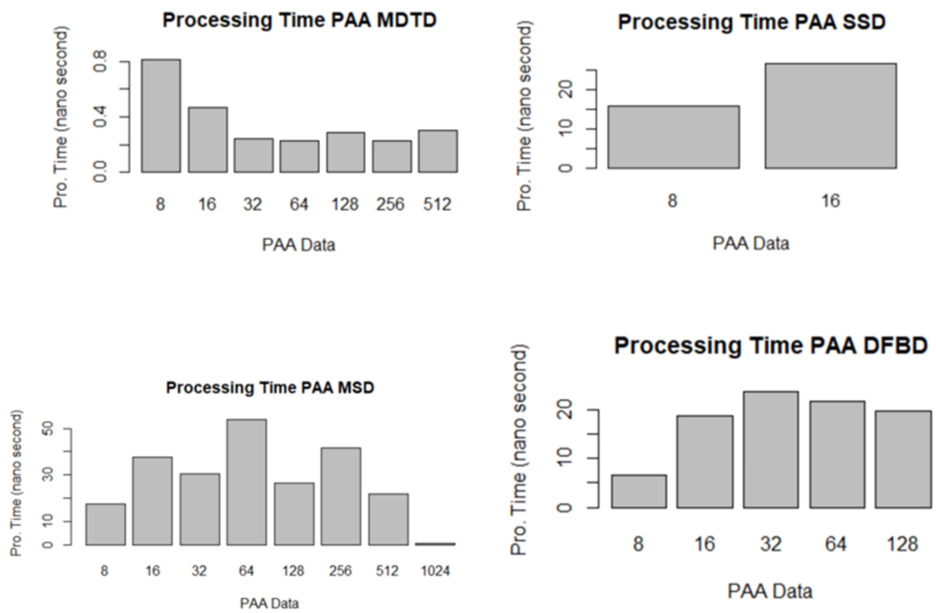

Figure18 PAA Processing Time for Different Dataset with Window Size

Figure 18 shows that execution time of the different dataset against piecewise aggregate approximation. The execution time is measured in nano second. The dataset MDTD divided into window size range from 8, 16, 32, 64, 128, 256, 512 compare with processing time in nano second. The dataset SSD divided into window size of 8 and 16 compare with processing time in nano second. The dataset MSD is divided into window size of 8, 16, 32, 64, 128, 256, 512, 1024 compare with processing time in nano second. The dataset DFBD is divided into 8, 16, 32, 64 and128 compare with processing time in nano second.

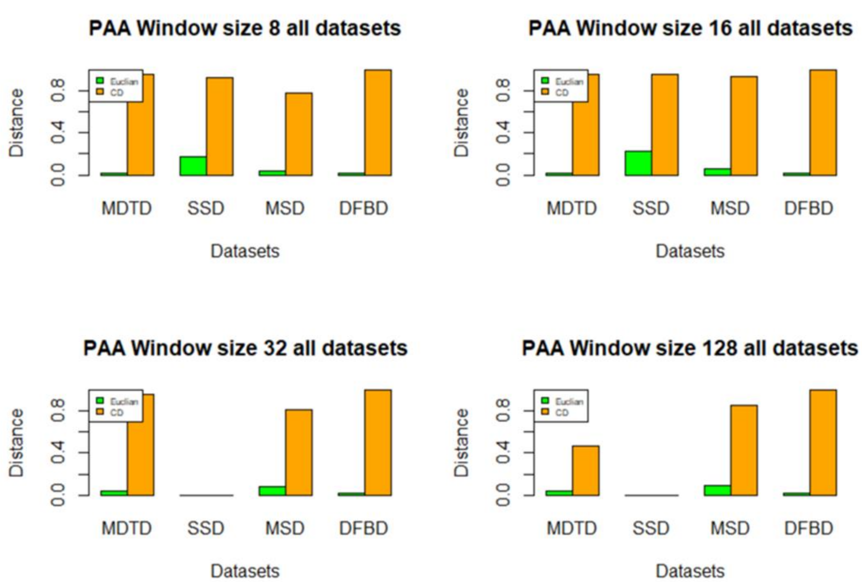

Figure19 PAA Euclidean and Cosine Distance vs Dataset with Different Segmenting Window Size

Figure 19 shows PAA Euclidean and Cosine Distance vs Dataset with different segmenting window size of 8, 16,32,128 graph. Cosine Distance is high value compare with the Euclidean distance.

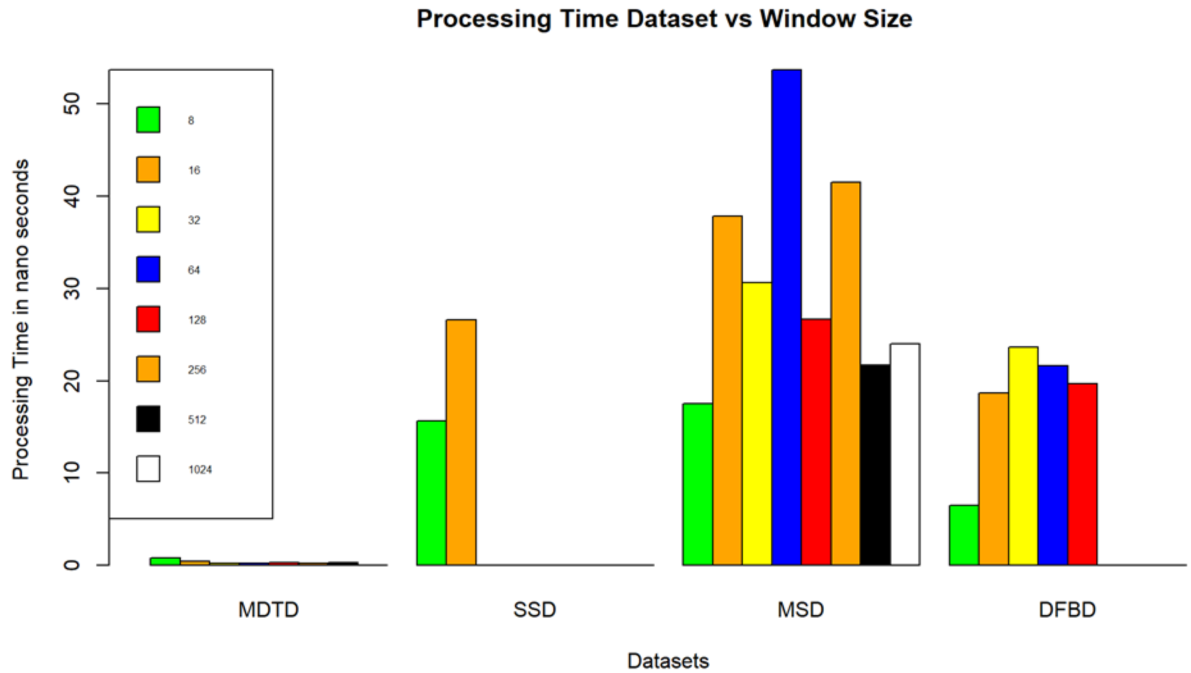

Figure 20 Processing Time vs Dataset with Different Segmenting Window Size

The processing time is the important factor to find the performance of the calculation of the expression or function. Figure 20 shows the overall processing time in nano second with different PAA window size 8, 16,32, 64, 128, 256,512 and 1024 against MDTD, SSD, MSD, DFBD datasets.

Table 1 Processing Time against PAA Window Size for Different Dataset

Window Size	MDTD	SSD	MSD	DFBD
8	.81	15.65	17.53	6.53
16	.47	26.6	37.81	18.65
32	.24	-	30.63	23.62
64	.23	-	53.69	21.65
128	.29	-	26.69	19.65
256	.23	-	41.54	-
512	.30	-	21.71	-
1024	-	-	24	-

The table1 shows the processing time in nano second against the different piecewise aggregate approximation for different univariate dataset.

The research work proposed the adaptive window size selection of piecewise aggregate approximation implemented using univariate time series data for representing time series. In earlier research work, the researchers window size of piecewise aggregate approximation as fixed value. Our research work is proposed a method to choose adaptive window size for piecewise aggregate approximation.

The shampoo Sales dataset (SSD), Monthly Sunspot Dataset (MSD), Minimum Daily Temperature Dataset (MDTD), Daily Female Birth Dataset (DFBD) are used for experiment. The adaptive window size selection algorithm first generates different ranges of window size for each dataset. Then Piecewise Aggregate Approximation (PAA), Simple Moving Average (SMA) is calculated based on the ranges of window size for different dataset. The euclidean distance, cosine distance is calculated from PAA and SMA. The optimize value corresponding to the window size is selected as adaptive window size for particular dataset. Our research results show 32 window size for SSD, 32 window size for MSD, 16 window size for MDTD, 32 window size for DFBD.

6. CONCLUSION

The proposed adaptive piecewise aggregate approximation method is implemented with univariate time series dataset Shampoo Sales Dataset, Minimum Daily Temperature Dataset, Monthly Sunspot Dataset and Daily Female Birth Dataset. Our research work gives solution to the fixed window size selection problem in piecewise aggregate approximation earlier researches. The results are discussed and suggested the adaptive window size for each dataset. Our research may act differently for multivariate time series dataset. Further research required for non-stationary time series dataset. In future, our plan is to extend our research work dynamic IOT based stream environment and multivariate time series and also, we can study the prediction performance with this work.

REFERENCES

[1] https://cran.r-project.org/web/packages/TSrepr/vignettes/ TSrepr_representations_of_time_series.html
[2] Bagnall, Anthony, Chotirat Ratanamahatana, Eamonn Keogh, Stefano Lonardi, and Gareth Janacek. 2006. “A bit level representation for time series data mining with shape based similarity.” Data Mining and Knowledge Discovery 13 (1): 11–40.
[3] Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. “Time-series clustering - A decade review.” Information Systems 53: 16–38.
[4] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra,“Dimensionality reduction for fast similarity search in large time series databases,” Knowl. Inf. Syst., vol. 3, pp. 263–286, 2001.
[5] K.-P. Chan and W.-C. Fu, “Efficient time-series matching by wavelets,” in Proc. 15th Int. Conf. Data Eng., 1999, Art. no. 126.
[6] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, “Fast subsequence matching in time-series databases,” ACM SIGMOD Rec., vol. 23, no. 2, pp. 419–429, 1994.
[7] A. Gonzalez-Vidal, P. Barnaghi, and A. F. Skarmeta, “BEATS: Blocks of eigenvalues algorithm for time-series segmentation,” IEEE Trans. Knowl. Data Eng., vol. 30, no. 11, pp. 2051–2064,Nov. 2018.
[8] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, “A symbolic representation of time-series, with implications for streaming algorithms,” in Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery, 2003, pp. 2–11.
[9] R. Duvignau, V. Gulisano, M. Papatriantafilou, and V. Savic,“Streaming piecewise linear approximation for efficient data management in edge computing,” in Proc. 34th ACM/SIGAPP Symp. Appl. Comput., 2019, pp. 593–596.
[10] D. Wu, A. Singh, D. Agrawal, A. El Abbadi, and T. R. Smith,“Efficient retrieval for browsing large image databases,” in Proc.5th Conf. Inf. Knowl. Manage., 1996, pp. 11–18.
[11] Bagnall, Anthony, Chotirat Ratanamahatana, Eamonn Keogh, Stefano Lonardi, and Gareth Janacek. 2006. “A bit level representation for time series data mining with shape based similarity.” Data Mining and Knowledge Discovery 13.
[12] Tang, Yongqiang, et al. "Tensor multi-elastic kernel self-paced learning for time series clustering." IEEE Transactions on Knowledge and Data Engineering (2019).
[13] Duvignau, Romaric, et al. "Streaming piecewise linear approximation for efficient data management in edge computing." Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. 2019.
[14] Havers, Bastian, et al. "Driven: a framework for efficient data retrieval and clustering in vehicular networks." 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, 2019.
[15] R. Rezvani, P. Barnaghi and S. Enshaeifar, "A New Pattern Representation Method for Time-Series Data," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 7, pp. 2818-2832, 1 July 2021, doi: 10.1109/TKDE.2019.2961097.
[16] H. Jiang, H. Nai, Y. Jiang, W. Du, J. Yang and L. Wu, "An Adversarial Examples Identification Method for Time Series in Internet-of-Things System," in IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9495-9510, 15 June15, 2021, doi: 10.1109/JIOT.2020.3005688.
[17] Laurinec, Peter, and Mária Lucká. 2016. “Comparison of Representations of Time Series for Clustering Smart Meter Data.” In Lecture Notes in Engineering and Computer Science: Proceedings of the World Congress on Engineering and Computer Science 2016, 458–63.

Authors

Dinesh Raja received the MCA degree from Anna University Chennai, India. He is currently pursuing an ME in Computer Science and Engineering from Noorul Islam Centre For Higher Education, India. His areas of interest include time series analysis, data science, artificial intelligence, and machine learning.

Dr. Anitha A has completed her Ph.D from the Noorul Islam Centre for Higher Education and has completed her M.E Computer Science and Engineering from the Dr. Sivanthi Adithanar College of Engineering. She has teaching experience for 13 years and research experience for 11 years. Her specialised areas in research are Wireless Communication and Networking, Image Processing, Network Security. She has the credit of having published 13 research papers in Web of Science/Scopus Indexed Journals and 6 papers in UGC CARE listed, refereed and reviewed journals. She has guided 0 Ph.D scholars and she is now guiding 6 Ph.D scholars.