ASTD: Automatic Seasonal-Trend Decomposition for Time Series

Bowen Chen[1,2], Hancheng Lu[2],Yuang Chen[2], Haoyue Yuan[2], and Minghui Wang[3]
[1] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
[2] University of Science and Technology of China, Hefei, China
[3] AI Institute of H3C Technologies Co., Ltd.,
{chenbowen, yuangchen21, yhyue}@mail.ustc.edu.cn, hclu@ustc.edu.cn, mhwang@h3c.com

***Abstract*—The rapid and accurate decomposition of multi-period time series is crucial for reliable forecasting, anomaly detection, and classification. However, the traditional approach of first detecting the periodicity and then selecting from a range of decomposition algorithms based on the periodicity results in inefficiencies and complexity. To address this challenge, we propose the automatic seasonal-trend decomposition (ASTD), a unified method for automatic time series decomposition. With the ASTD, users no longer need to worry about whether their time series is multi-period, single-period, or aperiodic. They simply provide the time series, and the ASTD automatically returns the final decomposition results. Careful consideration of runtime cost and accuracy requirements has been taken in the design of the ASTD, which has an overall time complexity of O($NlogN$). Extensive experimental results show that the proposed ASTD outperforms other state-of-the-art decomposition algorithms in terms of minimum mean square error (MSE) and mean absolute error (MAE). Notably, when applied to the Taylor dataset, the ASTD is approximately 3 times faster than other baseline decomposition algorithms.**

***Index Terms*—time series, periodic detection, seasonal-trend decomposition**

I. Introduction

The digital age has led to an increase in recorded time series, such as Central Processing Unit (CPU) usage and request numbers in Operation and Maintenance, spending and purchasing indices in economics, and daily electricity consumption. Many time series have periodic behavior, such as human heartbeat and rush-hour traffic, as well as tides related to lunar and solar cycles. Accurate time series decomposition is essential for tasks like anomaly detection [1]–[4], forecasting [5]–[8], and classification [9]–[12], and can improve prediction accuracy by extracting seasonal and trend components. Many decomposition algorithms have been proposed for time series, including single-period and multi-period algorithms. Single-period algorithms include STL [13] and RobustSTL [14], while multi-period algorithms include MSTL [15], Fast RobustSTL [16], STR [17], and TBATS [18]. Non-periodic time series can have their trend extracted using filters like HP [19], Supermoother [20], and $l1$ [21] trend filter . If unsure of the type of time series, beginners can use periodic detection tools to identify the periodicity and choose the corresponding algorithm.

There are two types of periodic detection tools: single period and multiple periods. Nevertheless, existing algorithms can be affected by outliers, noise, and spectral leakage, which can result in inaccurate periodicity detection and decomposition. Auto-period [22] may select incorrect periods due to spectral leakage and limited resolution. CFD-Autoperiod [23] addresses this issue with clustering, filtering, and detrending, but noise can still cause the centroid to be offset from the actual period. Robust-period [24] uses Huber-periodogram and Huber autocorrelation function (Huber-ACF) to mitigate outlier effects, but it requires multiple iterations, making it time-consuming.

Seasonal-trend decomposition methods have been proposed to analyze periodic information, but STL is only suitable for single-period time series. MSTL was introduced in 2021 to handle multi-period time series, but it may include data from other seasons. Hyndman proposed STR and TBATS for robustness to outliers in multi-period time series, but they have high computational complexity. Robust-STL was introduced to handle outliers, noise, and abrupt trend changes, but it is not suitable for multi-period time series. To address this limitation, Fast Robust-STL was introduced, which extends Robust-STL and reduces computational complexity, but still requires multiple iterations. Existing methods for time series only focus on either periodic detection or decomposition, making it inconvenient and inefficient for users.

To address these issues, we propose a unified method called ASTD for automatic time series decomposition, which detects and decomposes time series regardless of their periodic characteristics. ASTD uses a pruning method and partition idea to detect the period robustly and accurately, while a comprehensive score is calculated to mitigate spectral leakage effects. For decomposition, ASTD uses the input data spectrum to extract trends and seasons and the spectral residual (SR) [25] to obtain complete and accurate residuals. ASTD greatly expands its application in long time series and speeds up decomposition.

Our main contributions are listed as follows:

- We propose ASTD, a unified method for the automatic decomposition of periodic time series. With ASTD, users are no longer required manually detect the periodicity of time series and then select different decomposition algorithms according to the detection results.
- In ASTD, we propose a low-complexity and high-accuracy algorithm for detecting periodicity in multi-

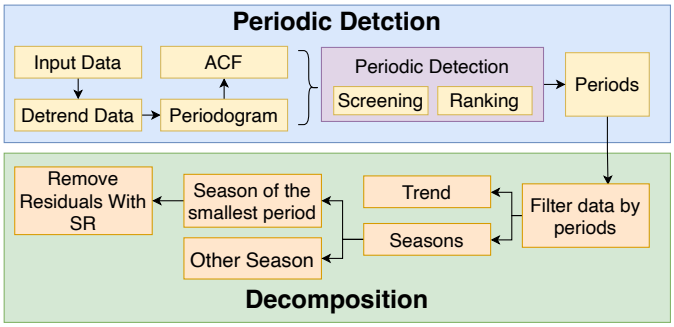

Fig. 1. Diagram of our methodology (ASTD method)

period time series, which effectively mitigates the effects of noise and limited resolution of DFT.

- Additionally, we also propose a novel method for fast and accurate decomposition of multi-period time series, which can accurately quantify each component and extract them in a single calculation, without the need for multiple iterations.
- Finally, our proposed method is validated through experiments on real datasets, which demonstrate its generality and effectiveness.

The remainder of this article is structured as follows: Section II presents the framework of ASTD, Section III reports the experimental results and analysis, and Section IV concludes the paper.

II. FRAMEWORK OF ASTD

A. Framework Overview

ASTD has two main parts: periodic detection and decomposition, as shown in Figure 1. It aims to automatically decompose periodic time series accurately and efficiently. To achieve this, ASTD prunes invalid periods, uses blocking to reduce time complexity, and applies a seasonal trend decomposition algorithm with only one iteration. The periodic time series is defined as:

$$X_t = T_t + \sum S_{i,t} + R_t, \quad t = 1, 2, ...N \tag{1}$$

where X_t is the observation at time t, T_t is the trend, S_t is the sum of seasonal components with periods, and R_t is the residual signal.

B. Precise Period Detection

Real-world time series have diverse trends that can impact periodic detection accuracy. LOWESS (Locally Weighted Scatterplot Smoothing) is used to estimate trends, particularly for complex trends with no theoretical models.

Once the trend is estimated, the Periodogram and ACF are computed for the detrended time series. This involves working with a detrended sequence x'.

$$A(f) = Amplitude(\mathcal{F}(x')) \tag{2a}$$
$$P(f) = A^2(f)/n \tag{2b}$$
$$ACF(p) = \mathcal{F}^{-1}(P) \tag{2c}$$

The Fourier Transform and Inverse Fourier Transform are denoted as $\mathcal{F}$ and $\mathcal{F}^{-1}$, respectively. The amplitude spectrum is denoted as $A(f)$, the Periodogram as $P(f)$, and the autocorrelation coefficient as $ACF(p)$.

As a result of spectrum leakage, a periodogram may produce multiple potential periods for a signal. Relying solely on the power of the periodogram to determine the candidate period is not reliable. To achieve greater accuracy, the ACF can be utilized in conjunction with the periodogram to provide a more refined estimate of the candidate period. There are two steps involved in Period Detection, which we will explain in detail in the following sections:

- **Step 1**: **Screening**. To identify potential periods, we first set a threshold power equal to $\lambda = 1/6$ of the maximum power in the Periodogram. Any period with power greater than the threshold is considered a candidate period. We then narrow down the range of candidate periods further by utilizing the ACF. If a candidate period is not present in the ACF peak, it is discarded.
- **Step 2**: **Ranking**. In this step, we calculate the comprehensive ranking of candidate periods. The candidate periods near each peak of ACF are grouped together, and their ranks in the ACF and Periodogram are calculated respectively. We then calculate a score for each period based on its ranking in the periodogram and ACF. The period with the smallest score is considered as the best period in the group. We set $\gamma = 0.6$ to determine the score, but we will discuss the range of γ later.

We derive the period from the following equation:

$$A(f) = Amplitude(\mathcal{F}(x')) \tag{3a}$$
$$P(f) = A^2(f)/n \tag{3b}$$
$$p_{thre} = \lambda \times arg\max_f(P) \tag{3c}$$
$$h_{peak}(x) = PeakHeight(ACF(x)) \tag{3d}$$
$$z = \lfloor \frac{1}{f} \rfloor \tag{3e}$$
$$f_{cand} = \{f | P(f) > p_{thre} \cap ACF(z) > \frac{1}{2} h_{peak}(z)\} \tag{3f}$$
$$p_{cand} = \lfloor 1/f_{cand} \rfloor \tag{3g}$$
$$score(i) = \gamma \times u_i + (1-\gamma) \times v_i, \quad i \in p_{cand} \tag{3h}$$

p_{thre} is the minimum power threshold for a period to be considered a candidate. h_{peak} is the ACF peak height. p_{cand} is a period that satisfies both the Periodogram and ACF conditions. u_i and v_i are the rankings of i in the Periodogram and ACF. $score(i)$ is the combined score of the Periodogram and ACF.

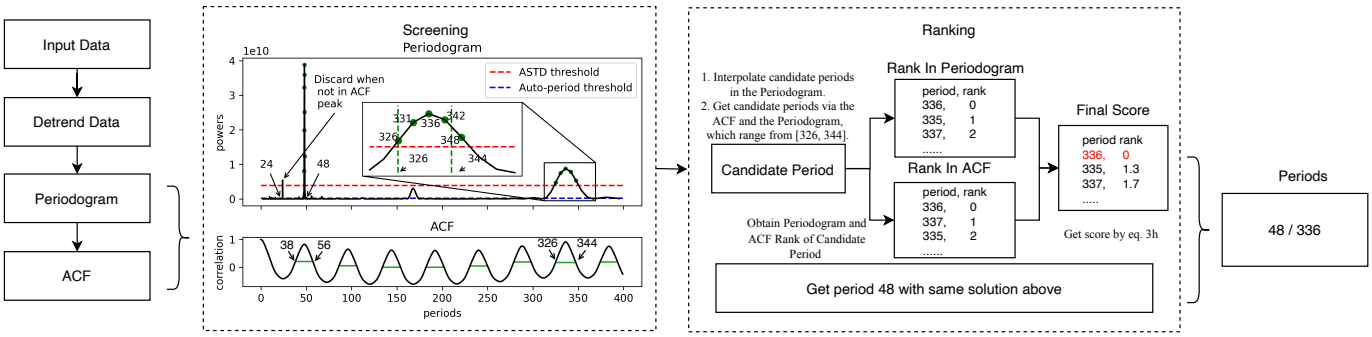

Fig. 2. Detect periodicity in electricity demand in England and Wales from June 5 to August 27, 2000. Screen by power threshold (red dash line) and ACF peak. Rank using ACF and Periodogram, with 336 ranked highest, and 48 also detected.

The true period is close to the periodogram and ACF peaks, despite spectral leakage and noise interference. Autoperiod and robust period methods use these peaks to find the period, but we are the first to rank candidate periods using both the ACF and periodogram. Higher rankings in both increase the probability of the true period. Extensive experiments showed that a score ranking with $\gamma = 0.6$ was most accurate. A larger γ implies a greater weight of the periodogram in the result, making the periodic detection more dependent on the ranking of the periodogram. Here, $\gamma = 0.6$, so the weight of the periodogram is slightly larger than that of the ACF, but they are very close.

Figure 2 shows the process of detecting periodicity in electricity demand data for England and Wales from June 5, 2000, to August 27, 2000. The data was sampled at a half-hour interval, and we hypothesized that the periods would be 48 (a day) and 336 (a week). The ACF plot revealed a peak at 24, which we disregarded. The Periodogram plot showed many green dots near 48 and 336, which we selected as candidate periods using a threshold of maximum power of $\lambda = 1/6$. However, due to noise in the data, it is inaccurate to rely solely on the ACF peak. Therefore, we used both the Periodogram and ACF to narrow down the candidate periods, selecting those whose power exceeded the threshold in the Periodogram and stayed near the ACF peak. In Figure 2, five points near 336 had greater power than the threshold in the Periodogram, but they were not continuous. To address this, we interpolated a cubic curve and identified the valid candidate period near 336 as ranging from 326 to 344 in the ACF. We then calculated the rankings of the candidate periods and obtained a score based on the two rankings. The period with the highest ranking (lowest score) was the best candidate period, which was 336.

C. Quantitative Decomposition

We utilize the period obtained through periodic detection to decompose the signal, resulting in two distinct components: trend extraction and season extraction. The decomposition process is illustrated in Figure 3.

Step 1: **Season Extraction**. The process of decomposing a multi-period time series is identical for each period. We decompose the series from high frequency (short period) to low frequency (long period) in a sequential manner. In particular, we avoid using classic filtering techniques such as Butterworth or Chebyshev filters, which require the setting of different parameters for different datasets. Incorrectly set parameters can lead to inaccurate data. Instead, we leverage the Fast Fourier Transform (FFT) to filter the data and extract seasons based on their respective periods. We obtain the season by using the following equation:

$$f_p = 1/period \tag{4a}$$

$$HF(f) = \begin{cases} 1, & f \geq f_p, \\ 0, & f < f_p. \end{cases} \tag{4b}$$

$$A(f) = Amplitude(\mathcal{F}(x)) \tag{4c}$$

$$AS(f) = A(f) \times HF(f) \tag{4d}$$

$$S(x) = ||\mathcal{F}^{-1}(AS(f))|| \tag{4e}$$

In this equation, f_p represents the cut-off frequency, $HF(f)$ is the function used to obtain the high frequency amplitude, $AS(f)$ represents the amplitude of the season, and $S(x)$ represents the season itself.

Step 2: **Extracting Trends**. After removing all of the seasonal components, the remaining parts consist of trends and residuals. To extract the trend, we apply a low-pass filter. The trend is derived from the following equation:

$$f_p = 1/period \tag{5a}$$

$$LF(f) = \begin{cases} 1, & f < f_p, \\ 0, & f \geq f_p. \end{cases} \tag{5b}$$

$$A(f) = Amplitude(\mathcal{F}(x)) \tag{5c}$$

$$AT(f) = A(f) \times LF(f) \tag{5d}$$

$$T(x) = ||\mathcal{F}^{-1}(AT(f))|| \tag{5e}$$

The symbol f_p represents the cut-off frequency, while $LF(f)$ refers to the function used to calculate the amplitude at low frequencies. The term $AT(f)$ denotes the amplitude of the seasonal variation, and $T(x)$ represents the trend.

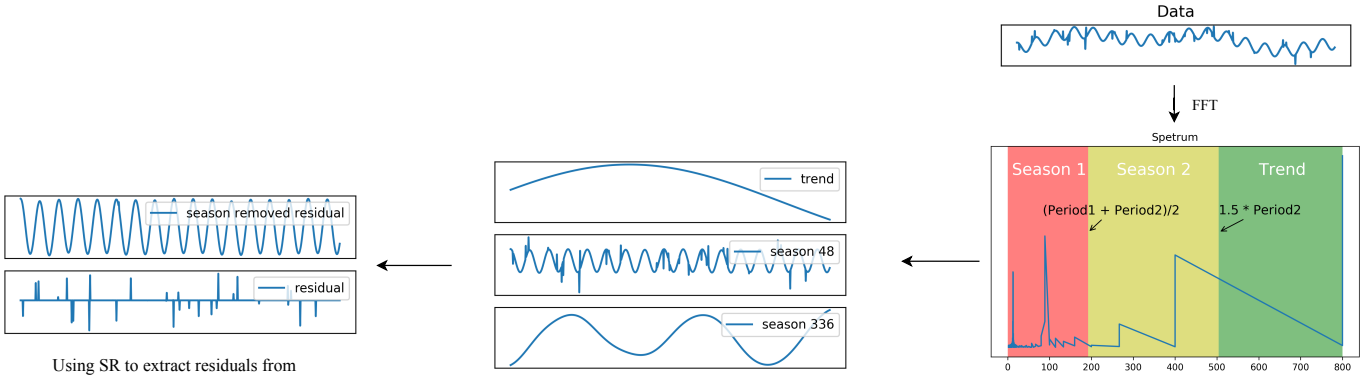

Fig. 3. Decomposition process: Extract Season 1, Season 2, Trend from Spectrum's red, yellow, and green using inverse fast Fourier transform (IFFT). Extract residual with smallest period using SR.

D. Accurate Residual

In this paper, we use the Spectral Residual (SR) model to detect outlier positions and extract them. Residuals, which represent noise, are typically high-frequency and random. By leveraging the SR model, we identify residual locations in the most high-frequency seasonal terms, smooth these points, and compute the difference between the original and smoothed seasons, which yields the residual. Notably, Microsoft introduced the use of the SR model for time series analysis in 2019, marking the first instance of applying the SR model from visual saliency detection to detect anomalies in time series data.

III. Experiments

A. Datasets

In our experiments, we utilized a combination of public and synthetic datasets, the characteristics of which are outlined in Table 1. For instance, the "M4-Hourly" dataset is a multi-period time series with a period distribution ranging from 24 to 168. This dataset consists of a total of 10 sequences, each with a sequence length of 700 data points.

To generate the synthetic data in Figure 4, we start by creating a trend signal with 700 data points. This signal includes a triangle wave to depict a gradual trend change. Then, we incorporate two cosine waves with periods of 20 and 70. Finally, we add random data for the residuals to simulate real-world situations.

TABLE I
Data Set Statistics

Data Set	Periodics	Periods	Count	Length
CRAN	Non-period	None	17	24 – 827
CRAN	Single-period	2–52	58	24 – 3024
M4-Hourly	Multi-period	(24,168)	10	700
AusGrid-Energy	Multi-period	(48,336)	5	4416
Synthetic	Multi-period	(20,70)	1	700

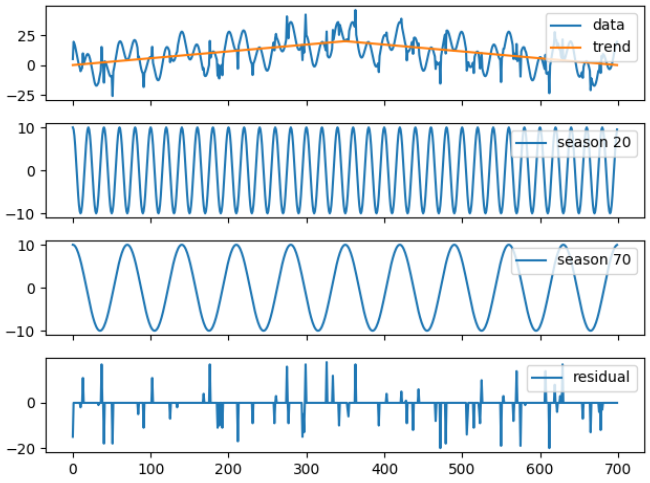

Fig. 4. The synthetic Data

B. Setup

We have selected four state-of-the-art (SOTA) algorithms for periodicity detection comparison: findfrequency, sazed, Autoperiod, and Robust-period. Both Autoperiod and Robust-period are capable of multi-period detection and single-period detection. For the purpose of decomposition comparison, we have chosen three SOTA decomposition algorithms: MSTL, STR, and TBATS. These are all multi-period time series decomposition algorithms and can also be used for decomposing single-period time series. The details of the baseline algorithms are presented in Table II. Through a grid search on the validation set, we have set $\lambda = 1/6$ and $\gamma = 0.6$.

C. Periodic Detection and Run-Time Cost

Table III compares different period detection algorithms. ASTD performs well in detecting periods for multi-period, single-period, and non-periodic time series with the lowest false positive rate. Sazed and robust-period have lower

TABLE II
Source of the baseline algorithms

Type	Name	Source
periodicity-detection	Sazed	https://cran.r-project.org/web/packages/sazedR/index.html
	Findfrequency	https://rdrr.io/cran/forecast/man/findfrequency.html
	Auto-period	https://github.com/akofke/autoperiod
	Robust-period	https://github.com/ariaghora/robust-period
decomposition	MSTL	https://github.com/KishManani/MSTL
	STR	https://cran.r-project.org/web/packages/stR/index.html
	TBATS	https://rdrr.io/cran/forecast/man/tbats.html

TABLE III
Detected periodicities on different dataset.

Algorithm	CRAN (Non-period)			CRAN (Single-period)			M4 (Multi-period)			AusGrid (Multi-period)		
	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
Findfrequncy	0.58	0.58	0.58	0.60	0.60	0.60	0.9	0.45	0.6	0	0	0
Sazed	0.0	0.0	0.0	0.65	0.65	0.65	1	0.5	0.66	0.4	0.2	0.26
Autoperiod	0.64	0.64	0.64	0.24	0.24	0.24	0.9	0.45	0.6	0.6	0.3	0.4
Robust-period	0.0	0.0	0.0	0.19	0.87	0.31	0.19	0.55	0.28	0.33	0.5	0.39
ASTD	**0.76**	**0.76**	**0.76**	**0.98**	**0.98**	**0.98**	**0.96**	**1**	**0.98**	**0.86**	**1**	**0.91**

precision for detecting single-period time series. Autoperiod recognizes some aperiodic sequences but has a higher false positives rate. In multi-period detection, ASTD achieves the highest F1 value, while Autoperiod and Sazed detect only one cycle, and robust-period detects multiple cycles. Findfrequency obtains the period using an autoregressive model, but it is prone to interference from multi-period data. ASTD effectively mitigates the side effects of spectral leakage and abnormal values, making it the only algorithm that can correctly detect periods in time series with an outlier ratio from 0% to 20%.

TABLE IV
Comparison of periodic detection for single-period time series with different outlier ratio.

Methods	$\eta = 0\%$	$\eta = 5\%$	$\eta = 10\%$	$\eta = 15\%$	$\eta = 20\%$
Findfrequency	71	71	83	91	83
Auto-period	69	69	69	67	67
ASTD	**70**	**70**	**70**	**70**	**70**

Table V shows the running time comparison of different multi-period detection algorithms in different datasets. From the table, we can see that the running time of each algorithm increases with the amount of data. In the same dataset, ASTD has the shortest running time and Robust-period the longest. In the Taylor dataset, Robust-period runs 6882 times longer than ASTD. Therefore, in the case of long time series, ASTD is faster than auto-period and Robust-period, which can better extend the application of long time series.

D. Decomposition and Run-Time Cost

To quantitatively evaluate the performance, we compared the mean squared error (MSE) and mean absolute error (MAE) in the synthetic dataset, and ASTD outperformed other algorithms (Table VI).

TABLE V
Running time comparison of periodic detection for multi-period time series with different lengths.

Methods	CRAN-cafe (length=114)	CRAN-cafe (length=114)	CRAN-cafe (length=114)
Auto-period	0.1631 s	0.5717 s	2.7062 s
Robust-period	7.7395 s	60.2719 s	838.9443 s
ASTD	**0.0428 s**	**0.0285 s**	**0.1219 s**

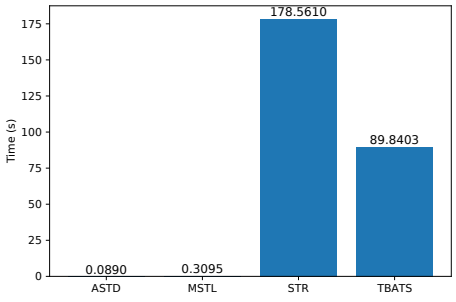

Fig. 5. Comparison of decomposition time of different algorithms in Taylor dataset.

In terms of decomposition time, ASTD was found to be 3 times faster than state-of-the-art algorithms (Figure 5). Overall, ASTD performed well in recovering the location and magnitude of seasonality and residuals. Its decomposed seasonal and residuals had smaller MAE and MSE than other algorithms, and its decomposition trend was close to the current state-of-the-art algorithms. ASTD's low time complexity makes it suitable for decomposing long time series.

TABLE VI
Comparison of MSE and MAE for trend, seasonality, and residual components of different decomposition algorithms.

Methods	MSE				MAE			
	trend	season=20	season=70	residual	trend	season=20	season=70	residual
MSTL	0.2108	3.1543	2.6806	6.1802	0.3610	1.0254	1.0605	1.4478
STR	**0.1714**	0.2161	0.5422	0.6984	**0.3400**	0.3812	0.5789	0.6541
TBATS	0.8824	0.1981	0.1590	0.9622	0.7815	0.3295	0.3566	0.8050
ASTD	0.2463	**0.0799**	**0.0081**	**0.2385**	0.4025	**0.2584**	**0.0814**	**0.3953**

IV. Conclusion

Multi-period time series decomposition is a fundamental technique widely used in anomaly detection, prediction, and classification. However, the current state-of-the-art decomposition algorithms require users to identify the periodicity of time series and select different decomposition methods accordingly. In this paper, we propose a unified method called ASTD, which provides an automatic decomposition of time series, greatly expanding the scope of applications for time series decomposition. In particular, the proposed ASTD requires only the input of time series data, and can automatically detect the periodicity and decompose the sequence. We have designed ASTD with low time complexity requirements to handle long time series that are commonly encountered in real-world scenarios. The complexity of ASTD is O($NlogN$). Extensive experiments have demonstrated that ASTD outperforms existing SOTA algorithms in terms of decomposition accuracy and run-time cost. Besides, it is highly robust to outliers. Furthermore, our method achieves the smallest Mean Absolute Error (MAE) and Mean Squared Error (MSE), and it is nearly 3 times faster than the state-of-the-art decomposition algorithm on the Taylor dataset. In the future, we will continue to improve the accuracy of trend extraction and explore the application of ASTD in various time series-related tasks.

V. Acknowledgement

This work was supported by National Key R&D Program of China under Grant 2020YFA0711400, and National Science Foundation of China under Grant U21A20452, U19B2044.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2013.

[3] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[4] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International conference on data mining (ICDM)*. IEEE, 2018, pp. 727–736.

[5] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An empirical comparison of machine learning models for time series forecasting," *Econometric reviews*, vol. 29, no. 5-6, pp. 594–621, 2010.

[6] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, vol. 90, p. 106181, 2020.

[7] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep learning for time series forecasting: a survey," *Big Data*, vol. 9, no. 1, pp. 3–21, 2021.

[8] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[9] H. Ismail 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.

[10] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data mining and knowledge discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[11] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[12] P. Schäfer, "Scalable time series classification," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1273–1298, 2016.

[13] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition," *J. Off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[14] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, "Robuststl: A robust seasonal-trend decomposition algorithm for long time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5409–5416.

[15] K. Bandara, R. J. Hyndman, and C. Bergmeir, "Mstl: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns," 2021.

[16] Q. Wen, Z. Zhang, Y. Li, and L. Sun, "Fast robuststl: Efficient and robust seasonal-trend decomposition for time series with complex patterns," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2203–2213.

[17] A. Dokumentov, R. J. Hyndman *et al.*, "Str: A seasonal-trend decomposition procedure based on regression," *Monash econometrics and business statistics working papers*, vol. 13, no. 15, pp. 2015–13, 2015.

[18] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American statistical association*, vol. 106, no. 496, pp. 1513–1527, 2011.

[19] J. D. Hamilton, "Why you should never use the hodrick-prescott filter," *Review of Economics and Statistics*, vol. 100, no. 5, pp. 831–843, 2018.

[20] J. H. Friedman and B. W. Silverman, "Flexible parsimonious smoothing and additive modeling," *Technometrics*, vol. 31, no. 1, pp. 3–21, 1989.

[21] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "l_1 trend filtering," *SIAM review*, vol. 51, no. 2, pp. 339–360, 2009.

[22] M. Vlachos, P. Yu, and V. Castelli, "On periodicity detection and structural periodic similarity," in *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 449–460.

[23] T. Puech, M. Boussard, A. D'Amato, and G. Millerand, "A fully automated periodicity detection in time series," in *International Workshop on Advanced Analysis and Learning on Temporal Data*. Springer, 2019, pp. 43–54.

[24] Q. Wen, K. He, L. Sun, Y. Zhang, M. Ke, and H. Xu, "Robustperiod: Robust time-frequency mining for multiple periodicity detection," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 2328–2337.

[25] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 3009–3017.