STD: A Seasonal-Trend-Dispersion Decomposition of Time Series

Grzegorz Dudek

***Abstract*—The decomposition of a time series is an essential task that helps to understand its very nature. It facilitates the analysis and forecasting of complex time series expressing various hidden components such as the trend, seasonal components, cyclic components and irregular fluctuations. Therefore, it is crucial in many fields for forecasting and decision-making processes. In recent years, many methods of time series decomposition have been developed, which extract and reveal different time series properties. Unfortunately, they neglect a very important property, i.e., time series variance. To deal with heteroscedasticity in time series, the method proposed in this work – a seasonal-trend-dispersion decomposition (STD) – extracts the trend, seasonal component and component related to the dispersion of the time series. We define STD decomposition in two ways: with and without an irregular component. We show how STD can be used for time series analysis and forecasting.**

***Index Terms*—Time series analysis, time series decomposition, time series forecasting.**

I. Introduction

A TIME series expresses states of a certain variable that describe a given phenomenon (economic, biological, physical, etc.) observed in subsequent periods. Time series analysis and forecasting is an extremely important task in many fields, including business, industry, government, politics, health and medicine [1]. However, this task can be difficult due to the complex nature of the time series. Time series can exhibit a variety of unobservable (latent) components that can be associated with different types of temporal variations. These include: (1) a long-term tendency or trend, (2) cyclical movements superimposed upon the long-term trend (usually non-periodical), (3) seasonal variations (periodical), and (4) irregular fluctuations. In economics, the seasonal variations represent the composite effect of climatic and institutional events which repeat more or less regularly each year [2]. The cycles appear to reach their peaks during periods of economic prosperity and their troughs during periods of depression. Their rise and fall constitute the business cycle.

Extracting the components of a time series can help us to understand the underlying process and to forecast it. Instead of building a complex forecasting model for the composed time series, after decomposition into basic components, we can built simpler specialized models for each component. This approach is very common in forecasting using both classical statistical methods and machine learning methods. Therefore, many methods of time series decomposition have been proposed.

A. Related Work

Time series decomposition has a long history dating back to the mid 19th century [3]. The idea of decomposing the time series into unobservable components appeared in the work of 19th century economists who drew their inspiration from astronomy and meteorology [4]. Much research back then was done to reveal the ”cycles” that made it possible to explain and predict economic crises. In 1884, Poynting proposed price averaging as a tool to eliminate trend and seasonal fluctuations [5]. Later his approach was extended by other researchers including Copeland who was the first to attempt to extract the seasonal component [6]. Persons was the first to define the various components of a time series, i.e., the trend, cycle, seasonal and irregular components, and proposed an algorithm to estimate them (link relatives method) [7]. The process of decomposition was refined by Macauley who proposed a way of smoothing time series, which has become a classic over time [8]. Based on Macauley's method, the Census II method was developed and its numerous variants are widely used today such as X-11, X-11-ARIMA, X-12-ARIMA, X-13ARIMA-SEATS, and TRAMO-SEATS. A detailed discussion of these methods is provided by [2].

Structural time series decomposition, which involves decomposing a series into components having a direct interpretation, is very useful from a practical point of view. A structural model is formulated directly in terms of unobserved components, such as the trend, cycles, seasonals and remaining component. These components can be combined additively or multiplicatively. An additive decomposition is applied if the variation around the trend-cycle, or the magnitude of seasonal variations, does not change with the time series level. When such variation is observed to be proportional to the time series level, multiplicative decomposition is more appropriate.

To extract the components of the series, both parametric or non-parametric methods are used. A parametric approach imposes a specific model on the component, e.g., linear or polynomial. The nonparametric approach offers more possibilities because it does not limit the model to a specific class. A

Manuscript received 21 April 2022; revised 14 February 2023; accepted 8 April 2023. Date of publication 19 April 2023; date of current version 15 September 2023. Recommended for acceptance by R.C.-W. Wong.

The author is with the Department of Electrical Engineering, Czestochowa University of Technology, 42-200 Czestochowa, Poland (e-mail: dudek@el.pcz.czest.pl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2023.3268125, provided by the author.

Digital Object Identifier 10.1109/TKDE.2023.3268125

popular example of a non-parametric method to extract a trend is smoothing with a moving average.

One of the most widely used methods of time series decomposition is STL (Seasonal and Trend decomposition using Loess) [9]. STL is additive. The STL decomposition procedure is iterative and relies on the alternate estimation of the trend and the seasonal components using locally estimated scatterplot smoothing (Loess), which can estimate nonlinear relationships. The seasonal component is allowed to change over time. It is composed of seasonal patterns estimated based on k consecutive seasonal cycles, where k controls how rapidly the seasonal component can change. Other attractive features of STL are: robustness to outliers and missing data, the ability to decompose time series with seasonality of any frequency, and the possibility of implementation using numerical methods instead of mathematical modeling.

Another popular method of additive time series decomposition uses a discrete wavelet transform. Wavelet-based multiresolution analysis decomposes the series in an iterative process into components with decreasing frequencies [10]. In the subsequent levels of decomposition, the series is processed by a pair of filters – high-pass and low-pass (two-channel subband coding). The result is a low-frequency component, the so-called approximation, representing the trend and a high-frequency component, the so-called detail, representing the detailed features of the series. In each iteration, the approximation from the previous iteration is decomposed into detail and new approximation. The sum of all the details produced at all levels, and the lowest-level approximation gives the input series. The decomposition depends on the form and parameters of the wavelet function, which is a function of both time and frequency.

In [11], Empirical Mode Decomposition (EMD) was proposed, which decomposes the time series in the time domain into components called Intrinsic Mode Functions (IMFs). These form a complete and nearly orthogonal basis for the original time series. An IMF amplitude and frequency can vary with time. The IMFs are obtained by applying a recursive so-called sifting process. This extracts the local minima and maxima of the series and then interpolates them separately using cubic splines. The IMFs extracted at subsequent levels are characterized by ever lower frequencies. Since the decomposition is based on the local characteristic time scale of the data, EMD is suitable for both non-linear and non-stationary time series.

Other, less popular, time series decomposition methods include: Variational Mode Decomposition (VMD) [12], Singular Spectrum Analysis (SSA) [13], and Seasonal-Trend Decomposition based on Regression (STR) [6]. VMD is a generalization of the classical Wiener filter into many adaptive signal bands. It extracts a set of IMFs defined in different frequency bands, which optimally reconstruct the time series. As an alternative to EMD, VMD is devoid of some EMD limitations, such as the lack of theoretical foundations, sensitivity to sampling and data disturbance, and the dependence of the result on the methods of extremes detection and envelope interpolation.

SSA is based on the matrix representation of the time series in the form of a so-called trajectory matrix (Hankel matrix) and its singular value decomposition (SVD). Using the SVD products, i.e., eigentriples, the trajectory matrix is expressed as the sum of elementary matrices. The time series components are obtained by appropriate grouping of the elementary matrices using eigentriples for this purpose. The SSA decomposition is additive. The components obtained as a result are interpretable. They express the trend, periodic components and random disturbances.

STR is an additive decomposition with a matrix representation of the seasonal component. The method can produce multiple seasonal and cyclic components. Seasonal components can be fractional, flexible over time, and can have complex topology. STR allows us to take into account the influence of additional external variables on decomposition and to estimate confidence intervals for components.

B. Motivation and Contribution

Existing methods of time series decomposition extract different components expressing different time series properties. However, to our knowledge, none of them extracts the component representing the series dispersion. To fill this gap, this work proposes a new method of time series decomposition that extracts the components of the trend, seasonality and dispersion. It can assist in the analysis and forecasting of heteroscedastic time series.

Our research contributions can be summarized as follows:

1) We propose a new method of time series decomposition. It has two variants. In the first, STD, it extracts the trend, seasonal component and dispersion component. In the second variant, STDR, it extracts additionally an irregular component (reminder).
2) We demonstrate how the proposed decomposition method can be used for simplifying and solving complex forecasting problems including those with multiple seasonality and variable variance.

The rest of the work is organized as follows. Section II describes decomposition of the heteroscedastic time series using standard methods. The proposed STD and STDR methods are presented in Section III. Section IV gives some application examples and shows how STD can be used for forecasting. Finally, Section V concludes the work.

II. Decomposition of Heteroscedastic Time Series Using Additive and Multiplicative Methods

Typically, time series decomposition can be expressed in an additive or multiplicative form as follows [2], [14]:

$$y_t = T_t + S_t + R_t \tag{1}$$

$$y_t = T_t \times S_t \times R_t \tag{2}$$

where y_t denotes the observed series, T_t is a trend-cycle component combining the trend and cycle (often just called the trend for simplicity), S_t is the seasonal component, and R_t is the irregular component (reminder), all at period t.

In the additive model, heteroscedasticity in y_t has to be expressed by heteroskadisticity in one or more decomposition products. Usually, the trend is a smoothed original time series, so it does not include short-term variations of varying variance. These variations appear in the seasonal and/or irregular components. If the decomposition method produces a regular seasonal component, i.e., composed of seasonal cycles of the same shape, which is a classical approach [14], the time series variance has to be expressed by the irregular component. But a desired property of the irregular component, which is often assumed for inferential purposes, is to be normally identically distributed and not correlated, which implies independence [2]. Hence, $R_t \sim NID(0, \sigma^2)$. When the variance of the irregular component changes in time, it does not express white noise in the strict sense. Therefore, the additive model (1) is not recommended for heteroscedastic time series.

In the multiplicative model, all components are multiplied, so the variations included in the irregular and seasonal components are amplified or weakened by the trend. An increasing trend increases these variations, while a decreasing trend decreases them. Thus, the multiplicative model is most useful when the variation in time series is proportional to the level of the series.

Fig. 1 shows decomposition of a time series expressing monthly electricity demand for Poland (17 years, observed from 1997 to 2014) using the most popular decomposition methods, i.e., classical additive and multiplicative methods, STL, wavelet transform, and EMD. Note that the times series has decreasing variations with the trend level. Mean values of the series and their standard deviations are shown in the bar chars shown in the right panel. They are calculated for successive sequences of length $n = 12$. To estimate the trend, the classical additive and multiplicative methods use two-sided moving averages. The negative effect of this is that the first and last few observations are missing from the trend and irregular components. The classical methods assume that the seasonal component is constant throughout the entire series. This constant seasonal pattern is determined as an average of all seasonal sequences of the detrended series. The long-term variability is expressed by the trend. Note how this variability changes over time in the std-chart. The short-term variability is expressed in the remainder component. The std-chart for this component shows that the variance is smallest in the middle part of the data period. In this part, the combined trend and seasonal components approximate the time series most accurately. In the first part, the amplitude of the combined components is smaller than the amplitude of the real series and must be increased by the irregular component. In this part, the extremes of the irregular component correspond to the extremes of the seasonal component. In the final part of the series, the amplitude of the combined trend-seasonal component is higher that the real amplitude. The irregular component compensates the amplitude. Its extremes are opposite to the extremes of the seasonal component. The compensation function of the irregular component results in its variable variance and autocorrelation.

STL produces a smoother trend than classical decomposition methods due to the use of local polynomial regression. The seasonal component in STL averages the real seasonal patterns but can still reflects its amplitude. Therefore, to compensate for the amplitude mismatch, the irregular component may be smaller than in classical decomposition methods. However, it still expresses the variable variance and autocorrelation.

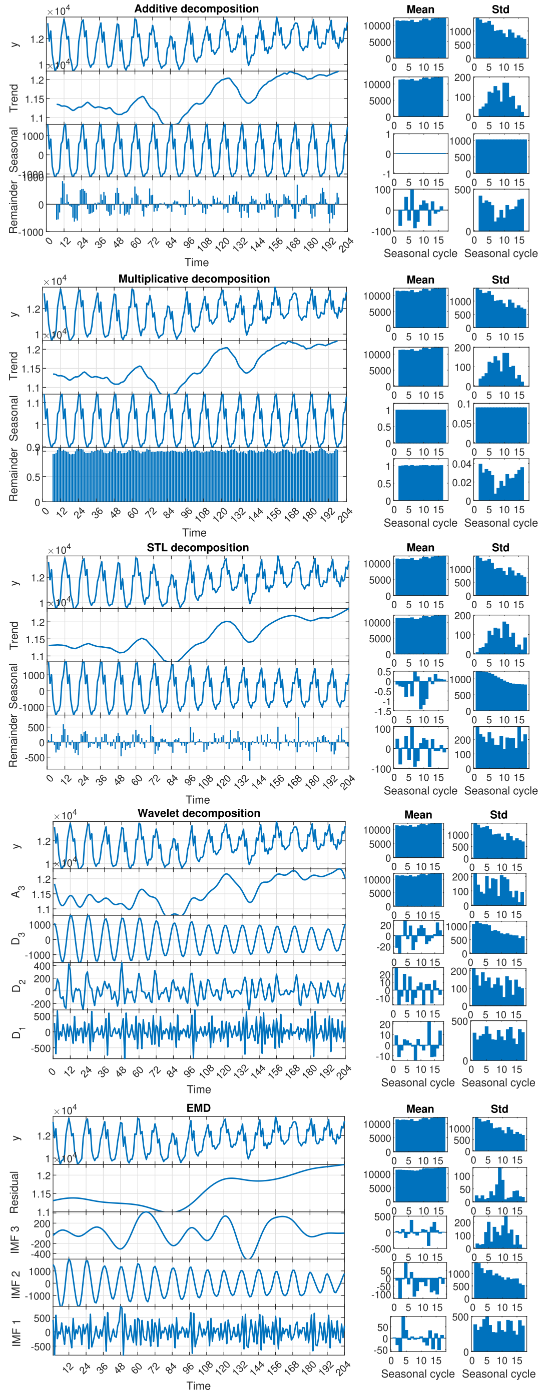

Fig. 1. Monthly electricity demand time series decomposition using standard methods.

Wavelet decomposition produces the components corresponding to the trend (A_3) and smoothed seasonal variations (D_3) as well as components expressing more detailed variations. Each of them expresses changing variance. As can be seen from Fig. 1, EMD produces the most smoothed trend (residual component) compared to other methods and a separate component representing non-periodical cyclical movements (IMF3). The seasonal component, IMF2, which is very similar to the D_3 component generated by wavelet transform, smooths the seasonal cycles significantly. The random component, IMF1, is very similar to the highest-level detail of the wavelet decomposition, D_1. The variance of the series is distributed between EMD components.

Note that the time series variance is not expressed explicitly in the decomposition products of the presented methods. It is hidden in the components. A separate dispersion component could be very useful for time series analysis and forecasting. In the next section, we propose a method which extracts this component.

III. Seasonal-Trend-Dispersion Decomposition

Let $\{y_t\}_{t=1}^{N}$ be a time series with a seasonality of period n. Assume that the length of the series is a multiple of the seasonal period, i.e., $N/n = K, K \in \mathbb{N}$. Time series y_t can be written as a series of successive seasonal sequences:

$$\{\{y_{i,j}\}_{j=1}^{n}\}_{i=1}^{K} = \{\{y_{1,j}\}_{j=1}^{n}, \ldots, \{y_{K,j}\}_{j=1}^{n}\} \tag{3}$$

where $i = 1, \ldots, K$ is the running number of the seasonal cycle, and $j = 1, \ldots, n$ is the time index inside the given seasonal cycle. The global time index $t = n(i-1) + j$.

The average value of the i-th seasonal sequence is:

$$\bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{i,j} \tag{4}$$

and its diversity measure is defined as:

$$\tilde{y}_i = \sqrt{\sum\nolimits_{j=1}^{n}(y_{i,j} - \bar{y}_i)^2} \tag{5}$$

The trend component is defined using averages of the seasonal sequences as follows:

$$\{T_t\}_{t=1}^{N} = \{\{\underbrace{\bar{y}_i, \ldots, \bar{y}_i}_{n\text{ times}}\}\}_{i=1}^{K} \tag{6}$$

while the dispersion component is defined using diversities of these sequences:

$$\{D_t\}_{t=1}^{N} = \{\{\underbrace{\tilde{y}_i, \ldots, \tilde{y}_i}_{n\text{ times}}\}\}_{i=1}^{K} \tag{7}$$

Based on the trend and dispersion components, we define the seasonal component:

$$S_t = \frac{y_t - T_t}{D_t} \tag{8}$$

The proposed STD decomposition is expressed as follows:

$$y_t = S_t \times D_t + T_t \tag{9}$$

Fig. 2 shows an example of STD decomposition of the time series of monthly electricity demand for Poland. Note that the trend and dispersion components are step functions, where the step length corresponds to seasonal period n. The trend expresses the level of the time series in successive seasonal periods, while the dispersion expresses the variation of the time series elements in these periods. The seasonal component is composed of the seasonal patterns, which are centered, i.e., their average value is zero, and unified in variance, i.e., their dispersion is the same. Moreover, when we express seasonal patterns by vectors, $\mathbf{s}_i = [S_{i,1}, \ldots, S_{i,n}]$, where $S_{i,j}$ is the j-th component of the i-th seasonal pattern, their length is equal to one. Thus, they are normalized vectors. Although unified, the seasonal patterns differ in ”shape”. Their ”shapes” express unified variations of the series in the successive seasonal periods. Note that the ”shapes” are not smoothed or averaged as in the standard decomposition methods.

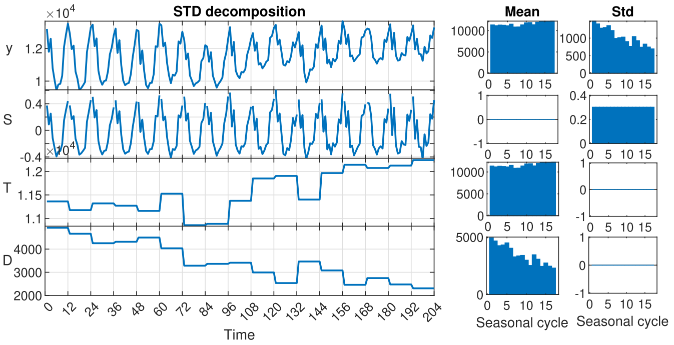

Fig. 2. Monthly electricity demand time series decomposition using STD.

A variant of STD is STD with a reminder component, STDR, defined as follows:

$$y_t = S'_t \times D_t + T_t + R_t \tag{10}$$

where S'_t is an averaged seasonal component and R_t is a reminder component.

In STDR, the trend and dispersion components are defined in the same way as in STD. The seasonal component is defined using an average seasonal pattern, $\{\bar{S}_j\}_{j=1}^{n}$, determined as follows:

$$\bar{S}_j = \frac{1}{K}\sum_{i=1}^{K} S_{i,j} \tag{11}$$

The seasonal component in STDR is a sequence of K averaged seasonal patterns:

$$\{S'_t\}_{t=1}^{N} = \{\underbrace{\{\bar{S}_j\}_{j=1}^{n}, \ldots, \{\bar{S}_j\}_{j=1}^{n}}_{K\text{ times}}\} \tag{12}$$

thus, it is identical across all seasonal periods.

The reminder component is calculated from (10):

$$R_t = y_t - (S'_t \times D_t + T_t) \tag{13}$$

An example of STDR decomposition is depicted in Fig. 3. Note the same trend and dispersion components as in Fig. 2 for STD, and the different seasonal component, which for STDR is composed of the same averaged seasonal pattern. Fig. 4 shows the seasonal patterns and the averaged pattern. The remainder corresponds to the mismatch between the original seasonal cycles and the averaged seasonal cycles. Thus, it contains additional dispersion resulting from averaging the seasonal cycles. This dispersion is lower for the cycles whose patterns are similar to the averaged pattern. Note that the reminder has a zero average value in each seasonal period. To assess its stationarity visually, Fig. 5 shows the plots of its sample autocorrelation function

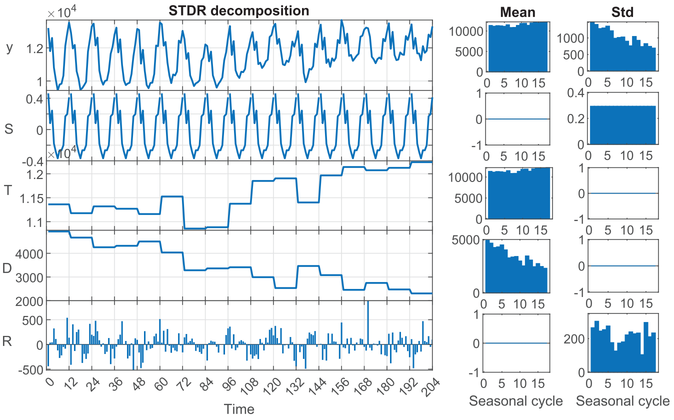

Fig. 3. Monthly electricity demand time series decomposition using STDR.

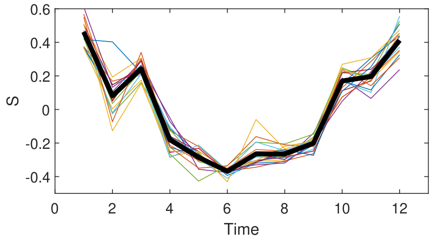

Fig. 4. Seasonal patterns of the monthly electricity demand time series (averaged pattern drawn with a thick line).

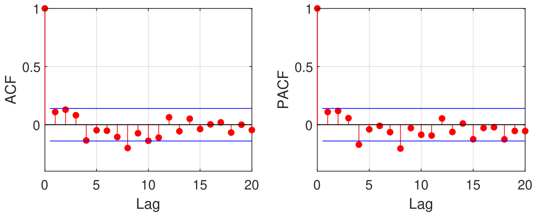

Fig. 5. ACF and PACF plots for the reminder component of STDR applied for the monthly electricity demand time series.

(ACF) and sample partial autocorrelation function (PACF). As can be seen from this figure, most of the spikes are not statistically significant, i.e., the reminder series is not highly correlated, which characterizes a stationary process. To confirm that the reminder is stationary, we apply three formal tests for a unit root in a univariate time series: augmented Dickey-Fuller test, Kwiatkowski, Phillips, Schmidt, and Shin test, and Phillips-Perron test. All tests confirmed stationarity at a 1% level of significance.

Remark: The dispersion component can be defined using a standard deviation as a diversity measure (which is diversity (5) divided by $\sqrt{n}$). In such a case, all components including the remainder have the same shape as in the standard formulation, but the dispersion component decreases its range $\sqrt{n}$ times, and the seasonal component increases its range $\sqrt{n}$ times.

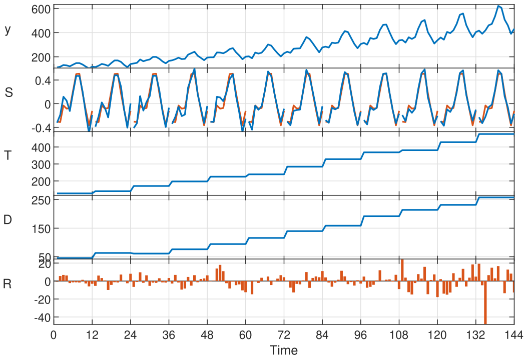

Fig. 6. Airline passengers time series (in thousands) and its decomposition using STD and STDR.

TABLE I

Results of the Stationarity Tests for the Reminder and the Ratio of the Reminder to the Time Series

Data	aDF	KPSS	PP	Median(r_t)± IQR(r_t)
Airline	+	+	+	1.78 ± 2.26
Unemployment	+	+	+	2.29 ± 3.24
Electricity	+	+	+	2.04 ± 2.95
S&P 500	+	+	+	1.12 ± 1.46
S&P 500 returns	+	+	+	92.96 ± 72.57
Mackey-Glass	+	+	+	6.51 ± 10.99

IV. Application Examples

In this section, we apply the proposed decomposition method to time series of different nature, including multiple seasonality and no seasonality. We also present forecasting approaches based on STD decomposition.

A. Time Series Analysis Using STD

As a first example we use the classic Box & Jenkins airline data [15], i.e., monthly totals of international airline passengers from 1949 to 1960. This time series expresses an increasing trend and strong yearly periodicity ($n = 12$) that increases in size with the series level – see top panel in Fig. 6. Fig. 6 shows both STD and STDR decompositions. They have the same trend and dispersion components. The seasonal component for STD is shown in blue, and the seasonal component for STDR as well as the reminder component are shown in red. Note that the seasonal patterns generated by STD are very similar in shape.

Table I shows the results of stationarity tests for the reminder, i.e., augmented Dickey-Fuller test (aDF), Kwiatkowski, Phillips, Schmidt, and Shin test (KPSS), and Phillips-Perron test (PP). All the tests confirm stationarity with 1% significance level. Table I also shows the median and interquartile range of the ratio of the reminder to the time series defined as follows:

$$r_t = \left| \frac{R_t}{y_t} \right| * 100 \tag{14}$$

The ratio of the reminder to the time series for Airline data is relatively small, 1.78%.

The second example uses data for the US unemployment rate for males (16 years and over) observed from January 1992 to

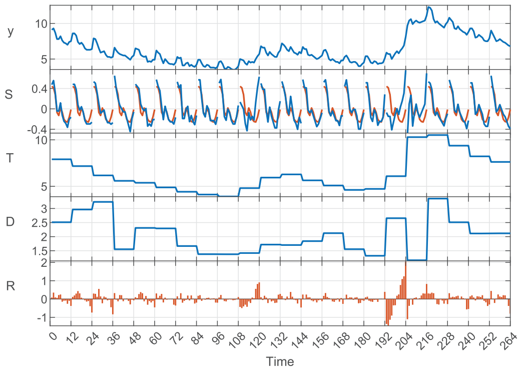

Fig. 7. Unemployment time series decomposition using STD and STDR.

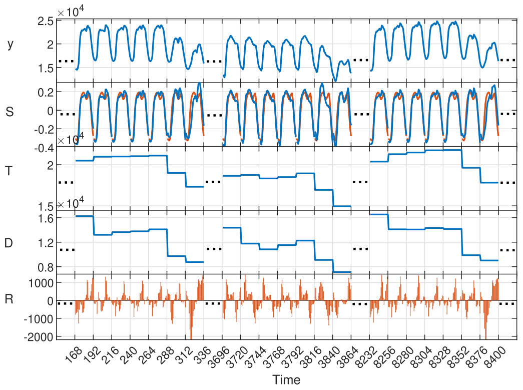

Fig. 8. Hourly electricity demand time series decomposition using STD and STDR.

December 2013 ($n = 12$). This series was analysed extensively in [2]. It exhibits yearly seasonality with strong asymmetric behavior, i.e., it displays steep increases that end in sharp peaks and alternate with much more gradual and longer declines that end in mild troughs [16]. Thus the seasonal patterns are generally similar to each other. The seasonal patterns observed in Fig. 7 are similar in shape, except for three patterns, which reflect sharp spikes in unemployment in the final months of the year, i.e., sequences 109-120, 193-204 and 205-2016. Due to a deviation from the typical shape for these three sequences, the reminder takes larger values for them than for other annual sequences. Nevertheless, it passes the stationarity tests, see Table I. The ratio of the reminder to the time series for unemployment data is 2.29%.

The third example concerns hourly electricity demand. Time series of this type exhibit triple seasonality: yearly, weekly and daily. The seasonalities are related to the local climate, weather variability and the nature of a country's economy. Fig. 8 shows decomposition products of the hourly electricity demand for Poland in 2018. We assumed a seasonal period as a daily one ($n = 24$). In Fig. 8, we show three weekly sequences of the time series, from January, July and December. As can be seen from this figure, the seasonal component expresses daily patterns whose shapes are related to the day of the week and season of the year. The daily patterns representing the working days from Tuesday to Friday for the same period of the year are similar to each other. Patterns for Mondays are slightly different from them. Saturdays and Sundays have their own shapes. Note that the trend and dispersion components have both weekly and yearly seasonalities. These two components can be further decomposed using STD or STDR. The ratio of the reminder to the time series is only 2.04%. The reminder passes all the stationarity tests.

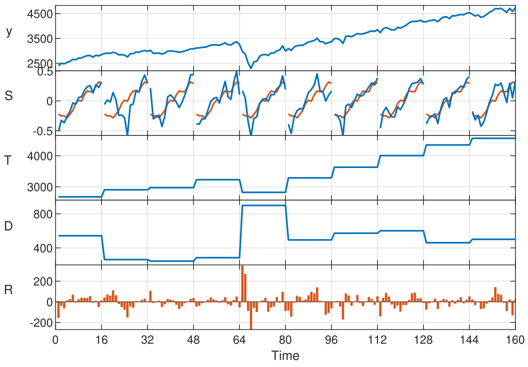

Fig. 9. S&P 500 Index time series decomposition using STD and STDR.

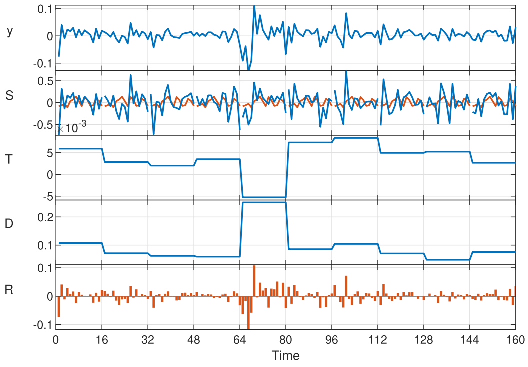

Fig. 10. Decomposition of S&P 500 weekly log returns using STD and STDR.

The next two examples are for financial time series. We analyse one of the most important stock market indexes, S&P 500. It tracks the performance of the 500 largest companies listed on stock exchanges in the United States. Fig. 9 shows decomposition of the weekly S&P 500 Index over the period 2019-2021. S&P 500 Index shows development within a rising trend that dips at the beginning of 2020 due to the Covid-19 crisis. The time series does not express seasonality. We assume $n = 16$ weeks for STD decomposition. Because of the rising trend, the 16-week patterns forming the seasonal component have a rising character, but differ due to significant random noise. For the pattern representing the Covid-19 fall period (sequence 65-80) the highest remainder values are observed as well as the highest dispersion. The ratio of the reminder to the time series is low, 1.12%. The reminder passes all stationarity tests (see Table I).

Fig. 10 shows decomposition of the S&P 500 returns calculated as $\ln(y_t/y_{t-1})$, where y_t represents the original time series. While the original time series of S&P 500 Index is nonstationary,

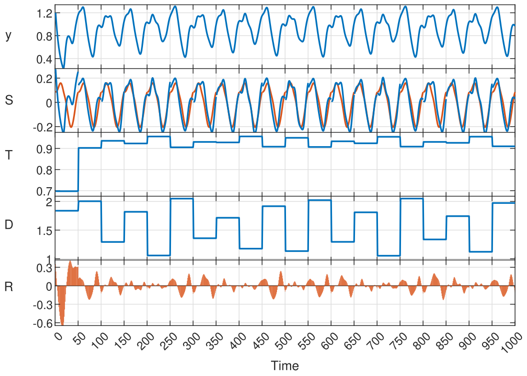

Fig. 11. Mackey-Glass time series decomposition using STD and STDR.

the returns fluctuate around a stable mean level [15]. However, their variability around the mean changes. In the period 2019-21, it is highest during the Covid-19 crisis, see Fig. 10, where the dispersion and remainder are highest for the crisis period, i.e., sequence 65-80. The ratio of the reminder to the time series is high (around 93%), which indicate the dominant content of the noise in the series of returns. The reminder passes all the stationarity tests (see Table I).

The last example concerns decomposition of a synthetic time series – a Mackey-Glass time series which is produced by delay differential equation [17], i.e., $\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t)$, where we assumed typical values for parameters: $a = 0.2$, $b = 0.1$, $x(0) = 1.2$, and $\tau = 17$. With these parameters, the time series is chaotic and exhibits a cyclic behavior. This time series is commonly used as a benchmark for testing different forecasting methods, because it has a simple definition, and yet its elements are hard to predict [1].

Fig. 11 depicts the Mackey-Glass time series decomposition. The series was computed with a time sampling of 1. First 1000 points of the series is shown. We assumed a seasonal pattern length as $n = 50$. Note the irregular character of the seasonal patterns and also the chaotic variability in the trend and dispersion components. The ratio of the reminder to the time series is 6.51%. The reminder passes all the stationarity tests (see Table I).

B. Time Series Forecasting Using STD and STDR

Decomposition not only helps to improve understanding of the time series, but it can also be used to facilitate forecasting complex series. Extracted components have lower complexity than the original time series and so can be modelled independently using simple models. In the case of STDR, the seasonal pattern does not change and we can use a naive approach to predict it for the next period. In STD, the seasonal pattern changes and we should use an appropriate forecasting method to predict it. Note that in the examples considered above the reminder was a stationary process. Thus it can be predicted even by those methods that require stationarity such as ARIMA. Trend and dispersion components can exhibit seasonality such as in the case of hourly electricity demand shown in Fig. 8. Such time series can be predicted using seasonal models or can be further decomposed into simple components using STD or STDR.

1) Forecasting Based on Patterns: To avoid the troublesome task of forecasting all the components extracted by STD, in [18], a method was described which combines all components into an output pattern (in fact in [18] many input and output patterns were proposed. We focus on the patterns denoted as X3.1 and Y3.1, which are related to STD). The forecasting model predicts output patterns based on the input patterns which are seasonal patterns expressed by vectors $\mathbf{s}_i = [S_{i,1}, \ldots, S_{i,n}]$, where $S_{i,j}$ is the j-th component of the i-th seasonal pattern. They are defined as follows (this is an alternative notation to (8)):

$$\mathbf{s}_i = \frac{\mathbf{y}_i - \bar{y}_i}{\tilde{y}_i} \tag{15}$$

where $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,n}]$ is a vector representing the i-th seasonal sequence of the time series.

Thus, the input patterns are centered and normalized seasonal sequences. The output pattern represents a forecasted seasonal pattern. It is defined as:

$$\mathbf{s}_{i+\tau} = \frac{\mathbf{y}_{i+\tau} - \bar{y}_i}{\tilde{y}_i} \tag{16}$$

where $\mathbf{s}_{i+\tau} = [S_{i+\tau,1}, \ldots, S_{i+\tau,n}]$ and $\tau \geq 1$ is a forecast horizon.

Note that in (16) to calculate the output pattern, we use the average and dispersion for sequence i and not for sequence $i + \tau$. This is because these two coding variables for future sequence $i + \tau$, which has been just forecasted, are not known. Using the coding variables for the previous period has consequences: the output patterns are no longer centered and normalized vectors like the input patterns are. But if the mean value of the series and its dispersion do not change significantly in the short period, i.e., $\bar{y}_{i+\tau} \approx \bar{y}_i$ and $\tilde{y}_{i+\tau} \approx \tilde{y}_i$, the output patterns are close to centered and normalized. For time series with multiple seasonality, we cannot assume that the trend and dispersion are constant in the short term because they are influenced by additional seasonal fluctuations. For example, the average values and dispersions of daily sequences can changes with the weekly seasonality, see Fig. 8. This translates into output patterns. Referring to the example shown in Fig. 8, the output patterns for Mondays are coded with the averages and dispersions of Sunday sequences (for $\tau = 1$), which are lower than those for Mondays. This has the effect of shifting the output patterns for Mondays up and stretching them. For similar reasons, output patterns for Saturdays and Sundays are placed lower than output patterns for the other days of the week and are less stretched (compare this in Fig. 12). Thus, the output patterns are not unified globally but are unified in groups composed of the same days of the week (unified means that they have a similar average value and dispersion). For this reason, it is reasonable to construct the forecasting models that learn from data representing the same days of the week. For example, when we train the model to forecast the daily sequence for Monday, a training set for it is composed of the output patterns representing all Mondays from history and the corresponding input patterns representing the

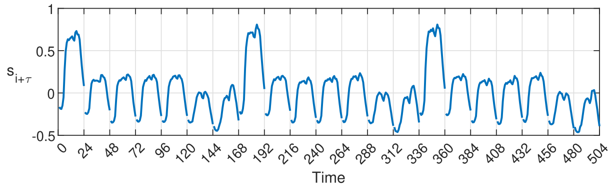

Fig. 12. Examples of output patterns for hourly electricity demand time series (first day is Monday, last day is Sunday).

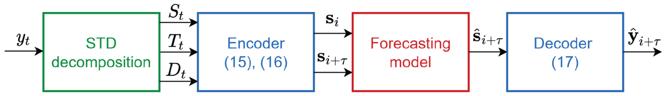

Fig. 13. Diagram of forecasting based on STD.

TABLE II
FORECASTING QUALITY METRICS FOR SHORT-TERM LOAD FORECASTING: MAPE - MEAN ABSOLUTE PERCENTAGE ERROR, MDAPE - MEDIAN OF APE, IQRAPE - INTERQUARTILE RANGE OF APE, AND RMSE - ROOT MEAN SQUARE ERROR

Model	MAPE	MdAPE	IqrAPE	RMSE
FNM*	2.50	2.30	2.29	334.08
GRNN*	2.48	2.28	2.27	332.91
SVM*	2.55	2.29	2.52	357.24
ARIMA	3.30	3.01	3.00	475.09
ES	3.11	2.88	2.73	439.26
Prophet	4.53	4.32	3.03	619.39
MTGNN	2.99	2.74	2.69	405.18
XGBoost	2.69	2.43	2.42	366.97
LightGBM	2.87	2.60	2.52	391.16
ES-adRNNe	**2.20**	**2.01**	**2.13**	**303.70**

previous days (depending on the forecast horizon; Sundays for $\tau = 1$).

The forecasting model fits function $f : \mathbf{s}_i \rightarrow \mathbf{s}_{i+\tau}$. The forecasted output pattern, $\hat{\mathbf{s}}_{i+\tau}$, is postprocessed to obtain the real forecasted sequence using transformed (16):

$$\hat{\mathbf{y}}_{i+\tau} = \hat{\mathbf{s}}_{i+\tau}\tilde{y}_i + \bar{y}_i \tag{17}$$

Note that in (17), the coding variables, $\bar{y}_i$ and $\tilde{y}_i$, are known from the most recent history. This enables us to perform the postprocessing (decoding).

Note that (15) and (16) filter out the current process variability from the data, i.e., filter out the local average and dispersion. The model learns on filtered (unified) patterns and forecasts output pattern $\hat{\mathbf{s}}_{i+\tau}$. Equation (17) introduces information about the process variability in sequence i (the most recent historical sequence) into the output data. This approach, depicted in Fig. 13, enables us to take into account the local variability of the process when constructing the forecast.

Due to representation of the time series by unified patterns $\mathbf{s}_i$ and $\mathbf{s}_{i+\tau}$, the forecasting problem simplifies and can be solved using simple models. The models proposed in [19] and [20], which were developed for short-term electrical load forecasting, are based on the similarity between the patterns. They assume that similarity in the input space is related to the similarity in the output space. Thus the forecasted output pattern is constructed from the training output patterns paired with the most similar input training patterns to the query pattern. To model function f, which in this approach has a nonparametric form, many models have been investigated such as the nearest-neighbor model, fuzzy neighborhood model, kernel regression model, general regression neural network, and pattern clustering-based models (including classical clustering methods and artificial immune systems).

In [21], function f was modeled locally using different linear models including stepwise and lasso regressions, principal components regression and partial least-squares regression. In [22], a random forest was used to model f, and in [23], different neural network architectures were compared. In [24], it was shown that STD decomposition improves the forecasting accuracy of an advanced hybrid and hierarchical deep learning model which combines exponential smoothing (ES) and residual dilated long short-term memory network (LSTM).

Table II shows the results for the short-term electrical load forecasting problem described in [25]. It concerns the prediction of hourly loads for the next day for 35 European countries. The models, which use pattern representation of input and output data described above are marked with an asterisk in Table II. They include the fuzzy neighborhood model (FNM), general regression neural network (GRNN), and support vector machine (SVM). The comparative models include standard statistical models such as ARIMA and ES, modern statistical model, Prophet, and state-of-the-art machine learning models such as boosted tree-based models (XGBoost and LightGBM; note the excellent performance of tree-based models in the forecasting competitions [26]), graph NN for multivariate TS forecasting (MTGNN) and a hybrid model combining ES and gated recurrent NN (RNN). The last model was developed in [25] especially for short-term load forecasting and is equipped with many procedures and mechanisms increasing its accuracy such as hierarchical RNN architecture, new recurrent cells with dilation and attention mechanisms, dynamic ES model, cross-learning and ensembling. As can be seen from Table II, the domain-adjusted ES-adRNNe model produces the most accurate forecasts. But when we compare the results of other models with the results of pattern-based models, we can see that the latter outperform the former in terms of accuracy. This proves that the pattern-based representation of the input and output data for STLF is advantageous. It simplifies the forecasting problem and enables the model to deal efficiently with multiple seasonality, nonlinear trend and variable variance.

2) Forecasting Individual Components of STDR: An alternative way of forecasting using the proposed decomposition approach is to predict each component of STD or STDR individually. Let us focus on STDR in this example. To predict time series for the next period using STDR, the seasonal component is just copied for this period, while the T, D and R components are forecasted using certain models. In this study, for these three components, we use a simple neural network, multilayer perceptron (MLP), composed of a single hidden layer with sigmoid nonlinearity and a linear output neuron. The predictors

TABLE III
RESULTS (sMAPE) OF FORECASTING M4 HOURLY TIME SERIES

MLP-STDR	SARIMA	ES-RNN	FFORMA	WASM	ESML	GROEC
12.26	14.45	8.86	10.63	9.44	9.57	14.56

TABLE IV
RESULTS OF FORECASTING FOR TIME SERIES SHOWN IN FIGS. 6, 7, 8, 9, 10, AND 11

	Airline	Unemp.	Electr.	S&P	S&Pr.	M-G
	sMAPE					
MLP-patterns	2.19	**2.13**	**1.71**	**1.06**	**127.14**	**2.77**
MLP-STDR	**2.15**	2.35	4.66	1.51	130.90	8.78
MLP-STL	3.32	2.47	12.76	1.27	128.55	12.35
MLP-raw	4.16	3.71	3.49	2.73	146.11	13.81
	RMSE					
MLP-patterns	**60.4**	**1.04**	**2376**	**350**	**0.097**	**0.316**
MLP-STDR	70.0	1.16	5720	537	0.114	0.926
MLP-STL	96.4	1.15	17503	370	0.126	1.254
MLP-raw	116.1	1.60	5103	835	0.440	1.481

are defined as the lagged points of the component series, and expressed by vectors $\mathbf{x}$ as follows:

- $\mathbf{x}_T = [\bar{y}_i, \ldots, \bar{y}_{i-l_T}]$ for T,
- $\mathbf{x}_D = [\tilde{y}_i, \ldots, \tilde{y}_{i-l_D}]$ for D,
- $\mathbf{x}_R = [R_t, \ldots, R_{t-l_R}]$ for R,

where l_T, l_D and l_R are the lag values for T, D and R, respectively.

The T and D components are predicted for $i+1$ period, while R is predicted for $t+1$. To cover period $i+1$, R is predicted recursively K times (the predictions of the previous steps are used to create the new lagged predictors). The proposed approach was applied to forecasting 414 hourly time series from the M4 competition dataset [28]. The time series were collected from a number of diverse sources and express daily and weekly seasonalities. The length of the time series is 700 or 960 (training part). We assumed a forecast horizon equal to 24 hours. The model hyperparameters were selected as follows: $l_T = l_D = 3$, $l_R = 24$, and the number of hidden neurons is 1.

Table III shows the results (symmetric mean absolute percentage error, sMAPE, which was the main accuracy measure in M4) for our method, SARIMA (ARIMA(1,1,1) model seasonally integrated with seasonal AR(24) and MA(24)) and the five top ranked methods in M4. They include:

- ES-RNN – a hybrid and hierarchical model combining ES and LSTM [29]. It includes such state-of-the-art solutions as: dilated LSTM layers, on-the-fly preprocessing, cross-learning, and three levels of ensembling.
- FFORMA – an automated method for obtaining weighted forecast combinations using time series features [30]. It employs a meta-model for assigning weights to various forecasting methods which include nine methods such as naive, random walk, Theta method, ARIMA, ES, TBATS model, STL with AR modeling and NN.
- WASM – a weighted aggregation of statistical methods such as naive, ES, Theta method, ARIMA and linear regression [31].
- ESML – an ensemble of statistical and machine learning methods such as TBATS model, THIEF ARIMA method, THIEF naive method, general ES, double seasonal Holt-Winters, multilayer perceptron, and extreme learning machine [32].
- GROEC – a combination method via generalized rolling origin evaluation. It produces a weighed combination of the Theta method, ES and ARIMA [33].

As can be seen from Table III, our model, MLP-STDR, outperforms in terms of accuracy the standard statistical model, SARIMA, but does not match the specialized M4 models except for the least sophisticated of them, GROEC. Note that the M4 models combine many different forecasting methods using ensembling. This gives them an advantage over individual methods like MLP-STDR and SARIMA.

Table IV compares forecasting results for six time series analyzed in Section IV-A (Figs. 6, 7, 8, 9, 10, and 11) using the following approaches: MLP-Patterns, which uses MLP based on patterns defined in Section IV-B1; MLP-STDR, which uses MLPs for individual forecasting of STDR components; MLP-STL, which uses MLPs for individual forecasting of STL components; and MLP-raw, which uses multilayer perceptron for forecasting the raw time series. The time series are forecasted in two final sequences of length $2n$ (sequence by sequence). A variant of the STL decomposition is used in which the seasonal component does not change over time (the same as in the STDR decomposition), allowing for the use of a naive method to predict this component. To predict the trend and residual components for the next period, MLPs are used. The input vector covers the recent historical sequence of length n, and the output vector includes the forecasted sequence of the same length. Similar definitions of the input and output vectors are used for forecasting the raw series. In MLP-STDR $l_T = l_D = 5$, and $l_R = n$. The number of hidden neurons was selected through experimentation. Fig. 14 shows the forecasted sequences.

Note that MLP-Patterns gives the lowest errors for all datasets except Airline data, where MLP-STDR provides a slightly lower sMAPE (but this was not confirmed statistically by Wilcoxon test). The greatest advantage in accuracy between MLP-Patterns and the other methods is observed for the Electricity and Mackey-Glass data, so for time series with the most complex and chaotic seasonality. Usually the largest errors are generated by MLP-raw, i.e., the method based on the raw time series without decomposition.

C. Discussion

The advantage of STD over the standard decomposition methods is that it extracts a dispersion component showing the short-term variability of the time series over time, i.e., variability of the series in seasonal periods. This is very useful for analysing heteroscedastic time series, which are very common in different domains such as finance, business, industry, meteorology etc. For example in finance, a statistical measure of the dispersion

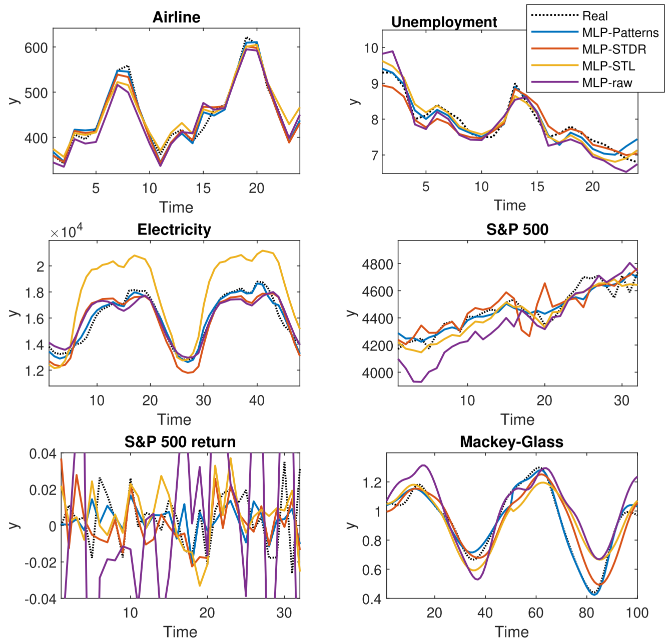

Fig. 14. Forecasts of time series shown in Figs. 6, 7, 8, 9, 10, and 11.

of returns for a given security or market index (a volatility) is extremely important. An asset's volatility is a key factor when pricing options contracts. The dispersion component enables direct evaluation, analysis and prediction of the series variance, which is not possible using standard methods, where this variance is expressed in many components, and additional calculations are required to estimate it. The graphical representation of the dispersion component provided by our method allows a quick visual evaluation of the properties of the series such as some regularities, periods of ups and downs, cyclicality, and seasonality. The picture given by the decomposition components helps us to choose an appropriate forecasting model, e.g., models requiring stationarity will not be appropriate in the case of variable dispersion.

The standard additive and multiplicative methods as well as STL produce an averaged seasonal component. It can be useful when our goal is to extract an averaged seasonal pattern but at the same time it can be viewed as a limitation when we want to evaluate how the seasonal pattern changes over time (averaged pattern masks these changes). Our proposed approach provides us with a choice: STDR produces an averaged seasonal pattern and STD produces seasonal components composed of the different seasonal patterns reflecting the real shapes of the seasonal sequences. So we can investigate the stability of the seasonal pattern or its variability over time which can be very useful in many fields. For example we can observe how the shape of the daily electricity demand curve differs for different days of the week and changes from season to season or year to year. We can also compare the shapes for different countries. Note that to measure shape variability, a simple euclidean distance between the seasonal patterns can be used because the patterns are represented by normalized vectors. The distance between a seasonal pattern and an averaged pattern can be a measure of atypicality. On this basis, outliers in seasonal shape can be detected. Comparing the shapes of seasonal cycles is impossible when using standard decomposition methods. This is because these methods either average the seasonal cycles, like the classical additive and multiplicative methods and STL, or express these cycles in many components, such as wavelet decomposition and EMD.

STDR averages the normalized seasonal patterns and delivers the reminder component. This component expresses the difference between real time series and series with unified seasonal cycles, i.e the series which has the same trend and dispersion components as the real series but its seasonal pattern is averaged. Analysing the reminder, we can detect periods in which the seasonal patterns differ from the averaged pattern most. For example, the unemployment time series shows increased differences in the shapes of seasonal cycles in periods 109-120 and 193-204. In these periods, the falling series temporarily increases (see Fig. 7). Patterns in the reminder can be further investigated in order to analyze the magnitudes and directions of deviations of seasonal cycles from the averaged cycles.

It is worth emphasizing the high interpretability of STD. It extracts easy to understand and informative components expressing the main properties of the series, i.e., the tendency of the series (trend averaged in seasonal periods), local variability of a series (dispersion in seasonal periods) and shapes of seasonal cycles (unified seasonal patterns). Compared to STD components, the components produced by standard methods, such as high frequency IMFs and details, are not easy to interpret. They do not express clear patterns.

Another very important advantage of STD and STDR are their simple algorithms, which can be coded in less then 30 lines of code in Matlab, as shown in Appendix A, available in the online supplemental material. The algorithms do not require complex computation. The components can be extracted using simple formulas (see mathematical formulation composed of just three equations for STD: (4), (5) and (8), and additional two for STDR: (11), (13)). Note that both versions, STD and STDR, have no parameters when used for seasonal time series. For non-seasonal series only one parameter should be selected, i.e., the "seasonality" period n. The simplest methods among the standard methods, the classical additive and multiplicative methods, require the selection of one parameter, i.e., the order of the moving average. More sophisticated methods, such as STL, wavelet decomposition and EMD, require more parameters to be selected. For STL these include: the spans of the Loess windows for the trend, seasonality and low-pass filter, and the degrees of the locally-fitted polynomials for trend, seasonality and low-pass filter. Wavelet decomposition requires the number of decomposition levels and wavelet type (or alternatively the coefficients of the low-pass and high-pass filters), while EMD requires selection of the interpolation method for envelope construction, decomposition stop criteria and shifting stop criteria. EMD suffers from a boundary problem which results in anomalously high amplitudes of the IMFs and artifact wave peaks towards the boundaries [27]. Another boundary problem occurs for classical additive and multiplicative decompositions. Due to the need to estimate the moving average using a two-sided window, the estimate of the trend and reminder are unavailable for observations near boundaries. In the proposed STD and STDR there are no boundary problems.

The existing methods use some form of smoothing to extract components. Thus the resulting components are smooth. STD/STDR does not need any smoothing algorithm (like Loess, high-pass and low-pass filtering or cubic splines), just averaging (4) or diversity calculation (5), which are extremely simple and fast. The resulting T and D components have a step-wise form, which can be useful for comparing the mean and dispersion of successive seasonal periods. Thanks to this we can evaluate changes in mean and dispersion for successive periods instead of "continuous" changes in the time series which are expressed by its smoothed representation. So the step-wise components produced by the proposed approach does not have to be considered a disadvantage.

Although STD and STDR were designed for time series with single seasonality, they can be used for non-seasonal time series. In such a case, the seasonal component does not express a regular pattern such as for S&P 500 returns (see Fig. 10) or expresses a pattern resulting from the general tendency of the time series such as for S&P 500 Index, where the rising "seasonal" patterns reflect the rising trend of the series (see Fig. 9). For non-seasonal time series parameter n should be selected in some way. When STD/STDR is used in the forecasting model, n is treated as an additional hyperparameter of the model and is selected in the grid search/Bayesian optimization and some variant of cross-validation to minimize the forecasting error. STD and STDR can also be useful for decomposition of time series with multiple seasonality. In such a case, the seasonal component expresses the seasonal patterns of the shortest period, and trend and dispersion components express seasonalities of the longer periods, see the example in Fig. 8. To extract all seasonal components, the STD/STDR decomposition can be applied for trend and dispersion components again.

Based on STD decomposition, we can define the input and output variables for the forecasting models. The input variables are just the seasonal patterns for period i, while the output variables are the seasonal cycles for period $i + \tau$ encoded using the average and dispersion for period i. Such encoding of both input and output variables filters out the trend and variability of the time series. This makes the relationship between the variables simpler. Thus this relationship can be modeled using simpler models such as linear regression or similarity-based models. Forecasting models using STD-based coding are great at dealing with time series with multiple seasonality, as has been proven in many papers [18], [19], [21], [22], [23].

V. Conclusion

Time series decomposition into several components representing an underlying pattern category is a key procedure for time series analysis and forecasting. In this work, we propose a new decomposition method, seasonal-trend-dispersion decomposition. It has two variants: with (STDR) and without (STD) the reminder component. The proposed decomposition can be summarized as follows:

1) It distinguishes itself from existing methods in that it extracts the dispersion component which expresses the short-term variability of the time series. A separate dispersion component is very useful for heteroscedastic time series analysis.
2) It produces interpretable components which express the main properties of the time series: the trend, dispersion and seasonal patterns.
3) In STD, the seasonal component is composed of centered and normalized seasonal patterns, which express the "shapes" of the seasonal cycles. By emphasizing these shapes, STD facilitates comparison and analysis of the seasonal cycles.
4) In STDR, the remainder component expresses the difference between the real seasonal cycles and the averaged cycles. It enables the detection of outlier seasonal cycles that differ in shape from the averaged cycles.
5) It has no parameters to adjust for seasonal time series. For non-seasonal time series, only one parameter should be selected.
6) The algorithms of STD and STDR are very simple and easy to implement. The computation time is extremely fast.
7) STD can be used for encoding the input and output variables for the forecasting models. STD-based encoding facilitates forecasting complex heteroscedastic time series with multiple seasonality. It simplifies the relationship between variables, which translates into simpler models.

References

[1] A. K. Palit and D. Popovic, *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications*. London, U.K.: Springer-Verlag, 2005.
[2] E. B. Dagum and S. Bianconcini, *Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation*. Cham, Switzerland: Springer, 2016.
[3] G. L. Mazzi, Ed., *Handbook on Seasonal Adjustment*. Luxembourg: Pub. Office Eur. Union, 2018.
[4] C. Buys-Ballot, *Les Changements Périodiques de Température*. Utrecht, The Netherlands: Kemink et Fils, 1847.
[5] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and Applications*, 3rd ed. New York, NY, USA: Wiley, 1998.
[6] A. Dokumento and R. J. Hyndman, "STR: Seasonal-trend decomposition using regression," *Informs J. Data Sci.*, vol. 1, no. 1, pp. 50–62, 2021.
[7] W. M. Persons, "Indices of general business conditions," *Rev. Econ. Statist.*, vol. 1, pp. 111–205, 1919.
[8] F. R. Macaulay, *The Smoothing of Time Series*. New York, NY, USA: Nat. Bur. Econ. Res., 1931.
[9] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. J. Terpenning, "STL: A seasonal-trend decomposition procedure based on loess," *J. Official Statist.*, vol. 6, no. 1, pp. 3–33, 1990.
[10] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[11] N. E. Huang et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proc. Roy. Soc. London Ser. A: Math. Phys. Eng. Sci.*, vol. 454, no. 1971, pp. 903–995, 1998.
[12] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, Feb. 2014.
[13] N. Golyandina, V. Nekrutkin, and A. A. Zhigljavsky, *Analysis of Time Series Structure: SSA and Related Techniques*, 1st ed. London, U.K./Boca Raton, FL, USA: Chapman and Hall/CRC, 2001.
[14] R. J. Hyndman and C. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts. Accessed: Mar. 09, 2022. [Online]. Available: OTexts.com/fpp3

[15] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2016.

[16] J. G. De Gooijer, *Elements of Nonlinear Time Series Analysis and Forecasting*. Cham, Switzerland: Springer, 2017.

[17] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 287, pp. 287–289, 1977.

[18] G. Dudek, "Pattern similarity-based methods for short-term load forecasting – Part 1: Principles," *Appl. Soft Comput.*, vol. 37, pp. 277–287, 2015.

[19] G. Dudek, "Pattern similarity-based methods for short-term load forecasting – Part 2: Models," *Appl. Soft Comput.*, vol. 36, pp. 422–441, 2015.

[20] G. Dudek and P. Pełka, "Pattern similarity-based machine learning methods for mid-term load forecasting: A comparative study," *Appl. Soft Comput.*, vol. 104, 2021, Art. no. 107223.

[21] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Syst. Res.*, vol. 130, pp. 139–147, 2016.

[22] G. Dudek, "Short-term load forecasting using random forests," in *Proc. Adv. Intell. Syst. Comput.*, 2015, pp. 821–828.

[23] G. Dudek, "Neural networks for pattern-based short-term load forecasting: A comparative study," *Neurocomputing*, vol. 2015, pp. 64–74, 2016.

[24] G. Dudek, P. Pełka, and S. Smyl, "3ETS+RD-LSTM: A new hybrid model for electrical energy consumption forecasting," in *Proc. Int. Conf. Neural Inf. Process.*, 2020, pp. 519–531.

[25] S. Smyl, G. Dudek, and P. Pelka, "ES-dRNN with dynamic attention for short-term load forecasting," in *Proc. Int. Joint Conf. Neural Netw.*, 2022, pp. 1–8, doi: 10.1109/IJCNN55064.2022.9889791.

[26] C. S. Bojer and J. P. Meldgaard, "Kaggle forecasting competitions: An overlooked learning opportunity," *Int. J. Forecasting*, vol. 37, pp. 587–603, 2021.

[27] A. Stallone, A. Cicone, and M. Materassi, "New insights and best practices for the successful use of empirical mode decomposition, iterative filtering and derived algorithms," *Sci. Rep.*, vol. 10, 2020, Art. no. 15161.

[28] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 competition: Results, findings, conclusion and way forward," *Int. J. Forecasting*, vol. 34, no. 4, pp. 802–808, 2018.

[29] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *Int. J. Forecasting*, vol. 36, no. 1, pp. 75–85, 2020.

[30] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, "FFORMA: Feature-based forecast model averaging," *Int. J. Forecasting*, vol. 36, no. 1, pp. 86–92, 2020.

[31] M. Pawlikowski and A. Chorowska, "Weighted ensemble of statistical models," *Int. J. Forecasting*, vol. 36, no. 1, pp. 93–97, 2020.

[32] S. Jaganathan and P. K. S. Prakash, "A combination-based forecasting method for the M4-competition," *Int. J. Forecasting*, vol. 36, no. 1, pp. 98–104, 2020.

[33] J. A. Fiorucci and F. Louzada, "GROEC: Combination method via generalized rolling origin evaluation," *Int. J. Forecasting*, vol. 36, no. 1, pp. 105–109, 2020.

Grzegorz Dudek received the PhD degree in electrical engineering from the Czestochowa University of Technology (CUT), Poland, in 2003, and the habilitation degree in computer science from the Lodz University of Technology, Poland, in 2013. Currently, he is an associate professor with the Department of Electrical Engineering, CUT. He is the author of two books concerning machine learning methods for load forecasting and evolutionary algorithms for unit commitment and more than 120 scientific papers. He came third in the Global Energy Forecasting Competition 2014 (price forecasting track). His research interests include pattern recognition, machine learning, artificial intelligence, and their application to practical classification, regression, forecasting, and optimization problems.