Introducing a Conditional GAN-Based Approach for Seasonality adjustment

November 29, 2024

Abstract

This paper evaluates the performance of seasonality adjustment methods using Monte Carlo simulations of synthetic time series with RBC-Slutzky-type business cycles and a stochastic trend. I introduce a new conditional Generative Adversarial Network (cGAN)-based method, cGAN-LOESS, for seasonal extraction. Unlike traditional smoothing or filtering methods, cGAN-LOESS learns the general underlying distribution of a data-generating process, which must be specified by the analyst. Once trained, the model requires only three parameters to adjust. The analysis compares cGAN-LOESS with X-13-ARIMA-SEATS and STL by calculating mean squared error (MSE) against known seasonal components in 2,000 simulated series. Results show that cGAN-LOESS outperforms traditional methods, particularly when combined with ensemble approaches.

Contents

List of Figures

1 Introduction

Many time series exhibit seasonal patterns that can bias the analysis if the primary interest lies in identifying the cycle or trend components. For example, a decrease in manufacturing production from June to July should not be immediately interpreted as an economic slowdown, as production cutbacks are common at the start of the summer holiday season. Filtering for seasonality is therefore an essential preprocessing step for many time series analyses. Various methods exist for filtering seasonality, but statistical offices around the world have largely concentrated X-13-ARIMA-SEATS (X13), Tramo/Seats, Berliner Verfahren and STL.

Although these methods are statistically well-supported, no systematic analysis of their performance currently exists. In real-world data, researchers only have access to the additive time series, not to its underlying, true components. But without a ground truth at hand, it is impossible to distinguish between stylized facts and statistical artifacts. The same problem exists for business cycle filters, where the filtering method generates an unobservable cycle component. However, the performance and degree of distortions of business cycle filters can be analyzed in the frequency domain, where a ideal high-pass filter, serves as the benchmark or "true filter", as demonstrated by Pedersen (2001). A ideal high-pass filter has

a power transfer function of the form $H^*_{hp}(\omega) = 0$ if $|\omega| < \omega_1$, 1 if $|\omega| \geq \omega_1$. Unfortunately, an ideal seasonality highpass (or bandpass) filter would also catch the harmonic frequencies (or "overtones") above the fundamental frequency of the cyclic component, making it unsuitable as an approximation for the true seasonality component.

To accurately evaluate these seasonality adjustment techniques, access to the true seasonal component or an "observed components" framework is necessary for a reliable comparison. In this paper, the performance of the X13, STL and a new introduced cGAN-LOESS method for seasonality prediction are evaluated on a set of 2000 monte carlo simulated synthetic time series with the known seasonal ground truth. To evaluate each method's performance, I estimated the MSE between the extracted seasonality and the ground truth. Additionally, a combined ensemble model is constructed by averaging the extracted seasonality components from the three base methods.

I introduce a new method for time series seasonality prediction based on a conditional generative adversarial network (cGAN), which learns a mapping from input vectors to output vectors, as described by Isola et al. (2016). A cGAN pix2pix model can be used in a wide range of tasks, including removing noise from images, generating colorized photos from black-and-white images or transforming sketches into photos. The ability to extract noise in images can be analogously applied to time series by treating the time series as a single-channel image to isolate different components.

2 Conventional Methods for seasonal adjustment

To accurately perform seasonality adjustments or generating synthetic seasonality, it is essential first to define what seasonality represents within time series data. Seasonality refers to recurrent patterns that consistently appear at regular intervals, driven by predictable external factors like weather or institutional calendars, such as holidays. These recurring patterns must exhibit a fixed frequency, typically daily, weekly, or yearly, to qualify as seasonality, which distinguishes seasonality from other time series components. In contrast, business cycles exhibit a frequency within a range rather than a fixed, specific frequency. Random occurrences, even if related to the seasonal factor, do not inherently create seasonality. Suppose store revenue tends to peak on Saturdays, primarily because consumers have more time to shop on weekends. This weekly seasonality follows a regular pattern, which we recognize as seasonality. However, if an unseasonably warm Saturday leads to lower-than-expected store revenue due to outdoor activities, this deviation should not alter the seasonal pattern. Here, the decline is due to a weather shock—a random influence external to the underlying seasonality. Similarly, if a particular

Friday becomes a national holiday, temporarily increasing revenue due to early shopping, this would represent an outlier rather than a change in the seasonal structure, unless this Friday holiday became a recurring event. Thus, random disturbances should be distinguished from the inherent seasonality and treated as either noise or influences within the cyclical or trend components. While seasonality may shift over time due to gradual structural changes—such as evolving social norms around work and shopping habits—these shifts must exhibit clear structure. The structure itself is often modelable through autoregressive processes or deterministic trends. Methods like STL decomposition try to capture these structure shifts by smoothing trends in the seasonal subseries.

2.1 X-12 Family

The X-12 Family of seasonal adjustment methods includes the original X-11 method and its successors X-11-ARIMA, X-12-ARIMA and X-13-ARIMA-SEATS. The Census Bureau's X-11 program for seasonal adjustment, introduced in 1978 (Shiskin (1978)), has become a widely used tool globally. This method employs a filtering process of iterative weighted symmetric moving averages to separate trend-cycle and seasonal components from a time series. Key features contributing to X-11's popularity include its handling of extreme observations, diverse moving averages for estimating trend-cycle and seasonal components, refined asymmetric moving averages for endpoints, and techniques for estimating trading-day effects.

The subsequent X-11-ARIMA program (Dagum (1980)) enhanced X-11 by extending the time series with ARIMA-based forecasts and backcasts, improving stability of symmetric filters, especially at series endpoints.

The next development was the X-12-ARIMA and is introduced by the Census Bureau and detailed in Findley et al. (1998). The most important enhancements of X-12-ARIMA over X-11-ARIMA is the model-based calendar effect adjustment using a RegARIMA model, and several new diagnostic tools to test the quality of seasonal and calendar effect adjustment. Similarly to the X-11-ARIMA a forecasts and backcasts are used before seasonal extraction with filters. The RegARIMA model used in X-12-ARIMA combines regression with ARIMA (AutoRegressive Integrated Moving Average) to model calendar and outlier influences. The RegARIMA model is represented as:

$$\ln(X_t) = \sum \alpha_l \cdot k_{lt} + \sum \beta_m \cdot LS_{mt} + \sum \gamma_n \cdot AO_{nt} + \sum \lambda_v \cdot TC_{vt} + Z_t \qquad (1)$$

where k_{lt} are calendar regressors (e.g., the number of working days per month), LS_{mt} represents level shifts, AO_{nt} are additive outliers, TC_{vt} indicates temporary changes, and Z_t is an ARIMA$(p, d, q)(P, D, Q)$ model component. Common ARIMA specifications, such as (011)(011), apply a difference operation both to

the previous period ($d = 1$) and to the same period in the prior year ($D = 1$), with moving averages ($q = 1, Q = 1$) that depend on recent and seasonal influences. The RegARIMA modeling within X-12-ARIMA serves two key functions. First, it facilitates the identification and removal of calendar effects, ensuring more accurate seasonal component estimation. Second, it enables forward and backward projections, enhancing the stability of endpoint values in the filtering process and reducing distortions at the series boundaries. The second stage in X-12-ARIMA involves applying a Henderson filter to estimate the trend-cycle component, followed by smoothing of the seasonal factors. The seasonal factors (or seasonal-cycle subseries), created for each calendar month (e.g., January values across years), are smoothed with a series of moving averages to establish consistent seasonal factors. These seasonal-cycle subseries serve as an essential diagnostic tool, enabling a more refined seasonal adjustment process.

The recent version X-13-ARIMA-SEATS uses the procedures of the predecessors and additionally uses the SEATS method developed by Gómez and Maravall (2001) at the bank of spain.

2.2 STL

STL is a filtering procedure for decomposing time series into trend, seasonal, and remainder components by applying a loess smoother and various moving averages. Proposed by Cleveland et al. (1990), the method allows for individualized specification of the seasonal component's period, robust estimates of the trend and seasonal components, and the ability to decompose time series with missing values. STL consists of an inner and an outer loop. In each iteration of the inner loop, the seasonal and trend components are updated. Each outer loop includes the inner loop followed by a calculation of robustness weights, which reduce the influence of transients in the next inner loop iteration. Within each iteration of the inner loop, a detrended series is first computed. Then, each seasonal-cycle subseries of the detrended series is smoothed by loess, and a low-pass filter is applied to the smoothed seasonal-cycle subseries. The seasonal-cycle subseries is a key concept and diagnostic tool in STL. For instance, with monthly data and a yearly periodicity, the seasonal-cycle subseries comprises values at each position in the seasonal cycle. The first subseries contains January values for all years, the second contains February values, and so forth.

In this analysis, I will use the concept of the seasonal-cycle subseries (sometimes also called seasonal factors) in the cGAN-LOESS approach for seasonality smoothing and for diagnostics.

3 A short introduction to Generative Adversarial Networks

Goodfellow et al. (2014) A generative adversarial network (GAN) belongs to the family of generative models, which includes other methods such as restricted Boltzmann machines, principal component analysis or naive Bayes. Generative models learn the joint probability distribution, $P(X, Y)$ (or $P(X)$ if unsupervised), of inputs X and outputs (or labels) Y. By learning $P(X, Y)$, these models can predict the most likely output (e.g. by using Bayes' theorem for classification tasks). In contrast, discriminative (or supervised) models aim to model the conditional probability distribution $P(Y \mid X)$ directly. This approach serves as a direct mapping from inputs X to outputs Y, bypassing the need to model the full underlying data distribution.

Unconditional Generative Adversarial Networks (GANs), initially proposed by Goodfellow et al. (2014), are generative models that learn a mapping from a random noise vector z to an output image y, $G : z \to y$. A GAN consists of two simultaneously trained models, a generator and a discriminator, which are typically both multilayer perceptrons. The generator aims to approximate the distribution of real data, p_{data}, by learning its own distribution, p_g, over data samples x. To achieve this, the generator utilizes a mapping function $G(z; \theta_g)$, where z are random noise variables typically sampled from a uniform distribution $p_z(z)$, and θ_g represents the learnable parameters of the generator's multilayer perceptron. This function transforms the noise z into a synthetic sample in the data space, attempting to resemble the real data. Simultaneously, the discriminator $D(x; \theta_d)$, where θ_d denotes the discriminator's parameters, is trained to estimate the probability that a given sample x originates from the real data distribution p_{data} rather than from the generator's distribution p_g. In essence, $D(x)$ outputs the probability that x is a real data point rather than a synthetic sample generated by G. The training process is framed as a two-player minimax game, where G and D are optimized according to the following objective:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Here, D is trained to maximize the probability of correctly distinguishing real samples from generated ones, while G is trained to maximize the probability of D making a mistake. Through this adversarial process, G learns to produce samples that are increasingly difficult for D to distinguish from the real data, effectively bringing p_g closer to p_{data} as training progresses. One major drawback of GANs has been the challenge of controlling their output.

Conditional GANs (cGANs), first proposed by Mirza and Osindero (2014), extend the GAN by introducing additional input information y, enabling control

over the characteristics of the generated data. The conditioning is performed by feeding y into both the discriminator and generator as additional input layer. The generator therefore learns a mapping from both y and the random noise z, to a target output x, $G : (y, z) \rightarrow x$. The objective of a cGAN is given by:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))]$$

Conditional GANs (cGANs), compared to unconditional GANs and most other methods is their use of a structured loss function, instead of an unstructured loss. Conditional GANs learn a structured loss that penalizes the joint configuration of output pixels, instead of penalizing each data point conditionally independent from all other data points, resulting in a per-pixel or per-data point regression Isola et al. (2016). That allow cGANs to capture complex pattern within the output space. The architecture of the GAN compared to a cGAN is illustrated in Figure 1 and Figure 2. The discriminator learns to classify between fake tuples (x,G(x)) and real tuples (x,y).

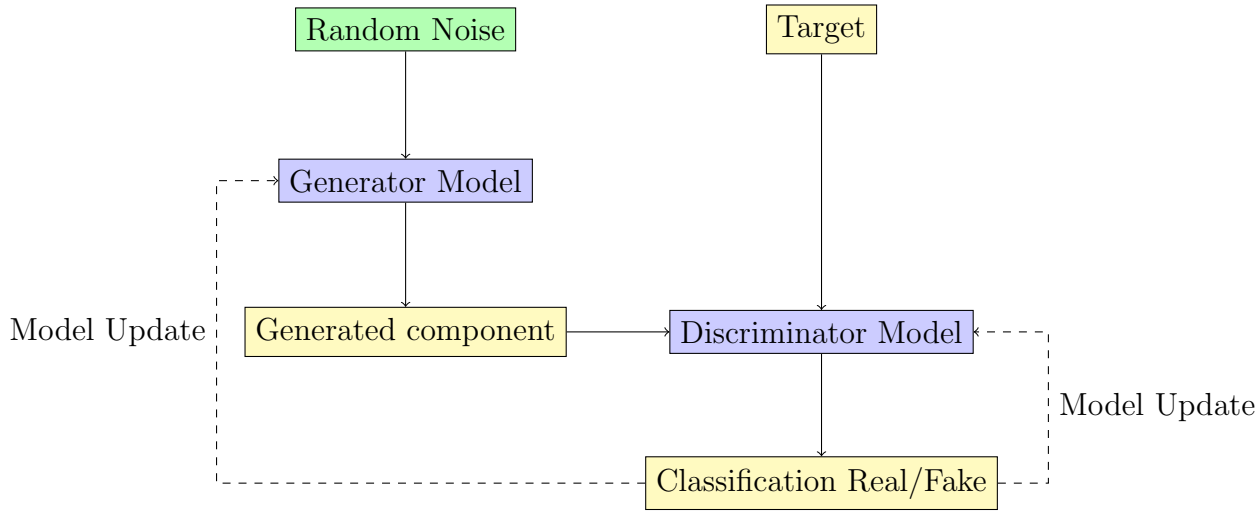

Figure 1: Flowchart of a GAN

4 A conditional generative adversarial network for seasonality adjustment

An unconditional GAN could be used to generate new, unseen seasonality structures (or any other time series component) by training the model on various seasonality patterns as output x. A conditional GAN (cGAN) additionally allows

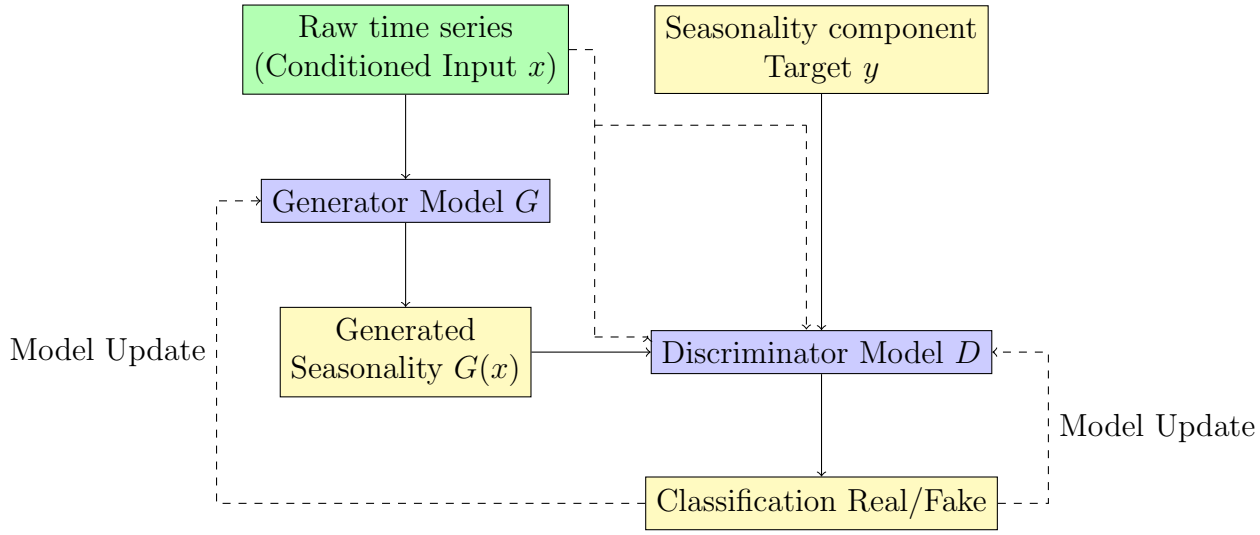

Figure 2: Flowchart of a Conditional GAN

to condition the generation of seasonality patterns on the whole time series that includes the seasonality a a sub-component. As described above, the cGAN generator learns a mapping from the whole time series with all components y (the condition) and random noise z to the seasonality component x (the target output). This architecture assumes a additive time series model $y_t = x_t + c_t + t_t + i_t$ independent components, t_t, the trend component and i_t is an irregular part.

In our cGAN implementation, we adopted the same general architecture as proposed by Isola et al. (2016), meaning a U-net based generator and a convolutional PatchGAN discriminator. The "randomjitter" was removed and the input/output dimensions were modified from the original 256x256x3 (an RGB image) to 256x1, which matches the dimension of a time series with 256 observations.

Most methods for seasonality extraction aim to isolate or extract the seasonal component from other components through various filtering procedures. In contrast, the cGAN approach is a model-based method that actively predicts the seasonality component. Seasonality extraction using cGANs requires pre-trained models, and the prediction accuracy depends on the quality and suitability of the training data, defined by the researcher. Ideally, the training data should have a similar structure to the time series being deseasonalized. Furthermore an important characteristic of the pix2pix cgan architecture is, that some layers behave stochastically (i.e. dropout and batch normalization), meaning that the predictions of the same input will differ slightly even after finished training phase. The most important cGAN parameter to be specified by the researcher is the kernel size of the convolutional layers specifying the size of the transposed convolution window. This parameter should be set equal to the periodicity of the seasonality,

e.g. 12 for monthly data and yearly seasonality. The cGAN-LOESS seasonality extraction involves the following steps:

I. Time series pre processing and model Training:

1. **Synthetic time series generation**: Generartion of synthetic time series with a separate seasonality component.
2. **Time series pre-processing**: Detrending and Normalization to match the input format of the cGAN.
3. **Model Training**: Train the cGAN model on the pre-processed synthetic data to learn seasonality patterns.

II. Seasonality Prediction:

1. **Time series pre-processing**: Detrending with the Hodrick-Prescott filter and Normalization to match the input format of the pre-trained cGAN.
2. **Prediction**: Use the pre-trained cGAN model to predict the seasonality component from the normalized input.
3. **Inverse Scaling**: Reverse the scaling to obtain the seasonality component in its original scale.
4. **LOESS Smoothing**: Apply LOESS smoothing to each seasonal factor group and find optimal bandwidth using cross validation.

As a first preprocessing step, the time series is detrended using the Hodrick-Prescott filter as introduced by Hodrick and Prescott (1997) with a cutoff frequency set above the periodicity of the seasonality. For yearly seasonality a cutoff of 2 years can be used. The HP filter minimizes the following objective function:

$$\min_{\{\tilde{y}_t\}_{t=1}^{T}} \left\{ \sum_{t=1}^{T} \left(y_t - \tilde{y}_t\right)^2 + \lambda \sum_{t=2}^{T-1} \left[(\tilde{y}_{t+1} - \tilde{y}_t) - (\tilde{y}_t - \tilde{y}_{t-1})\right]^2 \right\}$$

The filter computes a stochastic trend $\{\tilde{y}_t\}_{t=1}^{T}$ by minimizing the squared deviations from the trend $\left(y_t - \tilde{y}_t\right)^2$, constrained to keep the squared second differences small $\left[(\tilde{y}_{t+1} - \tilde{y}_t) - (\tilde{y}_t - \tilde{y}_{t-1})\right]^2$.

The frequency response function of the Hodrick-Prescott filter is given by:

$$R(\omega) = \frac{1}{1 + 4\lambda[1 - \cos(\omega)]^2}$$

Now solving for the HP-filter parameter λ:

$$\lambda = \frac{1 - R(\omega)}{4R(\omega)[1 - \cos(\omega)]^2}$$

It can be shown that in the case of the HP filter, the gain function equals the frequency response function. Using a gain of $R(\omega) = 0.5$ and replacing the frequency in radians (ω) with the ordinary frequency (f) multiplied by 2π, the equation can be simplified as follows:

$$\lambda = \frac{1 - 0.5}{4 \cdot 0.5 \cdot [1 - \cos(\omega)]^2} = \frac{0.25}{[1 - \cos(\omega)]^2} = \frac{0.25}{[1 - \cos(2\pi f)]^2}$$

The lambda parameters are set so that the frequency cut-off occurs at frequencies higher than 2 years. Therefore, the corresponding ordinary frequencies of $f_1 = 1/2$ is used:

$$\lambda_1 = \frac{0.25}{\left(1 - \cos\left(\frac{1}{12}2\pi\right)\right)^2} = 13.93$$

After detrending, the time series is normalized by dividing it by the largest absolute value. This ensures that the time series remains within the range [-1,1] and that the normalization can be easily reversed.

After predicting the time series, LOESS smoothing, similar to that used in the STL and X-12-ARIMA methods, is applied separately to each seasonal factor group (e.g., to each month across all years for yearly seasonality, or each day across all weeks for weekly seasonality). This approach removes high frequency fluctuations from the seasonality component while allowing for smooth transitions in the seasonal pattern over time. The LOESS algorithm, proposed by Cleveland (1979), estimates a smooth curve through local linear regression. The local linear regression for a specific seasonal factor (e.g. Jan. 2000) x is defined as:

$$\min_{\alpha,\beta} \sum_{i}^{N} \left(Y_i - (\alpha + \beta X_i)\right)^2 K(\mathbf{X}_i, \mathbf{x}, \mathbf{H}), \tag{2}$$

where $i = 1, \ldots, N$ are the specific seasonal factors within a seasonal factor group, $K(\mathbf{X}_i, \mathbf{x}, \mathbf{H})$ is a kernel function that weights the closeness of all other seasonal factors (e.g. Jan. 2001, Jan. 2002 ...) $\mathbf{X}_i$ to the specific seasonal factor (e.g. Jan. 2000) $\mathbf{x}$. The minimization problem estimates the optimal parameters α, β, such that the squared distance between the seasonality value Y_i and $(\alpha + \beta X_i)$ is minimized, taking into account the weights of each seasonal factor. In this case, a tricube weighting function with bandwidth h is used:

$$K\left(\frac{|X_i - x|}{h}\right) = \left(1 - \left(\frac{|X_i - x|}{h}\right)^3\right)^3 \tag{3}$$

Choosing the Bandwidth in a LOESS model has a great impact because a small bandwidth will reduce bias but introduce overfitting and a high variance. On the other hand a high bandwidth will create a low variance estimator, in extreme case a linear one, with possible failing to capture the shifts in seasonality.

5 Handling the Hodrick-Prescott filter endpoint bias

The HP filter is widely used because of its advantegous properties such as the ability to generate stationary time series when time series integrated up to order four (King and Rebelo (1993)), or of course the symmetry of the filter, such that no phase shift is introduced. Finally, as shown by Pedersen (2001), there is no cycle in the power transfer function, meaning that the filter does not induce spurious cycles such as the Slutzky or Kuznets filters. In fact the HP filter is an close approximation to an ideal high-pass filter, which is a filter which sharply cuts off components at frequencies below and above the bandpass cutoffs. However there are some drawbacks of the HP filter as clearly shown by Hamilton (2017). For example, the HP filter leaks unwanted frequencies and suppresses some fluctuations it should preserve. Especially the fact that Filtered values at both ends of the sample are very different from those in the middle because the HP filter is an symmetric infinite dimensional moving average filter in the time domain, allpied to non infinite time series. Therefore, some authors even advise to cut of the first and last observations to minimize the problem (Baxter and King (1999)). Of course this is not an option for small size time series. Furthermore especially the last observations are in most scenarios of highest interest, and removing them would reduce the worth of the analysis.

To reduce endpoint bias, three-step are conducted. First, forecasts are generated for both ends of the time series (a backcast for the beginning and a forecast for the end). Second, the HP filter is applied to this extended series. Finally, the forecasted values are trimmed to restore the original series length.

To evaluate the performance of this method, I generated $N = 1,000$ synthetic time series Y_i with the same characteristics, of the synthetic time series used for the cGAN. Each series Y_i has a length of $t = 10,000$ and was reduced to a 120 observations middle segment y_i. An SARIMA(p,d,q)(P,D,Q,s) model with $p = 11, d = 1, q = 0, P = 1, D = 1, Q = 0, s = 12$ was fitted to the middle segment y_i to forecast the next 24 values. For the backcast, y_i was reversed, and then forecasted; the forecast was reversed back to its original. Forecasts and backcasts are added to the end and beginning of y_i to create the extended series y_i^*.

Then, the HP filter generates the extended series cycle c_i^* from y_i^* , the orig-

inal series cycle c_i from y_i and a "true" cycle C_i from Y_i. Figure 3 shows the average absolute error of both methods, given by $ME_t = \frac{1}{N}\sum_{i=1}^{N} C_{it} - c_{it}$ and $ME_t^{ext.} = \frac{1}{N}\sum_{i=1}^{N} C_{it} - c_{it}^{*}$ The effectiveness of this method in reducing the HP filter's endpoint bias naturally depends on the predictability of the series. In the simulation, the mean error over all 120 time points for the standard method was 0.01506, whereas the mean error for the extended method was 0.01172. This represents an error reduction of 22%.

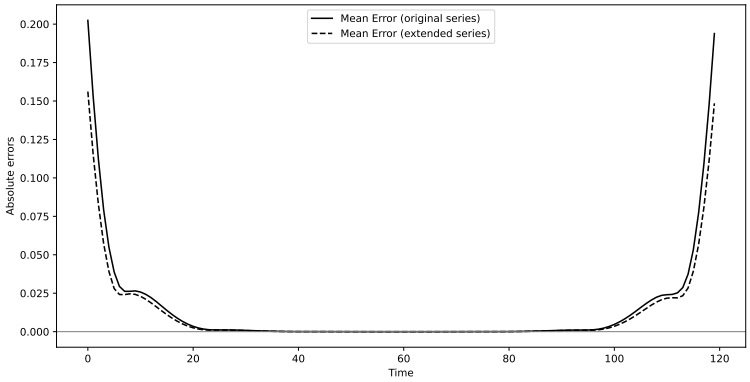

Figure 3: Hodrick-Prescott endpoint error simulation

6 Synthetic Time Series generation and Construction

It is essential to construct time series that closely mimic the dynamics observed in real-world economies, because both the accuracy of the cGAN-LOESS seasonality predictions and the validity of the performance comparison for the time series methods depend on the synthetic data. The synthetic time series model combines several key components: a stochastic trend (T_t), two cycles (LC_t and SC_t)—incorporating characteristics from both the Real Business Cycle (RBC) model and Slutsky-type cycles—a stochastic seasonality component (S_t), and an outlier component. The outlier component accounts for additive outliers (ao_t), temporary changes (tc_t) and structural breaks or level shifts (ls_t). The components are combined additively as follows:

$$Y_t = T_t + LC_t + SC_t + S_t + (ao_t + tc_t + ls_t) \tag{4}$$

It is important that in this type of model, the components are uncorrelated. As described above, the seasonality component is not affected by any short term

shocks and is a deterministic recurrent pattern with a specific periodicity. A zero correlation between the trend component and the cycle components can be generated with a standart text book RBC model where technology shocks drive stationary fluctuations around a deterministic exogenous growth trend. Shocks to the cyclical and secular components are orthogonal. That would be not the case in endogenous growth models where shocks to the trend and the cyclical component are perfectly correlated. The two cycle components (LC_t and SC_t) are generated using two theories, namely RBC and Slutzky.

In 1937, E.E. Slutsky's article, "The Summation of Random Causes as the Source of Cyclic Processes," was published in Econometrica, following its initial publication in the theoretical journal of the Moscow Conjuncture Institute. In this work (Slutzky (1937)), he investigated if it is possible "that a definite structure of a connection between random fluctuations could form them into a system of more or less regular waves?" (Slutsky: 106). The article was considered groundbreaking for a good reason: If Slutsky was correct, then business cycles could potentially be explained as the result of the summation of random causes without the need to specify causal factors behind these cyclical fluctuations. While underlying causes of business cycles certainly exist, Slutsky demonstrated that such causes need not occur periodically to generate cyclical behavior. For example, he noted that the cumulative effect of random events—such as rainfall on crop yield—could produce regular patterns.

Slutsky transformed as series of random numbers by applying single or multiple different moving average and differencing procedures sequentially. Kuznets (1929) discussed and supported Slutsky's ideas and added that extremely large deviation can explain cyclical swings when moving averages are applied. "In a moving average or in a cumulation, this deviation, exceptional in its size, will raise or depress the level of all the members that include it, and instead of one high or low point we get a high or low plateau." (Kuznets, S. 267) One year before Slutsky's publication, in the year 1926, G.U. Yule independently discovered similar characteristics, introducing the concept of autoregression in his analysis. Spanos (1990) mentioned that Wold (1938) established a connection between the autoregressive (AR(p)) and moving average (MA(q)) formulations of Yule and Slutsky.

However, Slutsky's explanation appears incomplete. Slutsky attributes economic fluctuations to aggregated, random individual choices that lead to oscillating patterns, but this overlooks the fact that individual choices are rarely random; current choices often depend on past decisions, and sometimes on future expectations. While Slutsky's MA model does not capture these dynamics, the RBC model captures prior shocks through an autoregressive process.

Shocks in the RBC model are represented by shifts in technology or productivity, rooted in the decisions and behaviors of individual economic agents. These

individual actions accumulate to create random shocks in each period. The choices of economic agents impact total economic output each period, either positively or negatively. This results in volatile fluctuations around the economy's equilibrium, which is defined by factors such as infrastructure, industry capacity, know-how, and other elements that determine productive potential.

Here, Slutsky's insights on moving averages should be considered. Due to long-term contracts, imperfect information, labor market restrictions, and other factors, the economy does not respond instantly to shocks; rather, it reflects an average of prior shocks.

Thus, in the two cycles of the synthetic time series, the autocorrelated decisions or shocks of agents are modeled by first constructing a time series in the style of an RBC model, followed by applying moving averages to capture the economy's slow response to these shocks. The long cycle applies a moving average with a larger window size, capturing more extended cycles, similar to a super-cycle or Kondratieff cycle, while the short cycle employs a smaller window, capturing fluctuations typical of a business cycle.

The stochastic trend component T_t is a random walk with drift, defined by:

$$\Delta T_t = T_t - T_{t-1} = \delta + \epsilon_t, \tag{5}$$

Recursively substituting for T shows that this trend builds cumulatively over time as:

$$T_t = T_0 + t\delta + \sum_{i=1}^{t} \epsilon_{T,i} \tag{6}$$

where δ is the drift parameter, with $\delta \sim \mathcal{N}(0, 0.025^2)$ and $\epsilon_{T,i}$ is random noise with $\epsilon_{T,i} \sim \mathcal{N}(0, \sigma_T^2)$, and $\sigma_T \sim \mathcal{U}(0.01, 0.2)$.

The long cycle component LC_t is constructed as an AR(p) process with moving average smoothing applied afterwards:

$$LC_t = \epsilon_{LC,t} + \sum_{i=1}^{p} \phi_i \cdot LC_{t-i} \tag{7}$$

where $\epsilon_{LC} \sim \mathcal{N}(0, \sigma_{LC}^2)$, with $\sigma_{LC} \sim \mathcal{U}(2, 5)$, and geometrically decaying parameters $\phi_i = \phi_{i,0} \cdot (0.5)^i$ with $\phi_{i,0} \sim \mathcal{N}(0, 0.5)$.

After generating the AR series, a moving average with window size $W_{LC} \sim \mathcal{U}(200, 250)$ is applied to smooth the long cycle component:

$$LC_t = \frac{1}{W_{LC}} \sum_{j=0}^{W_{LC}-1} LC_{t-j} \tag{8}$$

Similarly, the short cycle component SC_t is defined as:

$$SC_t = \epsilon_{SC,t} + \sum_{i=1}^{p} \psi_i \cdot SC_{t-i} \tag{9}$$

where $\epsilon_{SC} \sim \mathcal{N}(0, \sigma_{SC}^2)$, with $\sigma_{SC} \sim \mathcal{U}(3, 7)$, and each $\psi_i = \psi_{i,0} \cdot (0.5)^i$ with $\psi_{i,0} \sim \mathcal{N}(0, 0.5)$.

Subsequentially, a moving average with window size $W_{SC} \sim \mathcal{U}(48, 72)$ is applied:

$$SC_t = \frac{1}{W_{SC}} \sum_{j=0}^{W_{SC}-1} SC_{t-j} \tag{10}$$

If $\sum_{i=1}^{p} \psi_i > 1$ or $\sum_{i=1}^{p} \phi_i > 1$, the parameter series' are scaled to stabilize the process.

The seasonality component S_t is composed of two seasonal patterns, $S_{1,t}$ and $S_{2,t}$, which change in relative weight over time according to w_t:

$$S_t = S_{1,t} \cdot w_t + S_{2,t} \cdot (1 - w_t) \tag{11}$$

Each seasonal pattern is a cumulative random walk with seasonal shocks over 12 periods. For the pattern $S_{1,t}$:

$$S_{1,t} = \sum_{i=1}^{12} \epsilon_{S_1,i} \tag{12}$$

where $\epsilon_{S_1} \sim \mathcal{N}(0, \sigma_{S_1}^2)$ with $\sigma_{S_1} \sim \mathcal{N}(0, 0.1^2)$.

Similarly, the seasonal pattern $S_{2,t}$ is defined as:

$$S_{2,t} = \sum_{i=1}^{12} \epsilon_{S_2,i} \tag{13}$$

where $\epsilon_{S_2} \sim \mathcal{N}(0, \sigma_{S_2}^2)$ with $\sigma_{S_2} \sim \mathcal{N}(0, 0.1^2)$.

The weight w_t is itself a random walk and is bounded by two parameters, $w_{\min}$ and $w_{\max}$:

$$w_t = w_{t-1} + \epsilon_t \tag{14}$$

where $\epsilon_t \sim \mathcal{N}(0, 0.15)$. The bounds are given by $w_{\min} \sim \mathcal{U}(0, 0.5)$ and $w_{\max} \sim \mathcal{U}(0.5, 1)$, ensuring w_t remains within the interval $[w_{\min}, w_{\max}]$.

Additive outliers ao_t affect only a single observation or data point due to singular events like unusual holiday timing or extreme weather conditions. The number of occurrences n_{ao} follows a uniform distribution $n_{ao} \sim \mathcal{U}(0, 10)$, and the strength of each shock is normally distributed as $\mathcal{N}(0, 1)$. Temporary changes represent

transitory shocks that weaken over time, such as the impact of major events like the World Cup or the Olympics on employment. The temporary changes tc_t is a random shock $\mathcal{N}(0,1)$ with a randomly determined duration $d_{tc,i} \sim \mathcal{U}(1,20)$ and linearly decaying effect over the duration. The number of occurrences is determined by $n_{tc} \sim \mathcal{U}(0,5)$. The permanent level shift ls_t is also a random shock $\mathcal{N}(0,1)$, applied as a shift to all values from its occurrence onward with $n_{sb} \sim \mathcal{U}(0,3)$ allowing up to 3 permanent shifts in the time series.

This configuration defines how each time series component evolves through a set of distributional properties. The generation of diverse time series structures that capture a wide range of patterns is possible. Figure 4 shows the components of the synthetic time series.

7 Performance evaluation of seasonality adjustment methods

In this section, I evaluate the performance of the tested seasonality extraction methods: the employed cGAN-LOESS, X13, and STL. For X13, I used maximal seasonal ARIMA orders of (4,1,4)(1,1,1), enabled automatic outlier detection and testing, and included internal forecasting for five periods. Since the synthetic time series does not exhibit trading day effects, the automatic trading day adjustment was deactivated. For STL, I specified a periodicity of 12, a seasonal smoother length of 7, a trend smoother length of $1.5 * periodicity/(1 - 1.5/seasonal) = 23$ (following the suggestion in the original implementation), and a low pass filter length of 13. The cGAN-LOESS model training set consists of 2000 synthetic time series, generated with stochastic properties as described in section X The cGAN model training involves $50,000$ steps and the kernel size is set to 12, matching the seasonal periodicity. For the seasonal factor smoothing in the cGAN-LOESS, the 6 nearest values are considered in the LOESS model. Additionally, two combined ensemble models are constructed by taking the average and median of the extracted seasonality components from the three base methods. Ensemble methods are generally more reliable and robust, and they form decisions analogously to (rational) humans - by seeking opinions from multiple experts. The predicted seasonalities of the ensemble models are calculated from the seasonality components from the three base methods for each t: $S_t^{\text{mean ensemble}} = \text{mean}(S_t^{\text{cGAN-LOESS}}, S_t^{\text{X13}}, S_t^{\text{STL}})$ and $S_t^{\text{median ensemble}} = \text{median}(S_t^{\text{cGAN-LOESS}}, S_t^{\text{X13}}, S_t^{\text{STL}})$

The mean squared error (MSE) is used as the metric to quantify distortion. The MSE is estimated for each method on a set of 2,000 synthetic test time series. The distribution of the methods MSE on all test time series is shown in Figure 6, and the distributions means and medians are displayed in Table 1.

The results show notable differences in mean and median MSE values across methods. The mean ensemble method has the lowest mean MSE (0.0045), followed by CGAN-LOESS and the median ensemble, while STL has the highest mean MSE (0.0093). Similarly, the median ensemble achieves the lowest median MSE (0.0040), whereas STL again has the highest median value (0.0083). Considering both mean and median, the ensemble methods (mean and median) consistently perform better than the other methods in terms of accuracy. However, while CGAN-LOESS achieves a low mean and median MSE, it exhibits substantial variability with extreme outliers, as reflected in its standard deviation of 0.0051. The ensemble methods have the lowest variability, making them more robust, while CGAN-LOESS and STL show the highest variances.

Table 1: Mean and Median MSE of Different Methods

	cGAN-LOESS	X13	STL	Mn. ensemble	Med. ensemble
Mean	0.0048	0.0053	0.0093	0.0045	0.0048
Median	0.0038	0.0046	0.0083	0.0040	0.0043
St. dev.	0.0051	0.0031	0.0051	0.0025	0.0027

The bootstrap analysis was conducted to determine if the differences in performance between methods are statistically significant, as the MSE distributions shown in Figure 5 do not provide sufficient clarity. This was achieved by generating 10,000 bootstrap samples from the 2,000 time series, estimating the mean MSE for each bootstrap sample. The resulting bootstrap distributions along with the 5% and 95% quantiles for each method are presented in Figure 6 and Table 2. The results demonstrate that the mean ensemble method has a significantly lower mean MSE than all other methods, as its 90% CI does not overlap with that of any other method. Similarly, both the cGAN-LOESS and median ensemble methods have significantly lower mean MSEs than X13 and STL.

Table 2: 5% and 95% Quantiles of Bootstrapped Mean MSE for Different Methods

	cGAN-LOESS	X13	STL	Mn. esemble	Med. ensemble
Q 5%	0.00471	0.00530	0.00931	0.00448	0.00482
Q 95%	0.00515	0.00558	0.00974	0.00471	0.00508

Figure 7 and Figure 8 show the seasonal subseries of the base and ensemble methods for a synthetic time series In general, the extracted seasonality of X13 and STL have similar patterns while the cGAN-LOESS have a different parttern.

The general level of the extracted seasonal factor matches the true seasonal factors great. However, in this example the seasonal pattern shift is not catched appropriately. The methods are applied on real world example, namely the manufacturing orders (Auftragseingänge) which exhibit strong seasonality. Figure 9 shows the methods seasonal components for manufacturing orders.

7.1 Meta learner

Monte Carlo simulations often require the researcher to define specific distributional properties and other parameters for the data-generating process. Determining these parameters introduces a level of subjectivity, which can influence the results. Meta-learning, a concept widely applied in machine learning, involves fitting models to metadata about an experiment to better understand such relationships.

In this section, I investigate the relationship between the parameters of the data-generating process and the performance of each seasonality extraction method. This is achieved using the following model:

$$MSE_i = \alpha + \mathbf{parameter}_i^\top \boldsymbol{\beta} + \epsilon_i,$$

where MSE_i is the mean squared error (MSE) of a method for time series i, $\mathbf{parameter}_i$ is the vector of parameters from the data-generating process selected by the researcher, $\boldsymbol{\beta}$ is the corresponding vector of coefficients, and ϵ_i is the error term.

The model summaries for each seasonality adjustment method are shown in Table 3. The R^2 values indicate that the performance of cGAN-LOESS depends weakly on the parameters ($R^2 = 0.07$), whereas the performance of X13 ($R^2 = 0.36$) and STL ($R^2 = 0.5$) is more strongly influenced. Among the parameters, the "trend scale" emerges as the most influential, with consistently positive coefficients across all methods. This suggests that higher noise in the trend increases errors for all methods. Notably, X13 and STL exhibit higher coefficients for "trend scale," indicating their relative inability to filter out trend-related noise compared to cGAN-LOESS. The characteristics of the short cycle, specifically "short cycle window size" and "short cycle amplitude," also play a significant role. A larger window size, corresponding to cycles with higher periodicity, reduces MSE for all methods, likely because short-period cycles are harder to distinguish from seasonality. Finally, the presence of zero seasonality reduces MSE, particularly for cGAN-LOESS, which handles non-seasonal time series more effectively than the other methods. This finding aligns with further evidence from MSE comparisons for zero seasonality time series in Figure 10.

Table 3: OLS Summary Table

Variable	cGAN-LOESS	X13	STL	Mn. ens.	Med. ens.
const	-0.0037	-0.0020	-0.0000	-0.0015	-0.0016
T drift	0.0016	-0.0017	-0.0018	-0.0018	-0.0022
T scale	0.0259***	0.0513***	0.0858***	0.0398***	0.0442***
C_l win. size	-0.0000*	-0.0000	-0.0000	-0.0000	-0.0000
C_l amp	0.0004***	0.0002***	0.0001	0.0002***	0.0002***
C_s win. size	-0.0000**	-0.0001***	-0.0002***	-0.0001***	-0.0001***
C_s amp	0.0005***	0.0003***	0.0007***	0.0004***	0.0003***
S_1 scale	0.0031**	0.0000	0.0011	0.0005	0.0005
S_2 scale	0.0037**	0.0008	0.0005	0.0010	0.0009
S_1 ar	0.0031**	0.0000	0.0011	0.0005	0.0005
S_2 ar	0.0037**	0.0008	0.0005	0.0010	0.0009
AO	-0.0001	0.0000	0.0002	0.0001	0.0001
TC	0.0001	0.0001	0.0003	0.0001	0.0001
LS	-0.0000	0.0000	0.0004*	0.0000	0.0000
Zero seasonality	-0.0031***	-0.0008*	-0.0002	-0.0011***	-0.0010***
R-squared	0.0724	0.3616	0.4896	0.3617	0.3664
N	1259	1259	1259	1259	1259

8 Discussion and Conclusion

The cGAN-LOESS model requires specifying only three parameters: the HP high-pass filter's smoothing parameter, the LOESS smoothing parameter, and the seasonality periodicity. Once trained, the model is straightforward to use, and these parameters are easy to adjust. However, model performance heavily depends on the data-generating process chosen during training, which introduces significant researcher freedom due to the need to set distributional parameters.

In our analysis, the cGAN-LOESS model was trained on fixed time series of 256 observations. Future work should explore its performance with varying time series lengths. Unlike STL and X13, which are filter procedures and directly extract seasonality, cGAN-LOESS learns the underlying distribution of a data-generating process to predict seasonality. In tests on synthetic time series, cGAN-LOESS outperformed STL and X13, with ensemble methods further improving performance.

References

Baxter, M. and King, R. G. (1999). Measuring Business Cycles: Approximate Band-Pass Filters for Economic Time Series. *The Review of Economics and Statistics*, 81(4):575–593.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on LOESS. *Journal of Official Statistics*, 6:3–73.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

Dagum, E. B. (1980). *The X-11-ARIMA Seasonal Adjustment Method*. Number No. 12-564E. Statistics Canada, Ottawa.

Findley, D. F., Monsell, B. C., Bell, W. R., Otto, M. C., and Chen, B. (1998). New capabilities and methods of the x-12-arima seasonal-adjustment program. *Journal of Business & Economic Statistics*, 16(2):127–52.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial nets. In *Neural Information Processing Systems*.

Gómez, V. and Maravall, A. (2001). Seasonal adjustment and signal extraction in economic time series. In Peña, D., Tiao, G. C., and Tsay, R. S., editors, *A Course in Time Series Analysis*, chapter 8. John Wiley & Sons, New York.

Hamilton, J. D. (2017). Why you should never use the hodrick-prescott filter. Working Paper 23429, National Bureau of Economic Research, Cambridge, MA.

Hodrick, R. J. and Prescott, E. C. (1997). Postwar u.s. business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1):1–16.

Isola, P., Zhu, J., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004.

King, R. G. and Rebelo, S. T. (1993). Low frequency filtering and real business cycles. *Journal of Economic Dynamics and Control*, 17(1):207–231.

Kuznets, S. (1929). Random events and cyclical oscillations. *Journal of the American Statistical Association*, 24:258–275.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Pedersen, T. M. (2001). The hodrick–prescott filter, the slutzky effect, and the distortionary effect of filters. *Journal of Economic Dynamics and Control*, 25(8):1081–1101.

Shiskin, J. (1978). Seasonal adjustment of sensitive indicators. In Zellner, A., editor, *Seasonal Analysis of Economic Time Series*, pages 97–103. U.S. Department of Commerce, Bureau of the Census, Washington, DC.

Slutzky, E. (1937). The summation of random causes as the source of cyclic processes. *Econometrica*, 5(2):105–146.

Spanos, A. (1990). Towards a unifying methodological framework for econometric modelling. In Granger, C., editor, *Modelling Economic Series*. Clarendon Press, Oxford.

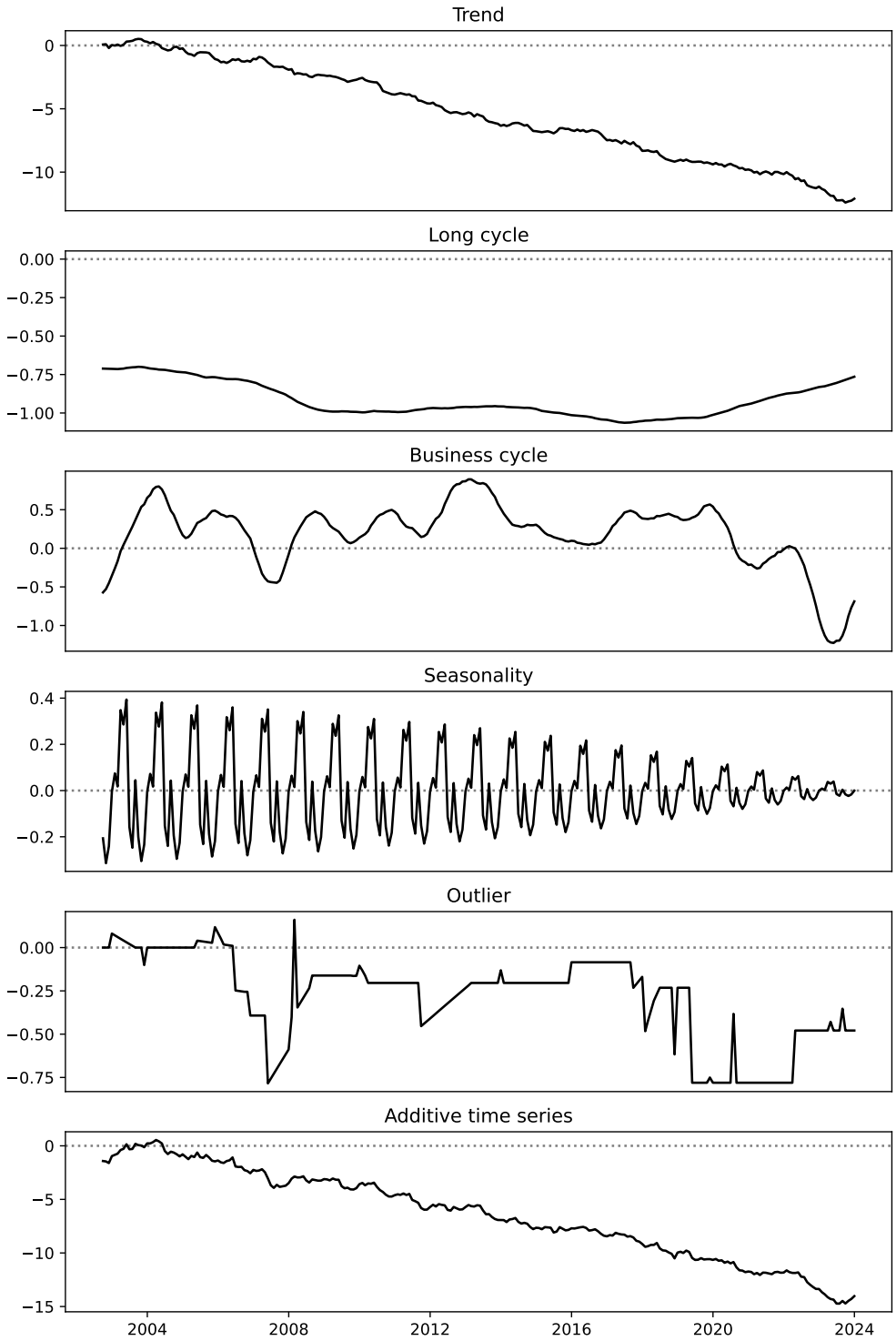

Figure 4: Synthetic time series components

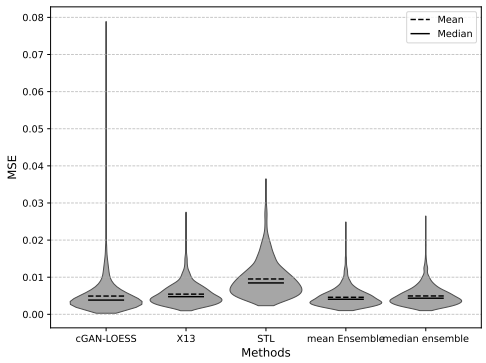

Figure 5: Violin plots of MSE for different methods

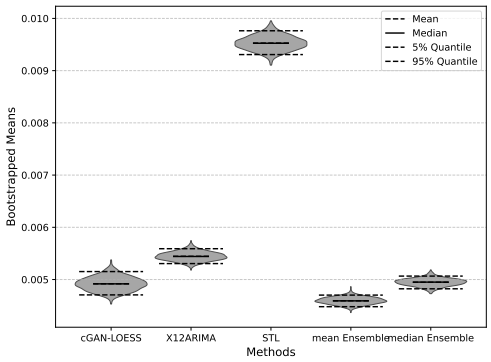

Figure 6: Violin plots of bootstrapped MSE means for different methods

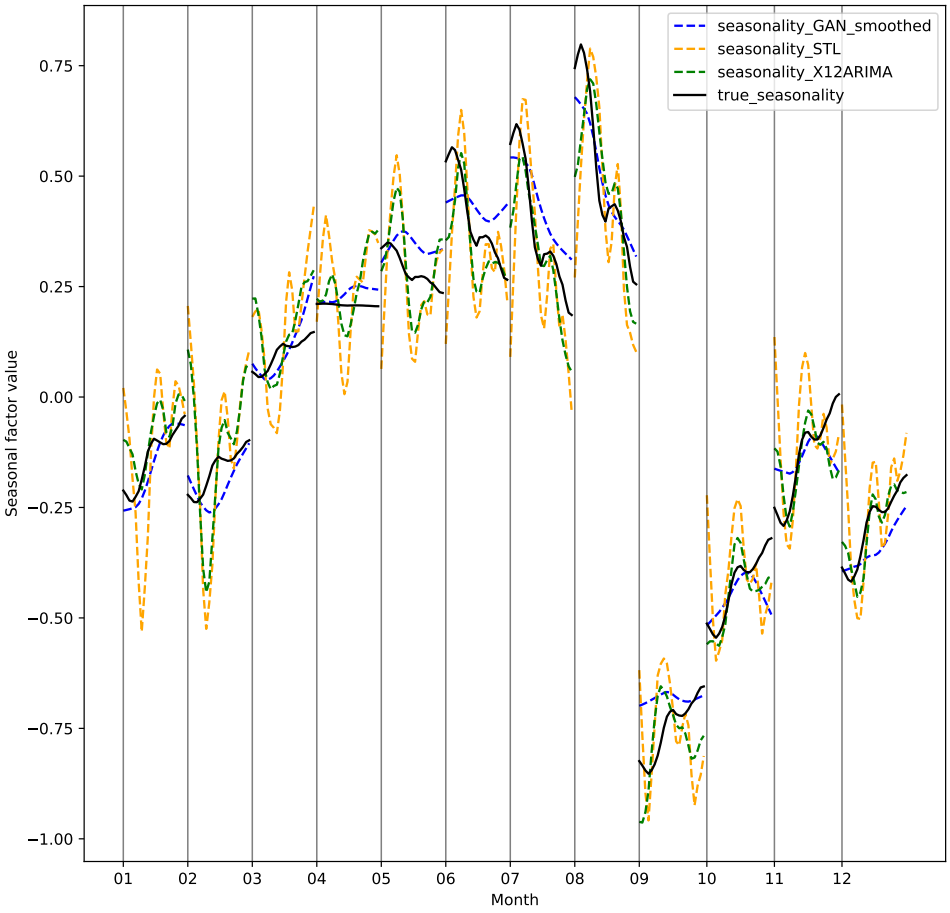

Figure 7: Seasonal factor plot

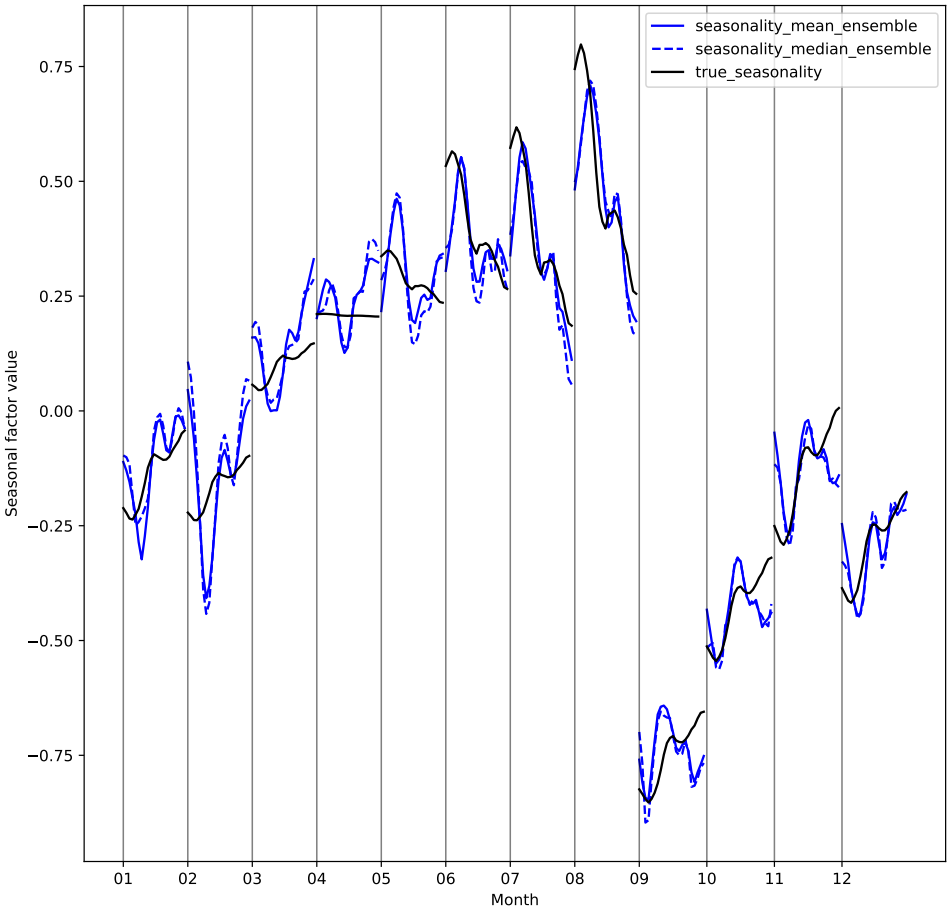

Figure 8: Seasonal factor plot

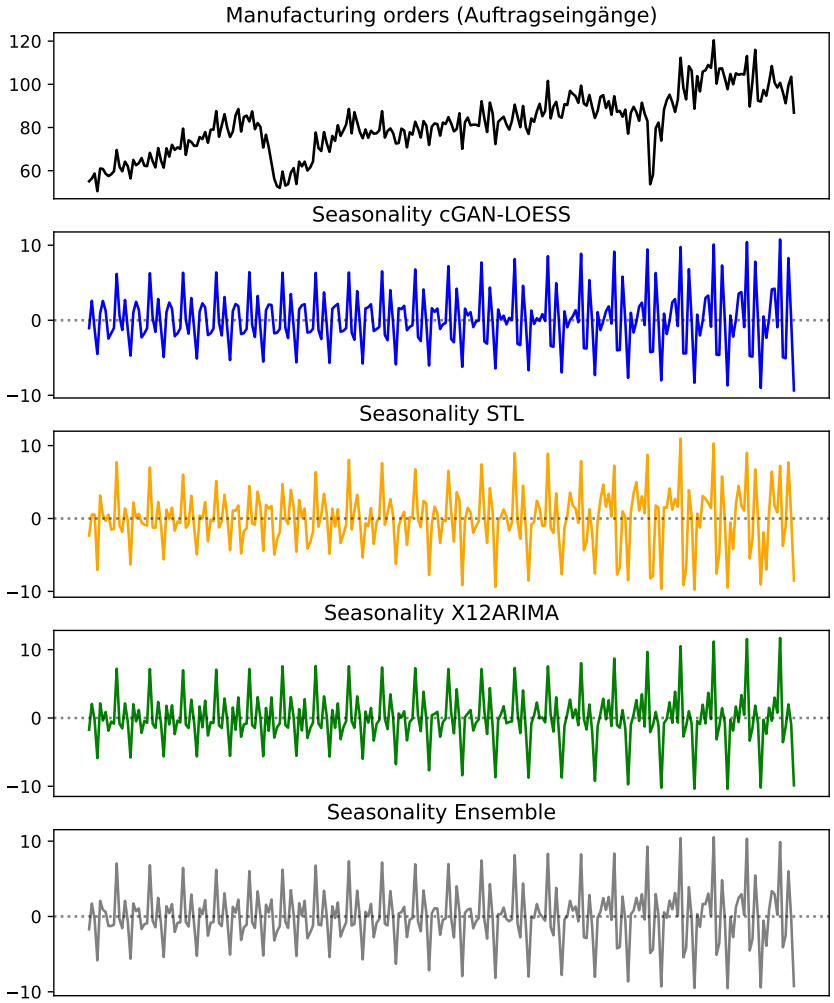

Figure 9: Time series seasonal component real

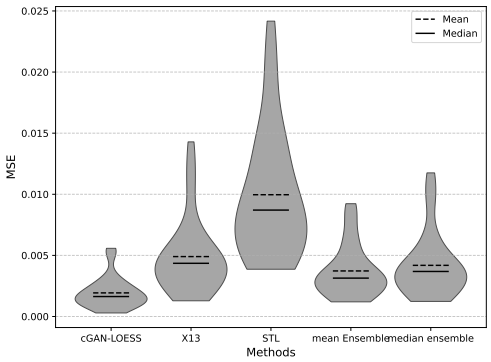

Figure 10: Violin plots of MSE for time series with zero seasonality