

Automatic locally stationary time series forecasting with application to predicting UK gross value added time series

Rebecca Killick¹ , Marina I. Knight² , Guy P. Nason³ ,
Matthew A. Nunes⁴  and Idris A. Eckley¹ 

¹School of Mathematical Sciences, Lancaster University, Lancaster, UK

²Department of Mathematics, University of York, York, UK

³Department of Mathematics, Imperial College London, London, UK

⁴Department of Mathematical Sciences, University of Bath, Bath, UK

Address for correspondence: Matthew A. Nunes, Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, UK. Email: m.a.nunes@bath.ac.uk

Abstract

Accurate forecasting of the UK gross value added (GVA) is fundamental for measuring the growth of the UK economy. A common nonstationarity in GVA data, such as the ABML series, is its increase in variance over time due to inflation. Transformed or inflation-adjusted series can still be challenging for classical stationarity-assuming forecasters. We adopt a different approach that works directly with the GVA series by advancing recent forecasting methods for locally stationary time series. Our approach results in more accurate and reliable forecasts, and continues to work well even when the ABML series becomes highly variable during the COVID pandemic.

Keywords: classical forecasting, COVID shocks, local partial autocorrelation, spectral estimation, wavelets

1 Introduction

The literature on forecasting stationary time series has been established for many years. See, for example, [Gardner \(1985\)](#) or [Box and Jenkins \(1970\)](#) with easily implemented code readily available on a variety of platforms. Rather surprisingly, the same cannot be so readily said when it comes to forecasting of nonstationary time series. Indeed, it is not uncommon for analysts to forecast time series assuming, but not testing for, second-order stationarity. Yet, as [Janeway \(2009\)](#) describes, there can be grave consequences for ignoring this nonstationary structure. This article seeks to address this by providing practical, implemented forecasting methods for nonstationary series.

Our work is motivated by a problem arising from the accurate forecasting of economic time series. Specifically we consider the ABML time series, which contains values of the UK gross value added (GVA), a major component of the UK gross domestic product (GDP). Both are vitally important economic statistics, with accurate forecasts being crucial in measuring the size of and growth in the UK economy. Our ABML series is recorded quarterly from quarter one 1955 until quarter four 2020, consists of $T = 264$ observations and is plotted in [Figure 1](#). The impact of the ‘great financial crisis’ of 2008 and the COVID pandemic during 2020 can be clearly seen. The data can be acquired from the Office for National Statistics website <https://www.ons.gov.uk>.

Received: April 26, 2023. Revised: July 9, 2024. Accepted: July 30, 2024

© The Royal Statistical Society 2024.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

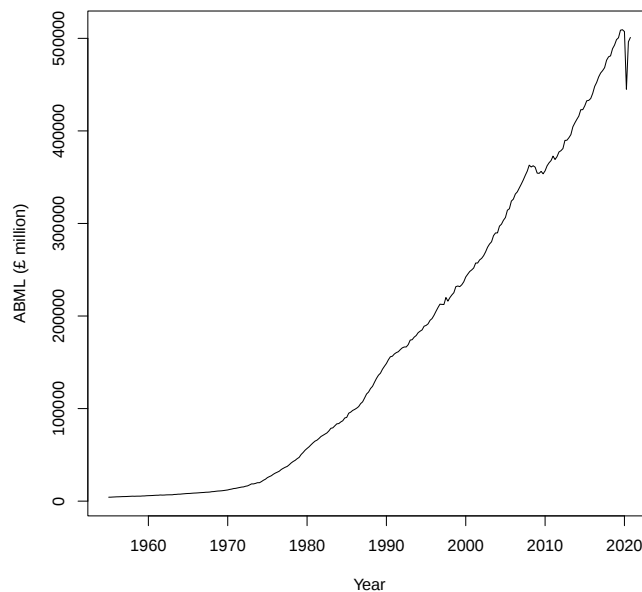


Figure 1. The ABML time series.

As with many economic time series, ABML exhibits a clear polynomial-like trend, which is characteristic of an integrated economic time series. Using standard statistical time series procedure, e.g. [Chatfield \(2003\)](#), we remove the trend using second-order differences. The second differences of our ABML series, including and not including the COVID period (up to Q4 2019), are shown in [Figure 2](#), the latter amply demonstrating the dramatic impact of the COVID pandemic.

Both figures strongly suggest that the series is not second-order stationary, in that the variance of the series increases markedly over time. Methods from [Nason \(2013a\)](#) show that the correlation also changes over time. Much of the increase in variance observed in [Figure 1](#) is probably due to inflation. However, we also analysed two different inflation-corrected versions of ABML, one provided by the UK Office of National Statistics, and both of these reject the null hypothesis of second-order stationarity, as determined by tests of stationarity in [Priestley and Subba Rao \(1969\)](#) and [Nason \(2013a\)](#). Consequently, given that our series is nonstationary and shows no evidence of nonlinear behaviour, to attempt forecasting we ought to use methods designed for such series.

The fundamentals of nonstationary forecasting have been considered by several authors, but mostly from a theoretical standpoint. See, for example, [Whittle \(1963\)](#), [Abdrabbo and Priestley \(1967\)](#), [Subba Rao \(1973\)](#), and [Hallin \(1986\)](#). [Dahlhaus \(1996\)](#) uses a version of Kolmogorov's formula ([Brockwell & Davis, 1991](#), Theorem 5.8.1) for zero-mean locally stationary time series, paralleling the [Subba Rao \(1973\)](#) result for the oscillatory process model. Several other methods for nonstationary series forecasting exist, such as the neural network method of [Chow and Leung \(1996\)](#) and [Mercurio and Spokoiny \(2004\)](#) or the simple, but effective, time-varying unconditional variance model of [Van Bellegem and von Sachs \(2004\)](#). Recent developments also include the work of [Kley et al. \(2019\)](#), who use local autoregressive approximations to forecast nonstationary series; [Ding and Zhou \(2023\)](#), who provide a theoretical contribution using a similar autoregressive approximation for series exhibiting both nonstationary and nonlinear characteristics; and finally, [Wu and Dette \(2022\)](#) who propose local kernel-based forecasting in the case where data is nonstationary and non-centred.

[Fryzlewicz et al. \(2003\)](#) derived time-varying Yule–Walker equations for the locally stationary wavelet (LSW) model of [Nason et al. \(2000\)](#), introducing local Yule–Walker estimators and solvers. The authors carefully established the theoretical properties of their estimators and proposed a practical forecasting methodology, which included parameter selection advice. However, as a grid search based approach, it can be computationally intensive to implement in practice.

Our aim in this article is to develop a new, readily implemented, automated approach to forecasting LSW time series. Here, we assume future observations can be modelled in terms of present

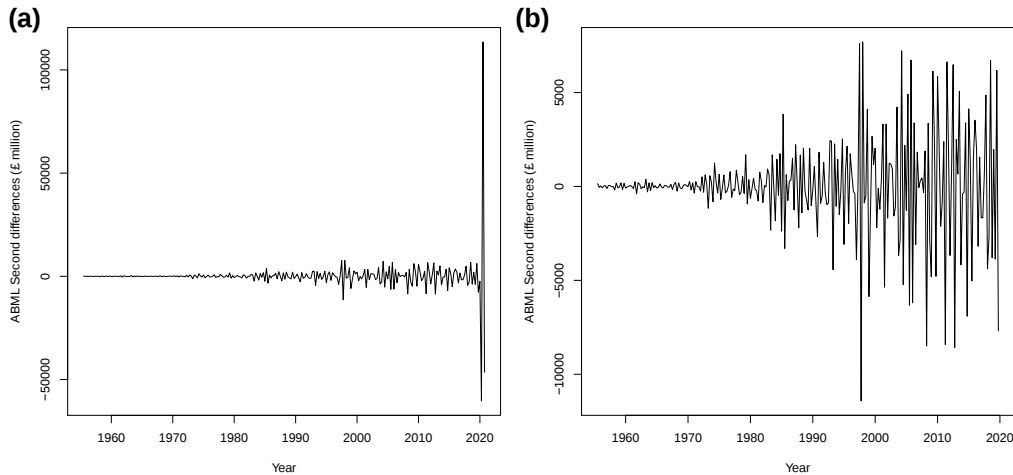


Figure 2. The second differences of the ABML time series (a) with and (b) without the COVID pandemic period.

and recent past values and adopt the stance of [Fryzlewicz et al. \(2003\)](#), thus restricting our attention to predictors that are linear functions of the data and seeking to minimize their associated mean square prediction error. We use the recently proposed local partial autocorrelation ([Killick et al., 2020](#)) to dynamically select the number of past time series observations to use in forecasting (denoted by p below), rather than relying on the grid-type search proposed by [Fryzlewicz et al. \(2003\)](#). Our approach is both computationally simpler, has minimal tuning parameters, and provides much improved forecasting results, which we demonstrate on both simulated and real data.

Section 2 details the LSW forecasting framework from [Fryzlewicz et al. \(2003\)](#). Section 3 introduces our improvement, uses of the local partial autocorrelation function to estimate the local structure required for forecasting. Section 4 demonstrates this improved performance on a range of simulated stationary and nonstationary time series. Such a simulation study is required as although we strongly suspect our ABML gross value added series is nonstationary, we cannot be absolutely sure. Hence, we need at least some validation of our new methodology on both stationary and nonstationary series. Section 5 provides our forecasting results on the ABML economic data series. Section 6 provides further discussion and concludes.

2 Review of locally stationary wavelet time series forecasting

We begin by reviewing the locally stationary wavelet (LSW) model ([Nason et al., 2000](#)) and its associated forecasting framework ([Fryzlewicz et al., 2003](#)). The LSW approach for modelling non-stationary time series has been used in many fields, from climatology ([Fryzlewicz, 2003](#)) and ocean engineering ([Killick et al., 2013](#)) to biology ([Hargreaves et al., 2019](#)), medicine ([Embleton et al., 2022a; Nason & Stevens, 2015](#)) and finance ([Fryzlewicz, 2005](#)).

2.1 The locally stationary wavelet model

The LSW model ([Fryzlewicz, 2003; Nason et al., 2000](#)) encompasses sequences of doubly indexed stochastic processes $\{X_{t,T}\}_{t=0,\dots,T-1}$, $T = 2^J \geq 1$, having the following representation in the mean-square sense

$$X_{t,T} = \sum_{j=1}^J \sum_k w_{j,k;T} \psi_{j,k}(t) \zeta_{j,k}, \quad (1)$$

where $\zeta_{j,k}$ is a random orthonormal increment sequence and the $\{\psi_{j,k}(t)\}_{j,k}$ form a discrete non-decimated family of wavelets based on a mother wavelet, $\psi(t)$, of compact support. The quantities

in (1) are assumed to observe a number of key properties. Most notably $\mathbb{E}(\xi_{j,k}) = 0$ for all j, k , and hence $E(X_{t,T}) = 0$ for all t, T , together with an assumption that $\text{Cov}(\xi_{j,k}, \xi_{\ell,m}) = \delta_{j,\ell} \delta_{k,m}$, where $\delta_{i,j}$ is the Kronecker delta. In addition, Nason et al. (2000) also introduces a number of conditions to ensure the amplitudes $\{w_{j,k;T}\}_k$ vary slowly within each level, thereby controlling the degree of local stationarity of the process. Relevant to our setting here, amplitude-based assumptions are subsequently revisited by Fryzlewicz and Nason (2006) in order to allow for jumps, an assumption also useful in the LSW changepoint literature (Killick et al., 2013; Nam et al., 2015).

Within the LSW framework the evolutionary wavelet spectrum (EWS), defined as $S_j(z) = |W_j(z)|^2$, at each scale $j = 1, \dots, J$ and rescaled time $z = k/T \in (0, 1)$, plays an analogous role to that of spectrum in the stationary time series setting. The EWS quantifies the process power distribution over time and scale, and is connected to a localized autocovariance function defined for each (rescaled) time z and lag $\tau \in \mathbb{Z}$ as follows

$$c(z, \tau) = \sum_{j=1}^{\infty} S_j(z) \Psi_j(\tau). \quad (2)$$

The $\{\Psi_j(\tau)\}_j$ is a family of compactly supported autocorrelation wavelets, see Nason et al. (2000).

Spectral estimation is usually carried out by means of the raw wavelet periodogram, defined as $I_{j,k;T} = |d_{j,k;T}|^2$, where $d_{j,k;T} = \sum_{t=0}^T X_{t,T} \Psi_{j,k}(t)$ are the empirical nondecimated wavelet coefficients. For notational simplicity, we shall refer to the raw periodogram as $I_{j,k}$.

An asymptotically unbiased estimator of the EWS is provided by the (corrected) empirical wavelet spectrum

$$\hat{\mathbf{L}}(z) = A^{-1} \mathbf{I}(z), \quad (3)$$

for all $z \in (0, 1)$, where $\mathbf{I}(z) := (I_{j,[zT]})_{j=1}^J$ is the raw wavelet periodogram vector and A is a $J \times J$ symmetric matrix with entries $A_{j,\ell} = \sum_{\tau} \Psi_j(\tau) \Psi_{\ell}(\tau)$. As in the stationary setting, the wavelet periodogram is not a consistent estimator of the wavelet spectrum (Nason et al., 2000). One way to overcome this is to smooth the raw wavelet periodogram as a function of (rescaled) time within each scale j , and then to apply the correction by A^{-1} as in (3).

Once a well-behaved spectral estimator, $\hat{\mathbf{L}}$, has been obtained, equation (2) can be used to obtain a local autocovariance estimator $\hat{c}(z, \tau)$, e.g. in the notation above $\hat{c}(z, \tau) = \sum_{j=1}^J \hat{\mathbf{L}}_j(z) \Psi_j(\tau)$.

The LSW framework has proved to be useful across a variety of tasks when compared to competitor methods, e.g. for dimension reduction (Knight et al., 2024), classification (Fryzlewicz & Ombao, 2009; Krzemieniewska et al., 2014), clustering (Hargreaves et al., 2018), testing for stationarity (Nason, 2013a), spectral equality (Hargreaves et al., 2019) and replicate-effect (Embleton et al., 2022b), changepoint detection (Nam et al., 2015) and testing for white noise and aliasing (Eckley & Nason, 2018). We focus here on forecasting nonstationary time series, and in particular on the work of Fryzlewicz et al. (2003).

2.2 Forecasting within the LSW framework

2.2.1 Existing work

Given observations $X_{0,T}, \dots, X_{t-1,T}$ of a zero-mean locally stationary wavelet process, the method of Fryzlewicz et al. (2003) (FVBvS algorithm henceforth) proposes to predict the next observation $X_{t,T}$ by taking a linear combination of the most recent p observations

$$\hat{X}_{t,T} = \sum_{s=t-p}^{t-1} b_{t-1-s,T} X_{s,T}. \quad (4)$$

Here, the predictor coefficients $\mathbf{b} = (b_{0,T}, \dots, b_{p-1,T})$ are chosen to minimize the mean square prediction error (MSPE). This is similar to the typical forecasting approach in the stationary context, except the weights $\mathbf{b}$ depend on time. Intuitively, the reason for not using the whole series when

predicting the next observation lies in the nonstationary character of the process: the beginning of the series might have a different structure to the end and hence may not be useful for forecasting.

Fryzlewicz et al. (2003) showed that $\text{MSPE}(\hat{X}_{t,T}, X_{t,T}) = \mathbb{E}(\hat{X}_{t,T} - X_{t,T})^2$ can be approximated by $\mathbf{b}^T B_t \mathbf{b}$, where B_t is a $(p+1) \times (p+1)$ matrix whose (m, n) th entry is given by $(B_t)_{m,n} = \sum_{j=1}^J S_j(\frac{m+n}{2T}) \Psi_j(m-n) = c(\frac{m+n}{2T}, m-n)$. The weights $\mathbf{b}$ can then be obtained by solving the ‘generalized’ set of Yule–Walker equations

$$\sum_{s=t-p}^{t-1} b_{t-1-s,T} c\left(\frac{n+s}{2T}, s-n\right) = c\left(\frac{n+t}{2T}, t-n\right), \quad \forall n = t-p, \dots, t-1, \quad (5)$$

which can then be extended to h -steps-ahead prediction. These weights may be estimated by plugging in a good estimator of the local autocovariance function, with the forecast quality obviously highly reliant on the quality of these estimators.

Contrasting the usual approach in LSW estimation discussed in Section 2.1, the FVBvS method proposes to obtain an estimate of the wavelet spectrum $\{\hat{S}_j(k/T)\}_j$ at all rescaled times corresponding to observed times (up to $(t-1)$) without smoothing and then to obtain an estimated local autocovariance $\tilde{c}(k/T, \tau)$ at each time $k = t-p, \dots, t-1$ and lags τ . For consistency, Fryzlewicz et al. (2003) smooth the estimated local autocovariance by means of a standard kernel smoothing method (using a normal or box kernel) with a chosen bandwidth that the authors denote by g . As well as providing a consistent estimator for the observed times, smoothing the local autocovariance additionally allows the forward estimation of the local autocovariance at rescaled time t/T . This (smoothed) estimated local autocovariance can then be used in the generalized Yule–Walker equations (5) and the weights $\hat{\mathbf{b}}$ derived. The forecast $\hat{X}_{t,T} = \sum_{s=t-p}^{t-1} \hat{b}_{t-1-s,T} X_{s,T}$ is then obtained, as well as its associated prediction error. Fryzlewicz et al. (2003) generalized the above one-step-ahead prediction to an h -steps ahead.

2.2.2 Criticisms of the FVBvS method

Whilst theoretically tractable, Xie et al. (2009) noticed that the FVBvS algorithm sometimes produces abnormally large forecasts. They identified the cause to be the occasional near-singularity of local covariance matrices and proposed a modification of the algorithm in order to stabilize the forecasts. Their proposal constrains the Yule–Walker solution vector $\mathbf{b}$ to have unit norm. The authors note that the revised method consistently produces better forecasts for a variety of prediction horizons.

A key quantity in calculating the forecast in equation (4) is p : the amount of recent data used for prediction. Fryzlewicz et al. (2003) and Xie et al. (2009) used an adaptive grid search method to select p , which requires a starting value p_0 . In their approach, from one time point to the next, p can only increase or decrease by one thus sometimes resulting in a slow adaption to the dynamics of the evolving series.

FVBvS suggested a procedure, also adopted by Xie et al. (2009), for simultaneously selecting the parameters p (the number of most recent observations from the past used to inform the forecast $\hat{X}_{t,T}$) and g (the bandwidth of the smoothing kernel). Briefly, their algorithm starts with some initial values for parameters p and g , and the pair (p, g) then are updated in an iterative process that evaluates their corresponding prediction performance for known data. The underlying idea is that the (p, g) pair gets trained over a segment of length m at the end of the series. Fryzlewicz et al. (2003) proposed to choose m to be the length of the longest segment at the end of the observed series with an apparent stationary behaviour, judged by *visual inspection*. Having made a choice for the parameter m , the FVBvS algorithm can be summarized as follows

- Make an initial choice of parameters, say (p_0, g_0) , and use it to obtain predicted $\hat{X}_{t-m,T}$ by means of equation (4) using the previous p_0 process observations.
- Predict $X_{t-m,T}$ also by using pairs of parameters around (p_0, g_0) , i.e. $(p_0 \pm 1, g_0 \pm \delta)$ for some fixed constant δ .
- The pair that gives ‘closest’ forecasts (in the sense of the minimum relative absolute prediction error) to the observed process is chosen, (p_1, g_1) , say.

- (d) Repeat steps (a)–(c). The updated pair of parameters is then itself updated through predicting $X_{t-m+1,T}$ from the previous p_1 observations and, by re-iterating this process, a parameter pair (p, g) is obtained for predicting the desired $X_{t,T}$.

Hence, despite this ‘automatic’ tuning, the practitioner must decide on the initial parameter pair (p_0, g_0) , the training length m and the smoothing kernel (normal or box) used in the FVBvS method.

3 Automating the locally stationary wavelet forecast

Our proposal departs from the currently adopted practice by automating the forecasting procedure. Instead of the smoothing and forward estimation being undertaken at the level of the local autocovariance function, we take advantage of the recent advances in local spectral estimation (Nason, 2013a) and propose to perform running mean *periodogram* smoothing, from which forward estimation followed by correction are straightforward. This avenue affords local autocovariance estimation through equation (2) and yields estimators with desirable properties (Nason, 2013a). In turn, this approach also removes the need for the initial bandwidth choice (g_0) and training process since our proposed periodogram smoothing contains an automatic bandwidth selection. In addition, we propose to remove the choice of p_0 and segment length m by adopting a *localized* estimator for p in order to determine the (time-dependent) forecasting window. Specifically, we use the local partial autocorrelation function (lpacf) proposed by Killick et al. (2020), as a measure of the localized conditional correlation structure. Details of this approach are provided below and throughout the article we shall refer to the proposed method for FORecasting Localized Autocorrelation Processes under the acronym FORLAP.

3.1 Proposed lpacf-based forecasting

Intuitively, the choice of p amounts to establishing the length of a (sub)interval over which the process displays stationary behaviour, so that data over this interval can feasibly contribute to the linear prediction. This is evocative of the stationary autoregressive setting where the unknown dependence order p is chosen via the partial autocorrelation function, q , using the theoretical property $p = \min \{\tau : q(\tau) = 0\} - 1$ (Tsay, 2002, p. 36). Of course, in practice, estimates $\hat{q}(\tau)$ are computed and, as these will never be exactly zero, their associated confidence intervals can be used to obtain an estimator $\hat{p}$ (see e.g. Theorem 8.1.2 from Brockwell & Davis, 1991). This method of selecting the number of past observations to feed into prediction is widely used, even when the assumed underlying process is not autoregressive. In contrast, on a backdrop of model choice, Kley et al. (2017) propose to choose the order of time-varying autoregressive processes by means of the MSPE.

Our proposal is to use a similar approach in which a *localized version of p* (and its estimate) is used to inform the length of the forecasting window. In our approach, we adopt the recently proposed local partial autocorrelation function (lpacf) of Killick et al. (2020), as well as a corresponding estimator. We next define their lpacf, $q(z, \tau)$ at rescaled time z and lag τ , before demonstrating how we can apply it to nonstationary forecasting.

Definition 3.1 Let $\{X_{t,T}\}$ be a zero-mean locally stationary wavelet process with local autocovariance $c(z, \tau)$ and spectrum $\{S_j(z)\}_j$ that satisfy

$$\sum_{\tau=0}^{\infty} \sup_z |c(z, \tau)| < \infty, \quad C_1 := \text{ess inf}_{z, \omega} \sum_{j>0} S_j(z) |\hat{\psi}_j(\omega)|^2 > 0,$$

where ess inf is the essential infimum and $\hat{\psi}_j(\omega) = \sum_s \psi_{j,0}(s) \exp(i\omega s)$. Then, the local partial autocorrelation function (lpacf) at (rescaled) time z and lag τ is given by:

$$q(z, \tau) = \varphi_{[zT], \tau, \tau} \left\{ \frac{\text{MSPE}(\hat{X}_{[zT], T}^{(b)}, X_{[zT], T})}{\text{MSPE}(\hat{X}_{[zT] + \tau, T}^{(f)}, X_{[zT] + \tau, T})} \right\}^{1/2}, \quad (6)$$

where the quantity $\varphi_{[zT],\tau}$ is the last element in the vector $\boldsymbol{\varphi}_{[zT],\tau}$ (of length τ) obtained as the solution to the local Yule–Walker equations i.e. $\mathbf{B}_{[zT]}\boldsymbol{\varphi}_{[zT],\tau} = \mathbf{r}_{[zT]}$. Furthermore, the estimates $\hat{X}_{[zT],T}^{(b)}$ and $\hat{X}_{[zT],T}^{(f)}$ are defined as the forecasting and backcasting predictions; $\hat{X}_{[zT],T}^{(b)} = \sum_{p=1}^{\tau-1} b_{p,T}^{(b)} X_{[zT]+p,T}$ and $\hat{X}_{[zT],T}^{(f)} = \sum_{p=1}^{\tau-1} b_{\tau-1-p,T}^{(f)} X_{[zT]+p,T}$.

The right hand term under the square root in (6) is a measure of nonstationarity. For a stationary process, the square root term equals one and the localized $q(z, \tau)$ coincides with the classical partial autocorrelation measure $q(\tau)$. If there is a degree of nonstationarity within the data, then $q(z, \tau)$ will be modified by the nonstationarity factor.

Killick et al. (2020) propose two methods for estimating the lpacf. For the purposes of forecasting we prefer to use the second, more stable windowed estimator, denoted $\tilde{q}_W(z, \tau)$, which has been shown to have an asymptotically normal distribution and practically to work well both in simulated and real data settings (Killick et al., 2020). Crucially, this windowed estimator allows for the *local* estimation of the partial autocorrelation at the last observation in the process ($t - 1$), as this is the point around which the prediction is made. Corollary 1 from Killick et al. (2020) then allows us to construct (pointwise) confidence bounds for the local partial autocorrelation function. Recall that choosing p in the stationary setting is akin to estimating the number of significant lags in the partial autocorrelation function. We mimic this idea and obtain the estimate $\hat{p}$ as the smallest value τ for which the pointwise confidence interval of $\tilde{q}_W(z, \tau)$ at rescaled time $z = (t - 1)/T$ contains zero, minus 1. However, we stress that we do not necessarily assume that the underlying process is necessarily autoregressive of any order. Future work could investigate whether $\hat{p}$ is a consistent estimator of the true value of p if the underlying process was indeed locally autoregressive.

Algorithm 1 details the steps of our proposed forecasting algorithm, FORLAP. Whilst our theoretical framework is based on Fryzlewicz et al. (2003), our practical implementation differs considerably. Specifically, in addition to (i) using $\hat{p}$ as an appropriate value of p , we also (ii) use the recent locally stationary wavelet covariance from Nason (2013a) and bandwidth estimation as implemented in the `AutoBestBW` function in the `locits` R package (Nason, 2013b), and (iii) permit the covariance matrix regularization method of Xie et al. (2009) as an option.

Algorithm 1 Proposed FORLAP algorithm for nonstationary time series forecasting.

lpacf-based forecasting algorithm (FORLAP)

Assume we observed $\{X_{0,T}, \dots, X_{t-1,T}\}$ with $T = 2^J$.

- (a) *Determine p via lpacf estimation*: obtain the lpacf estimate $\tilde{q}_W$ (Killick et al., 2023b) corresponding to time $(t - 1)$ and set $\hat{p}$ to be the smallest non-significant lag in its pointwise confidence interval minus 1.
 - (b) *Spectral estimation*: estimate the spectral content of the observed signal by correcting (with the matrix A^{-1} , see equation (3)) the running mean smoothed raw periodogram $\tilde{I}_{j,k} = (2s + 1)^{-1} \sum_{u=k-s}^{k+s} I_{j,u}$, where the bandwidth s is obtained automatically as described in Nason (2013a, 2013b). Note that this procedure also embeds forward smoothing of the raw periodogram and thus enables the estimation of $\hat{S}_j(k/T)$ for $k = 0, \dots, t$.
 - (c) *Autocovariance estimation*: Estimate the local autocovariance $c(z, \tau)$ by means of equation (2) at rescaled times z corresponding to observed times up to $(t - 1)$ and lags τ as dictated by the generalized Yule–Walker equation (5). Also obtain $\hat{c}(t/T, \tau)$ by making use of the extrapolated spectrum $\hat{S}_j(k/T)$.
 - (d) *Solve the generalized Yule–Walker equations*: obtain the estimated time-dependent weight vectors $\hat{\mathbf{b}}$ by solving equations (5) over the most recent $\hat{p}$ observations, subject to the regularization constraint of Xie et al. (2009) if desired.
 - (e) *Forecast $X_{t,T}$* : by using the linear combination of the last $\hat{p}$ observations with weights $\hat{\mathbf{b}}$, as described in equation (4). Following Fryzlewicz et al. (2003) (equation (4.6)), the $(1 - \alpha)$ prediction interval uses the corresponding estimated MSPE: $\hat{X}_{t,T} \pm \Phi^{-1}(1 - \alpha/2) \left(\hat{\mathbf{b}}^T \hat{\mathbf{B}}_t \hat{\mathbf{b}} \right)^{1/2}$.
-

4 Simulation study

We assess the performance of FORLAP by simulating from a variety of scenarios, including both stationary and nonstationary examples, as detailed below (see also [Appendix A](#) for their mathematical definition). Our overarching goal is to demonstrate the versatility of our proposed FORLAP forecasting technique and its utility in the ‘toolkit’ of any data analyst interested in forecasting. Our test set includes locally stationary wavelet processes as well as processes not represented by this framework. In so doing, we aim to assess the forecast performance of our approach both on realizations of wavelet processes but also a variety of other important model classes.

Below, we compare FORLAP to (i) the [Fryzlewicz et al. \(2003\)](#) method (FVBvS), (ii) forecasting using time-varying autoregressive processes (TVAR) of order 2 and higher, (iii) the Box–Jenkins forecasting procedure (B–J), designed for stationary time series but commonly used by analysts even on nonstationary data (see e.g. the report from [U.K. Centre for the Measurement of Government Activity, 2008](#)), as well as (iv) the often employed exponential smoothing (ES) ([Hyndman et al., 2008](#)).

The simulations, and real data example in later sections, use the `forecast` package ([Hyndman & Khandakar, 2008](#)), `tvReg` package ([Casas & Fernandez-Casal, 2019](#)), and the `smooth` package in the R statistical programming language ([R Core Team, 2022](#)). Specifically, we use the `auto.arima` function in the former package that automatically chooses and fits an ARIMA model and in addition its forecast function for forecasting, while time-varying autoregressive forecasting is carried out using the `forecast` function in the latter package. Our FORLAP algorithm is implemented within the `forecastLSW` package ([Killick et al., 2023a](#)), also in R. Forecasts are produced using the method described in Section 3 and implemented in the `forecast.lpacf` function, which makes use of the `lpacf` function estimate $\hat{q}_w$ implemented in the `lpacf` R package ([Killick et al., 2023b](#)).

In the simulations, we consider one-step-ahead forecasts over the stretch of last 20 observations, $h = 1$, although other horizons can be used. In addition, the methods and associated software implementation work for arbitrary length time series even though some example series below happen to be of dyadic length.

Our study presents empirically computed measures for the prediction interval coverage rates and accuracy for all forecasting methods. Nominal rates ranging from 40% to 90% in steps of 10% are used. The accuracy measure we adopt is the interval score of [Gneiting and Raftery \(2007\)](#) which simultaneously penalizes wide prediction intervals and lack of coverage. In this setup, lower interval scores correspond to better prediction intervals. More specifically, [Gneiting and Raftery \(2007\)](#) define the interval score for a $(1 - \alpha)\%$ prediction interval (l, u) and observed value x to be

$$S_{\alpha}^{\text{int}}(l, u; x) = (u - l) + 2\alpha(l - x)\mathbf{1}(x < l) + 2\alpha(x - u)\mathbf{1}(x > u), \quad (7)$$

where $\mathbf{1}$ is the standard indicator function.

The results in [Tables 1, 2, and 3](#) are based on averages taken over $K = 500$ runs. Specifically, the final columns of each table contain for each method (i) the mean of the prediction coverage ratios (MCR) and (ii) the mean interval score (MIS) at the 90% level, both relative to B–J. An MCR value greater than one demonstrates that the respective method provides better predictive coverage than the classical B–J, while MIS values less than one indicate that the respective method provides better prediction intervals, and balance between narrow range and coverage. We note that the tables only show results corresponding to the TVAR2 method, but similar results are also obtained when using higher orders (e.g. order 5), hence these are not reported here.

4.1 Stationary series

[Table 1](#) shows empirical prediction interval coverage rates for three models each with $T = 128$ and standard normal innovations. Model A simulates independent $N(0, 1)$ variates, Model B simulates from a stationary AR(1) model with parameter $\alpha = 0.7$, and Model C simulates from a stationary MA(1) model with parameter $\beta = -0.5$.

The prediction coverage rates and accuracy for B–J are best across all models, with FORLAP and ES almost matching for coverage across all models. The accuracy of FORLAP and ES for Models A and C is close to their nominal values (a very similar behaviour to TVAR2). For

Table 1. Empirical prediction interval coverage rates and accuracy: stationary underlying series

		Nominal						MCR	MIS
		40%	50%	60%	70%	80%	90%		
Model A N(0, 1)	B-J	39.3	48.5	58.5	68.6	79.0	89.2		
	FORLAP	38.8	48.0	58.1	68.2	78.3	88.8	1.00	1.00
	FVBvS	36.7	46.4	55.7	65.4	74.9	86.1	0.97	1.09
	TVAR2	39.1	48.4	58.5	68.8	78.9	89.2	1.00	0.99
	ES	39.5	49.4	59.4	69.3	79.5	89.7	1.01	1.00
Model B AR(1)	B-J	38.9	48.9	58.2	68.2	78.7	88.9		
	FORLAP	36.8	46.0	54.9	65.0	75.1	85.5	0.96	1.38
	FVBvS	25.9	32.5	39.3	46.9	55.4	65.2	0.74	5.22
	TVAR2	29.4	37.2	45.3	53.4	62.8	74.7	0.84	1.66
	ES	39.7	49.4	59.3	69.0	79.3	89.9	1.01	1.05
Model C MA(1)	B-J	39.2	49.3	59.1	69.1	79.8	89.4		
	FORLAP	36.7	45.8	56.0	65.9	76.3	87.0	0.97	1.11
	FVBvS	32.2	41.1	49.6	58.8	68.3	79.3	0.89	1.49
	TVAR2	35.9	45.0	54.5	64.7	74.9	86.2	0.96	1.10
	ES	40.1	50.2	60.2	70.4	80.7	90.6	1.02	1.09

Note. MCR = mean prediction coverage. MIS = mean interval score. (Superior behaviour relative to B-J when MCR > 1, MIS < 1.)

Model B, ES has better accuracy, but still lower than that of B-J. While all methods do well for Model A, FVBvS and TVAR are markedly inferior for Model B, and to a lesser extent for FVBvS on Model C.

As we designed this setup specifically to include stationary processes, *a priori* one would have expected B-J to be significantly better as it is designed for stationary series. However, the results show that our proposed FORLAP method is competitive even though it is not designed for stationary data, while the other nonstationary methods underperform.

4.2 Nonstationary series

Model specification. Tables 2 and 3 show coverage rates for ten nonstationary models, summarized as follows. Models D and E are realizations of TVAR(1) processes of different forms, whilst Models F and G are higher-order TVAR processes of orders two and twelve, respectively. Models H-J correspond to different time-varying MA processes. Model K is a uniformly modulated white noise process (Priestley, 1983, p. 826), whilst Models L and M are locally stationary wavelet processes. Complete model descriptions can be found in A. In each case the underlying innovations are independent zero-mean normally distributed random variables, and all realizations are of length $T = 128$, except for Model L where $T = 512$, and Model M where $T = 350$ for variety.

Discussion of simulation results. The FORLAP method provides much better coverage ($MCR \geq 1.05$) than B-J for five of the models considered (F, H, J, K, M). For the Models D, G, I, and L the two methods are broadly comparable. The method accuracy as measured by MIS provides a broadly similar picture, with FORLAP superior to B-J for five Models F, H, I, J, and K ($MIS \leq 0.95$), similar calibration for Models G, L, and M (MIS in the range 0.95–1.05) and lower for Models D and E ($MIS \geq 1.05$).

For Model E, no approach proves particularly competitive. However, this is not altogether surprising as Model E is neither stationary nor, in fact, locally stationary with Lipschitz smoothness constraints as it experiences six changes in parameter, $\alpha(z)$, over only $T = 128$ observations. Hence, a stationary forecasting system is clearly not appropriate and estimation by the nonstationary models is likely to be poor, as few observations contribute to any region of local stationarity.

Table 2. Models D–H. Empirical prediction interval coverage rates and accuracy: nonstationary underlying series

		Nominal						MCR	MIS
		40%	50%	60%	70%	80%	90%		
Model D TVAR(1)	B-J	31.4	39.6	47.8	57.1	67.5	79.1		
	FORLAP	30.1	37.5	45.5	54.4	64.5	76.3	0.96	1.08
	FVBvS	21.6	27.3	33.3	39.7	47.6	57.7	0.73	3.88
	TVAR2	27.6	34.6	42.2	50.1	59.6	71.3	0.90	1.21
	ES	29.8	37.8	46.1	54.9	64.9	77.4	0.98	1.13
Model E TVAR(1)	B-J	25.2	32.2	39.2	47.8	56.8	69.4		
	FORLAP	19.4	24.6	30.2	36.0	43.1	51.9	0.75	2.49
	FVBvS	22.9	28.5	35.1	41.5	49.1	57.9	0.84	502.36
	TVAR2	19.2	24.6	29.8	36.1	44.3	54.3	0.77	1.92
	ES	18.6	24.1	30.1	36.9	44.8	56.1	0.80	1.92
Model F TVAR(2)	B-J	29.8	38.2	46.6	55.4	65.7	77.80		
	FORLAP	35.4	44.8	53.9	63.9	74.1	84.4	1.09	0.84
	FVBvS	29.5	37.4	46.3	55.0	64.8	76.0	0.98	1.45
	TVAR2	29.8	37.8	46.2	55.3	65.4	76.9	0.99	0.93
	ES	47.4	58.2	68.8	78.2	86.6	93.6	1.22	0.82
Model G TVAR(12)	B-J	36.6	45.8	55.5	65.0	75.4	86.1		
	FORLAP	35.8	44.8	54.3	64.0	74.4	85.4	0.99	1.01
	FVBvS	33.3	41.7	50.2	59.5	69.4	80.5	0.94	1.32
	TVAR2	35.10	44.1	52.9	62.8	73.3	84.4	0.98	1.00
	ES	37.8	47.3	57.3	67.1	77.5	88.1	1.03	0.96
Model H TVMA(1)	B-J	27.3	34.7	42.0	50.4	59.1	69.9		
	FORLAP	33.1	42.4	50.9	60.1	69.4	80.2	1.17	0.66
	FVBvS	30.7	38.9	47.0	55.4	64.7	74.8	1.09	5.53
	TVAR2	32.5	42.3	49.8	59.4	71.0	82.6	1.20	0.53
	ES	36.5	46.2	55.6	65.1	75.6	85.9	1.25	0.60

Note. MCR = mean prediction coverage. MIS = mean interval score. (Superior behaviour relative to B–J when MCR > 1, MIS < 1.)

When compared to the B–J method, the FVBvS method provides much better coverage ($MCR \geq 1.05$) for three of the models considered (H, J, K); of these, FORLAP outperforms FVBvS on Model H (1.17 vs. 1.09) and has lower performance on Models J and K (1.22 vs. 1.32 and 1.23 vs. 1.35, respectively). The accuracy of the FVBvS prediction intervals is lower ($MIS \geq 1.05$) than that of B–J for all models except for Models J (superior) and K (the same).

When comparing TVAR coverage rates to those achieved by the B–J baseline, TVAR outperforms B–J on Model H and delivers comparable results (MCR values in the range 0.95–1.05) on all other models except for underperforming on models D and E. Unlike FVBvS, TVAR achieves similar levels of accuracy to B–J (MIS in the range 0.95–1.05) for Models G, I, J, K, L, and M, worse accuracy for Models D, E, and better performance for Models F and H.

The forecasting accuracy measure (MIS) consistently indicates that our proposed FORLAP technique delivers better calibrated prediction intervals than FVBvS on *all* models, with sizeable performance gaps in favour of FORLAP. This metric indicates that FORLAP and TVAR deliver comparably accurate prediction intervals for Models G, L, and M (ratio in the range 0.95–1.05), FORLAP is better for Models D (1.08 vs. 1.21), F (0.84 vs. 0.93), I (0.93 vs. 1.01), J (0.80 vs. 0.96), and K (0.79 vs. 0.95), and worse for Models E (2.49 vs. 1.92) and H (0.66 vs. 0.53). According to the coverage rate measure, FORLAP is superior to TVAR on all models,

Table 3. Models I–M. Empirical prediction interval coverage rates and accuracy: nonstationary underlying series

		Nominal						MCR	MIS
		40%	50%	60%	70%	80%	90%		
Model I TVMA(1)	B-J	33.1	41.9	50.7	60.2	70.6	82.3		
	FORLAP	33.2	41.7	50.4	59.9	70.2	82.1	1.00	0.93
	FVBvS	28.9	36.2	43.8	52.0	61.4	73.0	0.89	1.66
	TVAR2	30.8	38.5	46.9	55.8	65.9	78.1	0.95	1.01
Model J TVMA(2)	ES	34.1	42.8	51.6	61.4	72.0	83.9	1.02	0.96
	B-J	22.2	28.2	34.7	41.5	49.7	60.6		
	FORLAP	28.0	35.3	42.9	51.6	60.9	73.0	1.22	0.80
	FVBvS	31.4	39.4	48.0	56.9	67.0	78.2	1.32	0.88
Model K TVWN	TVAR2	22.0	28.2	34.5	42.0	50.7	61.7	1.03	0.96
	ES	22.6	29.1	35.4	43.0	51.6	62.9	1.05	0.96
	B-J	20.6	26.0	31.9	38.9	47.3	58.3		
	FORLAP	26.3	33.4	40.5	48.7	58.2	70.4	1.23	0.79
Model L LSW-P3	FVBvS	29.8	37.9	46.3	55.4	65.6	77.0	1.35	1.00
	TVAR2	21.0	26.8	32.9	39.8	48.4	59.2	1.02	0.95
	ES	21.3	27.0	33.2	40.1	48.8	59.7	1.03	0.96
	B-J	41.6	51.4	61.1	71.2	81.5	91.0		
Model M LSW-P4	FORLAP	41.9	52.3	62.5	72.6	82.4	91.7	1.01	1.05
	FVBvS	31.3	39.1	47.0	56.0	65.9	77.4	0.85	4.03
	TVAR2	41.5	51.0	61.1	71.2	81.2	91.0	1.00	1.02
	ES	43.3	53.8	64.4	74.7	84.5	93.4	1.03	1.02
Model N	B-J	34.2	42.7	51.7	60.2	70.6	82.4		
	FORLAP	36.8	45.8	55.5	65.1	75.1	85.4	1.05	1.02
	FVBvS	33.7	42.1	50.9	59.7	69.7	80.8	0.99	2.24
	TVAR2	31.8	40.0	48.6	57.9	67.6	79.3	0.97	1.02
Model O	ES	37.2	47.1	56.5	66.0	76.4	86.8	1.06	0.95

Note. MCR = mean prediction coverage. MIS = mean interval score. (Superior behaviour relative to B–J when MCR > 1, MIS < 1.)

except for models E, G, H, I, and L where they deliver similar rates (corresponding to ratios in the range 0.95–1.05).

The simulations for Model F are particularly revealing. The underlying model is a TVAR(2) and, indeed, the forecasting method based on that (TVAR2) performs somewhat better (MCR 0.99, MIS 0.93) than the B–J method. However, it is interesting that our new method, FORLAP, produces best coverage (1.09) and accuracy (MIS 0.84). Note that the other wavelet-based method, FVBvS, has comparable coverage to TVAR2 (0.98) at the price of sacrificing some accuracy (MIS 1.45).

In the TVAR2 setting, the real competitor for FORLAP turns out to be ES, which produces comparable results in terms of coverage and accuracy rates (corresponding ratios in the range 0.95–1.05) to FORLAP for all models, except for ES obtaining superior results on Model H and FORLAP on Models J and K, where ES performs closer to the B–J benchmark.

The coverage rates are also reasonably stable for all methods. The standard deviation of coverage rates for each model and coverage level is roughly between 7% and 19%, with the majority around 12%. This is true for all models except for the ‘pathological’ Model E, which ranges from about 12% to 23%, which is not surprising given the difficulty of forecasting data from this model.

In addition to the simulations described above, we repeated our simulations on these models with t_4 -distributed innovations scaled to have unit variance for B-J, FORLAP and FVBvS. With the heavier-tailed innovations the relative performance of B-J compared to FORLAP forecasting changes negligibly: the largest MCR change is 0.05 and 70% of the changes are less than 0.02.

Overall, one could conclude that, for zero-mean locally stationary series, the FORLAP method is much better than the B-J method in two-thirds of cases and about the same in the remaining one-third, and provides an improvement over all of the other nonstationary-based methods across the majority of our models.

5 Forecasting the UK national accounts time series

We now return to consider the ABML series that helped motivate this work in Section 1. The B-J method is sometimes used to forecast the ABML series, see [U.K. Centre for the Measurement of Government Activity \(2008\)](#) for example. [Table 4](#) shows that FORLAP seriously outperforms the B-J methodology for forecasting the second differences of ABML series at one-step ahead. Here, the input data are provided to both methods up to $t = T - n$ and $T - n + 1$ is forecast and the percentages are average success rates over $n = T - 51, \dots, T - 1$.

[Figure 3](#) compares FORLAP against B-J forecasting for ABML in a similar way to [Table 4](#) except for the last 20 time points. The good performance of FORLAP relative to B-J can be discerned, e.g. particularly Q1 and Q4 of 2020, but the sheer size of the movements during the pandemic obscures what the forecasts are doing earlier in the plot. So, [Figure 4](#) shows a similar plot except the vertical axis is plotted on a signed square-root transformed scale to de-emphasize the COVID period. Here, it is clear that FORLAP succeeds 16 times out of 20 (80%), whereas B-J succeeds 13 times (65%), although some point estimates for B-J perform better than the point estimates for FORLAP.

After extensive experiments with the ABML data it seems that Haar wavelets perform best. Also, FORLAP's advantage persists over high step-ahead forecasts (e.g. $h = 2, 3$), but diminishes and becomes worse the further into the future. This is perhaps not surprising, due to the quite strong non-stationarity and B-J, in this case, still providing some degree of 'catch all' forecast accuracy.

6 Discussion

We have demonstrated how the recently proposed local partial autocorrelation function introduced in [Killick et al. \(2020\)](#) can be used to select the p parameter, i.e. how much recent data are relevant, in the locally stationary forecasting method of [Fryzlewicz et al. \(2003\)](#). Our subsequent modified forecasting method, FORLAP, outperforms not only the original [Fryzlewicz et al. \(2003\)](#) approach but also the method based on direct time-varying autoregressive estimation and the B-J method, on the majority of our simulated examples and on the practical data application.

On average, FORLAP takes about 0.15 s for a $T = 256$ length data set, 0.26 s for $T = 512$, 0.52 s for $T = 1,024$, and 1.28 s for $T = 2,048$. On average, FORLAP is about 10 to 20 times slower than the basic forecasting found in the `forecast` package.

A bootstrap test might provide an alternative to using asymptotic-based confidence intervals to select $\hat{p}$. One could view p as a kind of local autoregressive model order, even if the underlying process was not autoregressive, and attempt to develop a theory of how well $\hat{p}$ estimates it if the underlying process was autoregressive, but this is beyond the scope of the current work. Additionally, the simulation results in [Section 4](#) provide good examples of the phenomenon

Table 4. Percentage of times that the B-J and the FORLAP methods' 95% one-step ahead forecast prediction intervals contain the truth over the last 50 time points

Method	ABML series	
	With COVID	Without COVID
Box-Jenkins	66	72
FORLAP	90	90

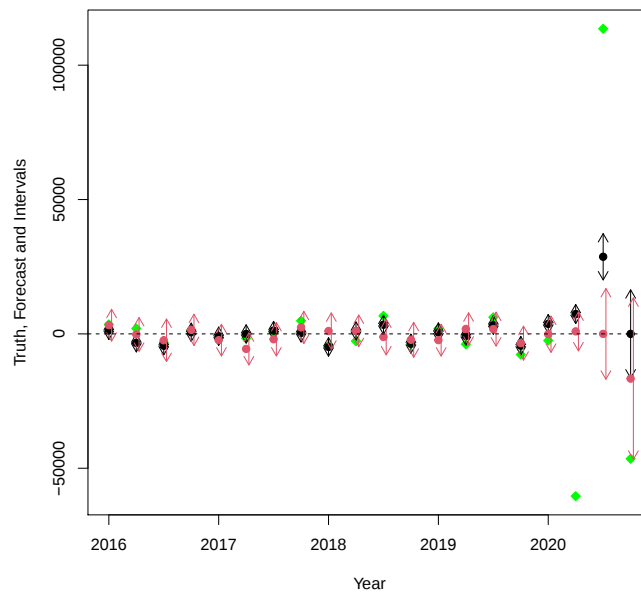


Figure 3. Forecast results for one-step ahead forecasting of the final twenty observations of the ABML series including the COVID period. Green diamond = True value of series. Black = Box-Jenkins. Red = FORLAP. Solid circle indicates point forecast and double-headed arrows indicate 95% prediction intervals.

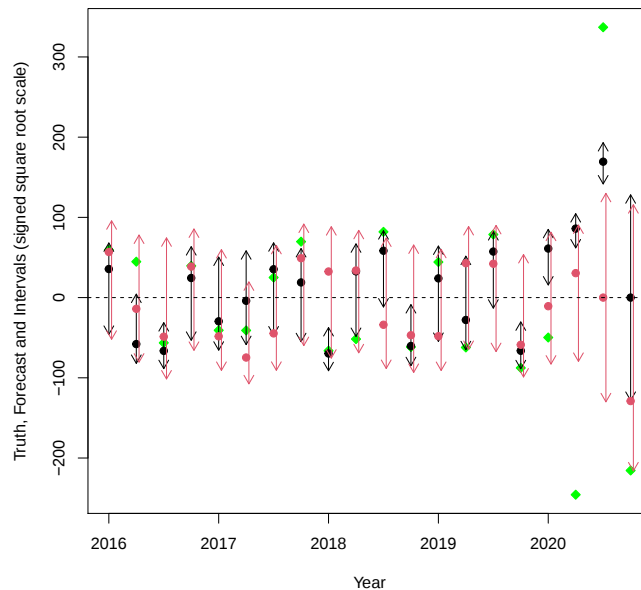


Figure 4. Forecast results for one-step ahead forecasting of the final twenty observations of the ABML series including the COVID period plotted on a signed square root scale. Green diamond = True value of series. Black = Box-Jenkins. Red = FORLAP. Solid circle indicates point forecast and double-headed arrows indicate 95% prediction intervals.

highlighted by [Kley et al. \(2017\)](#), where the best predictor of a nonstationary time series might be a stationary predictor, and vice versa, e.g. Model E in [Table 2](#). Finally, the predictive interval coverage, and our use of the least-squares loss function are not necessarily the only, or the best, methods for assessing predictive performance and problem-specific/problem-tailored measures are of value. For a more detailed discussion, see [Gneiting \(2011\)](#), for example.

For the ABML time series, it is fascinating to see how well FORLAP does against B–J even during periods of significant change, as, for example, during the COVID pandemic. We can only put this down to the flexible modelling and forecasting afforded by a method that explicitly acknowledges the nonstationary nature of the time series. Part of FORLAP's success is that it can position its forecast intervals better, but it can respond much quicker to variance changes and this can be seen in the forecast intervals shown in Figures 3 and 4.

Further research is also necessary to properly develop a mature understanding of the proposed forecasting methodology, particularly on comparisons with other methods for forecasting of nonstationary series (where code is not freely available), on different types of series, and at forecast horizons other than $h = 1$. Additionally, while the focus of this work is channelled on short-memory locally stationary processes, further extensions to processes that exhibit features such as long-memory or nonlinearity would form interesting avenues for future development.

Acknowledgments

The authors are grateful to the Editor, an Associate Editor and two anonymous referees for their constructive suggestions on the earlier version of this article.

Conflicts of interest: None declared.

Funding

The authors were partially supported by the Research Councils UK (RCUK) Energy Programme. The Energy Programme is an RCUK cross-council initiative led by Engineering and Physical Sciences Research Council (EPSRC) and contributed to by Economic and Social Research Council (ESRC), Natural Environment Research Council (NERC), Biotechnology and Biological Sciences Research Council (BBSRC), and Science and Technology Facilities Council (STFC). G.P.N. gratefully acknowledges support from EPSRC grants EP/I01697X/1 and K020951/1. I.A.E. gratefully acknowledges support from EPSRC grant EP/I016368/1.

Data availability

The ABML data can be obtained from the Office for National Statistics website <https://www.ons.gov.uk>. The (differenced) ABML dataset for the period examined in this article, as well as the FORLAP forecasting method, is available within the `forecastLSW` R package on CRAN.

Appendix. Simulation study models for nonstationary series

Model D corresponds to the time-varying autoregressive TVAR(1) model $X_t = \alpha_t X_{t-1} + Z_t$ for $t = 1, \dots, 128$ and $\alpha_t = \alpha(t/T)$ where $\alpha(z) = 1.8z - 0.9$ for $z \in (0, 1)$.

Model E corresponds to the TVAR(1) model with the same specification as for Model D except that

$$\alpha(z) = \begin{cases} 5.6z - 0.9 & \text{for } z \in (0, 1/8), \\ 4.8z - 0.8 & \text{for } z \in (1/8, 2/8), \\ 3.2z - 0.4 & \text{for } z \in (2/8, 3/8), \\ 0.8 & \text{for } z \in (3/8, 5/8), \\ -2.4z + 2.6 & \text{for } z \in (5/8, 6/8), \\ -7.2z + 5.4 & \text{for } z \in (6/8, 7/8), \\ -1.6z + 0.5 & \text{for } z \in (7/8, 1). \end{cases}$$

Model F corresponds to the TVAR(2) model $X_t = \alpha_{1,t} X_{t-1} + \alpha_{2,t} X_{t-2} + Z_t$ with $\alpha_{i,t} = \alpha_i(t/T)$ for $i = 1, 2$ and $\alpha_1(z) = \alpha_2(z) = 1.6z - 1.1$ for $z \in (0, 1)$.

Model G corresponds to the TVAR(12) model $X_t = \alpha_{1,t} X_{t-1} + \alpha_{2,t} X_{t-2} + \alpha_{12,t} X_{t-12} + Z_t$, where $\alpha_{i,t} = \alpha_i(t/T)$ for $i = 1, 2, 12$ and $\alpha_1(z) = \alpha_2(z) = 0.7z - 0.4$ and $\alpha_{12}(z) = 0.3z$ for $z \in (0, 1)$.

Model H corresponds to the TVMA(1) model $X_t = Z_t + \beta_t Z_{t-1}$, where $\beta_t = \beta(t/T)$, where $\beta(z) = 1$ for $z \in (0, 0.9)$ and $\beta(z) = -1$ for $z \in (0.9, 1)$.

Model I corresponds to the TVMA(1) model as in Model H but with $\beta(z) = 2z - 1$ for $z \in (0, 1)$.

Model J corresponds to the TVMA(2) model $X_t = Z_t + \beta_{1,t}Z_{t-1} + \beta_{2,t}Z_{t-2}$ with $\beta_{i,t} = \beta_i(t/T)$ for $i = 1, 2$, where $\beta_1(z) = 2z - 1$ and $\beta_2(z) = 9z - 0.8$ for $z \in (0, 1)$.

Model K corresponds to uniformly modulated white noise, see Priestley (1983, page 826), where $X_t = \sigma_t^2 Z_t$, where $\sigma_t^2 = \sigma^2(t/T)$, where $\sigma^2(z) = (9z + 1)^{3/2}$ for $z \in (0, 1)$. Note, this model might be a good fit for the ABML time series from Section 1.

Model L is a locally stationary wavelet (LSW) process from Nason et al. (2000) with spectrum P3 from Nason (2013a) defined by $S_j(z) = 0$ for $j > 2$, $S_1(z) = \frac{1}{4} - (z - \frac{1}{2})^2$, and $S_2(z) = S_1(z + \frac{1}{2})$ for $z \in (0, 1)$, assuming periodic boundaries for spectrum construction only.

Model M is a LSW process from Nason et al. (2000) with spectrum P4 from Nason (2013a) defined by $S_j(z) = 0$ for $j = 2, j > 4$ and $S_1(z) = \exp\{-4(z - \frac{1}{4})^2\}$, $S_3(z) = S_1(z - \frac{1}{4})$, and $S_4(z) = S_1(z + \frac{1}{4})$, for $z \in (0, 1)$, assuming periodic boundaries for spectrum construction only. The process, X_t , is computed for $t = 1, \dots, 512$ and the first $T = 350$ values are returned.

References

- Abdrabbo N., & Priestley M. (1967). On the prediction of non-stationary processes. *Journal of the Royal Statistical Society B*, 29(3), 570–585. <https://doi.org/10.1111/j.2517-6161.1967.tb00718.x>
- Box G., & Jenkins G. (1970). *Time series analysis: Forecasting and control*. Holden-Day.
- Brockwell P. J., & Davis R. A. (1991). *Time series: Theory and methods*. Springer.
- Casas I., & Fernandez-Casal R. (2019). *tvReg: Time-varying coefficients linear regression for single and multi-equations*. <https://CRAN.R-project.org/package=tvReg>. R package version 0.4.2.
- Chatfield C. (2003). *The analysis of time series: An introduction*. Chapman and Hall/CRC.
- Chow T. W. S., & Leung C. T. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, 11(4), 1736–1742. <https://doi.org/10.1109/59.544636>
- Dahlhaus R. (1996). On the Kullback-Leibler information divergence of locally stationary processes. *Stochastic Processes and their Applications*, 62(1), 139–168. [https://doi.org/10.1016/0304-4149\(95\)00090-9](https://doi.org/10.1016/0304-4149(95)00090-9)
- Ding X., & Zhou Z. (2023). Autoregressive approximations to nonstationary time series with inference and applications. *The Annals of Statistics*, 51, 1207–1231. <https://doi.org/10.1214/23-AOS2288>
- Eckley I. A., & Nason G. P. (2018). A test for the absence of aliasing or local white noise in locally stationary wavelet time series. *Biometrika*, 105, 833–848. <https://doi.org/10.1093/biomet/asy040>
- Embleton J., Knight M. I., & Ombao H. (2022a). Multiscale spectral modelling for nonstationary time series within an ordered multiple-trial experiment. *The Annals of Applied Statistics*, 16(4), 2774–2803. <https://doi.org/10.1214/22-AOAS1614>
- Embleton J., Knight M. I., & Ombao H. (2022b). Wavelet testing for a replicate-effect within an ordered multiple-trial experiment. *Computational Statistics & Data Analysis*, 174, Article 107456. <https://doi.org/10.1016/j.csda.2022.107456>
- Fryzlewicz P. (2005). Modelling and forecasting financial log-returns as locally stationary wavelet processes. *Journal of Applied Statistics*, 32, 503–528. <https://doi.org/10.1080/02664760500079563>
- Fryzlewicz P., & Nason G. P. (2006). Haar-Fisz estimation of evolutionary wavelet spectra. *Journal of the Royal Statistical Society B*, 68(4), 611–634. <https://doi.org/10.1111/j.1467-9868.2006.00558.x>
- Fryzlewicz P., & Ombao H. (2009). Consistent classification of nonstationary time series using stochastic wavelet representations. *Journal of the American Statistical Association*, 104(485), 299–312. <https://doi.org/10.1198/jasa.2009.0110>
- Fryzlewicz P., Van Bellegem S., & von Sachs R. (2003). Forecasting non-stationary time series by wavelet process modelling. *The Annals of the Institute of Statistical Mathematics*, 55(4), 737–764. <https://doi.org/10.1007/BF02523391>
- Fryzlewicz P. Z. (2003). *Wavelet techniques for time series and poisson data* [Ph.D. thesis]. University of Bristol, U.K.
- Gardner E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1), 1–28. <https://doi.org/10.1002/for.v4:1>
- Gneiting T. (2011). Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494), 746–762. <https://doi.org/10.1198/jasa.2011.r10138>
- Gneiting T., & Raftery A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378. <https://doi.org/10.1198/016214506000001437>
- Hallin M. (1986). Non-stationary q -dependent processes and time-varying moving-average models. *Advances in Applied Probability*, 18(1), 170–210. <https://doi.org/10.2307/1427242>
- Hargreaves J. K., Knight M. I., Pitchford J. W., Oakenfull R., Chawla S., Munns J., & Davis S. J. (2019). Wavelet spectral testing: Application to nonstationary circadian plant rhythms. *The Annals of Applied Statistics*, 13(3), 1817–1846. <https://doi.org/10.1214/19-AOAS1246>

- Hargreaves J. K., Knight M. I., Pitchford J. W., Oakenfull R., & Davis S. J. (2018). Clustering nonstationary circadian plant rhythms using locally stationary wavelet representations. *Multiscale Modeling and Simulation*, 16(1), 184–214. <https://doi.org/10.1137/16M1108078>
- Hyndman R., Koehler A., Ord K., & Snyder R. (2008). *Forecasting with exponential smoothing*. Springer Series in Statistics. Springer.
- Hyndman R. J., & Khandakar Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3), 1–22. <https://doi.org/10.18637/jss.v027.i03>
- Janeway W. (2009). Six impossible things before breakfast. *Significance*, 6(1), 28–31. <https://doi.org/10.1111/j.1740-9713.2009.00339.x>
- Killick R., Eckley I., & Jonathan P. (2013). A wavelet-based approach for detecting changes in second order structure within nonstationary time series. *Electronic Journal of Statistics*, 7, 1167–1183. <https://doi.org/10.1214/13-EJS799>
- Killick R., Knight M. I., Nason G. P., & Eckley I. A. (2020). The local partial autocorrelation function and some applications. *Electronic Journal of Statistics*, 14(2), 3268–3314. <https://doi.org/10.1214/20-EJS1748>
- Killick R., Nason G. P., Nunes M. A., & Knight M. I. (2023a). *forecastLSW: Forecasting routines for locally stationary wavelet processes*. R package version 1.0.
- Killick R., Nason G. P., Knight M. I., & Nunes M. A. (2023b). *lpcf: Local partial autocorrelation function estimation for locally stationary wavelet processes*. R package version 1.5.
- Kley T., Preuss P., & Fryzlewicz P. (2017). Predictive, finite-sample model choice for time series under stationarity and non-stationarity. stats.lse.ac.uk/fryzlewicz/articles.html.
- Kley T., Preuß P., & Fryzlewicz P. (2019). Predictive, finite-sample model choice for time series under stationarity and non-stationarity. *Electronic Journal of Statistics*, 13, 3710–3774. <https://doi.org/10.1214/19-EJS1606>
- Knight M. I., Nunes M. A., & Hargreaves J. K. (2024). Adaptive wavelet domain principal component analysis for nonstationary time series. *Journal of Computational and Graphical Statistics*, 1–14. <https://doi.org/10.1080/10618600.2023.2301069>
- Krzemieniewska K., Eckley I. A., & Fearnhead P. (2014). Classification of non-stationary time series. *Stat*, 3, 144–157. <https://doi.org/10.1002/sta4.v3.1>
- Mercurio D., & Spokoiny V. (2004). Statistical inference for time-inhomogeneous volatility models. *The Annals of Statistics*, 32, 577–602. <https://doi.org/10.1214/009053604000000102>
- Nam C., Aston J., Eckley I., & Killick R. (2015). The uncertainty of storm season changes: Quantifying the uncertainty of autocovariance changepoints. *Technometrics*, 57, 194–206. <https://doi.org/10.1080/00401706.2014.902776>
- Nason G. P. (2013a). A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series. *Journal of the Royal Statistical Society B*, 75, 879–904. <https://doi.org/10.1111/rssb.12015>
- Nason G. P. (2013b). *locits: Tests of stationarity and localized autocovariance*. <https://CRAN.R-project.org/package=locits>. R package version 1.4.
- Nason G. P., von Sachs R., & Kroisandt G. (2000). Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *Journal of the Royal Statistical Society B*, 62(2), 271–292. <https://doi.org/10.1111/1467-9868.00231>
- Nason G. P., & Stevens K. (2015). Bayesian wavelet shrinkage of the Haar-Fisz transformed wavelet periodogram. *PLoS One*, 10(9), Article e0137662. <https://doi.org/10.1371/journal.pone.0137662>
- Priestley M. B. (1983). *Spectral analysis and time series*. Academic Press.
- Priestley M. B., & Subba Rao T. (1969). A test for stationarity of time series. *Journal of the Royal Statistical Society B*, 31(1), 140–149. <https://doi.org/10.1111/j.2517-6161.1969.tb00775.x>
- R Core Team (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Subba Rao T. (1973). Discussion of the paper by Priestley and Tong. *Journal of the Royal Statistical Society B*, 35, 182–183. <https://doi.org/10.1111/j.2517-6161.1973.tb00951.x>
- Tsay R. S. (2002). *Analysis of financial time series*. Wiley.
- U.K. Centre for the Measurement of Government Activity (2008). From Holt-Winters to ARIMA modelling: Measuring the impact of forecasting errors for components of quarterly estimates of public service output. (Technical Report) Office for National Statistics, U.K.
- Van Bellegem S., & von Sachs R. (2004). Forecasting economic time series with unconditional time-varying variance. *International Journal of Forecasting*, 20(4), 611–627. <https://doi.org/10.1016/j.ijforecast.2003.10.002>
- Whittle P. (1963). Recursive relations for predictors of non-stationary processes. *Journal of the Royal Statistical Society B*, 27(3), 523–532. <https://doi.org/10.1111/j.2517-6161.1965.tb00612.x>
- Wu W., & Dette H. (2022). Prediction in locally stationary time series. *Journal of Business & Economic Statistics*, 40, 370–381. <https://doi.org/10.1080/07350015.2020.1819296>
- Xie Y., Yu J., & Ranneby B. (2009). Forecasting using locally stationary wavelet processes. *Journal of Statistical Computation and Simulation*, 79(9), 1067–1082. <https://doi.org/10.1080/00949650802087003>