Leveraging Non-Decimated Wavelet Packet Features and Transformer Models for Time Series Forecasting

Guy P. Nason *and James L. Wei †

March 14, 2024

Abstract

This article combines wavelet analysis techniques with machine learning methods for univariate time series forecasting, focusing on three main contributions. Firstly, we consider the use of Daubechies wavelets with different numbers of vanishing moments as input features to both non-temporal and temporal forecasting methods, by selecting these numbers during the cross-validation phase. Secondly, we compare the use of both the non-decimated wavelet transform and the non-decimated wavelet packet transform for computing these features, the latter providing a much larger set of potentially useful coefficient vectors. The wavelet coefficients are computed using a shifted version of the typical pyramidal algorithm to ensure no leakage of future information into these inputs. Thirdly, we evaluate the use of these wavelet features on a significantly wider set of forecasting methods than previous studies, including both temporal and non-temporal models, and both statistical and deep learning-based methods. The latter include state-of-the-art transformer-based neural network architectures. Our experiments suggest significant benefit in replacing higher-order lagged features with wavelet features across all examined non-temporal methods for one-step-forward forecasting, and modest benefit when used as inputs for temporal deep learning-based models for long-horizon forecasting.

Keywords: Time series forecasting, wavelets, wavelet packets, non-decimated wavelets, transformers.

1 Introduction

Univariate time series forecasting is a crucial area of research with important applications across numerous fields, such as electricity load forecasting and environmental forecasting. Recently, there has been increased interest in hybrid methods that combine traditional statistical methods and more advanced machine learning methods to generate more accurate forecasts (Lim and Zohren (2021)), which have achieved state-of-the-art performance in time series forecasting competitions (Makridakis et al. (2020)). Our study investigates the use of wavelet transforms to generate features for a wide array of machine learning methods, including deep learning architectures, demonstrating large gains in forecasting performance across different data sets for the majority of benchmark models.

*g.nason@imperial.ac.uk, Department of Mathematics Imperial College London, London, United Kingdom

†james.wei19@imperial.ac.uk Department of Mathematics Imperial College London London, United Kingdom

The application of wavelets to time series forecasting problems has received considerable attention in the past two decades, with the main approaches involving either the use of wavelet-based denoising and decomposition on the input time series (Wong et al. (2003), Conejo et al. (2005), Schlüter and Deuschle (2010), Wang and Guo (2020)) or by directly using the wavelet coefficients as additional features to be used by the forecasting model (Nason and Sapatinas (2002), Adjoumani (2021)).

We build upon efforts in the latter category by considering the use of Daubechies wavelets with different numbers of vanishing moments as input features to both non-temporal and temporal forecasting methods. We also investigate the utility of both the non-decimated wavelet transform and the non-decimated wavelet packet transform for computing these features, where the latter has already been successfully employed in classification tasks (Nason et al. (2001)). Our approach uses a shifted version of the pyramidal algorithm to avoid information leakage from future observations that can be implemented in an online fashion.

Moreover, our experiments demonstrate the usefulness of these wavelet features for both short- and long-horizon forecasting applications, by combining them with a far wider set of forecasting methods than have previously been investigated in the literature. These include both temporal and non-temporal models, and both statistical and deep learning-based methods. The latter include recently-developed transformer-based neural network architectures, including the Temporal Fusion Transformer (Lim et al. (2021)), Informer (Zhou et al. (2021)), Autoformer Wu et al. (2021)), and Patch Time Series Transformer (Nie et al. (2022)).

Section 2 provides a brief introduction to wavelet analysis, including non-decimated wavelet transforms and wavelet packet transforms, and their application to time series forecasting problems. The section concludes with a summary of the machine learning methods investigated in our wavelet-machine learning (wavelet-ML) approach. Section 3 introduces our simple online algorithm for computing the the non-decimated wavelet and wavelet packet coefficients. Section 4 describes empirical experiments evaluating the performance benefits to using wavelet features, and Section 5 concludes with a discussion of future avenues of research.

2 Background

2.1 Discrete wavelet transforms

In time series analysis, wavelets can be used to decompose a time series into localised components at multiple scales. To introduce the wavelet transform in this context, let us assume that a dyadic sequence of length $T = 2^J$ for some integer $J \geq 0$, $\mathbf{y} = (y_1, ..., y_T)^T$, is observed from some univariate time series $\{Y_t\}$. We first motivate the multiscale analysis of time series by Haar wavelets, before generalising to all wavelets, in line with the introductions of Nason (2008) and La Cour-Harbo and Jensen (2009).

Following Daubechies (1988) or Mallat (1989), the finest level of 'detail' in $\mathbf{y}$ can be obtained by the differencing operations

$$d_{J-1,k} = (y_{2k} - y_{2k-1})/\sqrt{2}, \tag{1}$$

for $k = 1, 2, ..., T/2$, where the $J-1$ subscript relates to the 2^{J-1}-length of the resulting sequence. The next coarser 'smoothed' sequence is generated by the summations

$$c_{J-1,k} = (y_{2k} + y_{2k-1})/\sqrt{2}, \tag{2}$$

again for $k = 1, 2, ..., T/2$. The scaling by $\sqrt{2}$ in (1) and (2) conserves the energy in the original time series. Similarly, detail $\{d_{j,k}\}$ and smoothed sequences $\{c_{j,k}\}$ at coarser scales $j < J - 1$ may be obtained from

$$d_{j,k} = (c_{j+1,2k} - c_{j+1,2k-1})/\sqrt{2} \tag{3}$$

and

$$c_{j,k} = (c_{j+1,2k} + c_{j+1,2k-1})/\sqrt{2}, \tag{4}$$

for $k = 1, ..., T/2^{J-j}$. Hence, smaller j corresponds to coarser scales.

In wavelet terminology, $\{d_{j,k}\}$ and $\{c_{j,k}\}$ are (mother) wavelet coefficients and scaling (or father wavelet) coefficients respectively, at scale j and location k, from the discrete wavelet transform (DWT) using Haar wavelets (Haar (1910)). The operations that perform the inverse of (1)-(4) constitute the corresponding inverse discrete wavelet transform (IDWT).

We now extend the previous discussion to any wavelet function $\psi(x)$ and scaling function $\phi(x)$, where $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$ and $\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$. For Haar wavelets,

$$\psi(x) = \begin{cases} 1 & 0 \le x < 1/2, \\ -1 & 1/2 \le x < 1, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

and

$$\phi(x) = \begin{cases} 1 & 0 \le x < 1, \\ 0 & \text{otherwise}. \end{cases} \tag{6}$$

The design of wavelet functions that provide a multiresolution analysis for any given function space is outside the scope of this paper; we instead refer readers to Daubechies (1992) for a theoretical treatment of this topic. We only mention here that the collection of translated and dilated wavelet functions $\{\psi_{j,k}(x)\}_{j,k}$ forms a basis of the function space $L^2(\mathbb{R})$ by construction, as per Daubechies (1992).

Daubechies (1988) showed that we can obtain wavelet and scaling coefficients for general wavelet functions from the general DWT, whose operations are given by

$$d_{j,k} = \sum_{n \in \mathbb{Z}} g_{n-2k} c_{j+1,n-1}, \tag{7}$$

and

$$c_{j,k} = \sum_{n \in \mathbb{Z}} h_{n-2k} c_{j+1,n-1}, \tag{8}$$

where the coefficients h_n originate from the dilation equation of the wavelet function, which is

$$\phi(x) = \sum_{n \in \mathbb{Z}} h_n \phi_{1,n}(x), \tag{9}$$

where $\phi_{1,n}$ denotes the scaling function that forms a basis for the next-finer scale resolution space, and

$$g_n = (-1)^n h_{1-n}. \tag{10}$$

Adopting the more concise vector notation of Nason and Silverman (1995), we can rewrite (7) and (8) as the filtering operations

$$\mathbf{d}_j = \mathcal{D}_0 \mathcal{G} \mathbf{c}_{j+1} \tag{11}$$

and

$$\mathbf{c}_j = \mathcal{D}_0 \mathcal{H} \mathbf{c}_{j+1} \tag{12}$$

respectively, where $\mathbf{d}_j = \left(d_{1,j}, ..., d_{T/2^{J-j},j}\right)^T$, $\mathbf{c}_j = \left(c_{1,j}, ..., c_{T/2^{J-j},j}\right)^T$, $\mathcal{D}_0$ denotes the even dyadic decimation operator defined by $(\mathcal{D}_0 \mathbf{x})_i = x_{2i}$ for $\mathbf{x} = (x_1, ..., x_N)^T$, $\mathcal{G}$ denotes the filtering operation using $\{g_n\}$ and $\mathcal{H}$ denotes the filtering operation with $\{h_n\}$.

We conclude the introduction of the general wavelet transform by drawing attention to the issue of computing coefficients when the filter operations $\mathcal{G}$ and $\mathcal{H}$ extend beyond the available time series observations — the so-called 'boundary problem' described in Chapter 2.8 of Nason (2008). As the choice of solution is extremely important for the design of an online algorithm to compute the wavelet and scaling coefficients at all levels, we refer readers to Section 3 for a detailed discussion of this topic.

2.2 Non-decimated wavelet transforms

The non-decimated wavelet transform (NDWT) differs from the standard DWT by applying both odd and even dyadic decimations to a given sequence, see Nason and Silverman (1995) or Coifman and Donoho (1995). More precisely, given a vector of observations $\mathbf{y} = (y_1, ..., y_T)^T$ from our time series $\{Y_t\}$, the NDWT keeps the wavelet coefficients from both $\mathcal{D}_0 \mathcal{G} \mathbf{y}$ and $\mathcal{D}_1 \mathcal{G} \mathbf{y}$, where $\mathcal{D}_1$ denotes the odd dyadic decimation operator defined by $(\mathcal{D}_1 \mathbf{x})_i = x_{2i-1}$. The scaling coefficients are similarly obtained from $\mathcal{D}_0 \mathcal{H} \mathbf{y}$ and $\mathcal{D}_1 \mathcal{H} \mathbf{y}$. To then compute the next coarser-scale set of wavelet coefficients, $\mathcal{D}_0 \mathcal{G}$ and $\mathcal{D}_1 \mathcal{G}$ are applied to both these sets of scaling coefficients. Repeating these operations for all J scales results in a total of JT coefficients. The wavelet coefficient vector at every scale has the same length as the original time series, which can be useful when computing predictions at a specific time index. Given that $T = 2^J$, the time complexity of the NDWT is $\mathcal{O}(T \log_2 T)$; not much more intensive than the DWT for large T. A given permutation of choices of $\mathcal{D}_0$ and $\mathcal{D}_1$ at each level characterises a basis; the collection of all such permutations forms a particular library of bases. This library is extended further with the wavelet packet transforms described in the following section.

The reader may wonder what is gained by these extra computations, when no information is lost with the standard DWT; that is, the original sequence can be perfectly recovered from the wavelet coefficients and coarsest scaling coefficient of the DWT. The most obvious advantage is the fact that, because a wavelet coefficient can be found for each scale at each time point, the resulting coefficient vectors will be the same length as the original signal, allowing us to directly treat these vectors as time series regressors. Another key benefit of the NDWT is translation equivariance: that applying a shift operator $\mathcal{S}$ to $\mathbf{y}$ before applying the NDWT, where $(\mathcal{S}\mathbf{x})_i = x_{i+1}$, would result in the output of the NDWT on the original sequence shifted by one position. Nason (2008) suggest that the NDWT is superior to the DWT for time series analysis due to the improved retention of features corresponding to oscillations at lower frequencies. Outperformance of the NDWT compared to the DWT has also been demonstrated in several practical applications including electrocardiogram data denoising (Raj and Venkateswarlu (2011)) and image denoising (Gyaourova et al. (2002)).

2.3 Wavelet packet and non-decimated wavelet packet transforms

Wavelet packet transforms (WPT) involve the application of the $\mathcal{G}$ and $\mathcal{H}$ filters to both the wavelet and scaling coefficients of the next-finer scale, rather than just the scaling coefficients

as in the standard DWT, resulting in function bases that contain additional oscillations compared to the wavelet basis functions that characterise those filters. Coifman and Wickerhauser (1992) define a sequence of functions $\{W_n\}_{n\in\mathbb{Z}}$ according to the set of recursive equations

$$W_{2n}(x) = \sqrt{2}\sum_k h_k W_n(2x-k) \tag{13}$$

$$W_{2n+1}(x) = \sqrt{2}\sum_k g_k W_n(2x-k), \tag{14}$$

where $W_0(x) = \phi(x)$ and $W_1(x) = \psi(x)$. The library of wavelet packet bases is defined by Coifman and Wickerhauser (1992) to be the collection of orthonormal bases generated by functions of the form $W_n\left(2^j x - k\right)$, where j, k and n are integers and $n \geq 0$ approximately equals the number of oscillations in the function. Like the NDWT, the wavelet packet transform for a length T sequence for a fixed selection of a basis can be computed using $\mathcal{O}(T\log_2 T)$ operations. Coifman and Wickerhauser (1992) also propose an algorithm for selecting a 'best basis' from the library of wavelet packets, which they define as the basis that minimises the Shannon entropy of the vector of wavelet coefficients, hence favouring sparsity in the representation of the signal. Their 'best basis algorithm' starts from the finest scale, selecting the basis at that scale that minimises entropy. This is repeated until some given maximum scale is reached, resulting in a best basis at each scale.

As with the standard DWT, the non-decimated wavelet packet transform (NWPT) introduced by Nason et al. (1997) involves the application of both even and odd decimation operators to the coefficients at each scale. Cardinali and Nason (2018) demonstrate the utility of using wavelet packet basis libraries (rather than a single wavelet or Fourier basis) to detect nonstationarities in locally stationary processes, where avoiding decimation ensures there are no implicit gaps in the analysis where changes in the underlying process should take place. Finally, when treating wavelet coefficients as features in regression or forecasting problems, it is again convenient to have wavelet packet coefficients vectors of the same length as the original time series, which would not be the case for the decimated wavelet packet transform.

2.4 Forecasting time series with wavelets

In practice, a key advantage of analysing time series with wavelets rather than with Fourier methods is that wavelets, which have finite support, can capture local information from non-stationary time series. Moreover, this analysis is performed at multiple scales simultaneously. This allows wavelets to capture some seasonality that has time-varying impact without any additional assumptions regarding the structure of the seasonality, such as the need to specify observation frequency. For example, if a seasonal pattern exists in the data, it will manifest as a recurring pattern in the wavelet coefficients at a corresponding scale, but a trend in these coefficients will reflect changes in the influence of seasonality over time.

Furthermore, it can be shown that the wavelet coefficients obtained by the DWT no longer contain long-term dependencies that are present in the original time series under certain weak assumptions, which is referred to as the 'decorrelating' property of the wavelet transform (see Johnstone and Silverman (1997), Soltani et al. (2000)).

Finally, as previously mentioned, there exist fast algorithms for the computation of the NDWT and NWPT, allowing us to quickly generate time series features of the same length as the original time series. This section will provide a brief overview of extant forecasting methods that utilise wavelet analysis.

Nason and Sapatinas (2002) used the NWPT to predict the wind speeds at one geographical location, represented by $\{Y_t\}$, by wind speeds at another location, represented by $\{X_t\}$. They first applying the NWPT to $\{X_t\}$, resulting in a large $2T-2$ set of length T coefficient vectors that are treated as candidate regressors. In order to reduce the dimension of the input space, the authors proposed selecting some small subset of the regressors that have strongest correlation to $\{Y_t\}$ to use as inputs to their regression models, and further reducing complexity by the use of backwards variable selection.

Wong et al. (2003) suggest decomposing a nonstationary exchange rate time series into a trend component and irregular component. The trend component is obtained by the application of a wavelet-based filter and forecasts are generated by extrapolating a polynomial function of time fitted to the trend. Conejo et al. (2005) also decompose nonstationary time series using the DWT, but instead fit ARIMA models to each component wavelet coefficient vector and a scaling coefficient vector. The inverse DWT is then applied to the the ARIMA forecasts for each component.

Schlüter and Deuschle (2010) tested several different wavelet-based approaches. One such method involves using the DWT to first denoise the target time series, before forecasting the denoised time series with autoregressive integrated moving average (ARIMA) models. They use hard thresholding, which involve setting any wavelet coefficients from the DWT below a given threshold to zero, followed by performing the inverse DWT to return the denoised series. Readers are referred to Donoho and Johnstone (1994) for a discussion of appropriate threshold levels. Other methods included the decomposition approach of Conejo et al. (2005) and modelling the time series as locally stationary wavelet processes, as introduced in Nason et al. (2000). Schlüter and Deuschle (2010) conclude that classical time series forecasting methods like ARIMA may be improved by including an initial wavelet transform step.

Wang and Guo (2020) also propose hybrid forecasting methods combining wavelet analysis with classical time series modelling. The authors also use the DWT to decompose the time series and forecast the denoised component using an ARIMA model. However, the error component is instead forecasted using the XGBoost algorithm, a highly efficient implementation of the gradient boosted decision trees introduced by Chen and Guestrin (2016).

Finally, Adjoumani (2021) also utilise the XGBoost algorithm in a hybrid approach, but instead use it in the final forecasting step. Firstly, the Haar NDWT or NWPT are performed on the target time series to obtain a high-dimensional set of time series regressors. These features, which may themselves be denoised, are then used as inputs for the XGBoost algorithm to obtain direct forecasts of the target time series.

At this point, one may wonder how a wavelet function is selected in the first place. Nason (2002) suggests that cross-validation can be used to determine a suitable smoothness for the wavelet function. Nunes et al. (2006) instead propose an adaptive lifting scheme similar to wavelet decomposition that would allow for time-varying smoothness of the denoised time series, resulting in attractive compression properties and dispensing with the need to directly select a wavelet function at all. Their scheme is based on Jansen et al. (2004) and involves 'lifting one coefficient at a time', where for a given scale, scaling coefficients are one-by-one predicted using 'neighbouring' scaling coefficients. Points may be classified as neighbours based on distance, with prediction performed using polynomial regression up to order 3. The residuals from these predictions are analogous to the detail coefficients from classical wavelet analysis.

2.5 Machine learning methods

Table 1 provides a summary of the machine learning techniques to be implemented as part of our wavelet-ML framework. We consider two categories of methods: non-temporal methods that treat all lags of a given input time series simply as a set of unordered features, and temporal methods that do take time-order of the inputs into account. We now provide a brief overview of each method and previous examples of their use in time series forecasting problems. In this section, we will introduce the temporal, deep learning-based methods, including several state-of-the-art architectures.

Non-temporal	**Temporal, Statistical**	**Temporal, Deep Learning-Based**
Ridge regression	Persistence	RNN
Support vector regression	ARIMA	GRU
Random forests	Exponential smoothing	LSTM
XGBoost	Theta	Dilated LSTM
Multilayer Perceptrons		TCN
		TFT
		Informer
		Autoformer
		PatchTST

Table 1: Machine learning methods evaluated for the wavelet-ML framework

Recurrent Neural Networks (RNNs), first introduced in Rumelhart et al. (1986) and popularised by Elman (1990), are a type of artificial neural network specifically designed to recognise patterns in arbitrarily long sequences of data. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs, making them effective for tasks where context and historical information play a critical role. As a result, even simple RNNs have been wide applied in both univariate and multivariate forecasting problems. Early examples include Kuan and Liu (1995) and Vermaak and Botha (1998).

However, simple RNNs suffer from the 'vanishing gradient problem', where gradients often get smaller and smaller as they are propagated backwards through time, presenting a problem for modelling long-term dependencies (Bengio et al. (1994)). This issue led to the development of improved versions of RNNs such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks, which introduce and develop the concept of gates to control the flow of information and memory. Readers are referred to Hochreiter and Schmidhuber (1997) and Cho et al. (2014) for more details on these architectures. A further extension is the dilated LSTM network (Chang et al. (2017)), where each layer contains dilated recurrent skip connections at different scales, enabling more efficient learning of long-term patterns in the input.

Dilated Temporal Convolutional Networks (TCNs), introduced by Lea et al. (2017), offer a useful alternative to RNNs for time series forecasting tasks. Standard Convolutional Neural Networks (CNNs) involve convolutional layers that apply sliding filters to capture local spatial hierarchies in the input data. In TCNs, causal convolution operations ensure outputs of each layer depend only on past and present input values at various time scales. TCNs are constructed by stacking multiple causal convolutional layers with exponentially increasing dilation factors, allowing the receptive field of the network to also grow exponentially. Yan et al. (2020) demonstrated the outperformance of TCNs over LSTMs for forecasting El Niño-

Southern Oscillation indices.

Since their introduction in Vaswani et al. (2017), transformer architectures are most commonly known for their great success in the field of natural language processing (Kalyan et al. (2021)). The primary innovation of Transformers is the self-attention mechanism, which infers context for each token in an input sequence by considering all other tokens in the same sequence. This context is determined by calculating scaled dot product attention weights between query and key vectors, which are derived from the original token embeddings. In the realm of time series forecasting, 'tokens' are akin to continuous-valued observations of the time series of interest, hence self-attention endows transformers with the ability to exploit diverse temporal patterns in multivariate input data, but suffer from insensitivity to local context, as well as time and space complexity quadratic in the length of the input time series (Li et al. (2019)).

The Temporal Fusion Transformer (TFT) is an extension of the Transformer specifically designed by Lim et al. (2021) for multi-horizon time series forecasting tasks, capturing dependencies at both local and long-run scales. Each input to TFT is passed through a variable selection network and multiple gating mechanisms that learn which components of the input and network architecture respectively can be ignored, also resulting in greater interpretability of the model. An additional LSTM-based sequence-to-sequence encoder-decoder structure is used for locality enhancement by generating temporal features from the input sequence.

The Informer model, proposed by Zhou et al. (2021), addresses the issue of space complexity associated with the Transformer architecture by implementing a novel self-attention mechanism known as ProbSparse self-attention. In contrast to the traditional approach, the query matrix in ProbSparse self-attention retains only the top-u queries that exhibit the highest relevance to a given key. This relevance is approximated by the max-mean of the dot-product similarity, computed from a randomly selected sample of query-key pairs of size $T\ln(T)$, where T represents the length of the query and key sequences. Zhou et al. (2021) further suggest setting $u = c\ln T$ for some constant sampling factor c, which results in significantly reduced time and space complexities of $O(L\ln L)$.

Wu et al. (2021) propose an alternative solution to the space complexity problem with the Autoformer architecture. In the Autoformer, the self-attention blocks of the vanilla Tranformer are replaced with *Auto-Correlation blocks*, which replace dot product in the self-attention mechanism with sample autocorrelation statistics. In a similar vein to the Informer, only the top $u = c\ln T$ lag orders for time series of length T and hyperparameter c are used to compute the output of the block. This again results in space complexity of $O(T\ln T)$. The output of these *Auto-Correlation blocks* are fed into *series decomposition blocks*, which explicitly decompose the outputs of the hidden layers into a seasonal component and a trend-cyclical component, with the latter being aggregated at every layer of the decoder.

Finally, the channel-independent Patch Time Series Transformer (PatchTST) recently introduced by Nie et al. (2022) aims to reduce the time and memory complexity of the vanilla Transformer, while extracting more local information, by utilising a simple 'patching design' that reduces the length of the sequence fed into the Transformer encoder by only using fixed-length segments of the time series, separated by a constant stride factor.

3 Online computation of wavelet packet features

We propose a simple online algorithm for computing the wavelet coefficient vectors based on the pyramidal algorithm, for both the NDWT and the NWPT. Recall the length of the coefficient vectors from non-decimated wavelet transforms always equal the length of the original time series. Our implementation ensures that wavelet coefficient computed at time t will never use information from data at times $t+1, t+2, ...$ etc. This is achieved by shifting the windows over which data are convolved with the wavelet filters, such that the last input to the filter is the time t finer-scale coefficient.

To avoid the separate storage of an excessive number of matrices to contain each packet of coefficients and to negate the need to 're-thread' coefficients to time-order, we derive new equations to compute the time-ordered wavelet coefficients. For the NDWT, the time-ordered shifted wavelet and scaling coefficients can be computed using

$$d_{j,t} = \sum_{n=0}^{W-1} g_n c_{j+1,2^{J-j-1}(n-W+1)+t}, \tag{15}$$

$$c_{j,t} = \sum_{n=0}^{W-1} h_n c_{j+1,2^{J-j-1}(n-W+1)+t}, \tag{16}$$

where

$$g_n = (-1)^n h_{1-n}, \tag{17}$$

W is the total number of wavelet filter coefficients, time scale $j \in \{0, 1, ..., J-1\}$ and time index $t \in 1, ...T$ where T is the length of the input time series. Similarly, the time-ordered NWPT coefficients are computed using the equations

$$p_{j,2l,t} = \sqrt{2} \sum_{n=0}^{W-1} h_n p_{j+1,l,2^{J-j-1}(n-W+1)+t}, \tag{18}$$

$$p_{j,2l+1,t} = \sqrt{2} \sum_{n=0}^{W-1} g_n p_{j+1,l,2^{J-j-1}(n-W+1)+t}, \tag{19}$$

for packet index $l \in \left\{0, 1, ..., 2^{J-j} - 1\right\}$. At each time step starting from $t = 1$, all NDWT and NWPT coefficients are computed for all scales using the above equations.

Where the computation requires inputs that do not exist ($t \leq 0$), we impute the missing values as the first available coefficient at that scale, referred to as constant-end extension. Using constant-end extension rather than symmetric-end or periodic reflection allows us to compute coefficients in a single pass, rather than requiring the updating of previously computed coefficients when more data become available. In practical situations, it may not be possible to update forecasts for a given time period as new data become available, such as when one-step-ahead forecasts are used immediately for decision making.

Given that the algorithm sequentially proceeds from finer to coarser scales, there will always be at least one non-missing value to compute any given wavelet coefficient. The NDWT and NWPT online pyramidal algorithms are illustrated in Figures 1 and 2 respectively.

The resulting NWDT and NWPT coefficient vectors can then be treated as features for downstream time series forecasting tasks, although dimension reduction methods may be

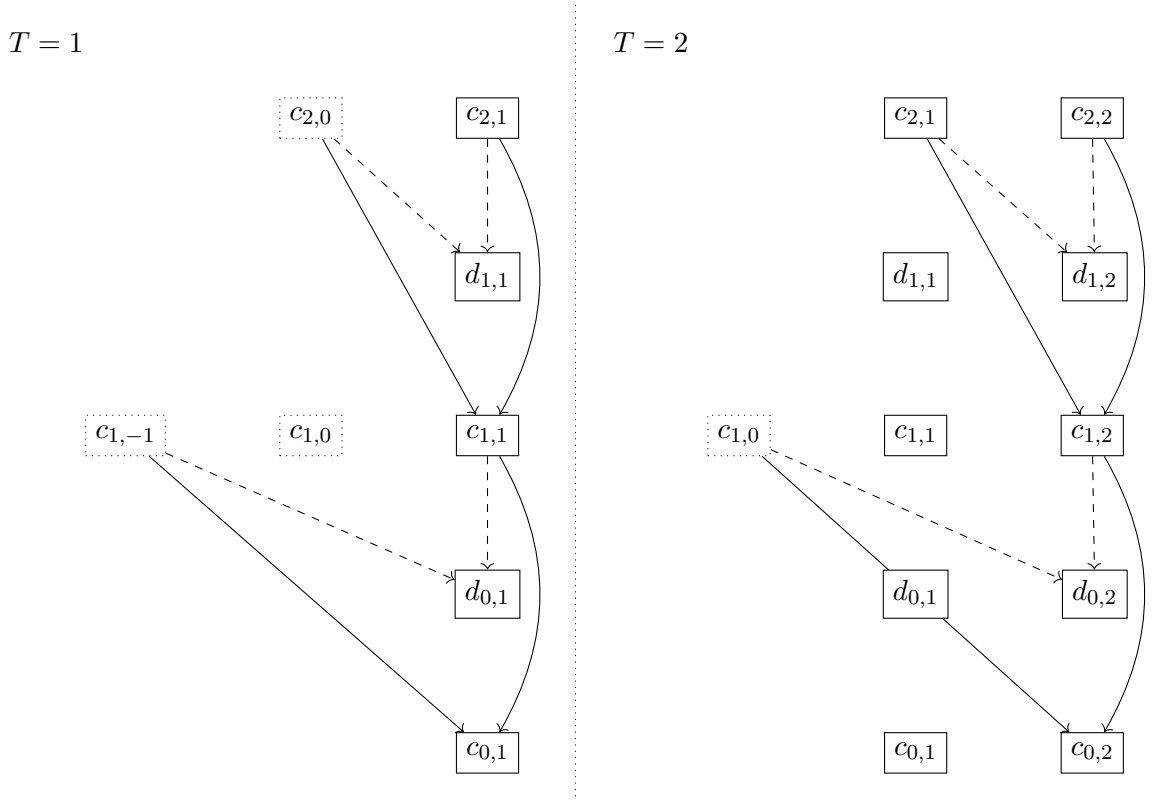

Figure 1: Illustration of online NDWT pyramidal algorithm for $J = 3$, $W = 2$. Input time series given by top row of coefficients. Coefficients with dotted borders obtained by constant-end extension. Dashed arrows denote filtering operations with $\mathcal{G}$, solid arrows denote filtering operations with $\mathcal{H}$.

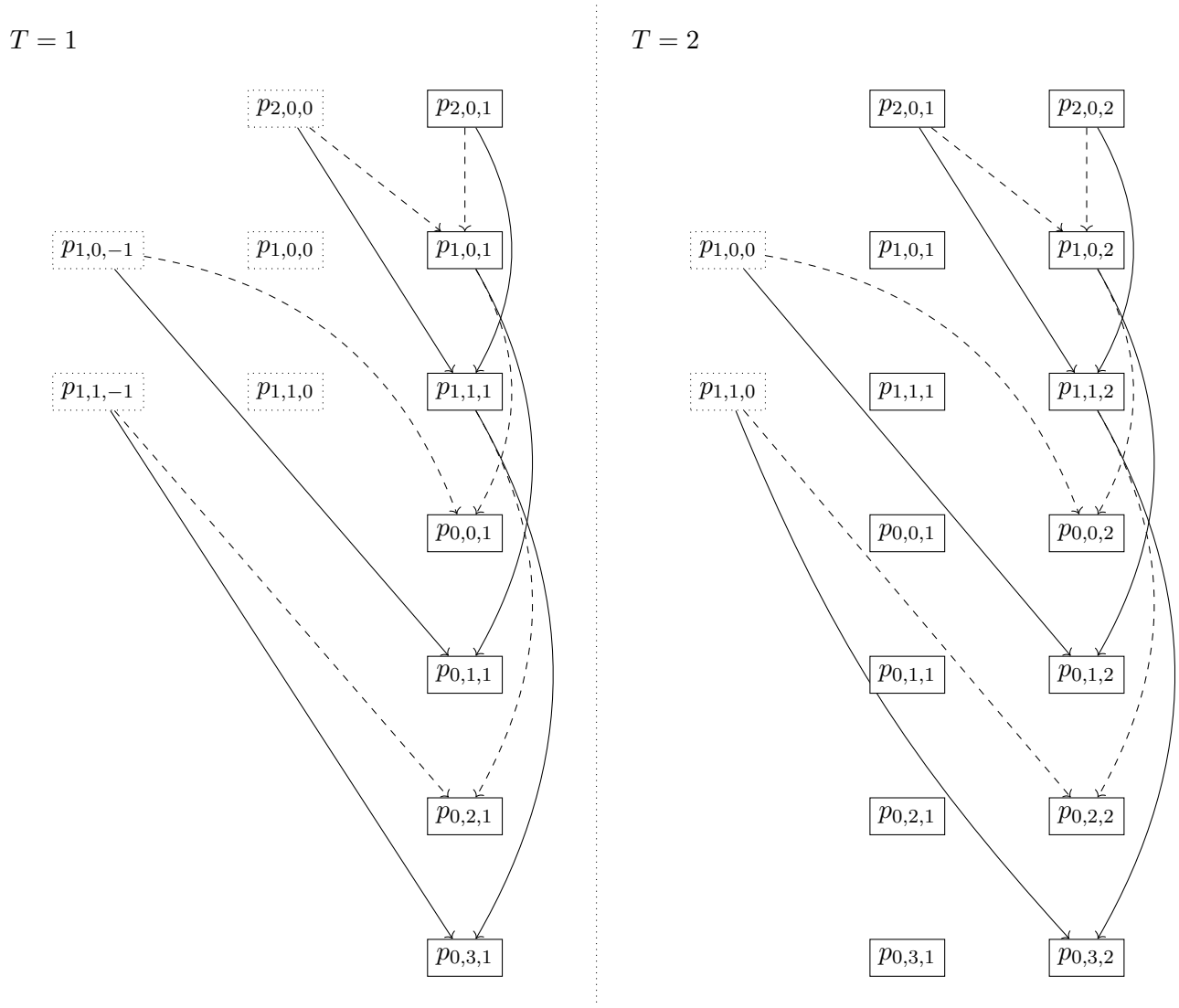

Figure 2: Illustration of online NWPT pyramidal algorithm for $J = 3$, $W = 2$. Input time series given by top row of coefficients. Coefficients with dotted borders obtained by constant-end extension. Dashed arrows denote filtering operations with $\mathcal{H}$, solid arrows denote filtering operations with $\mathcal{G}$.

required to avoid overfitting. In Section 4, we examine the use of principal components analysis or regularised regression to select NWPT coefficient vectors.

Finally, we use a simple example to show that online wavelet denoising with thresholding using our algorithm is only feasible for Haar wavelets. Let $J = 3$ and $W = 4$ (recall the Haar wavelet transform corresponds to $W = 2$). The forward transform is represented by the linear system

$$\begin{bmatrix} c_{j-1,t-2\times 2^{J-j}} \\ d_{j-1,t-2\times 2^{J-j}} \\ c_{j-1,t-2^{J-j}} \\ d_{j-1,t-2^{J-j}} \\ c_{j-1,t} \\ d_{j-1,t} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & 0 \\ 0 & g_0 & g_1 & g_2 & g_3 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{bmatrix} \times \begin{bmatrix} c_{j,t-5\times 2^{J-j}} \\ c_{j,t-4\times 2^{J-j}} \\ c_{j,t-3\times 2^{J-j}} \\ c_{j,t-2\times 2^{J-j}} \\ c_{j,t-2^{J-j}} \\ c_{j,t} \end{bmatrix}. \tag{20}$$

The transformation matrix has determinant below one, hence the inverse matrix, corresponding to the inverse transform, has determinant above one. In practice, this means that if we apply thresholding, the scaling coefficients we obtain from the inverse transform explode in size. If we instead modify the forward transform so that the rows of the trasnformation matrix are orthonormal, at time $t = 1$ and scale $j = 1$ we have

$$\begin{bmatrix} c_{0,1} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \\ g_0 & g_1 & g_2 & g_3 \end{bmatrix} \times \begin{bmatrix} c_{1,-11} \\ c_{1,-7} \\ c_{1,-3} \\ c_{1,1} \end{bmatrix}. \tag{21}$$

In this case, the transformation matrix is not invertible. The solution would be to set $c_{1,-11} = c_{1,-7} = 0$, so that we can rewrite the equation as

$$\begin{bmatrix} c_{0,1} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} h_2 & h_3 \\ g_2 & g_3 \end{bmatrix} \times \begin{bmatrix} c_{1,-3} \\ c_{1,1} \end{bmatrix}. \tag{22}$$

In the above, we have set half of the scaling coefficients on the RHS to equal zero. If we do not want to set such a high proportion of coefficients to zero, we can extend the matrix so that the transformation makes use of more coefficients (note that the blocks are shifted by two columns rather than one column in the first equation to ensure orthogonality):

$$\begin{bmatrix} c_{0,-7} \\ d_{0,-7} \\ c_{0,1} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ g_0 & g_1 & g_2 & g_3 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 \end{bmatrix} \times \begin{bmatrix} c_{1,-19} \\ c_{1,-15} \\ c_{1,-11} \\ c_{1,-7} \\ c_{1,-3} \\ c_{1,1} \end{bmatrix}. \tag{23}$$

Similarly, by setting $c_{1,-19} = c_{1,-15} = 0$, we can again simplify the system so that the transformation matrix is orthogonal:

$$\begin{bmatrix} c_{0,-7} \\ d_{0,-7} \\ c_{0,1} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} h_2 & h_3 & 0 & 0 \\ g_2 & g_3 & 0 & 0 \\ h_0 & h_1 & h_2 & h_3 \\ g_0 & g_1 & g_2 & g_3 \end{bmatrix} \times \begin{bmatrix} c_{1,-11} \\ c_{1,-7} \\ c_{1,-3} \\ c_{1,1} \end{bmatrix}. \tag{24}$$

The reader may notice a problem: the first two scaling coefficients on the RHS of the original system must always be set to zero to obtain an orthogonal transformation matrix. This would not be a problem for offline wavelet transforms, since we can just create a matrix to transform the entire input time series at once and arbitrarily set the first two coefficients to equal zero (or be a symmetric extension of the first two scaling coefficients). But for online wavelet transforms, this approach is not possible. As the 'window' of scaling coefficients on the RHS shifts during the operation of the online algorithm, we would need to set different coefficients equal to zero to obtain the orthogonal transformation matrix.

Of course, in practice we do not use matrix multiplication to obtain the coefficients in the forward transform. But if we want to obtain a formula for the finer coefficients using the inverse transform in the process of wavelet denoising, we would need to solve the above system, which requires us to incorrectly set the first two scaling coefficients on the RHS to zero. Hence thresholding is not possible using the online algorithm for wavelet functions with more than two filter coefficients.

Readers that are primarily interested in analysis rather than forecasting applications, and therefore do not require online computation of coefficients, are referred to the *wavethresh* package in **R** (Nason et al. (2016)), which contains a comprehensive suite of options for both non-decimated wavelet transforms and non-decimated wavelet packet transforms.

4 Experiments

4.1 Datasets

Our data consist of simulated time series, energy supply time series and meteorological time series. Each of these three groups contains three time series of length $100,000$, split into ten contiguous segments of equal length, resulting in a total of 90 time series. The length of our samples has been chosen so that our experiments can feasibly be replicated on a personal desktop computer. For example, the entire experiment described in Section 4.2.1 takes approximately 140 hours to run on a computing setup featuring an Intel i9-9920X processor with 24 cores operating at 3.50GHz, coupled with a NVIDIA GeForce RTX 3080 Ti graphics processing unit.

Our energy data consists of time series relating to UK National Grid electricity supply (Elexon (2022)), all observed at 5-minute intervals from October 2020. These include 1) total electricity demand in MW, which equals the sum of the electricity generated from all sources, 2) electricity generation in MW from non-pumped storage hydropower plants, and 3) electricity generation in MW from wind power.

Meteorological time series consist of hourly measurements taken at the weather station at Heathrow, UK, from January 1950 (Centre for Environmental Data Analysis (2022)). These three time series include: 1) relative humidity, 2) air temperature in degrees Celsius, and 3) wind speed in knots.

Simulated time series follow the bumps, Doppler and heavisine functions described in Donoho and Johnstone (1994), generated using the *wavethresh* **R** package (Nason et al. (2016)). The bumps function is characterised by localised step changes over time around zero, allowing us to assess which methods fit to noise. The Doppler function creates a harmonic time series with frequency decreasing over time. Finally, the heavisine function is the sum of a sine wave and a step function that introduces discontinuity at given intervals. Examples of these functions are given in Figure 3.

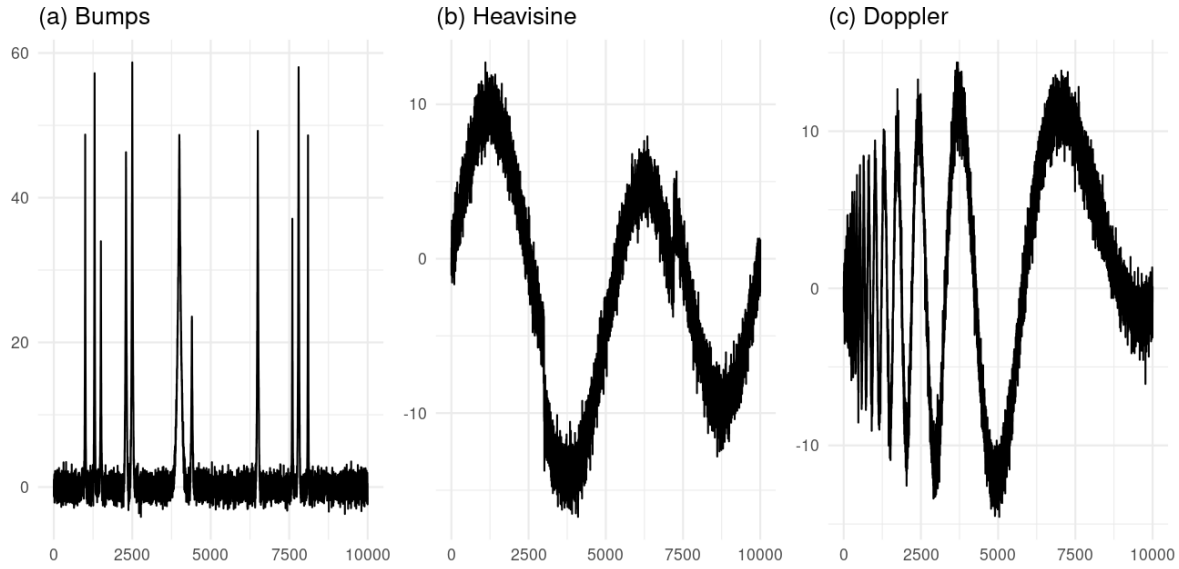

Figure 3: Examples of simulated time series corresponding to each function.

4.2 Results

4.2.1 Experiment 1: Non-temporal machine learning methods for one-step-ahead forecasts

In this experiment, the training set consists of the first 9000 observations of the time series and the models produce one-step-ahead forecasts. Out-of-sample forecasting performance is evaluated on the test set containing the remaining 1000 data points. We compare the performance for the non-temporal machine learning methods listed in 1 when using one of the following feature sets, all containing 3000 features for consistency:

1. **Lags-only**: Time series lags of up to 3000 periods, which allows the models to consider very long-run dependencies that span a significant proportion of the training set, comparable to the receptive field of the coarsest wavelet scale.

2. **NDWT**: Non-decimated wavelet coefficient vectors for wavelet numbers between 0 and 10, up to a scale of $J = 13$, resulting in $J + 1 = 14$ coefficient vectors (one for each scale plus the original time series). For each coefficient vector, we create features using up to 215 lags ($14 \times 215 = 3010$ features), then using ridge regression to select for the most promising 3000 features by selecting those corresponding to the largest regression coefficients.

3. **NWPT**: Non-decimated wavelet packet coefficient vectors for wavelet numbers between 0 and 10, up to a scale of $J = 13$, resulting in $2^0 + 2^1 + ... + 2^J = 2^{J+1} - 1 = 16,384$ coefficient vectors, then using ridge regression to select for the most promising 3000 features by selecting those corresponding to the largest regression coefficients.

Models are tuned using two-fold cross-validation for 10 random samples from a predefined search space of hyperparameters to reduce computational burden. Details of the sampled hyperparameter sets are provided in Section C. Before fitting the models, input variables and

the target variable are normalised by subtracting their mean and dividing by their standard deviation. For the NDWT and NWPT feature sets, an additional cross-validation step is taken to choose the best wavelet number based on out-of-sample symmetric mean absolute percentage error (SMAPE) for the last 1000 observations of the training set. We use the following definition for SMAPE:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{T}\frac{|\hat{y}_t - y_t|}{(|\hat{y}_t| - |y_t|)/2}, \tag{25}$$

where $\hat{y}_t$ denotes the forecast values and y_t are the actual values. For terms where the denominator equals zero, that term is set to zero.

Table 2 shows mean SMAPE of the one-step-ahead out-of-sample forecasts across all 90 time series, as well as the modal wavelet numbers selected by cross-validation. We find that using the NDWT feature set outperforms using only lags for all five machine learning methods examined, including a 11% reduction in SMAPE for XGBoost models and 31% reduction in SMAPE for MLP models. The NWPT feature set also outperforms using only lags for four of the five examined methods, although by smaller margins on average. The most commonly selected wavelet number was 1 for both NDWT and NWPT feature sets. More granular results for each of the nine categories of time series data are available in Section A, where we find that models trained on NDWT or NWPT feature sets are superior in 42 of 45 combinations of categories and models. In particular, we find that the use of wavelet features produces dramatic improvements in performance over multiple windows when forecasting total electricity demand or hydropower electricity supply, with the best overall models using MLP and ridge regression with NDWT features respectively.

Model	**Feature Set**	**Modal Wavelet Number**	**Mean SMAPE % (SE)**
Ridge	Lags	-	35.57 (4.87)
Ridge	**NDWT**	**1**	**33.23 (4.90)**
Ridge	NWPT	7	44.71 (5.43)
SVR	Lags	-	45.00 (4.79)
SVR	**NDWT**	**1**	**42.51 (6.37)**
SVR	NWPT	1	44.04 (5.52)
Forest	Lags	-	40.56 (5.80)
Forest	NDWT	1	38.27 (5.79)
Forest	**NWPT**	**1**	**36.94 (5.64)**
XGBoost	Lags	-	40.73 (5.49)
XGBoost	NDWT	1	36.31 (5.20)
XGBoost	**NWPT**	**1**	**36.15 (5.06)**
MLP	Lags	-	52.91 (4.98)
MLP	**NDWT**	**1**	**36.49 (5.35)**
MLP	NWPT	1	48.14 (5.48)

Table 2: Average Out-of-Sample One-Step-Ahead Forecast Performance Across All Time Series. The top-performing feature set for each model has been bolded.

4.2.2 Experiment 2: Temporal machine learning methods for long-run forecasts

In this experiment, the training set again consists of the first 9000 observations of each of the 90 time series, but the models produce 1- to 1000-step-ahead forecasts. The very long range of forecasts allows us to better exploit the multi-scale featurisation provided by wavelet analysis. Out-of-sample forecasting performance is evaluated on the test set containing the remaining 1000 data points, averaged across all 1000 forecast horizons. We report forecasting results for the temporal statistical and deep learning-based methods listed in 1. For the deep learning methods, which can all handle multivariate input, we utilise each of the following sets of input time series for a length 3000 lookback window:

1. **Univariate**: The time series of interest.
2. **NDWT**: The non-decimated wavelet coefficient vectors for wavelet numbers between 0 and 10, up to a scale of $J = 13$, resulting in $J + 1 = 14$ time series (one for each scale plus the original time series).
3. **NWPT**: Non-decimated wavelet packet coefficient vectors for wavelet numbers between 0 and 10, up to a scale of $J = 13$, resulting in $2^0 + 2^1 + ... + 2^J = 2^{J+1} - 1 = 16,384$ coefficient vectors, then using the top 13 principal components, again resulting in $J+1 = 14$ time series (one for each principal component plus the original time series).

Just as in Experiment 1, for the NDWT and NWPT feature sets, an additional cross-validation step is taken to choose the best wavelet number based on out-of-sample symmetric mean absolute percentage error (SMAPE) for the last 1000 observations of the training set.

Table 3 shows mean SMAPE of the 1- to 1000-step-ahead out-of-sample forecasts across all 90 time series, as well as the modal wavelet numbers selected by cross-validation. We find that using NDWT and NWPT multivariate inputs result in superior forecasts for seven out of nine deep learning models across our datasets compared to the corresponding univariate models, with the best method being the GRU architecture with the NWPT feature set. Most importantly, we find no consistent evidence that using the additional thirteen multi-scale features compared to the univariate approach leads to overfitting, despite including no extra information beyond the original time series. We also note that the wavelet number 1 (Haar wavelets) is not selected in the majority of cases, suggesting that wavelets of greater complexity should be considered using our cross-validation approach.

As with the non-temporal methods, more granular results for each of the nine categories of time series data are available in Section B, where we demonstrate that models using NDWT or NWPT feature sets outperform in 53 of 81 combinations of categories and temporal models. Of these, we find that wavelet features provide most benefit for wind electricity supply and humidity forecasting.

5 Conclusion

We explored the benefits of using wavelet analysis techniques combined with machine learning methods for time series forecasting problems, building on existing literature in three ways. Firstly, we investigated the the use of Daubechies wavelets with varying numbers of vanishing moments as input features into both non-temporal and temporal forecasting methods, with wavelet number selected during cross-validation. Secondly, we assessed the use of both non-decimated wavelet transform and non-decimated wavelet packet transform to compute these

Model	Feature Set	Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	69.30 (4.78)
ARIMA	Univariate	-	77.34 (6.52)
ETS	Univariate	-	70.31 (4.88)
Theta	Univariate	-	73.53 (5.36)
RNN	Univariate	-	66.93 (6.04)
RNN	**NDWT**	**1**	**65.93 (6.04)**
RNN	NWPT	1	66.71 (5.92)
GRU	Univariate	-	66.59 (6.00)
GRU	NDWT	1	64.98 (5.78)
GRU	**NWPT**	**5**	**62.88 (5.68)**
LSTM	Univariate	-	65.90 (5.98)
LSTM	**NDWT**	**1**	**65.18 (6.01)**
LSTM	NWPT	9	65.59 (5.86)
DilatedRNN	Univariate	-	67.96 (6.06)
DilatedRNN	NDWT	1	65.63 (6.13)
DilatedRNN	**NWPT**	**1**	**65.12 (5.97)**
TCN	Univariate	-	65.88 (6.00)
TCN	NDWT	1	65.52 (6.07)
TCN	**NWPT**	**1**	**63.78 (5.73)**
TFT	Univariate	-	70.74 (6.02)
TFT	NDWT	1	69.70 (5.87)
TFT	**NWPT**	**3**	**68.99 (5.92)**
Informer	**Univariate**	**-**	**118.63 (6.54)**
Informer	NDWT	1	120.69 (6.80)
Informer	NWPT	9	159.99 (5.92)
Autoformer	**Univariate**	**-**	**77.71 (6.34)**
Autoformer	NDWT	1	86.73 (6.59)
Autoformer	NWPT	1	151.39 (7.33)
PatchTST	Univariate	-	81.93 (6.09)
PatchTST	**NDWT**	**8**	**79.09 (6.27)**
PatchTST	NWPT	9	89.82 (6.60)

Table 3: Average Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance Across All Time Series with Lookback Period of Length 3000. The top-performing feature set for each model has been bolded.

features, using a shifted version of the pyramidal algorithm to ensure no future information leakage into these inputs. Lastly, these wavelet features were evaluated on a broad array of forecasting methods, encompassing temporal and non-temporal models, statistical and deep learning-based methods. These included state-of-the-art transformer-based neural network architectures.

Our results demonstrate a significant advantage to replacing higher order lagged features with wavelet features across all examined non-temporal methods for one-step-forward forecasting. In the case of temporal deep learning-based models for long-horizon forecasting, the addition of wavelet coefficient features shows modest benefit for the majority of example time

series, and relatively larger performance gains across most models for wind electricity supply and humidity forecasting. Therefore, we suggest researchers consider computing wavelet features for all time series forecasting tasks, rather than only using lagged features, even for models with recurrent architectures.

Further research would be needed to evaluate the effectiveness of different selection methods across coefficient vectors of all wavelet numbers, rather than selecting a specific wavelet number during cross validation. Moreover, a detailed comparison between performance on the original time series and deseasonalised time series is warranted, to assess the proportion of performance gains of wavelet features on non-temporal methods that can be attributed to the seasonality captured by wavelets of different scales.

Appendix

A Experiment 1 results for individual time series categories

See Tables 4-12 for Experiment 1 results for each category of time series, for each model, for each feature set.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	0.71 (0.08)
Ridge	**NDWT**	**1**	**0.37 (0.02)**
Ridge	NWPT	7	1.35 (0.32)
SVR	Lags	-	3.87 (0.65)
SVR	**NDWT**	**1**	**0.54 (0.11)**
SVR	NWPT	7	1.04 (0.08)
Forest	Lags	-	0.54 (0.05)
Forest	**NDWT**	**1**	**0.45 (0.05)**
Forest	NWPT	2	0.52 (0.08)
XGBoost	Lags	-	0.99 (0.11)
XGBoost	**NDWT**	**1**	**0.75 (0.10)**
XGBoost	NWPT	3	0.92 (0.14)
MLP	Lags	-	6.06 (0.80)
MLP	**NDWT**	**1**	**0.36 (0.02)**
MLP	NWPT	4	2.54 (1.03)

Table 4: UK Total Electricity Demand Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	5.05 (0.85)
Ridge	**NDWT**	**1**	**2.65 (0.32)**
Ridge	NWPT	2	9.49 (1.45)
SVR	Lags	-	24.60 (6.07)
SVR	**NDWT**	**1**	**2.68 (0.30)**
SVR	NWPT	9	16.26 (4.09)
Forest	Lags	-	4.32 (0.87)
Forest	NDWT	5	3.37 (0.36)
Forest	**NWPT**	**7**	**3.14 (0.33)**
XGBoost	Lags	-	11.20 (2.73)
XGBoost	NDWT	1	6.17 (1.05)
XGBoost	**NWPT**	**2**	**5.69 (0.70)**
MLP	Lags	-	33.94 (3.24)
MLP	**NDWT**	**2**	**3.75 (0.88)**
MLP	NWPT	7	22.76 (6.56)

Table 5: UK Hydropower Electricity Supply Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	2.40 (0.23)
Ridge	**NDWT**	**1**	**0.83 (0.10)**
Ridge	NWPT	7	4.18 (1.82)
SVR	Lags	-	16.80 (3.81)
SVR	**NDWT**	**1**	**1.07 (0.10)**
SVR	NWPT	9	5.33 (1.58)
Forest	**Lags**	**-**	**1.11 (0.14)**
Forest	NDWT	1	1.54 (0.63)
Forest	NWPT	10	1.15 (0.16)
XGBoost	Lags	-	4.00 (0.85)
XGBoost	NDWT	1	2.98 (0.62)
XGBoost	**NWPT**	**2**	**2.34 (0.55)**
MLP	Lags	-	54.16 (11.43)
MLP	**NDWT**	**1**	**4.28 (2.46)**
MLP	NWPT	7	12.11 (3.92)

Table 6: UK Wind Electricity Supply Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	6.73 (0.90)
Ridge	**NDWT**	**1**	**5.18 (0.83)**
Ridge	NWPT	7	9.80 (0.84)
SVR	Lags	-	14.71 (2.03)
SVR	**NDWT**	**1**	**5.33 (0.84)**
SVR	NWPT	1	9.64 (1.06)
Forest	Lags	-	5.27 (0.73)
Forest	**NDWT**	**1**	**5.07 (0.69)**
Forest	NWPT	1	5.17 (0.70)
XGBoost	Lags	-	6.64 (0.86)
XGBoost	NDWT	1	6.01 (0.84)
XGBoost	**NWPT**	**1**	**5.99 (0.83)**
MLP	Lags	-	20.26 (2.60)
MLP	**NDWT**	**1**	**5.43 (0.83)**
MLP	NWPT	1	11.29 (1.28)

Table 7: Heathrow Relative Humidity Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

B Experiment 2 results for individual time series

See Tables 13-21 for Experiment 2 results for each time series, for each model, for each feature set. SMAPE is computed using the mean prediction error for the ten length-10,000 contiguous segments, while the SE is the standard error of those means.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	10.65 (2.73)
Ridge	**NDWT**	**1**	**8.15 (2.25)**
Ridge	NWPT	7	15.87 (3.84)
SVR	Lags	-	32.93 (7.01)
SVR	**NDWT**	**1**	**9.58 (2.64)**
SVR	NWPT	9	21.35 (4.56)
Forest	Lags	-	9.85 (2.43)
Forest	**NDWT**	**1**	**8.41 (2.29)**
Forest	NWPT	1	8.66 (2.31)
XGBoost	Lags	-	15.42 (3.35)
XGBoost	**NDWT**	**1**	**11.03 (3.18)**
XGBoost	NWPT	1	11.15 (3.04)
MLP	Lags	-	48.08 (21.13)
MLP	**NDWT**	**1**	**8.31 (2.27)**
MLP	NWPT	4	20.53 (5.13)

Table 8: Heathrow Temperature Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	35.90 (3.86)
Ridge	**NDWT**	**1**	**26.62 (3.15)**
Ridge	NWPT	7	49.81 (4.44)
SVR	Lags	-	41.24 (2.95)
SVR	**NDWT**	**1**	**28.50 (3.42)**
SVR	NWPT	1	29.19 (3.16)
Forest	Lags	-	26.68 (2.90)
Forest	**NDWT**	**6**	**26.40 (2.87)**
Forest	NWPT	1	26.58 (3.05)
XGBoost	Lags	-	27.14 (2.94)
XGBoost	**NDWT**	**1**	**26.44 (2.94)**
XGBoost	NWPT	1	26.49 (2.96)
MLP	Lags	-	50.70 (3.82)
MLP	**NDWT**	**1**	**26.89 (3.19)**
MLP	NWPT	1	36.51 (3.44)

Table 9: Heathrow Wind Speed Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

C Model settings

The hyperparameter search spaces for each non-temporal model are as follows (note that if hidden size is a scalar, the neural network has only a single hidden layer):

1. **Ridge Regression**. Regularisation parameter (alpha): 1/32, 1/16, 1/8, 1/4, 1/2, 1, 2, 4, 8, 16, 32.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	144.28 (0.56)
Ridge	**NDWT**	**4**	**143.83 (0.85)**
Ridge	NWPT	4	150.58 (0.88)
SVR	Lags	-	151.88 (1.01)
SVR	**NDWT**	**4**	**145.18 (1.08)**
SVR	NWPT	4	155.97 (1.83)
Forest	**Lags**	**-**	**162.64 (1.72)**
Forest	NDWT	2	163.30 (0.95)
Forest	NWPT	5	166.53 (0.97)
XGBoost	Lags	-	157.01 (1.01)
XGBoost	NDWT	10	155.43 (1.79)
XGBoost	**NWPT**	**2**	**144.73 (0.87)**
MLP	Lags	-	147.54 (6.30)
MLP	**NDWT**	**1**	**144.76 (0.66)**
MLP	NWPT	1	147.73 (1.29)

Table 10: Simulated Bumps Data Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	82.85 (0.65)
Ridge	**NDWT**	**6**	**81.53 (0.99)**
Ridge	NWPT	7	116.79 (1.93)
SVR	**Lags**	**-**	**85.69 (3.78)**
SVR	NDWT	3	158.58 (7.35)
SVR	NWPT	8	116.07 (4.20)
Forest	Lags	-	106.88 (7.82)
Forest	NDWT	9	105.42 (6.28)
Forest	**NWPT**	**5**	**90.06 (1.71)**
XGBoost	Lags	-	99.65 (10.73)
XGBoost	**NDWT**	**1**	**84.31 (0.50)**
XGBoost	NWPT	1	94.84 (3.30)
MLP	**Lags**	**-**	**83.99 (0.77)**
MLP	NDWT	1	105.26 (10.12)
MLP	NWPT	7	130.30 (4.39)

Table 11: Simulated Doppler Data Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

2. **Support Vector Regression**. Kernel: linear, polynomial, radial basis function, sigmoid. Regularisation parameter (C): $0.125, 0.25, 0.5, 1$. Error sensitivity (epsilon): $0.025, 0.05, 0.1, 0.2, 0.4$.

3. **Random Forest**. Number of trees: $5, 10, 20, 40$. Minimum number of samples to split: $8, 16, 32$. Minimum number of samples required at each leaf node: $4, 8, 16$. Maximum number of features to consider when splitting a node: all, square root of total, base 2

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Ridge	Lags	-	31.53 (0.23)
Ridge	**NDWT**	**1**	**29.92 (0.36)**
Ridge	NWPT	7	44.57 (0.65)
SVR	Lags	-	33.27 (0.23)
SVR	**NDWT**	**1**	**31.13 (0.29)**
SVR	NWPT	1	41.55 (0.35)
Forest	Lags	-	47.78 (0.54)
Forest	**NDWT**	**7**	**30.41 (0.34)**
Forest	NWPT	2	30.63 (0.44)
XGBoost	Lags	-	44.49 (1.39)
XGBoost	NDWT	3	33.68 (0.80)
XGBoost	**NWPT**	**4**	**33.23 (0.87)**
MLP	Lags	-	31.42 (0.30)
MLP	**NDWT**	**1**	**29.41 (0.35)**
MLP	NWPT	1	49.51 (2.75)

Table 12: Simulated Heavisine Data Out-of-Sample One-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

log of total.

4. **XGBoost**. Number of trees: $100, 200$. Maximum tree depth: $3, 6, 12$. Learning rate (eta): $0.15, 0.3, 0.6$. Minimum loss reduction (gamma): $0, 10, 100$. Minimum child weight: $1, 2, 4$. Regularisation parameter (lambda): $1, 10, 100$. Regularisation parameter (alpha): $0, 10, 100$. Fraction of features to consider when splitting a node: $0.25, 0.5, 1$. Fraction of training set used to train each tree: $0.25, 0.5, 1$.

5. **MLP**. Learning rate: $0.0001, 0.001, 0.01$. Maximum number of epochs: $1000, 2000$. Batch size: $1000, 10000$. Hidden size: $None, 60, [200, 14]$.

For temporal models, we fix the architectures as follows and batch sizes of 32 in order to ensure GPU memory usage remains below 12GB during training:

1. **Simple RNN, GRU, LSTM, TCN**. Encoder hidden size: 8. Decoder hidden size: 8.

2. **Dilated LSTM**. Encoder hidden size: 8. Decoder hidden size: 8. Dilation factors: $1, 2$.

3. **TFT**. Hidden state dimension: 32. Number of attention heads: 4.

4. **Informer, Autoformer**. Hidden state dimension: 32. Number of attention heads: 4. Convolutional encoder channels: 32. Number of encoder layers: 2. Number of decoder layers: 1.

5. **PatchTST**. Hidden state dimension: 128. Number of attention heads: 16. Number of encoder layers: 3. Linear layer hidden size: 256. Patch length: 32. Stride length: 16.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	18.59 (1.70)
ARIMA	Univariate	-	11.44 (1.45)
ETS	Univariate	-	19.65 (2.29)
Theta	Univariate	-	18.61 (1.70)
RNN	**Univariate**	**-**	**10.45 (0.84)**
RNN	NDWT	1	12.12 (0.91)
RNN	NWPT	1	11.47 (0.61)
GRU	**Univariate**	**-**	**9.55 (1.28)**
GRU	NDWT	1	34.89 (14.64)
GRU	NWPT	4	11.88 (1.01)
LSTM	**Univariate**	**-**	**9.78 (0.92)**
LSTM	NDWT	6	15.65 (3.14)
LSTM	NWPT	2	11.13 (0.58)
DilatedRNN	Univariate	-	11.21 (1.24)
DilatedRNN	NDWT	4	11.56 (0.74)
DilatedRNN	**NWPT**	**5**	**11.17 (0.63)**
TCN	**Univariate**	**-**	**9.55 (0.91)**
TCN	NDWT	5	12.98 (1.24)
TCN	NWPT	1	10.81 (0.52)
TFT	**Univariate**	**-**	**12.26 (0.83)**
TFT	NDWT	3	12.50 (0.87)
TFT	NWPT	6	12.64 (1.17)
Informer	Univariate	-	199.65 (0.01)
Informer	NDWT	2	199.06 (0.11)
Informer	**NWPT**	**4**	**198.94 (0.21)**
Autoformer	Univariate	-	13.21 (0.88)
Autoformer	**NDWT**	**3**	**13.13 (0.89)**
Autoformer	NWPT	1	13.40 (1.02)
PatchTST	Univariate	-	14.18 (2.34)
PatchTST	**NDWT**	**5**	**10.94 (1.21)**
PatchTST	NWPT	5	16.17 (4.61)

Table 13: UK Total Electricity Demand Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

References

Adjoumani, I. (2021) *Time Series Forecasting Using Boosting: An Application To The West African Stock Market*, Master's thesis, Imperial College London.

Bengio, Y., Simard, P., and Frasconi, P. (1994) Learning long-term dependencies with gradient descent is difficult, *IEEE Transactions on Neural Networks*, **5**, 157–166.

Cardinali, A. and Nason, G. P. (2018) Practical powerful wavelet packet tests for second-order stationarity, *Applied and Computational Harmonic Analysis*, **44**, 558–583.

Centre for Environmental Data Analysis (2022) Met Office MIDAS Open: UK

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	46.41 (5.05)
ARIMA	Univariate	-	38.28 (4.38)
ETS	Univariate	-	51.21 (5.20)
Theta	Univariate	-	47.06 (5.31)
RNN	**Univariate**	**-**	**45.62 (6.10)**
RNN	NDWT	1	49.73 (7.22)
RNN	NWPT	1	48.08 (5.96)
GRU	**Univariate**	**-**	**42.01 (6.53)**
GRU	NDWT	4	44.39 (5.07)
GRU	NWPT	5	45.31 (5.54)
LSTM	**Univariate**	-	**43.79 (6.70)**
LSTM	NDWT	1	48.81 (6.62)
LSTM	NWPT	9	47.15 (5.49)
DilatedRNN	Univariate	-	48.29 (6.17)
DilatedRNN	NDWT	4	46.89 (5.52)
DilatedRNN	**NWPT**	**1**	**41.78 (4.91)**
TCN	Univariate	-	46.25 (6.26)
TCN	NDWT	1	48.68 (6.53)
TCN	**NWPT**	**1**	**45.35 (5.31)**
TFT	Univariate	-	45.35 (4.52)
TFT	NDWT	3	45.38 (4.20)
TFT	**NWPT**	**4**	**43.06 (3.70)**
Informer	Univariate	-	161.77 (7.44)
Informer	**NDWT**	**1**	**115.50 (9.77)**
Informer	NWPT	9	144.27 (16.60)
Autoformer	**Univariate**	**-**	**46.30 (4.98)**
Autoformer	NDWT	1	47.85 (5.08)
Autoformer	NWPT	1	165.60 (10.07)
PatchTST	Univariate	-	64.24 (7.49)
PatchTST	**NDWT**	**8**	**59.60 (10.34)**
PatchTST	NWPT	7	69.36 (9.16)

Table 14: UK Hydropower Electricity Supply Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Land Surface Stations Data (1853-current), `http://catalogue.ceda.ac.uk/uuid/dbd451271eb04662beade68da43546e1`, accessed: 2022-11-07.

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., Cui, X., Witbrock, M., Hasegawa-Johnson, M. A., and Huang, T. S. (2017) Dilated recurrent neural networks, *Advances in Neural Information Processing Systems*, **30**.

Chen, T. and Guestrin, C. (2016) Xgboost: A scalable tree boosting system, in *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, Association for Computing Machinery.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	73.69 (9.00)
ARIMA	Univariate	-	73.41 (9.01)
ETS	Univariate	-	76.69 (9.90)
Theta	Univariate	-	73.89 (9.03)
RNN	Univariate	-	49.50 (3.42)
RNN	NDWT	1	48.79 (4.63)
RNN	**NWPT**	**3**	**46.86 (4.56)**
GRU	Univariate	-	50.49 (3.62)
GRU	NDWT	4	47.73 (5.10)
GRU	**NWPT**	**3**	**47.45 (3.78)**
LSTM	Univariate	-	51.26 (4.10)
LSTM	**NDWT**	**10**	**45.25 (4.87)**
LSTM	NWPT	2	49.00 (5.36)
DilatedRNN	Univariate	-	51.68 (3.72)
DilatedRNN	NDWT	1	49.13 (4.93)
DilatedRNN	**NWPT**	**9**	**49.05 (5.37)**
TCN	Univariate	-	49.29 (3.25)
TCN	**NDWT**	**1**	**45.24 (4.64)**
TCN	NWPT	8	48.56 (5.28)
TFT	Univariate	-	58.46 (7.91)
TFT	NDWT	1	58.79 (4.37)
TFT	**NWPT**	**3**	**51.27 (4.93)**
Informer	Univariate	-	197.70 (0.49)
Informer	NDWT	2	193.86 (1.39)
Informer	**NWPT**	**10**	**193.13 (2.05)**
Autoformer	Univariate	-	52.68 (5.79)
Autoformer	**NDWT**	**4**	**50.43 (5.42)**
Autoformer	NWPT	1	50.99 (5.49)
PatchTST	Univariate	-	96.09 (9.22)
PatchTST	**NDWT**	**2**	**64.67 (7.82)**
PatchTST	NWPT	2	70.13 (10.22)

Table 15: UK Wind Electricity Supply Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Bengio, Y. (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078*.

Coifman, R. R. and Donoho, D. L. (1995) Translation-invariant de-noising, in A. Antoniadis and G. Oppenheim, eds., *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pp. 125–150, Springer-Verlag, New York.

Coifman, R. R. and Wickerhauser, M. V. (1992) Entropy-based algorithms for best basis selection, *IEEE Transactions on Information Theory*, **38**, 713–718.

Conejo, A. J., Plazas, M. A., Espinola, R., and Molina, A. B. (2005) Day-ahead electricity

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	27.21 (5.00)
ARIMA	Univariate	-	22.61 (4.73)
ETS	Univariate	-	27.21 (5.00)
Theta	Univariate	-	27.33 (5.06)
RNN	Univariate	-	19.33 (2.18)
RNN	NDWT	2	17.90 (2.75)
RNN	**NWPT**	**9**	**17.72 (2.93)**
GRU	Univariate	-	19.55 (2.36)
GRU	NDWT	5	17.81 (2.77)
GRU	**NWPT**	**3**	**17.45 (2.95)**
LSTM	Univariate	-	20.32 (2.38)
LSTM	**NDWT**	**2**	**16.95 (2.61)**
LSTM	NWPT	9	17.38 (2.75)
DilatedRNN	Univariate	-	20.28 (2.64)
DilatedRNN	**NDWT**	**1**	**17.50 (2.62)**
DilatedRNN	NWPT	8	18.96 (2.76)
TCN	Univariate	-	17.47 (2.17)
TCN	NDWT	2	17.47 (2.79)
TCN	**NWPT**	**9**	**17.34 (2.82)**
TFT	Univariate	-	22.61 (2.89)
TFT	NDWT	5	21.60 (2.85)
TFT	**NWPT**	**8**	**21.31 (2.82)**
Informer	Univariate	-	103.50 (3.80)
Informer	**NDWT**	**3**	**28.32 (2.57)**
Informer	NWPT	2	33.88 (11.54)
Autoformer	**Univariate**	**-**	**20.47 (2.94)**
Autoformer	NDWT	6	21.09 (3.02)
Autoformer	NWPT	4	163.51 (15.37)
PatchTST	Univariate	-	26.48 (3.83)
PatchTST	**NDWT**	**3**	**22.34 (2.97)**
PatchTST	NWPT	5	28.12 (3.24)

Table 16: Heathrow Relative Humidity Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

price forecasting using the wavelet transform and arima models, *IEEE Transactions on Power Systems*, **20**, 1035–1042.

Daubechies, I. (1988) Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, **41**, 909–996.

Daubechies, I. (1992) *Ten lectures on wavelets*, SIAM, Philadelphia.

Donoho, D. L. and Johnstone, J. M. (1994) Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, **81**, 425–455.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	55.75 (12.99)
ARIMA	Univariate	-	45.15 (7.23)
ETS	Univariate	-	55.75 (12.99)
Theta	Univariate	-	56.56 (13.62)
RNN	Univariate	-	42.25 (6.53)
RNN	**NDWT**	**1**	**40.11 (6.79)**
RNN	NWPT	8	40.48 (7.54)
GRU	Univariate	-	42.94 (6.76)
GRU	NDWT	2	40.91 (7.33)
GRU	**NWPT**	**5**	**40.23 (6.94)**
LSTM	Univariate	-	41.54 (5.91)
LSTM	**NDWT**	**1**	**39.35 (7.40)**
LSTM	NWPT	8	42.53 (7.22)
DilatedRNN	Univariate	-	44.56 (6.43)
DilatedRNN	**NDWT**	**1**	**39.32 (6.98)**
DilatedRNN	NWPT	9	39.88 (7.16)
TCN	Univariate	-	42.16 (6.71)
TCN	**NDWT**	**2**	**40.23 (6.69)**
TCN	NWPT	10	40.89 (6.39)
TFT	Univariate	-	50.24 (6.79)
TFT	NDWT	1	45.35 (7.57)
TFT	**NWPT**	**10**	**44.41 (6.63)**
Informer	**Univariate**	**-**	**41.03 (5.93)**
Informer	NDWT	10	63.79 (10.33)
Informer	NWPT	9	154.67 (12.25)
Autoformer	**Univariate**	**-**	**71.87 (9.57)**
Autoformer	NDWT	3	88.15 (9.43)
Autoformer	NWPT	3	189.57 (2.02)
PatchTST	Univariate	-	57.77 (7.63)
PatchTST	**NDWT**	**1**	**55.32 (10.98)**
PatchTST	NWPT	9	62.47 (11.06)

Table 17: Heathrow Temperature Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Elexon (2022) Elexon BSC: Insights Solution, `https://bmrs.elexon.co.uk/`, accessed: 2022-11-07.

Elman, J. L. (1990) Finding structure in time, *Cognitive Science*, **14**, 179–211.

Gyaourova, A., Kamath, C., and Fodor, I. K. (2002) Undecimated wavelet transforms for image de-noising, *Report, Lawrence Livermore National Lab., CA*, **18**.

Haar, A. (1910) Zur theorie der orthogonalen funktionensysteme, *Mathematische Annalen*, **69**, 331–371.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	69.51 (7.42)
ARIMA	Univariate	-	112.75 (17.66)
ETS	Univariate	-	69.29 (7.26)
Theta	Univariate	-	69.27 (7.26)
RNN	**Univariate**	**-**	**48.25 (2.91)**
RNN	NDWT	2	51.59 (3.22)
RNN	NWPT	9	49.83 (3.06)
GRU	**Univariate**	**-**	**48.23 (2.98)**
GRU	NDWT	6	50.68 (2.77)
GRU	NWPT	9	50.30 (2.95)
LSTM	**Univariate**	**-**	**48.37 (3.15)**
LSTM	NDWT	6	50.12 (2.81)
LSTM	NWPT	6	50.10 (2.93)
DilatedRNN	Univariate	-	48.55 (2.95)
DilatedRNN	NDWT	2	49.20 (2.91)
DilatedRNN	NWPT	9	49.88 (2.98)
TCN	**Univariate**	**-**	**48.73 (2.78)**
TCN	NDWT	10	49.72 (2.92)
TCN	NWPT	7	50.89 (2.94)
TFT	**Univariate**	**-**	**50.51 (2.89)**
TFT	NDWT	3	51.38 (2.96)
TFT	NWPT	3	56.56 (4.54)
Informer	Univariate	-	53.35 (2.98)
Informer	**NDWT**	**7**	**53.32 (2.72)**
Informer	NWPT	8	174.59 (7.11)
Autoformer	**Univariate**	**-**	**51.25 (3.23)**
Autoformer	NDWT	2	65.69 (7.30)
Autoformer	NWPT	9	173.38 (13.56)
PatchTST	Univariate	-	78.09 (4.25)
PatchTST	**NDWT**	**9**	**70.92 (4.67)**
PatchTST	NWPT	10	84.74 (6.84)

Table 18: Heathrow Wind Speed Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory, *Neural Computation*, **9**, 1735–1780.

Jansen, M., Nason, G. P., and Silverman, B. W. (2004) Multivariate nonparametric regression using lifting, Technical report, University of Bristol.

Johnstone, I. M. and Silverman, B. W. (1997) Wavelet threshold estimators for data with correlated noise, *Journal of the Royal Statistical Society: Series B*, **59**, 319–351.

Kalyan, K. S., Rajasekharan, A., and Sangeetha, S. (2021) Ammus: A survey of transformer-based pretrained models in natural language processing, *arXiv preprint arXiv:2108.05542*.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	136.74 (3.11)
ARIMA	Univariate	-	193.57 (1.60)
ETS	Univariate	-	140.23 (2.90)
Theta	Univariate	-	143.80 (6.22)
RNN	Univariate	-	152.21 (0.57)
RNN	NDWT	4	142.30 (1.12)
RNN	**NWPT**	**1**	**136.88 (1.06)**
GRU	Univariate	-	152.80 (0.97)
GRU	NDWT	2	140.98 (1.33)
GRU	**NWPT**	**6**	**139.57 (1.80)**
LSTM	Univariate	-	150.51 (1.27)
LSTM	NDWT	2	143.47 (0.96)
LSTM	**NWPT**	**2**	**140.46 (1.42)**
DilatedRNN	Univariate	-	152.79 (1.47)
DilatedRNN	NDWT	2	145.97 (1.82)
DilatedRNN	**NWPT**	**1**	**139.64 (1.57)**
TCN	Univariate	-	148.85 (1.23)
TCN	**NDWT**	**3**	**140.48 (1.42)**
TCN	NWPT	2	141.37 (1.68)
TFT	Univariate	-	147.72 (0.57)
TFT	NDWT	4	142.85 (3.08)
TFT	**NWPT**	**6**	**138.81 (2.75)**
Informer	Univariate	-	159.05 (3.08)
Informer	**NDWT**	**1**	**156.30 (3.30)**
Informer	NWPT	9	193.51 (1.92)
Autoformer	Univariate	-	157.38 (1.33)
Autoformer	**NDWT**	**8**	**156.46 (1.81)**
Autoformer	NWPT	8	197.33 (0.98)
PatchTST	Univariate	-	173.75 (0.58)
PatchTST	**NDWT**	**8**	**164.05 (1.25)**
PatchTST	NWPT	9	187.46 (1.36)

Table 19: Simulated Bumps Data Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Kuan, C.-M. and Liu, T. (1995) Forecasting exchange rates using feedforward and recurrent neural networks, *Journal of Applied Econometrics*, **10**, 347–364.

La Cour-Harbo, A. and Jensen, A. (2009) *Wavelets and the Lifting Scheme*, pp. 10007–10031, Springer New York, New York.

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., and Hager, G. D. (2017) Temporal convolutional networks for action segmentation and detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. (2019) Enhancing

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	129.64 (13.68)
ARIMA	Univariate	-	133.28 (13.72)
ETS	Univariate	-	128.17 (15.85)
Theta	Univariate	-	158.90 (9.71)
RNN	**Univariate**	**-**	**183.13 (2.34)**
RNN	NDWT	10	188.12 (0.98)
RNN	NWPT	1	187.03 (0.59)
GRU	Univariate	-	180.23 (2.75)
GRU	NDWT	2	174.89 (4.63)
GRU	**NWPT**	**1**	**170.38 (9.87)**
LSTM	**Univariate**	**-**	**181.11 (1.37)**
LSTM	NDWT	10	185.82 (1.37)
LSTM	NWPT	1	181.69 (3.02)
DilatedRNN	**Univariate**	**-**	**185.41 (0.78)**
DilatedRNN	NDWT	9	188.80 (0.44)
DilatedRNN	NWPT	1	186.72 (0.69)
TCN	Univariate	-	182.54 (1.43)
TCN	NDWT	6	190.42 (0.56)
TCN	**NWPT**	**1**	**175.39 (2.67)**
TFT	Univariate	-	189.88 (0.34)
TFT	**NDWT**	**1**	**186.74 (3.73)**
TFT	NWPT	3	189.98 (1.80)
Informer	**Univariate**	**-**	**110.76 (4.53)**
Informer	NDWT	10	175.44 (7.99)
Informer	NWPT	6	195.23 (1.14)
Autoformer	Univariate	-	197.07 (0.17)
Autoformer	**NDWT**	**1**	**184.48 (3.35)**
Autoformer	NWPT	10	199.71 (0.08)
PatchTST	**Univariate**	**-**	**177.81 (2.02)**
PatchTST	NDWT	7	180.88 (10.69)
PatchTST	NWPT	10	186.00 (4.54)

Table 20: Simulated Doppler Data Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

the locality and breaking the memory bottleneck of transformer on time series forecasting, *Advances in Neural Information Processing Systems*, **32**.

Lim, B. and Zohren, S. (2021) Time-series forecasting with deep learning: a survey, *Philosophical Transactions of the Royal Society A*, **379**, 20200209.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting, *International Journal of Forecasting*, **37**, 1748–1764.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2020) The m4 competition: 100,000 time series and 61 forecasting methods, *International Journal of Forecasting*, **36**, 54–74.

Model	Feature Set	Modal Wavelet Number	Mean SMAPE % (SE)
Persistence	Univariate	-	66.17 (1.74)
ARIMA	Univariate	-	65.57 (0.29)
ETS	Univariate	-	64.61 (1.55)
Theta	Univariate	-	66.38 (0.40)
RNN	Univariate	-	51.66 (0.55)
RNN	**NDWT**	**1**	**42.67 (1.13)**
RNN	NWPT	10	62.02 (5.66)
GRU	Univariate	-	53.50 (0.39)
GRU	**NDWT**	**1**	**32.48 (1.14)**
GRU	NWPT	5	43.33 (4.82)
LSTM	Univariate	-	46.42 (0.20)
LSTM	**NDWT**	**1**	**41.16 (0.84)**
LSTM	NWPT	9	50.86 (5.31)
DilatedRNN	Univariate	-	48.87 (0.23)
DilatedRNN	**NDWT**	**1**	**42.33 (0.58)**
DilatedRNN	NWPT	9	49.00 (3.17)
TCN	Univariate	-	48.08 (0.23)
TCN	NDWT	1	44.48 (2.40)
TCN	**NWPT**	**9**	**43.44 (3.12)**
TFT	**Univariate**	**-**	**59.64 (2.79)**
TFT	NDWT	1	62.67 (2.20)
TFT	NWPT	7	62.89 (5.39)
Informer	**Univariate**	**-**	**40.82 (0.64)**
Informer	NDWT	6	100.66 (14.56)
Informer	NWPT	6	158.34 (3.58)
Autoformer	**Univariate**	**-**	**89.17 (3.80)**
Autoformer	NDWT	2	153.33 (10.25)
Autoformer	NWPT	8	199.00 (0.90)
PatchTST	**Univariate**	**-**	**48.98 (0.47)**
PatchTST	NDWT	5	83.08 (10.14)
PatchTST	NWPT	4	103.97 (12.34)

Table 21: Simulated Heavisine Data Out-of-Sample 1- to 1000-Step-Ahead Forecast Performance. The top-performing feature set for each model has been bolded.

Mallat, S. G. (1989) A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674–693, dOI: 10.1109/34.192463.

Nason, G., Sapatinas, T., and Sawczenko, A. (2001) Wavelet packet modelling of infant sleep state using heart rate data, *Sankhyā: The Indian Journal of Statistics, Series B*, pp. 199–217.

Nason, G., Barber, S., Downie, T., Fryzlewicz, P., Kovac, A., Ogden, T., and Silverman, B. (2016) wavethresh: Wavelets statistics and transforms, r package version 4.7.2.

Nason, G. P. (2002) Choice of wavelet smoothness, primary resolution and threshold in wavelet shrinkage, *Statistics and Computing*, **12**, 219–227.

Nason, G. P. (2008) *Wavelet Methods in Statistics with R*, volume 574, Springer New York, New York.

Nason, G. P. and Sapatinas, T. (2002) Wavelet packet transfer function modelling of nonstationary time series, *Statistics and Computing*, **12**, 45–56.

Nason, G. P. and Silverman, B. W. (1995) The stationary wavelet transform and some statistical applications, in A. Antoniadis and G. Oppenheim, eds., *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pp. 281–299, Springer-Verlag, New York.

Nason, G. P., Sapatinas, T., and Sawczenko, A. (1997) Statistical modelling of time series using non-decimated wavelet representations, Technical report, University of Bristol.

Nason, G. P., Von Sachs, R., and Kroisandt, G. (2000) Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum, *Journal of the Royal Statistical Society: Series B*, **62**, 271–292.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022) A time series is worth 64 words: Long-term forecasting with transformers, *arXiv preprint arXiv:2211.14730*.

Nunes, M. A., Knight, M. I., and Nason, G. P. (2006) Adaptive lifting for nonparametric regression, *Statistics and Computing*, **16**, 143–159.

Raj, V. N. P. and Venkateswarlu, T. (2011) ECG signal denoising using undecimated wavelet transform, in *2011 3rd International Conference on Electronics Computer Technology*, volume 3, pp. 94–98.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning representations by back-propagating errors, *Nature*, **323**, 533–536.

Schlüter, S. and Deuschle, C. (2010) Using wavelets for time series forecasting: Does it pay off?, Technical report, IWQW Discussion Papers.

Soltani, S., Boichu, D., Simard, P., and Canu, S. (2000) The long-term memory prediction by multiscale decomposition, *Signal Processing*, **80**, 2195–2205.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017) Attention is all you need, *Advances in Neural Information Processing Systems*, **30**.

Vermaak, J. and Botha, E. (1998) Recurrent neural networks for short-term load forecasting, *IEEE Transactions on Power Systems*, **13**, 126–132.

Wang, Y. and Guo, Y. (2020) Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost, *China Communications*, **17**, 205–221.

Wong, H., Ip, W.-C., Xie, Z., and Lui, X. (2003) Modelling and forecasting by wavelets, and the application to exchange rates, *Journal of Applied Statistics*, **30**, 537–553.

Wu, H., Xu, J., Wang, J., and Long, M. (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, *Advances in Neural Information Processing Systems*, **34**, 22419–22430.

Yan, J., Mu, L., Wang, L., Ranjan, R., and Zomaya, A. Y. (2020) Temporal convolutional networks for the advance prediction of enso, *Scientific reports*, **10**, 1–15.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115.