ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251157

Boosting Global Time Series Forecasting Models: A Two-Stage Modelling Framework

Junru Ren[a] **and Shaomin Wu**[a,*]

[a]Kent Business School, University of Kent, Canterbury, Kent, CT2 7FS, UK

Abstract. A time series forecasting model—which is typically built on a single time series—is known as a local time series model (tsLM). In contrast, a forecasting model trained on multiple time series is referred to as a global time series model (tsGM). tsGMs can enhance forecasting accuracy and improve generalisation by learning cross-series information. As such, developing tsGMs has become a prominent research focus within the time series forecasting community. However, the benefits of tsGMs may not always be realised if the given set of time series is heterogeneous. While increasing model complexity can help tsGMs adapt to such a set of data, it can also increase the risk of overfitting and forecasting error. Additionally, the definition of homogeneity remains ambiguous in the literature. To address these challenges, this paper explores how to define data heterogeneity and proposes a two-stage modelling framework: At stage one, a tsGM is learnt to identify homogeneous patterns; and at stage two, tsLMs (e.g., ARIMA) or sub-tsGMs tailored to different groups are learnt to capture the heterogeneity. Numerical experiments on four open datasets demonstrate that the proposed approach significantly outperforms six state-of-the-art models. These results highlight its effectiveness in unlocking the full potential of global forecasting models for heterogeneous datasets.

1 Introduction

Accurate time series forecasting facilitates data-driven decision-making in business, economy, and other industries [30]. Recently, global time series models (tsGMs), estimated on a set of time series with similar patterns, have been proposed [31]. Research on tsGMs is becoming more burgeoning in the time series forecasting community, compared with traditional local time series models (tsLMs) that are estimated on individual time series. By leveraging cross-series information, tsGMs effectively reduce generalisation errors and perform exceptionally well, especially for the cases where individual time series are too short to build a reliable model [27, 33].

The improved performance of a tsGM relies on the assumption that the time series the model learnt from are homogeneous, namely, they share a similar pattern. However, different time series may have heterogeneity, which may stem from the differences in seasonal patterns, trends, or underlying data structures [29]. Intuitively, it is challenging to build a well-performed tsGM on a set of different time series that contain heterogeneity. Wellens et al. [40] and Hewamalage et al. [16] found that the forecasting performance of tsGMs is closely related to both the degree of the homogeneity of the time series and the complexity of tsGMs, and that modelling methods such as recurrent neural networks (RNNs) and light gradient boosting models (LGBMs) can handle heterogeneity better than linear time series models. A sophisticated tsGM is required to accurately describe multiple highly heterogeneous time series simultaneously. However, such complex models are prone to overfitting, especially for the cases where the volume of the available data is limited. In summary, the forecasting performance of a tsGM is constrained by the level of heterogeneity of time series on which the tsGM is learnt, while constructing tsLMs on each individual time series completely overlooks the shared information among different time series [29]. This further prompts some new questions: *how to determine the heterogeneity level of a given set of time series and how to fully leverage the strength of tsGMs learnt on heterogeneous time series?*

To improve the capability of capturing the respective characteristics of the time series a tsGM learnt on, the divide-and-conquer methodology is employed in the literature. Clustering techniques, such as distance-based clustering [13] and feature-based clustering [1], are applied to divide the entire dataset into several sub-groups. Time series in a sub-group is regarded as homogeneous, and then a tsGM is learned on each sub-group. Additionally, some research considers combining local and global components, resulting in the proposal of hybrid models. Smyl [36] used exponential smoothing methods to capture the local level and seasonality of each series, and then a long short-term memory (LSTM) network was globally estimated on the remaining parts of the series after removing the obtained local characteristics. Their aim is to extract and separate non-homogeneous local patterns and homogeneous global patterns.

As evident from the above literature review, the existing methods analyse some characteristics of a set of time series to obtain clusters or remove individual-specific patterns—which can be referred to as data pre-processing—then they build forecasting models. A drawback of these methods is that the data pre-processing stage only considers characteristics of the set of time series but ignores the characteristics of modelling methods. Modelling methods play an important role as different modelling methods can capture different characteristics in the set of time series. To overcome this drawback, this paper proposes to boost forecasting models with a two-stage modelling framework: At stage one, we learn a tsGM on the entire set of time series to capture the homogeneous patterns. At stage two, we perform a cluster analysis on the residual time series—which are calculated from stage one and contain heterogeneity and noises—and then build either tsLMs (e.g., the autoregressive integrated moving average (ARIMA) models) or tsGMs to different clusters to capture heterogeneity.

* Corresponding Author. Email: s.m.wu@kent.ac.uk.

The novelty and contribution of this paper includes: (i) It proposes of a two-stage modelling framework to first capture the homogeneity of a given set of time series, and then the heterogeneity in the residuals is separated out; (ii) the degree of heterogeneity is quantified by considering characteristics of both the data and the learnt tsGM; (iii) it presents three propositions relating to the proposed two-stage modelling framework to provide a theoretical guarantee.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 proposes a two-stage modelling framework. Section 4 compares the cumulative errors of the proposed approach with six other models on four open datasets. Section 5 discusses relevant issues. Section 6 concludes the work.

2 Related Work

Clustering-based models. The clustering-and-then-model method is a commonly-used method in dealing with heterogeneous time series forecasting. Bandara et al. [1] and Semenoglou et al. [33] used k-means algorithms and series features on trends, seasonality, and autocorrelation to conduct feature-based clustering, and then trained a cluster-specific model on each cluster. Godahewa et al. [13] investigated feature-based clustering, distance-based clustering, and random clustering, where dynamic time warping (DTW) distances were used, and trained multiple tsGMs on each cluster of the time series, and then constructed an ensemble model to generate final forecasts. Chen et al. [4] constructed an adaptable channel clustering module that can be plugged into neural networks to realise Euclidean distance-based clustering using radial basis function kernels. The module consists of a cluster assigner and a cluster-aware feed forward module. Fröhwirth-Schnatter and Kaufmann [12] utilised a model-based clustering method and assumed the time series were originated from AR processes. A time series forecasting model was then integrated into the clustering. On the contrary, Neubauer and Filzmoser [29] considered a model-and-then-clustering mechanism and proposed an algorithm named TSAVG. In their work, specifically, tsLMs were first estimated on each time series independently, and then DTW distances were calculated to determine the neighbours of the target series. The forecast of the target series is generated by averaging local forecasts of its neighbours. The cluster-based models ignore the global information across the entire dataset. Meanwhile, clustering based on data similarity does not consider the model used, and it may divide the series that could be modelled using one tsGM into two clusters, leading to the loss of information and an increase in model complexity.

Local-global hybrid models. To take advantage of the strength of both tsLMs and tsGMs, a local-global hybrid time series modelling method is introduced. Semenoglou et al. [33] averaged the forecasts obtained by a local Theta method and a tsGM. Semenoglou et al. [33] used some outputs of the tsLM as inputs to feed into the tsGM. Smyl [36] proposed a hybrid method of the exponential smoothing and RNNs, that is, the exponential smoothing method was used to capture the components of level and seasonality of the individual series, and a global RNN was applied to model the remaining homogeneous parts after local characteristics were removed. However, heterogeneity is not solely attributable to differences in level and seasonality. Insufficient removal of heterogeneity results in data patterns not being fully modelled. There is also some work combining linear models and non-linear models parallelly or serially, see Hajirahimi and Khashei [14] and Zhang [41], for example. However, they are built on one single time series, instead of multiple time series.

Error correction methods. Error correction is a technique to improve forecasting accuracy by residual modelling. A correction procedure is implemented after building a forecasting model, during which the residuals are modelled recursively until they are white noises. Firmino et al. [10] used ARIMA models to recursively correct the forecasts obtained by neural networks, and discussed additive error models and multiplicative error models. Subsequently, da Silva et al. [5] used linear models (e.g., ARIMA) first and then employed non-linear models (e.g. support vector regression and LSTM) to correct errors and improve their model's forecasting accuracy. Boosting methods sequentially train weak learners, with each learner correcting the errors of predecessors at each iteration [3]. Although these methods have some similarities to our method in this study, the error correction in their methods is a recursive combination of a sequence of linear and non-linear models. Their aim was to correct errors and adjust forecasts. Nevertheless, the aim of the present study is to identify and model the homogeneity among multiple time series and then capture the heterogeneity of the time series.

3 Two-stage modelling framework

3.1 Problem Formulation

Given a set of n time series, we denote $\{x_{i,t}\}$ as the i-th time series for $i \in \{1, 2, \cdots, n\}$, $t \in \{t_0^{(i)}, t_0^{(i)}+1, \cdots, t_0^{(i)}+n_i-1\}$, and $t_0^{(i)}$ is the start time point, and n_i is the length of the i-th series. Given a look-back window $\boldsymbol{x}_{i,(t-q):(t-1)} = [x_{i,t-q}, \cdots, x_{i,t-1}]$, a tsGM, $x_{i,t} = g(\boldsymbol{x}_{i,(t-q):(t-1)}|\Theta_g) + \epsilon_{i,t}$, is learnt on the entire set of the time series, where Θ_g denotes the vector of all parameters in the function $g(.)$ and $\epsilon_{i,t}$ is the residual.

Heterogeneity among multiple time series poses a challenge for a tsGM in capturing all respective patterns of individual time series, even if the model is sufficiently complex. If the residuals of a tsGM model are tested as white noise, they are assumed not autocorrelated. The constructed tsGM is deemed to be statistically adequate to model the corresponding time series. Otherwise, the tsGM is statistically inadequate and the residuals needs further analysing and modelling. Let n_h denote the number of time series whose residuals are not white noise. Denote the heterogeneity level of the set of time series by R_h, which is defined as the ratio of n_h to the total number of time series in this dataset, as shown below,

$$R_h = R_h|(\{x_{i,t}\}, g) = \frac{n_h}{n}. \tag{1}$$

The heterogeneity level of a set of time series is relevant to the modelling method. For a given set of time series $\{x_{i,t}\}$ and a tsGM model $g(.)$, the larger R_h, the stronger heterogeneity.

In our proposed method, the tasks of learning a tsGM and identifying homogeneity are included at stage one, and those of modelling the heterogeneity are performed at stage two. Aggregating the results from two stages, the final time series model is updated to

$$x_{i,t} = f\left(\boldsymbol{x}_{i,(t-q):(t-1)}, g(\boldsymbol{x}_{i,(t-q):(t-1)}; \Theta_g), \Theta_g|\Theta_f\right) + \epsilon_{i,t}, \tag{2}$$

where Θ_f is the parameter vector to be estimated at stage two.

The loss function used in this study is the mean squared error (MSE), which has the form of

$$\ell(x_{i,t}, \widehat{x}_{i,t}) = \frac{1}{N}\sum_{i,t}(x_{i,t} - \widehat{x}_{i,t})^2, \tag{3}$$

where $\widehat{x}_{i,t}$ is the forecast and N is the total number of sample points.

The empirical loss obtained by Equation (3) over the training samples is denoted as R_{emp} and the expected loss over the entire data distribution $\mathcal{D}$ is given by $R_{\text{exp}} = \mathbb{E}_{(\boldsymbol{x}_{i,(t-q):(t-1)}, x_{i,t})\sim\mathcal{D}}[(x_{i,t} - \widehat{x}_{i,t})^2]$. The generalisation error of the model, denoted by ε, is defined as $\varepsilon = |R_{\text{emp}} - R_{\text{exp}}|$ [2], and it can be approximated as the difference between the in-sample error and the out-of-sample error.

3.2 Model construction

Figure 1 illustrates the proposed two-stage modelling framework. At stage one, a tsGM $g(.)$ is learnt on the entire set of time series, based on which the model residuals can be calculated. The residuals are considered as including heterogeneity within this dataset, and heterogeneity series (denoted as $\{h_{i,t}\}$) can be described using either additive (i.e., $h_{i,t} = x_{i,t} - \widehat{x}_{i,t}$) or multiplicative (i.e., $h_{i,t} = x_{i,t}/\widehat{x}_{i,t}$) modelling methods. If the residuals of all individual time series are white noise as a result of some statistical tests (e.g., the Ljung-Box test), stage one is completed and stage two is not needed. Otherwise, tsLMs and/or tsGMs must be estimated on the residuals at stage two to address the heterogeneity of the individual time series and extract remaining patterns. Two types of modelling methods are employed at stage two: Type-I and Type-II approaches.

The Type-I approach constructs individual tsLMs such as the ARIMA model on each heterogeneous residual series to supplement and update the forecasts generated by $g(.)$ at stage one. Theoretically, R_h can be updated to 0. However, the costs of training and maintaining models dramatically increase when the number of heterogeneous time series is excessively large, and thus it is not applicable in this case. Alternatively, the Type-II approach estimates multiple sub-tsGMs for sub-groups of heterogeneous series, respectively. The identified heterogeneous series presents various autocorrelation, non-linearity or heteroskedasticity. Thus, feature-based clustering is considered. The residuals of the individual time series are grouped according to the features including autocorrelation coefficients, non-linearity (using the statistic of Teräsvirta's non-linearity test [38]) and autoregressive conditional heteroskedasticity effects (using the statistic of Lagrange-multiplier test [8]). Other features such as spectral entropy and lumpiness can also be involved [19]. These features are commonly-used and identified as features which influence the forecasting modelling and accuracy the most [37].Through measuring the Euclidean distance between features, time series whose residual series exhibit similar features are put into a cluster. The number of clusters is denoted as K. A sub-tsGM, denoted by $g_k(.)$ ($k \in \{1, 2, \cdots, K\}$), is then trained on each cluster of time series. The structure of $g_k(.)$ is given in Figure 1, and Algorithm 1 presents the procedure of constructing it. After building all sub-tsGMs, an updated R_h can be calculated.

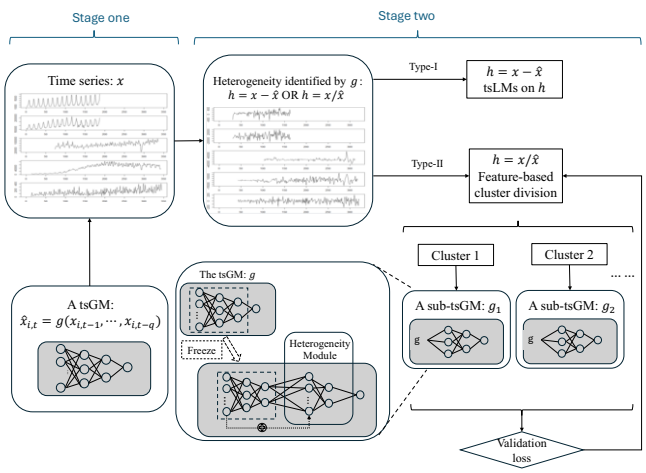


Figure 1. The two-stage modelling framework.

Algorithm 1: The procedure of constructing $g_k(.)$.

Input: The tsGM at stage one $g(.)$ with L layers
Output: Cluster-specific sub-tsGM $g_k(.)$ with L_k layers

for $l = 1$ **to** $L - 1$ **do**
  Copy weights from layer l of $g(.)$ to $g_k(.)$
  Freeze layer l in $g_k(.)$
Let $\mathbf{h}_{\text{global}} \leftarrow$ output of the first $L - 1$ layer of $g_k(.)$;
`/* Capture global features shared by the entire dataset */`
Let $\mathbf{x}_{\text{input}} \leftarrow$ input of $g(.)$;
Let $g_k^{(l)}(.) \leftarrow$ output of the l-th layer of $g_k(.)$;
Construct the heterogeneity module:
$\mathbf{z}_1 \leftarrow g_k^{(L)}(\mathbf{h}_{\text{global}})$;
$\mathbf{z}_2 \leftarrow \text{reshape}(\mathbf{z}_1) + \mathbf{x}_{\text{input}}$; `/* Incorporate cluster-specific information */`
$\widehat{x}_{i,t} \leftarrow g_k^{(L+1:L_k)}(\mathbf{z}_2)$; `/* Add more layers to capture cluster-specific patterns and adjust forecasts */`

We split the entire set of time series into training, validation, and test subsets, and consider the in-sample error, the generalisation error and the model maintenance cost to optimise the clustering. Let the clustering C based on m features $fea_1, \cdots, fea_m$ be denoted by $C(fea_1, \cdots, fea_m) = \{C_1, C_2, \cdots, C_K\}$. The objective of the clustering is

$$\min_{C(fea_1,\cdots,fea_m)} \quad \sum_{k=1}^{K} \sum_{i\in C_k} \sum_{t\in \text{val}} (x_{i,t} - \widehat{x}_{i,t})^2, \tag{4}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} |C_k| = n_h,$$

$$\sum_{k=1}^{K} f_m(|C_k|) \le U_m.$$

Based on the training dataset, the optimisation defined in Equation (4) aims to find a clustering C which minimises the square error on the validation set. $|C_k|$ is the number of series in cluster C_k. We set an upper limit of the model maintenance cost as U_m. In reality, it can be relevant to the data volume (or model complexity) and the function $f_m(\cdot)$ demonstrates the relationship.

The two-stage modelling framework is denoted as TS-X-I/II, where X represents the tsGM estimated at stage one and I/II indicates Type-I or Type-II used at stage two. The following propositions are derived on the in-sample error and the generalisation error.

Proposition 1. *Based on the tsGM trained at stage one, stage two further reduces the MSE loss defined in Equation (3) by descending the gradient in the function space.*

Proof. The tsGM constructed at stage one is denoted as $g(.)$. $\forall i, t$, the functional gradient of the MSE loss at $g(.)$ can be derived as

$$\frac{\partial \ell(x_{i,t}, g(\boldsymbol{x}_{i,(t-q):(t-1)}))}{\partial g(\boldsymbol{x}_{i,(t-q):(t-1)})} \propto x_{i,t} - g(\boldsymbol{x}_{i,(t-q):(t-1)}), \tag{5}$$

which is proportional to the heterogeneity (i.e., residuals). Thus, for additive modelling, adding a model describing heterogeneity at stage two is equivalent to conducting functional gradient descent in the function space. The MSE loss further decreases.

The $x_{i,t}$ and $g(.)$ can always be shifted to positive by adding a fixed small value and then shifted back. Thus, in the log-space,

$$\frac{\partial \ell(\log(x_{i,t}), \log\left(g(\boldsymbol{x}_{i,(t-q):(t-1)})\right))}{\partial \log\left(g(\boldsymbol{x}_{i,(t-q):(t-1)})\right)} \propto \log\left(\frac{x_{i,t}}{g(\boldsymbol{x}_{i,(t-q):(t-1)})}\right). \tag{6}$$

For multiplicative modelling, multiplying a model describing heterogeneity is equivalent to conducting functional gradient descent in the log-space. $\forall x_{i,t}$, the loss in the log-space, namely $\left(\log\left(\frac{x_{i,t}}{g(\boldsymbol{x}_{i,(t-q):(t-1)})}\right)\right)^2$, decreases, and the MSE loss decreases as well. □

Proposition 1 proves that stage two is conducive to reducing the in-sample error. The hypothesis set of the tsGM, which consists of all possible functions, is denoted as $\mathcal{H}$. The size of $\mathcal{H}$, which is the number of functions in $\mathcal{H}$ if $\mathcal{H}$ is finite, denoted as $|\mathcal{H}|$ [2]. The size of hypothesis set of the model with trainable parameters learnt at stage two for cluster k is denoted as $|\mathcal{H}_k|$, $k \in \{1, 2, \cdots, K\}$, respectively. $|\Phi|$ represents the size of the k-means clustering function class used at stage two and it is of order $\mathcal{O}(Km)$ [24, 34]. Proposition 2 provides a generalisation upper bound of the proposed two-stage modelling framework.

Proposition 2. *If the identified heterogeneity is divided into K clusters at stage two, with probability of at least $1-\delta$, it holds that*

$$\varepsilon \leq \sqrt{\frac{\log(|\mathcal{H}|) + \log(\frac{4}{\delta})}{2\sum_{i=1}^{n} n_i'}} + \sqrt{\frac{\log(|\Phi|) + \sum_{k=1}^{K}\log(|\mathcal{H}_k|) + \log(\frac{4}{\delta})}{2\sum_{i\in I_h} n_{i(h)}'}},$$

where $\delta \in (0,1)$, and n_i' and $n_{i(h)}'$ are the effective sample sizes [1] of the i-th original and heterogneity time series, respectively, and I_h is a set that contains the indices of all identified heterogeneity series.

Proof. For Type-I modelling at stage two, K can be regarded as equal to n_h, and $\mathcal{H}_k$ refers to the hypothesis set of each tsLM (e.g., ARIMA) in this case. While for the Type-II modelling method, $\mathcal{H}_k$ is the hypothesis set of the heterogeneity module constructed for cluster k. The MSE loss on normalised data can be regarded as bounded, and they can take values in [0, 1] after rescaling [27]. Following Theorem 1.17 of Berner et al. [2], with probability of at least $1-\delta/2$, the generalisation upper bound of each stage can be derived. A union bound of two stage is obtained and $P(\text{The upper bound of either stage fails}) \leq \delta/2 + \delta/2 = \delta$. The observations are sliding windows of time series and are non-i.i.d. The sample size is therefore replaced by the effective sample size [27]. The proposition is established. □

Proposition 3. *There exists an optimal clustering C at stage two that minimises the out-of-sample error.*

Proof. Assuming the tsGM constructed at stage one identifies n_h heterogeneous time series, the possible number of clusters at stage two ranges from 1 to n_h. Based on Propositions 1 and 2, with well-fitted sub-tsGMs, an increase in the number of clusters reduces the in-sample error while increasing the generalisation error. Take two extreme scenarios as examples. If $K = n_h$, a Type-I modelling method at stage two is implemented and n_h tsLMs are trained on each heterogeneous series, respectively. In-sample patterns can be completely captured as the residuals are checked as white noises. However, the risk of overfitting increases. If $K = 1$, a single tsGM is trained at stage two. All heterogeneous series are described by one single model, and the underlying heterogeneity among these series obscures the modelling ability of capturing series-specific patterns. The effectiveness of the tsGM at stage two is limited to further improve the in-sample forecasting accuracy. However, the generalisation capability of tsGMs is fully utilised. Therefore, there must exist a K ($1 \leq K \leq n_h$) that achieves a trade-off between in-sample error and generalisation error and minimises the out-of-sample error. □

Proposition 1 proves that modelling heterogeneity involves a one-step gradient descent in the loss function space. In fact, if we add more stages to implement clustering further, there always exists an equivalent clustering at stage two. Thus, the multi-stage setting is redundant and two stages suffice to approximate the target function. The key is to appropriately choose the features and the number of clusters and construct sub-tsGMs in light of out-of-sample error.

4 Experiments and Results

4.1 Experimental Setup

Datasets. Four open datasets are used to evaluate the two-stage modelling framework. **Tourism**: It is from the tourism forecasting competition and is publicly available through the *Tcomp* R package [7]. **M3**: It has been used in the M3 forecasting competition and is provided in the R package *Mcomp* [18]. It is divided into five subcategories: M3-demographic, M3-finance, M3-industry, M3-macro, and M3-micro. The models are constructed on each subcategory separately and aggregated results are reported. **CIF 2016**: It is from the CIF 2016 forecasting competition, including 24 real-world series from the banking cluster and 48 artificially generated series. The dataset can be obtained from Neubauer and Filzmoser [29]. **Hospital**: It tracks the number of patients for various medical problems and is publicly available from R package *expsmooth* [20].

The sampling frequency of these datasets is monthly and the statistical description is summarised in Table 1, including the domain, number of series (#Series), series length, mean trend and seasonality strength and Augmented Dick-Fuller (ADF) test statistics [6]. The trend and seasonality strength are calculated based on an STL decomposition [19], and the value close to 1 indicates strong trend or seasonality. A smaller ADF test statistics indicates a more stationary time series.

Table 1. The statistical description of datasets.

Dataset	Domain	#Series	Length			Trend	Seasonality	ADF
			min	max	mean			
Tourism	Tourism	366	91	333	299	0.855	0.752	-5.788
Hospital	Health care	767	84	84	84	0.484	0.343	-3.385
CIF 2016	Banking	72	28	120	99	0.903	0.480	-2.633
M3-demo	Population	111	71	138	123	0.944	0.348	-2.738
M3-finance	Markets	145	68	144	124	0.906	0.331	-1.925
M3-industry	Manufacturing	334	96	144	140	0.790	0.531	-3.538
M3-macro	Economy	312	66	144	131	0.947	0.279	-2.336
M3-micro	Business	474	68	126	93	0.493	0.371	-4.063
M3	-	1,376	66	144	119	0.748	0.383	-3.212

[1] The effective sample size n_i' can be calculated using $n_i' = n_i\left[\sum_{j=-(n_i-1)}^{+(n_i-1)}(1-\frac{|j|}{n_i})\rho(j)\right]^{-1}$, where $\rho(.)$ is the autocorrelation function [39]. And $n_{i(h)}'$ can be obtained similarly.

The time series in these datasets present characteristics of non-stationarity, including trends and seasonal patterns of different strengths. The problem of distribution shift affects the predictability of time series. Kim et al. [22] proposed reversible instance normalisation (RevIN) to forecast non-stationary time series, and it applied normalisation with learnable parameters to each individual time series, and restored the statistical information of mean and variance on the corresponding outputs. Subsequently, Liu et al. [25] experimentally found that this normalisation-and-denormalisation method is also effective without learnable parameters and named this revised design as *series stationarisation*. We apply series stationarisation to perform data pre-processing and post-processing. Concretely, normalisation is carried out on each sliding window over the temporal dimension. For instance, the normalisation for an individual input $\boldsymbol{x}$ can be formulated as $\boldsymbol{x}' = \frac{\boldsymbol{x}-\mu_x}{\sigma_x}$, where $\mu_x = \frac{1}{q}\sum_{j=1}^{q} x_{i,t-j}$ and $\sigma_x = \frac{1}{q}\sum_{j=1}^{q}(x_{i,t-j} - \mu_x)^2$. Suppose the corresponding forecasts $\widehat{x}'_{i,t}$ are obtained through inputting instance $\boldsymbol{x}'$ into the constructed model. Then, the denormalisation process transforms $\widehat{x}'_{i,t}$ into the eventual forecasting results using $\widehat{x}_{i,t} = \widehat{x}'_{i,t} \cdot \sigma_x + \mu_x$.

Baselines. The following baseline models are considered for model comparison.

- **TSAVG**: It was proposed by Neubauer and Filzmoser [29] in 2024. The k-nearest neighbour algorithm with DTW distances was utilised to form a neighbourhood of each time series, and the simple exponential smoothing is applied on each individual time series and the forecast is improved by averaging within its neighbourhood. Different averaging approaches include simple average, distance-weighted average and error-weighted average.
- **ARIMA**: Local ARIMA models are estimated on each individual time series [35] using *auto.arima*() in the R package *forecast* [21].
- **Pooled AR**: A pooled AR(q) model, formulated as $x_{i,t} = \beta_0 + \beta_1 x_{i,t-1} + \cdots + \beta_q x_{i,t-q} + \epsilon_{i,t}$, is constructed on the entire dataset as a tsGM, where $\beta_0, \beta_1, \cdots, \beta_q$ are fitted by ordinary least squares. The order q is determined following the practice of Neubauer and Filzmoser [29].
- **Multilayer Perceptron (MLP)**: It gains popularity due to its ability to model non-linear relationships [9]. We consider dense layers, and the used activation function is the tanh function.
- **LSTM**: It was introduced by Hochreiter and Schmidhuber [17] in response to the problem of long-term dependencies. We consider LSTM layers followed by dropout layers to avoid overfitting.
- **LSTM.Cluster**: Bandara et al. [1] proposed the cluster-and-then-model methodology. Feature-based clustering is performed via the k-means algorithm and an LSTM network is constructed per cluster. The optimal number of clusters is determined by evaluating the validation loss.

Evaluation metrics. We focus on one-step-ahead forecasting and use cumulative errors as evaluation metrics [29]. The cumulative one-step-ahead root mean squared error (RMSE), mean absolute error (MAE), and symmetric mean absolute percentage error (sMAPE) for the i-th time series have the following forms: $\text{RMSE} = \frac{1}{n_{test}^{(i)}}\sum_{\tau=1}^{n_{test}^{(i)}}\sqrt{\frac{1}{\tau}\sum_{t=T}^{T+\tau-1}(x_{i,t}-\widehat{x}_{i,t})^2}$, $\text{MAE} = \frac{1}{n_{test}^{(i)}}\sum_{\tau=1}^{n_{test}^{(i)}}\frac{1}{\tau}\sum_{t=T}^{T+\tau-1}|x_{i,t}-\widehat{x}_{i,t}|$, and $\text{sMAPE} = \frac{1}{n_{test}^{(i)}}\sum_{\tau=1}^{n_{test}^{(i)}}\frac{2}{\tau}\sum_{t=T}^{T+\tau-1}\frac{|x_{i,t}-\widehat{x}_{i,t}|}{|x_{i,t}|+|\widehat{x}_{i,t}|}$, respectively, where $n_{test}^{(i)}$ is the length of the test subset of the i-th time series, $i \in \{1, 2, \cdots, n\}$, and T is the forecasting origin.

4.2 Experimental Results

The in-sample and out-of-sample data, divided by the dataset itself, are used as the training and test datasets, respectively. For the TSAVG, ARIMA and Pooled AR models, the experiments are implemented consistently with the settings of Neubauer and Filzmoser [29]. While building MLP and LSTM neural networks, a validation dataset is needed to avoid overfitting and select the optimal hyperparameters. One subset of the training dataset is split as the validation dataset and the length is set as 10% of the training dataset. Time series cross-validation with the one-step-ahead rolling origin setup is performed [15]. The learning rate is set to be 0.002 at the beginning and it is automatically updated using LearningRateScheduler with the rule of $learning\ rate * 0.5^{epoch-1}$, and the number of epochs is 100 and early stopping is set based on the validation loss. Grid search is used to select the optimal hyperparameters and the ranges where the hyperparameters are selected are listed as follows: the input length $\in \{12, 24\}$, the number of layers $\in \{1, 2\}$, the number of nodes $\in \{4, 8, 16\}$, the dropout rate $\in \{0.2, 0.5\}$, and the batch size $\in \{32, 64\}$. The Adam optimiser is used [23]. All experiments are conducted on Google Colab using a GPU-enabled runtime. The primary model computations are executed on an NVIDIA Tesla T4 GPU (16 GB VRAM). The environment also included an Intel Xeon CPU (2 cores, 2.20GHz) and 12 GB RAM running Ubuntu 22.04.4.

The proposed two-stage models are developed based on the MLP and LSTM networks built above, namely, the MLP and LSTM models are the tsGM at stage one. The Ljung-Box test is performed, and the residual series with the p-value smaller than 0.05 indicate heterogeneity. At stage two, as shown in Figure 1, there are two options. The Type-I is to construct all tsLMs on non-white noise residuals, we use *auto.arima*() from the R package *forecast* to build ARIMA models. For the Type-II modelling method at stage two, the R package *tsfeatures* is utilised to extract the features of residual series, including acf_features, pacf_features, entropy, lumpiness, stl_features, arch_stat, nonlinearity, unitroot_kpss, unitroot_pp, holt_parameters, and hw_parameters, and one may refer to Hyndman et al. [19] for the meaning of each feature, and we then carry out k-means algorithm to divide the residual series into different clusters. We simplify the maintenance cost in Equation (4) to relate only to the number of models and ensure that the number of clusters does not exceed 10. After freezing weights and structure of the tsGM estimated at stage one, one more dense layer is added to it and the batch size at stage two decreases to 16, 8 or 4. We consider additive residuals for Type-I modelling and multiplicative residuals for Type-II according to experimental results. The forecasting performance is evaluated on the test dataset and the mean and median errors for each metric are calculated (see Table 2 [2]), where the best-performed ones are in bold and the second-best ones are underlined. The tsGMs, especially neural networks, present flexible forecasting and generalisation abilities and outperform tsLMs such as the ARIMA. The proposed models significantly outperform the TSAVG, which is designed specifically for heterogeneous time series forecasting. Compared with the pure tsGMs, the TS-X-I/II is able to effectively identify and model heterogeneity, and further boost the forecasting performance.

The M3 dataset comprises five subsets, resulting in a total of eight datasets. Based on the obtained mean sMAPE, the Friedman test [11] is conducted to detect significant differences among the MLP, LSTM, LSTM.Cluster, and TS-MLP/LSTM-I/II models. The p-value is 0.096, indicating that there are significant differences between the

[2] The code is available at https://github.com/R-jr-star/Two-stage-modelling-framework.

Table 2. Model comparison among proposed models and baselines in terms of cumulative RMSE, MAE and sMAPE.

			TSAVG	ARIMA	Pooled AR	MLP	LSTM	LSTM.Cluster	TS-MLP-I	TS-LSTM-I	TS-MLP-II	TS-LSTM-II
Tourism	RMSE	mean	4759.500	4119.036	2047.617	1913.457	1963.280	2192.981	1898.218	1903.357	**1892.543**	1916.188
		median	1005.795	929.144	597.842	511.050	495.335	509.936	503.737	**483.842**	499.367	490.785
	MAE	mean	3647.229	3190.488	1543.199	1539.330	1540.164	1809.114	1529.742	1536.206	**1513.960**	1515.407
		median	770.753	700.724	461.658	393.885	384.459	418.610	386.049	387.552	**375.761**	383.539
	sMAPE	mean	0.291	0.267	0.206	0.171	**0.167**	0.185	0.170	0.167	0.170	0.167
		median	0.262	0.241	0.158	0.136	0.136	0.153	0.138	0.142	0.136	**0.135**
M3	RMSE	mean	615.987	598.826	597.438	562.183	579.985	**557.876**	558.386	560.449	561.264	565.051
		median	406.703	395.709	389.837	328.168	358.744	356.812	**325.942**	337.669	329.176	338.711
	MAE	mean	484.660	470.654	475.890	464.153	478.653	463.335	**460.971**	462.290	462.659	465.618
		median	322.573	311.126	313.683	270.498	297.641	290.237	**268.922**	282.301	271.838	281.160
	sMAPE	mean	0.115	0.114	0.114	0.113	0.113	**0.110**	0.114	0.112	0.113	0.111
		median	0.069	0.065	0.066	0.055	0.062	0.061	**0.054**	0.057	0.055	0.058
CIF 2016	RMSE	mean	360462.500	301763.100	451571.800	293090.638	268568.973	279579.044	308556.696	266999.874	281854.068	**248097.484**
		median	93.481	96.834	32944.598	78.259	87.128	87.080	**52.036**	54.156	76.271	75.733
	MAE	mean	301584.200	233306.100	411428.100	239905.587	215052.708	227962.398	256551.538	207039.528	225569.258	**199321.157**
		median	82.064	79.007	32944.310	67.905	72.229	73.351	**44.982**	50.356	65.871	62.792
	sMAPE	mean	0.107	0.101	1.338	0.095	0.103	0.101	**0.082**	0.083	0.089	0.090
		median	0.084	0.080	1.592	0.071	0.086	0.082	0.057	**0.056**	0.071	0.069
Hospital	RMSE	mean	23.985	23.320	21.610	20.674	22.056	22.887	**20.596**	21.888	20.926	21.961
		median	8.080	8.022	8.357	7.833	7.725	7.663	7.825	**7.725**	7.944	7.753
	MAE	mean	19.610	19.040	17.643	17.651	18.739	19.408	**17.574**	18.654	17.816	18.635
		median	**6.437**	6.495	6.854	6.600	6.558	6.492	6.616	6.546	6.665	6.612
	sMAPE	mean	**0.168**	0.169	0.176	0.169	0.169	0.170	0.170	0.169	0.170	0.170
		median	0.158	0.161	0.169	0.154	0.154	0.159	0.154	**0.153**	0.154	0.156

performance of these models at a significance level of 0.1. The Nemenyi method [28] is employed as a post-hoc test to further evaluate four LSTM-involved models, and the Critical Distance (CD) diagram is plotted in Figure 2. The TS-LSTM-I and TS-LSTM-II are not significantly different from each other while the TS-LSTM-I model performs significantly better than the LSTM and LSTM.Cluster.

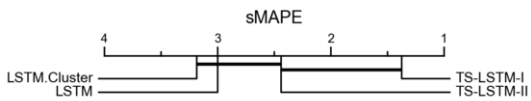


Figure 2. The CD diagram to visualise the differences among four LSTM-involved models in terms of mean sMAPE.

5 Discussion

Heterogeneity level. Table 3 presents the heterogeneity levels detected by taking the MLP or the LSTM as the tsGM at stage one, and the residuals are calculated using the additive model. The number of series (#Series), the number of sliding windows (#Samples), and the number of model parameters (#Parameters) are also listed. It can be found that even with the best-performed tsGM at stage one, the heterogeneity still exists and the level ranges from 9% to 76%. Stage two is necessary. Besides, we used the multiplicative model to obtain the heterogenous time series and applied the Type-II modelling method at stage two. Table 4 summarises the heterogeneity levels before and after stage two. Adding stage two is conducive to reducing the heterogeneity level.

Significance of global information. One commonly-used strategy to address heterogeneity is clustering. As shown in Figure 3(a), a sub-tsGM is trained per cluster and the LSTM.Cluster falls within this category. In this way, however, only sub-global information within cluster is considered. Beyond the methodology proposed in this study (see Figure 3(b)), there are several alternative approaches to incorporate global information. In Figure 3(c), heterogeneity is similarly identified by the developed tsGM. Rather than adding a heterogeneity module to the tsGM, the clustering-and-then-model method is directly applied to heterogeneity. In Figure 3(d), feature-based clustering is conducted on identified heterogeneity to form clusters. Within each cluster, the forecasts generated by the tsGM are concatenated with the corresponding time series data to serve as inputs for training a sub-tsGM, and the final forecasts are produced.

Table 5 presents a comparative analysis of these different strategies based on the mean values of sMAPE. To ensure a fair comparison, the sub-tsGMs employed in strategies (c) and (d) are developed as LSTM networks as well, and hyperparameters and architectural configurations keep consistent with the setting of heterogeneity module of the TS-LSTM-II model. One-tail pairs t-tests are carried out to compare

Table 3. The heterogeneity levels identified by tsGMs of the MLP and LSTM using the additive model.

Dataset	Global model	#Series	#Samples	#Parameters	n_h	R_h
Tourism	MLP	366	91712	241	186	50.82%
	LSTM			1993	198	54.10%
Hospital	MLP	767	36816	417	92	11.99%
	LSTM			361	91	11.86%
CIF 2016	MLP	72	5470	113	34	47.22%
	LSTM			361	37	51.39%
M3-demo	MLP	111	9028	417	70	63.06%
	LSTM			1225	85	76.58%
M3-finance	MLP	145	11948	417	59	40.69%
	LSTM			1225	73	50.34%
M3-industry	MLP	334	32739	417	125	37.43%
	LSTM			1225	171	51.20%
M3-macro	MLP	312	27731	417	133	42.63%
	LSTM			1225	158	50.64%
M3-micro	MLP	474	24009	417	56	11.81%
	LSTM			1225	44	9.28%

Table 4. The change of heterogeneity levels before and after the Type-II modelling method at stage two using the multiplicative model.

Dataset	Global model	Before n_h	Before R_h	K	#Parameters	After n_h	After R_h
Tourism	MLP	139	37.98%	5	173	64	17.49%
	LSTM	162	44.26%	5	173	74	20.22%
Hospital	MLP	81	10.56%	3	461	10	1.30%
	LSTM	88	11.47%	3	173	29	3.78%
CIF 2016	MLP	33	45.83%	2	137	16	22.22%
	LSTM	35	48.61%	2	89	14	19.44%
M3-demo	MLP	59	53.15%	3	461	34	30.63%
	LSTM	84	75.68%	3	173	52	46.85%
M3-finance	MLP	63	43.45%	3	461	18	12.41%
	LSTM	67	46.21%	3	173	31	21.38%
M3-industry	MLP	118	35.33%	5	461	46	13.77%
	LSTM	169	50.60%	3	173	77	23.05%
M3-macro	MLP	128	41.03%	3	461	53	16.99%
	LSTM	156	50.00%	3	173	64	20.51%
M3-micro	MLP	40	8.44%	3	461	14	2.95%
	LSTM	37	7.81%	3	173	16	3.38%

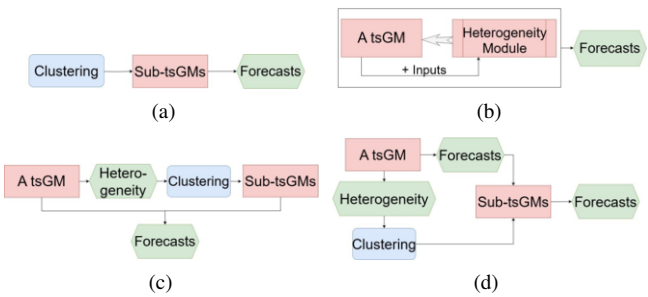


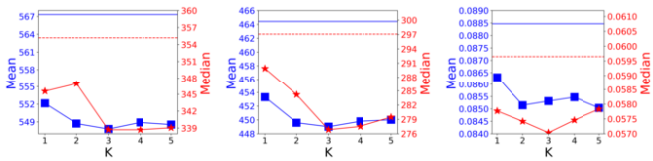
Figure 3. Different strategies to address heterogeneity.

strategies (b)–(d) and (a) with the significance level of 0.1. It can be obtained from p-values that integrating global information of the entire dataset leads to statistically significant accuracy improvements, and the improvement demonstrated by the proposed TS-LSTM-II is the most significant.

Table 5. Evaluation of various strategies for addressing heterogeneity in terms of mean values of cumulative sMAPE.

	(b) TS-LSTM-II	(c)	(d)	(a) LSTM.Cluster
Tourism	0.1672	0.1681	0.1681	0.1850
CIF 2016	0.0897	0.0843	0.0900	0.1018
Hospital	0.1696	0.1691	0.1690	0.1698
M3-demo	0.0414	0.0413	0.0421	0.0442
M3-finance	0.0722	0.0712	0.0716	0.0755
M3-industry	0.0851	0.0863	0.0854	0.0901
M3-macro	0.0366	0.0375	0.0365	0.0377
M3-micro	0.2069	0.2081	0.2073	0.1978
p-value	0.094*	0.102	0.097*	

Sensitivity analysis on K. The maintenance cost in Equation (4) constraints both K and the number of series in each cluster. In practice, a form of $f_m(.)$ can be linear or nonlinear as different companies have various strategies [32]. The sensitivity of the number of clusters K is analysed on the test dataset. M3-industry is taken as an example for illustration (see Figure 4). The tsGM used at stage one is LSTM, and two horizontal lines are the values of corresponding test errors when only the first stage is considered. It further affirms Proposition 3 that there exists an optimal K to realise a trade-off and minimise the out-of-sample error. Experimental results reveal the optimal number of clusters is generally between 2 to 5 (incl.).



(a) RMSE (b) MAE (c) sMAPE

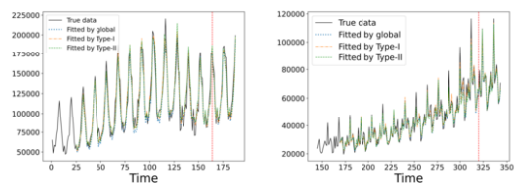
Figure 4. Mean and median RMSE, MAE and sMAPE on M3-industry with different K (Square markers and the solid horizontal line – mean error).

Runtime analysis. The results on runtime of LSTM-involved models across different datasets are summarised in Table 6. The Type-I modelling at stage two is time-efficient. The TS-LSTM-II involves strenuous computation although it achieves more accurate forecasting. The increase in running time is primarily due to the small batch size employed at stage two to prevent overfitting.

Table 6. Runtime (in minutes) for each model across different datasets.

	LSTM	LSTM.Cluster	TS-LSTM-I	TS-LSTM-II
Tourism	10.75	10.95	11.20	22.24
Hospital	4.17	6.82	4.66	7.94
CIF 2016	0.71	1.72	0.72	3.16
M3-demo	1.11	2.85	1.55	7.98
M3-finance	1.17	3.90	1.46	6.33
M3-industry	3.54	10.24	4.18	18.40
M3-macro	2.52	8.74	3.01	16.19
M3-micro	2.35	8.26	2.47	4.50

Visualisation of forecasting results. With the LSTM as the tsGM at stage one, the stage two leads to effective adjustments to generated forecasts (see Figure 5).



(a) The 1-st time series (b) The 112-nd time series

Figure 5. The plots of true values and forecasts on the Tourism dataset, where the red vertical dashed line separates the training and the test datasets.

6 Conclusion

Considering characteristics of both time series and modelling methods, this paper proposed a two-stage modelling framework to boost global time series forecasting models on heterogeneous datasets. It can leverage local information of individual series, sub-global information within each sub-group, and global information across the entire dataset. This provides an alternative modelling approach to practitioners and researchers. Alternative techniques on data normalisation such as SAN [26] can be considered in future work.

Acknowledgements

This work was supported by the Economic and Social Research Council of UK (ES/P00072X/1: 2617249, ESRC Standard Research Studentship: 22020946).

References

[1] K. Bandara, C. Bergmeir, and S. Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140:112896, Feb. 2020. ISSN 0957-4174. doi: 10.1016/j.eswa.2019.112896.

[2] J. Berner, P. Grohs, G. Kutyniok, and P. Petersen. *The Modern Mathematics of Deep Learning*, pages 1–111. Cambridge University Press, Dec. 2022. ISBN 9781316516782. doi: 10.1017/9781009025096.002.

[3] H. Binder, O. Gefeller, M. Schmid, and A. Mayr. The evolution of boosting algorithms: From machine learning to statistical modelling. *Methods of Information in Medicine*, 53(06):419–427, 2014. ISSN 2511-705X. doi: 10.3414/me13-01-0122.

[4] J. Chen, J. E. Lenssen, A. Feng, W. Hu, M. Fey, L. Tassiulas, J. Leskovec, and R. Ying. From similarity to superiority: Channel clustering for time series forecasting. *Advances in Neural Information Processing Systems*, 37:130635–130663, 2024.

[5] E. G. da Silva, P. S. de Mattos Neto, and J. F. de Oliveira. Hybrid system for time series using iterative residual forecasting models. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 872–877. IEEE, Oct. 2019. doi: 10.1109/bracis.2019.00155.

[6] G. Elliott, T. J. Rothenberg, and J. H. Stock. Efficient tests for an autoregressive unit root. *Econometrica*, 64(4):813, July 1996. ISSN 0012-9682. doi: 10.2307/2171846.

[7] P. Ellis. Tcomp: Data from the 2010 tourism forecasting competition. *R package version*, 1(1), 2018.

[8] R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50 (4):987–1007, 1982. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1912773.

[9] S. Etemadi, M. Khashei, and S. Tamizi. Etemadi reliability-based multilayer perceptrons for classification and forecasting. *Information Sciences*, 651:119716, Dec. 2023. ISSN 0020-0255. doi: 10.1016/j.ins.2023.119716.

[10] P. R. A. Firmino, P. S. de Mattos Neto, and T. A. Ferreira. Error modeling approach to improve time series forecasters. *Neurocomputing*, 153: 242–254, Apr. 2015. ISSN 0925-2312. doi: 10.1016/j.neucom.2014.11.030.

[11] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1): 86–92, Mar. 1940. ISSN 0003-4851. doi: 10.1214/aoms/1177731944.

[12] S. Fröhwirth-Schnatter and S. Kaufmann. Model-based clustering of multiple time series. *Journal of Business and Economic Statistics*, 26(1):78–89, Jan. 2008. ISSN 1537-2707. doi: 10.1198/073500107000000106.

[13] R. Godahewa, K. Bandara, G. I. Webb, S. Smyl, and C. Bergmeir. Ensembles of localised models for time series forecasting. *Knowledge-Based Systems*, 233:107518, Dec. 2021. ISSN 0950-7051. doi: 10.1016/j.knosys.2021.107518.

[14] Z. Hajirahimi and M. Khashei. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, 86:83–106, Nov. 2019. ISSN 0952-1976. doi: 10.1016/j.engappai.2019.08.018.

[15] H. Hewamalage, K. Ackermann, and C. Bergmeir. Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2):788–832, Dec. 2022. ISSN 1573-756X. doi: 10.1007/s10618-022-00894-5.

[16] H. Hewamalage, C. Bergmeir, and K. Bandara. Global models for time series forecasting: A simulation study. *Pattern Recognition*, 124: 108441, 2022. doi: 10.1016/j.patcog.2021.108441.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[18] R. Hyndman, M. Akram, C. Bergmeir, M. O'Hara-Wild, and M. R. Hyndman. Package 'mcomp'. *R package version*, 2, 2018.

[19] R. Hyndman, Y. Kang, P. Montero-Manso, T. Talagala, E. Wang, Y. Yang, M. O'Hara-Wild, et al. tsfeatures: Time series feature extraction. *R package version*, 1(0), 2019.

[20] R. J. Hyndman. Expsmooth: Data sets from forecasting with exponential smoothing. *R package version 2.3*, 2015.

[21] R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: Theforecastpackage forr. *Journal of Statistical Software*, 27(3), 2008. ISSN 1548-7660. doi: 10.18637/jss.v027.i03.

[22] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

[23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] S. Li and Y. Liu. Sharper generalization bounds for clustering. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6392–6402. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21k.html.

[25] Y. Liu, H. Wu, J. Wang, and M. Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[26] Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36:14273–14292, 2023.

[27] P. Montero-Manso and R. J. Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, 37(4):1632–1653, Oct. 2021. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2021.03.004.

[28] P. B. Nemenyi. *Distribution-free multiple comparisons.* Princeton University, 1963.

[29] L. Neubauer and P. Filzmoser. Improving forecasts for heterogeneous time series by "averaging", with application to food demand forecasts. *International Journal of Forecasting*, 40(4):1622–1645, Oct. 2024. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2024.02.002.

[30] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. B. Taieb, C. Bergmeir, R. J. Bessa, J. Bijak, J. E. Boylan, et al. Forecasting: theory and practice. *International Journal of Forecasting*, 38(3):705–871, 2022. doi: 10.1016/j.ijforecast.2021.11.001.

[31] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

[32] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28, 2015.

[33] A.-A. Semenoglou, E. Spiliotis, S. Makridakis, and V. Assimakopoulos. Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, 37(3):1072–1084, July 2021. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2020.11.009.

[34] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[35] R. H. Shumway and D. S. Stoffer. *Time Series Regression and ARIMA Models*, pages 89–212. Springer New York, 2000. ISBN 9781475732610. doi: 10.1007/978-1-4757-3261-0_2.

[36] S. Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, Jan. 2020. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2019.03.017.

[37] T. S. Talagala, F. Li, and Y. Kang. Fformpp: Feature-based forecast model performance prediction. *International Journal of Forecasting*, 38(3):920–943, July 2022. ISSN 0169-2070. doi: 10.1016/j.ijforecast.2021.07.002.

[38] T. Teräsvirta, C.-F. Lin, and C. W. Granger. Power of the neural network linearity test. *Journal of time series analysis*, 14(2):209–220, 1993.

[39] H. J. Thiébaux and F. W. Zwiers. The interpretation and estimation of effective sample size. *Journal of Climate and Applied Meteorology*, 23(5):800–811, May 1984. ISSN 0733-3021. doi: 10.1175/1520-0450(1984)023<0800:tiaeoe>2.0.co;2.

[40] A. P. Wellens, N. Kourentzes, and M. Udenio. When and how to use global forecasting methods on heterogeneous datasets. *Available at SSRN 4629272*, 2023.

[41] G. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, Jan. 2003. ISSN 0925-2312. doi: 10.1016/s0925-2312(01)00702-0.