Pattern Recognition xxx (xxxx) xxx

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Global models for time series forecasting: A Simulation study

Hansika Hewamalage[a,*], Christoph Bergmeir[a], Kasun Bandara[b]

[a] Dept of Data Science and AI, Faculty of IT, Monash University, Melbourne, Australia
[b] School of Computing and Information Systems, Melbourne Centre for Data Science, University of Melbourne, Melbourne, Australia

ARTICLE INFO

Article history:
Received 9 March 2021
Revised 3 November 2021
Accepted 18 November 2021

Keywords:
Time series forecasting
Global forecasting models
Time series simulation
Data generating processes

ABSTRACT

The recent advances in Big Data have opened up the opportunity to develop competitive Global Forecasting Models (GFM) that simultaneously learn from many time series. Although, the concept of series relatedness has been heavily exploited with GFMs to explain their superiority over local statistical benchmarks, this concept remains largely under-investigated in an empirical setting. Hence, this study attempts to explore the factors that affect GFM performance, by simulating a number of datasets having controllable characteristics. The factors being controlled are along the homogeneity/heterogeneity of series, the complexity of patterns in the series, the complexity of forecasting models, and the lengths/number of series. We simulate time series from simple Data Generating Processes (DGP), such as Auto Regressive (AR), Seasonal AR and Fourier Terms to complex DGPs, such as Chaotic Logistic Map, Self-Exciting Threshold Auto-Regressive and Mackey-Glass Equations. We perform experiments on these datasets using Recurrent Neural Networks (RNN), Feed-Forward Neural Networks, Pooled Regression models and Light Gradient Boosting Models (LGBM) built as GFMs, and compare their performance against standard statistical forecasting techniques. Our experiments demonstrate that with respect to GFM performance, relatedness is closely associated with other factors such as the availability of data, complexity of data and the complexity of the forecasting technique used. Also, techniques such as RNNs and LGBMs having complex non-linear modelling capabilities, when built as GFMs are competitive methods under challenging forecasting scenarios such as short series, heterogeneous series and having minimal prior knowledge of the data patterns.

1. Introduction

In many industries, such as retail, energy, ride-share, and tourism, generating accurate forecasts plays a crucial role in business decision-making scenarios. For example, for retailers such as Amazon and Walmart, sales demand forecasting is important as it provides better grounds for optimising their product inventories [5,52]. In the energy sector, demand forecasts are used to determine the fuel allocation, economic dispatch, and others [36], whereas accurate service demand forecasts across different geographies are essential for industries such as tourism and healthcare [2,11]. In ride-share services such as Uber, accurate prediction of passenger demand during extreme events can help with better resource allocation and with budget planning in advance [56].

The paradigm in time series forecasting throughout decades has been to treat every time series as an independent dataset. As a result, traditional forecasting techniques are local, consider each time series separately and forecast it in isolation. The Exponential Smoothing State Space Model [ETS, 23] and Auto-Regressive Integrated Moving Average Model [ARIMA, 9] are the most prominent examples of such methods. However, nowadays, many companies are collecting large quantities of time series from similar sources routinely, such as sales in retail of thousands of different products, measurements for predictive maintenance across many machines, smart meter data across many households, etc. Though traditional local forecasting techniques can still deal with forecasting under these circumstances, they leave the huge potential of learning patterns across time series untapped.

Due to this reason, there has been a paradigm shift in forecasting recently, where now a set of series, as opposed to just one series, is seen as a dataset. Then, a Global Forecasting Model (GFM) [25] is trained across all the series in the dataset. The global model has the same set of parameters (e.g., the weights if the global model is a neural network) for all the series in contrast to a local model which has a different set of parameters for every in-

* Corresponding author at: Faculty of Information Technology, PO Box 63 Monash University, Victoria 3800, Australia.
E-mail addresses: Hansika.Hewamalage@monash.edu (H. Hewamalage), Christoph.Bergmeir@monash.edu (C. Bergmeir), kasun.bandara@unimelb.edu.au (K. Bandara).

https://doi.org/10.1016/j.patcog.2021.108441
0031-3203/

dividual series [25]. Nowadays, such global models for forecasting are being introduced at a fast pace, e.g., in the works of Bandara et al. [4], Salinas et al. [42], Wen et al. [52]. They are quickly making their way into practice, and have won all recent prestigious forecasting competitions, such as the M4 [34] and M5 competition [35], and all competitions held in recent years on the Kaggle platform with a forecasting task [8].

The premise under which these works operate, and how they explain the almost unreasonable effectiveness of global models, is that the series have to be "related" in some way [4,41,42,51,52], so that the global model can extract patterns across the series. However, none of the works attempts to define or analyse the characteristics of such "relatedness". While some studies have associated high relatedness of series with higher similarity of the shapes/patterns of the series and higher cross-correlation among series [42,51], others have also suggested that the more a set of extracted features for the considered series are similar, the higher the relatedness [4]. Nevertheless, global models seem to work well even in situations where series are clearly unrelated, such as the M4 forecasting competition, whose dataset is a broad mix of unaligned time series across many different domains. The competition was won by ES-RNN, a global recurrent neural network [43].

Thus, the problem of understanding when and why GFMs work, is arguably the most important open problem currently in time series forecasting. The first work to offer explanations in this space is the recent work by Montero-Manso and Hyndman [39], which demonstrates theoretically that, no matter how heterogeneous the data are, there always exists a GFM for any dataset that can perform equally well or even better than a collection of local models. Thus, GFMs are in theory not in any way more restricted than local ones, and series do not have to be "related" for GFMs to perform well. However, the mere existence of such a global model does not mean it is straightforward to find or construct such a model. Instead of considering relatedness, that study focuses on model complexity. Due to exploiting more data, global models can afford to be more complex than local ones while still generalising better. Montero-Manso and Hyndman [39] then argue that the complexity of global models can be controlled mainly by using 1) more lags as inputs, 2) using non-linear, non-parametric models, and 3) using data partitioning. Those authors then proceed to confirm and illustrate their findings empirically using real-world datasets.

Despite the insights of the work of Montero-Manso and Hyndman [39], many questions in this space are still unanswered, around under which circumstances GFMs work, and how model complexity is best introduced for datasets with different amounts of data complexity, data availability, and relatedness. Our work is mainly motivated by this lack of understanding around the concept of relatedness for GFMs. In this work, we explore this concept by adopting a model-based definition for the relatedness of series. A model-based definition is a concept adopted often in situations when the concept to grasp is difficult. Examples for these are Teräsvirta NN test [46] and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) [31] test for the concepts of non-linearity and non-stationarity respectively.

In the work by Fulcher and Jones [16], the coefficients of a model fitted to a series are regarded as features of that particular time series. Hence, the similarity of the coefficients of local models fitted to each series can be used to estimate the relatedness of the respective series, in the same way as checking the similarity of features extracted from time series. The higher the similarity of the coefficients, the more related those series are, with respect to that particular model. This covers the gist of how GFMs function where the same model coefficients are applied to each time series in the dataset.

However, similar to the findings by Montero-Manso and Hyndman [39], this form of model-based relatedness of series is a concept that goes together with other factors such as the complexity of the underlying model, the amount of data available etc. This is because with a model-based definition of relatedness, the question is how well the model fits the data. Thus, to have full control of the relatedness of the time series in a dataset, we base our work on simulated time series with known Data Generating Processes (DGP). A homogeneous set of series are all simulated from the same DGP, whereas a heterogeneous set consists of series simulated from different DGPs. In this respect, we consider a homogeneous set of series as "related" and a heterogeneous set as "unrelated". Therefore, the model-based definition of relatedness of series in our work is associated with the parameters of the known DGPs. This also gives us full control over the complexity of the patterns in the data and the availability of data, rather than relying on real-world datasets which may contain a mixture of many different characteristics or none at all. We perform a comprehensive experimental study under controlled conditions, to carefully analyse and quantify the interplay between complexity of the DGP, complexity of the model, amount of data available, and the "relatedness" of the series.

Our study has the following key contributions. We simulate different datasets using a number of carefully chosen DGPs which cover a variety of time series characteristics, most of them closely simulating real-world scenarios. The DGPs that we have chosen vary from simple linear AR(3), Seasonal Auto-Regressive (SAR), Fourier Terms based DGPs to more complex and non-linear DGPs such as Chaotic Logistic Map, Mackey-Glass Equation and Self-Exciting Threshold Auto-Regressive (SETAR) models. The simulation settings that we use in our study along with the respective DGP implementations are available publicly as a code repository.[1] We have designed several experimental scenarios with each one of these DGPs. To explore the effect of data availability on the model performance, we vary the amount of data in the datasets in terms of both the number of series as well as the lengths of the individual series. For different degrees of data relatedness, we investigate both homogeneous and heterogeneous settings by changing the number of DGPs mixed within the same dataset. Thirdly, we perform experiments with a number of selected GFMs including Recurrent Neural Networks (RNN), Pooled Regression (PR) models, Feed-Forward Neural Networks (FFNN) and Light Gradient Boosting Machines [LGBM, 29] on the simulated datasets under the different experimental scenarios. The complexity of the different GFMs is further changed by introducing two model setups, namely the GFM.All setup where all the series are fitted using the same global model and the GFM.Cluster setup, which builds one GFM for each cluster of series from a different DGP within a heterogeneous context. A quantitative approximation of the relative complexities of these forecasting techniques is also provided in our work. Finally, the performance of the models is compared against each other as well as against the statistical techniques such as AR, SETAR DHR-ARIMA etc. Based on the empirical evidence, an analysis is provided around the different factors which affect the performance of different GFMs. In addition to the simulated datasets, we also include real-world datasets in our experimentation to reinforce our findings by using potentially homogeneous/heterogeneous datasets in the real-world.

The rest of the paper is structured as follows. Section 2 first details the overall methodology employed in the study including the different experimental scenarios designed. Next, Section 3 provides a brief review of the different studies which involve time series simulation, followed by the details of the DGPs and real-world datasets involved in this study. In Section 4, we provide the details of the different forecasting techniques that are compared

[1] Available at: https://github.com/HansikaPH/tssimulate

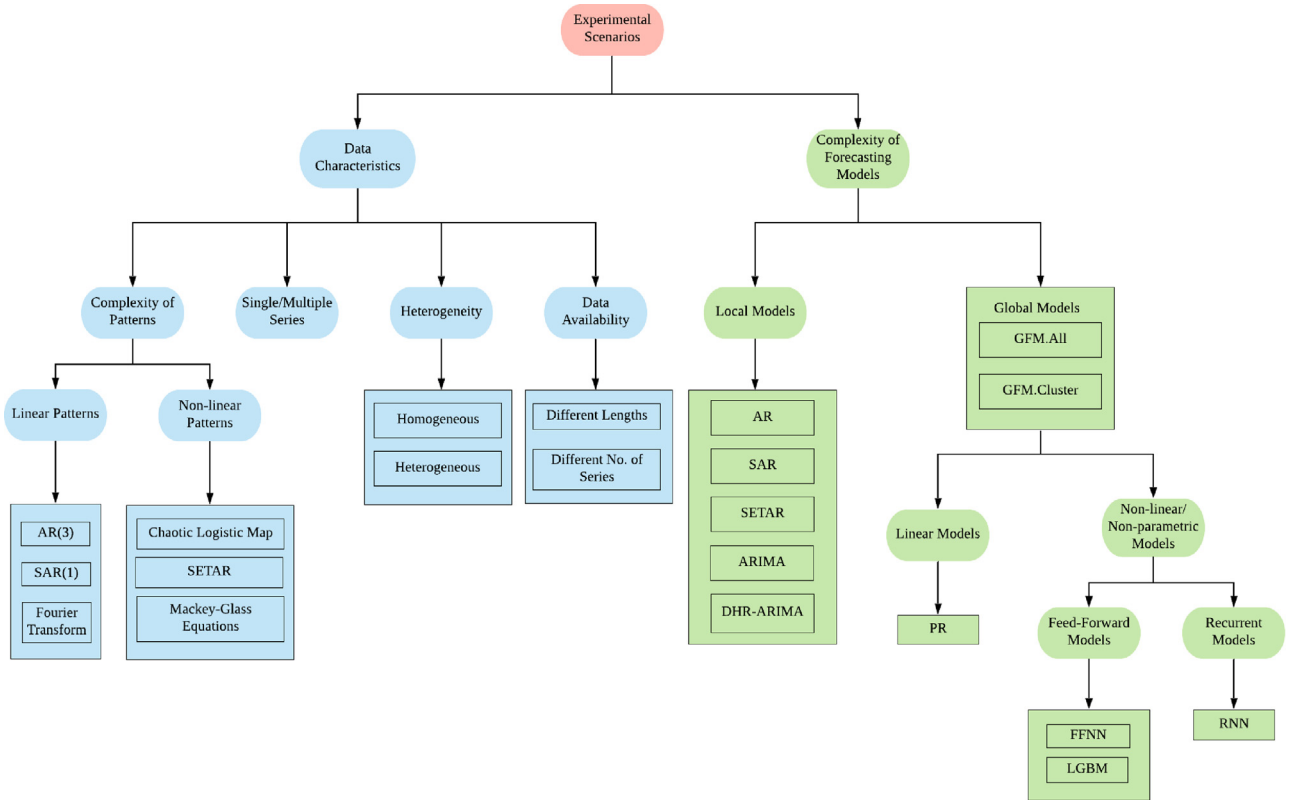

Fig. 1. Visualisation of experimental settings employed in the study for the simulation scenarios.

against each other in the study along with their data preprocessing, model training and testing information. Section 5 presents the details of the dataset characteristics along with the evaluation metrics as well as the tests for statistical significance of the performance differences between the methods. In Section 6, we present an analysis of the results of the experiments, a comparison of the complexities of the models used in the GFM.Cluster setup as well as a comparison of the computational times of the employed forecasting techniques. Section 7 concludes the paper with the final remarks of the overall study, the conclusions derived and the way forward. We also have an Online Appendix Supplementary materials where we include further details of the used DGPs, forecasting techniques, experimental setup and results analysis.

2. Overview of the methodology for simulated scenarios

The methodology of our study involves first simulating a number of datasets having many different characteristics. Once the datasets are simulated as required, different forecasting techniques are tested on them. We introduce a host of experimental scenarios to quantitatively explore the interplay between the complexity of different forecasting models and characteristics of the data in this manner. In our experiments, characteristics of the data are controlled along four dimensions: 1) complexity, 2) single or multiple series, 3) heterogeneity, and 4) amount of data points, whereas forecasting models are controlled using different levels of complexity. The specific details of these controlled experimental settings are explained in this section, and Fig. 1 gives an overview.

2.1. Complexity of the patterns in the data

The complexity of the simulated time series is determined by the complexity of the underlying DGP. We start from AR(3), the simplest form of DGP, which is linear with short memory. Then, we gradually make them more complex by increasing the number of lags to have a linear DGP with longer memory; the SAR(1). Longer memory DGPs are made further complex by having multiple seasonalities using Fourier Terms. Next, we introduce more complex, non-linear DGPs such as SETAR which is a threshold linear model that switches between different linear models based on certain conditions, a Chaotic Logistic Map DGP and another delay differential equation based model, the Mackey-Glass equations.

2.2. Single or multiple series

With every DGP used, series are both used for single series settings and split into multiple equal length series to constitute a scenario with multiple series. The purpose of this experimental scenario is to quantitatively evaluate whether it makes any difference for the models under consideration to learn from the same data either on one long series or across multiple series. For both the scenarios, we evaluate on the same forecast horizon. This means that for the multiple series scenarios, the evaluation is performed on the test forecast horizon of the series that corresponds to the last segment of the relevant long single series. In terms of the forecasting techniques, this indicates that the techniques that can be built as global models can leverage that facility when in the multiple series mode. However, for local forecasting models such as AR and SETAR models, learning is always restricted to use only one series at a time. Consequently, in single time series scenarios, such local models have more training data than in the multiple series scenarios.

2.3. Scale of heterogeneity

As explained in Section 1, the relatedness of series as we use it in this study is the same as the homogeneity of the series induced by the DGP. Homogeneity is when all the series are gener-

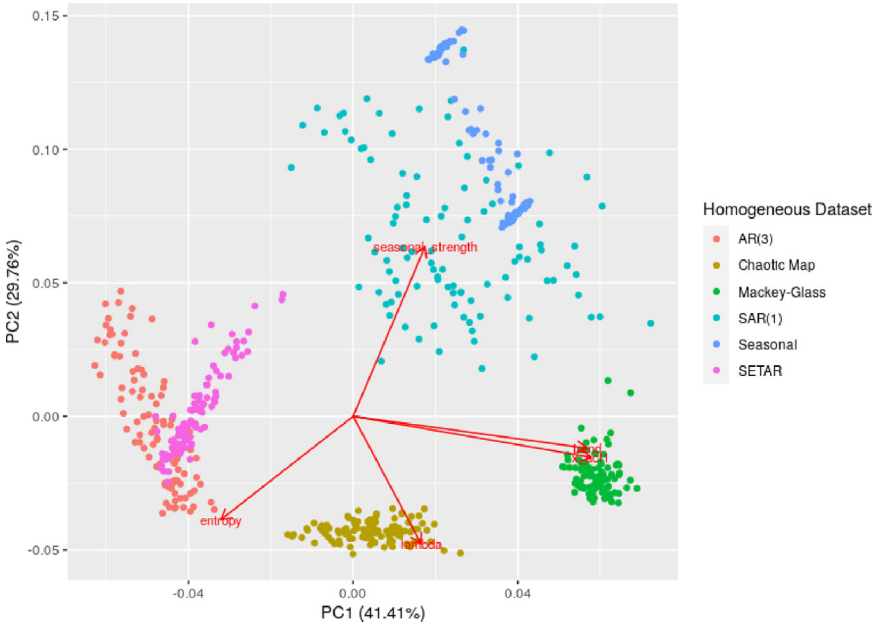

Fig. 2. Tsfeature plots for the homogeneous datasets.

ated from the same DGP (same set of coefficients) with different seeds (different initial values). Heterogeneous series are created by generating them from the same process (same DGP) with different coefficients. Different patterns can also be created by using the same DGP by adding Gaussian noise over the coefficients. Likewise, we control the scale of heterogeneity of time series by simulating the following scenarios using the aforementioned DGPs.

1. Homogeneous (Single) Pattern
2. Heterogeneous Patterns (Only for Multiple Series)

Homogeneity is considered equivalent to relatedness of the series in our work. On the other hand, heterogeneity is considered analogous to unrelatedness. There can also be different levels of relatedness/heterogeneity depending on how far the DGPs are mixed in together. To understand how this form of series relatedness compares with other common ideas in the domain such as time series feature similarity, we also include an analysis of *tsfeatures* [22] of the series belonging to different datasets [28]. As in the work by Kang et al. [27], we visualise this feature similarity of time series using a 2-D plot after reducing the feature dimensionality by applying Principal Component Analysis (PCA) [26]. Similar to the approach by Kang et al. [27], we select a subset of the most important features that drive the forecastability of a series, namely, spectral entropy, strength of trend, strength of seasonality, first order autocorrelation and optimal Box-Cox transformation parameter, and apply PCA on them.

As seen in Fig. 2, the series of the different types of datasets simulated from the relevant DGPs tend to be concentrated in different regions in the 2-D space with respect to *tsfeatures*. This indicates that the homogeneity of the series simulated from one specific DGP corresponds to the similarity of the same series with respect to *tsfeatures*. Fig. 3, shows the spread of the series generated from heterogeneous DGPs when projected on to the *tsfeature* space. As expected, in Fig. 3, when considering a single heterogeneous DGP, all the series generated from it tend to be more scattered in the *tsfeatures* space, which indicates their dissimilarity in terms of *tsfeatures*. This also shows that with the heterogeneous datasets, we are exploring a comparatively more diverse set of series (in terms of *tsfeatures*) within a single dataset.as seen

Another option for simulating heterogeneous datasets is by mixing different DGPs (such as AR(3), Chaotic Logistic Map) altogether within the same dataset. As seen from the *tsfeature* plot in Fig. 2, this technique can be used to achieve even more diversity in the datasets as well as to control the scale of relatedness/homogeneity by changing the number of DGPs mixed together. Therefore, we include an additional experiment where we keep increasing the number of DGPs mixed together as in 1 DGP, 2 DGPs up to 5 DGPs combined in a single dataset. The heterogeneity of the datasets also increases proportional to the number of DGPs mixed together. This allows to perform further analysis on the impact of relatedness on the GFM performance.

2.4. Amount of data available

We also control the availability of data by changing the lengths of the individual series or varying the number of series in the dataset. For single series and multiple series scenarios, this is performed using two different setups.

1. Single Series Scenarios - Varying the length of the series
2. Multiple Series Scenarios
   (a) Homogeneous Patterns - Varying the lengths of the series, Varying the number of series
   (b) Heterogeneous Patterns - Varying the lengths of the series

For the single series scenario, we can only change the length of the single series. However, for the multiple series scenarios with a single pattern (homogeneous), we simulate both with varying number of series and varying lengths of the series. For the multiple series scenarios with heterogeneous patterns, we only vary the amount of data by changing the lengths of the series.

Based on the different experimental considerations relevant to each DGP, such as data heterogeneity or single and multiple series inclusion, we introduce the terminology summarised in Table 1 that is used throughout the rest of this paper.

2.5. Complexity of forecasting techniques

The modelling capability of models is varied by conducting experiments using a number of techniques with different attributes

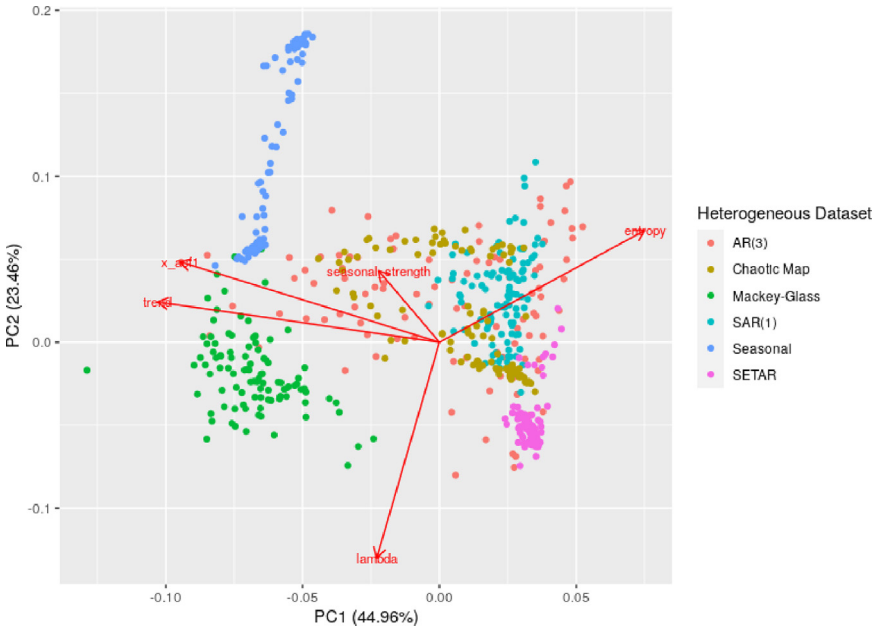

Fig. 3. Tsfeature plots for the heterogeneous datasets.

Table 1
Explanations of abbreviations.

Abbreviation	Meaning
SS	Single Series
MS-Hom-Short	Many Series Homogeneous with Short Lengths
MS-Hom-Long	Many Series Homogeneous with Long Lengths
MS-Het	Many Series Heterogeneous

as explained more under Section 4. The complexity of these base models can be further increased by using two techniques. The first is by increasing the number of parameters in the model. For NNs, the hyperparameters are tuned by using automated hyperparameter tuning techniques. However, the size of the input window can be increased as required to improve the complexity. For pooled regression models this can be done by increasing the lag size. The second approach is to introduce different training paradigms for global models as mentioned below.

1. GFM.All: Where the global models are trained using all the series available in the dataset irrespective of their potential heterogeneity. This setup is used in all the multiple series experimental scenarios.
2. GFM.Cluster: Where multiple global models are built for different subsets of the same dataset and the final accuracy computed by combining the forecasts from the different clusters. This is used to give special attention to existing heterogeneity in the data by clustering similar series together [4]. However, this training paradigm is irrelevant for the heterogeneous setting that we described before, since none of the series in the dataset share coefficients. Therefore, we introduce a separate experimental scenario named Cluster setup where we mix few patterns from several Chaotic Logistic Map DGPs within a single dataset. Also, how the model training is done with the clustering depends on the individual model classes.

The GFM.All setup is the base setup. The GFM.Cluster setup is expected to increase the complexity of the GFM.All setup. We also include a quantitative estimation for the complexity of the models. The idea of degrees of freedom has been introduced in the work by Franklin [15] to define the complexity of a model. Kolmogorov complexity, also known as algorithmic complexity and first defined formally by Kolmogorov [30], is another technique for this. Kolmogorov complexity of a certain object (such as a string of text) is defined as the length of the shortest program, with respect to a certain programming language, defined as a Universal Turing Machine (UTM), that produces the particular object as an output. Kolmogorov complexity is claimed to be theoretically incomputable [57]. However, the recent work by Cilibrasi and Vitanyi [10] has shown that the UTM of the Kolmogorov complexity can be approximated using the decompressor of modern real-world compression techniques. Thus, Kolmogorov complexity of an object can be approximated by the length of a minimum and efficient description for that object produced from a lossless compressor. Therefore, to compute model complexity, we use the size of the Gzip compressions of the models used specifically in our Cluster based experiment scenario. Although this measure of algorithmic complexity of models could be data-dependent similar to the Minimum Message Length concept [50], since in our setup all models involve the same data, we do not include the data in the compression. The flowchart in Fig. 4 shows the step by step process of estimating the model complexity this way.

3. Datasets

This section contains details of the simulation of the datasets as well as the real-world datasets used in this study.

3.1. Data generating processes

Table 2 summarises the literature related to time series simulation in different domains. Most of the work present in the literature builds local models for forecasting as opposed to GFMs. Our interest in this study lies specifically around those methods that can generate time series with controllable features, rather than random generation processes. Moreover, the DGPs are selected such that they can simulate practical forecasting scenarios as closely as possible. This section gives details of the DGPs that we use for our experiments in order to control the complexity of the patterns in the data. For further details on DGPs, refer to Appendix B of our Online Appendix.

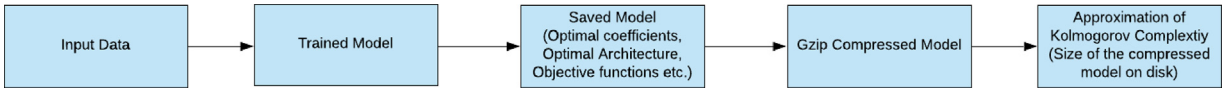

Fig. 4. Steps for approximating model complexity using the concept of Kolmogorov Complexity. 1) Prepare the input data 2) Train the models using the input data 3) Save the trained model including all its distinctive objects such as optimal coefficients, architecture, objective functions, metrics, optimizers etc. 4) Compress the saved model using Gzip compression technique 5) Use the size of the compressed model on the disk as the algorithmic complexity (Kolmogorov Complexity) of that model.

Table 2
Related Work of time series simulation in different domains.

Domain	Literature
Time Series Classification	Sun et al. [44], Zhao and Itti [55]
Time Series Forecasting	Bergmeir and Benítez [6], Bergmeir et al. [7]
Machine Learning	Lau and Wu [32], Ye and Dai [53], Vanli et al. [49], Fischer et al. [14], Zhang and Qi [54]

3.1.1. Linear Auto-Regressive models

AR models use a linear combination of the past values of the target series itself as regressors to predict the future values. Therefore, AR DGPs can simulate those time series common in the real world where the value at a certain time is a (linear) function of its own past values. In this study, we generate series using both Non-Seasonal AR models as well as Seasonal AR models to enforce linear relationships in the simulated data.

3.1.2. Chaotic Logistic Map

The Chaotic Logistic Map DGP is another technique used in this study to generate more complex patterns in the time series data. This type of logistic models was first introduced in biological research to describe the population growth of certain species over time [38]. Chaos theory is used in mathematics to deal with heavily non-linear dynamical systems. Thus, Chaotic Logistic Maps are well-suited techniques for simulating non-linear characteristics in time series which can also be found in the real world.

3.1.3. Self-Exciting Threshold Auto-Regressive models

The SETAR model belongs to the family of TAR models, first introduced by Tong [47]. In contrast to linear AR models, TAR models capture non-linear dynamics in time series by means of a regime-switching technique that changes the underlying AR coefficients, when a particular threshold value is met. When the threshold variable is a lagged value of the series itself, the model is known as a SETAR model. The SETAR models were first introduced by Tong and Lim [48]. Due to their regime-switching nature, SETAR models can capture, for example, real-world scenarios where patterns of a certain time series change due to policy interventions upon reaching a certain threshold value in the time series.

3.1.4. Mackey-Glass Equation

The equation of the Mackey-Glass DGP was first introduced in a research studying first-order non-linear Differential-Delay Equations (DDE) which describe physiological control systems [33]. The Mackey-Glass Equation was specifically used to explain the fluctuations of white blood cells in the human body under certain cases of chronic leukaemia. The solutions of these DDEs can be chaotic and thus cause complexity in the underlying time series.

3.1.5. Fourier Terms

Fourier Terms are used to model periodic effects in a time series [19]. They are especially helpful with very long or non-integer periods as well as multiple seasonalities. When used as a DGP, they can be used to generate time series having seasonal effects with long memory as well.

Table 3 provides an overall summary of the aforementioned techniques used for time series simulation. Here, the *Function* and *Package* columns provide the references to the respective software implementations of the simulation techniques used in our experiments. The Chaotic Logistic Map DGP and Fourier Terms DGP are implemented by ourselves and do not involve any already existing software packages. The table furthermore indicates a characterisation of the generated time series in terms of their linearity.

3.2. Real-world datasets

In addition to the simulated datasets, we conduct few experiments using real-world datasets as well. The objective is to derive conclusions based on these real-world datasets too, to better support practice. For this we use the datasets from the NN5 forecasting competition [12] and the Wikipedia Web Traffic (WWT) dataset [18]. To induce heterogeneity into the real-world datasets, we also conduct experiments on a mixture of series from each of the individual datasets. To ensure that our originally simulated datasets are close to these real-world datasets, we include plots of *tsfeatures* of the real-world datasets along with the simulated datasets. This is visualised in Fig. 5. Based on these plots, it is visible that the chosen real-world datasets are roughly homogeneous in terms of *tsfeatures*. The mixture of series from each of the datasets is indicated as Mixture in Fig. 5. Therefore, the data points of the Mixture dataset belong to both the WWT dataset as well as the NN5 dataset. Although we are not covering the whole space of *tsfeatures* from our simulated datasets, these plots indicate that our simulation scenarios are representative of the reality; the real-world series are close to our simulated scenarios in terms of *tsfeatures*. The WWT series are close to the Mackey-Glass series and NN5 series are close to the SAR(1) series.

4. Forecasting framework

Depending on the specific experimental scenario, we train models either as local models or GFMs. In this section, the different forecasting techniques employed are mentioned along with their associated data preprocessing methodologies, model training and testing.

4.1. Forecasting techniques

For the forecasting techniques of the study, we use RNNs, FFNNs, PR models, LGBMs, AR, SETAR, SAR, ARIMA, Dynamic Harmonic Regression with ARIMA Errors (DHR-ARIMA) models as appropriate. The summary of all the forecasting techniques used for the study along with their respective attributes as well as the corresponding software details are available in Table 4. Further details of these techniques can be found under Appendix C of our Online Appendix.

Table 3
Summary of techniques used for time series simulation.

Simulation Technique	Function	Package	Linearity
AR(3)	`arima.sim`	`stats` [40]	✓
SAR(1)	`simulate sim.ssarima`	`forecast` [21] `smooth` [45]	✓
Chaotic Logistic Map	-	-	✗
SETAR	`setar.sim`	`tsDyn` [13]	✗
Mackey-Glass Equation	`data.mackey_glass`	`nolitsa` [37]	✗
Fourier Terms	-	-	✓

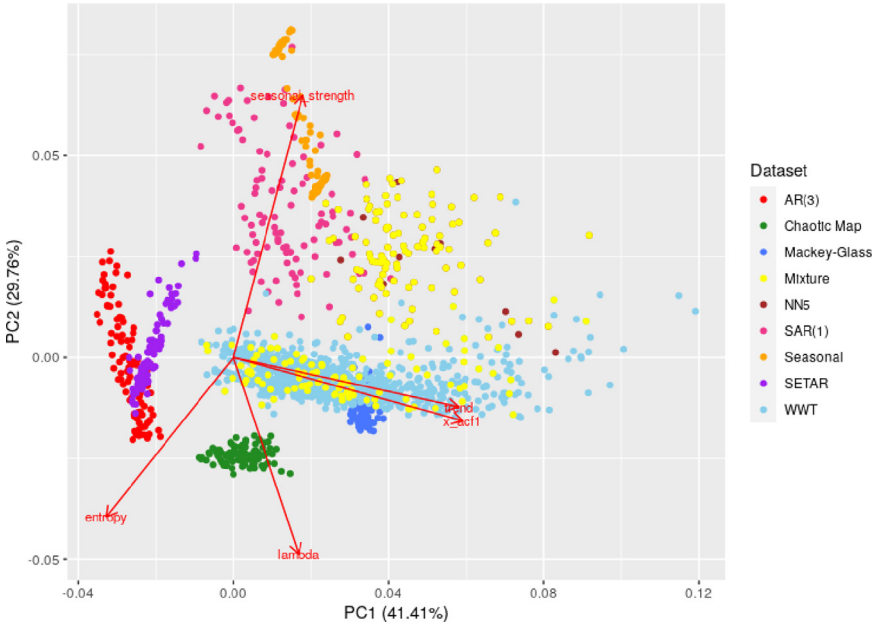

Fig. 5. Tsfeature plots for the real-world datasets along with the simulated homogeneous datasets.

Table 4
Summary of forecasting techniques.

Forecasting Technique		Software Details	Attributes
RNN		`Tensorflow` Framework [1]	Non-linear Modelling Per Series State
FFNN		`Tensorflow` Framework [1]	Non-linear Modelling
LGBM		`lightgbm` Python Package [29]	Non-linear Modelling
	SETAR	`tsDyn` R Package [13] (`setar` function)	Non-linear Modelling
	PR	`stats` R Package [40] (`glm` function)	Linear Modelling
	AR	`forecast` R Package [21] (`Arima` function)	Linear Modelling
	ARIMA	`forecast` R Package [21] (`auto.arima` function)	Linear Modelling
	DHR-ARIMA	`forecast` R Package [21] (`auto.arima` function)	Linear Modelling

4.2. Data preprocessing for forecasting

We apply a number of data preprocessing steps to the data corresponding to the different forecasting techniques. 1) We first perform a normalization of each series to avoid the varying scales of the series, especially when developing a global model. 2) Then, we perform a logarithmic transformation of the series in order to avoid non-stationarities in time series data such as exponential trends that machine learning models cannot handle properly. 3) Next, to produce a forecast horizon, we perform a moving window transformation of the data specifically for those models that can output windows; other models use either a recursive strategy or train one model per each step in the horizon. 4) Finally, to further reduce risks from remaining linear trends, a per-window normalization is applied to the data which is first moving window transformed. A detailed description of all the preprocessing steps can be found in Appendix D of the Online Appendix.

4.3. Model training & testing

For the NNs, once the optimal hyperparameters are found, they are applied for training the NNs once again and testing on the test data. During testing, the NNs are trained using 10 Tensorflow graph seeds and then the final NN forecasts are ensembled by taking the median, as discussed by Hewamalage et al. [20]. This approach effectively addresses the parameter uncertainty associated with the NNs by initialising the networks 10 times independently. For all the other models, once the model training is completed, the

Table 5
Characteristics of generated datasets from all DGPs.

Scenario	No. of DGPs	Min. Length	Max. Length	Min. No. of Series	Max. No. of Series	Forecast Horizon
AR(3) DGP						
SS	1	18	1800	1	1	3
MS-Hom-Short	1	18	18	1	100	3
MS-Hom-Long	1	18	180	100	100	3
MS-Het	100	18	180	100	100	3
SAR(1) DGP						
SS	1	24	2400	1	1	3
MS-Hom-Short	1	24	24	1	100	3
MS-Hom-Long	1	24	240	100	100	3
MS-Het	100	24	240	100	100	3
Chaotic Logistic Map DGP						
SS	1	60	6000	1	1	12
MS-Hom-Short	1	60	60	1	100	12
MS-Hom-Long	1	60	600	100	100	12
MS-Het	100	60	600	100	100	12
SETAR DGP						
SS	1	6000	6000	1	1	12
MS-Hom-Short	1	60	60	100	100	12
MS-Hom-Long	1	240	240	100	100	12
MS-Het	100	240	240	100	100	12
Mackey-Glass Equation DGP						
SS	1	6000	6000	1	1	12
MS-Hom-Short	1	60	60	100	100	12
MS-Hom-Long	1	240	240	100	100	12
MS-Het	100	240	240	100	100	12
Fourier Terms DGP						
MS-Hom-Long	1	17,600	17,600	100	100	48
MS-Het	100	17,600	17,600	100	100	48

trained models are then used to perform the forecasting on the intended forecast horizon.

4.4. Data post-processing

Once the forecasts are obtained from each individual model, a sequence of postprocessing is done on the forecasts, to reverse the preprocessing steps that were carried out on the data beforehand. This is done in the following order: 1) Add the last input point trend value back into the corresponding output windows. 2) Transform the data by taking the exponential. 3) Deduct 1, in case the original data contains zero values. 4) Re-scale the forecasts by multiplying by the series means. However, the exact postprocessing steps performed on the forecasts of each model, depend on which preprocessing was done in the beginning.

5. Experimental setup

This section details the experimental framework used in this study. To achieve significant results, for every scenario we generate 1000 datasets having the described characteristics and average the results over 1000 runs. In the rest of this section, we present the details of the datasets generation, error metrics used for the evaluation and the statistical tests conducted for the significance of the differences.

5.1. Characteristics of simulated datasets

The characteristics of different datasets generated from all DGPs according to the experimental scenarios explained in Section 2, are shown in Table 5. Here, the *Min. Length* and *Max. Length* columns refer to the range of the time series lengths we select when training our models. Similarly, *Min. No. of Series* and *Max. No. of Series* is the range of number of time series we use in our experiments.

Due to computational constraints arising from the variety of experiments designed, out of the six DGPs used in the study, we perform the data availability experiments only using three DGPs, namely AR(3), SAR(1) and Chaotic Logistic Map. For SETAR, Mackey-Glass Equation and Fourier Terms DGPs, we experiment using only one selected length and number of series as shown in Table 5. Here, the idea is that once the important conclusions are drawn about data availability in initial experiments, those conclusions can be used to fix a sufficient length and the number of series for the remaining experiments. Furthermore, for the Fourier Terms scenarios, only the MS-Hom-Long and MS-Het experiments are conducted due to computational limitations. To have all three seasonalities, one series in the MS-Hom-Short scenario of the Fourier Terms DGP needs to be at least 17,520 long to constitute two periods, which results in an ultra long time series for the SS scenario (17,520 * 100 length), making it computationally difficult to train models such as DHR-ARIMA and RNNs on it. The objective of the Fourier Terms DGP is to examine how the models behave with series all having either homogeneous or heterogeneous seasonal shapes. We also drop the number of datasets associated with each scenario to 100, to avoid heavy computational complexities of this study. This applies to the experiments on the three DGPs, Mackey-Glass Equation, Fourier Terms and SETAR and the Cluster setup of the Chaotic Logistic Map DGP.

Also, due to computational constraints, we perform the Cluster setup experiments only using the Chaotic Logistic Map DGP. In Table 6, are the characteristics of the datasets generated for this Cluster setup of the Chaotic Logistic Map DGP. However, for this setup, we use datasets of 10,000 time series each. This is done so, since 1) the recent forecasting competitions such as M4 and M5 were also involved with such very large datasets [34,35], 2) NNs are typically known to require sufficiently large datasets to learn a good model and 3) to induce even further diversity in a heterogeneous dataset compared to the homogeneous simulation scenarios.

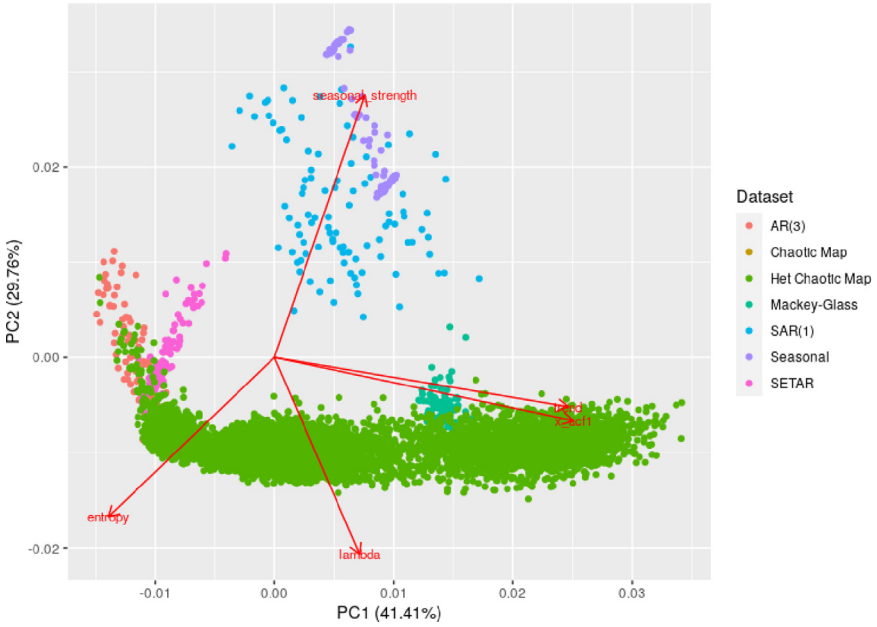

Fig. 6. Tsfeature plots for the Chaotic Logistic Map DGP dataset in the cluster setup along with the other simulated homogeneous datasets. Het Chaotic Map indicates the Chaotic Logistic Map DGP series with 4 DGPs mixed together. The diversity of this dataset is indicated by the span of the series across the *tsfeature* space.

Table 6
Characteristics of generated datasets for the cluster setup of the Chaotic Logistic Map DGP.

Characteristic	Value
No. of DGPs	4
Min. Length	240
Max. Length	240
Min. No. of Series	10,000
Max. No. of Series	10,000
Forecast Horizon	12

Table 7
Characteristics of generated datasets for relatedness analysis experiments.

Characteristic	1 DGP	2 DGP	3 DGP	4 DGP	5 DGP
AR(3) Series	1000	500	333	250	200
Chaotic Logistic Map Series	-	500	333	250	200
SETAR Series	-	-	334	250	200
Mackey-Glas Series	-	-	-	250	200
SAR(1) Series	-	-	-	-	200
Length	240	240	240	240	240
Forecast Horizon	12	12	12	12	12

Therefore, in this Cluster setup based experiment, we increase the dataset size significantly to draw conclusions based on such a large dataset. The scale of the heterogeneity of this dataset is depicted in Fig. 6.

For the experiment where we control the scale of heterogeneity by using different numbers of DGPs mixed together, the characteristics of the datasets are as shown in Table 7. We use 100 such simulated datasets from each number of DGPs mixed together, to make the results more robust. For simplicity and computational efficiency, to analyse the performance of GFMs with varying scales of heterogeneity, we conduct experiments on these datasets only using PR and LGBM models as global models. Since relatedness is closely tied to the complexity of a model, having both those models allows us to explore this phenomenon better.

Table 8
Characteristics of the real-world datasets.

Characteristic	NN5	WWT	Mixture
Min. Length	735	550	550
Max. Length	735	550	735
No. of Series	111	997	200
Forecast Horizon	56	56	56

5.2. Characteristics of real-world datasets

For the experiments involving real-world datasets, the characteristics of the datasets are as shown in Table 8. Both NN5 and WWT are daily datasets having equal length series. For the Mixture scenario, we combine 100 series from each of those datasets to create a potentially heterogeneous real-world dataset. The chosen forecast horizon for all the datasets is 56 days.

5.3. Performance measures

The relative performance of the models is evaluated with respect to two performance measures commonly found in the literature related to forecasting, namely SMAPE and Mean Absolute Scaled Error (MASE). Further details of the error measures can be found under Appendix E of the Online Appendix. However, According to, e.g., Hyndman and Koehler [24], the SMAPE metric is highly skewed with values close to zero. Therefore, as per the guidelines by Hewamalage et al. [20], on the Chaotic Logistic Map DGP datasets which have zero values, we use a variant of SMAPE to address this problem. Since our forecasting scenarios involve multiple series from many datasets, we consider the mean SMAPE and mean MASE measures to summarise the overall error distribution across all the series available.

5.4. Statistical tests for significance of the differences

The objective behind performing every experiment on 100/1000 datasets is that the individual differences between the models are significant for every scenario that way. However, to show systematically that this is the case, we perform non-parametric Friedman

rank-sum tests on a few of the most important scenarios as explained in Section 6, to estimate the significance of differences. This is followed by Hochberg's post hoc procedure to further analyse the differences relative to a control method, in particular the best method in every scenario [17]. A significance level of $\alpha = 0.05$ is used for all the tests.

6. Results and analysis

This section provides a detailed analysis of the results obtained for different experimental setups. Furthermore, we provide a comparison of the computational complexities of various forecasting techniques employed in this study.

6.1. Comparison of model performance

The results of all the DGPs as well as the percentage difference of every model from the best model in each scenario computed according to Eq. (1) are presented in Table 9 under the *Error* and *Diff* columns. In Eq. (1), D_m is the percentage difference of the performance of model m, E_m is the error of model m and E_b is the error of the best model in the respective scenario.

$$D_m = 100 * \frac{E_m - E_b}{E_b} \tag{1}$$

In every column of Table 9, the best-performing model under every DGP is indicated in boldface. Not all the models are relevant to all the experimental setups; hence the '-' in some of the table cells. Since PR models are the global versions of AR models, we run only the AR models in the SS scenarios. For the relevant models, we also indicate the number of lags used to train the model. For example, LGBM(3) refers to an LGBM model with an order of 3 lags.

Specifically for the AR(3) DGP, since we have sufficient lengths in the MS-Hom-Long scenario, we run the PR, RNN, LGBM and FFNN models with both 3 and 10 lags. For the DHR-ARIMA model in the Fourier Terms DGP scenario, the number of Fourier terms K is set to 5, which is equal to the original DGP. Also in the real-world data experiments, for the AR, MA and differencing orders of the ARIMA models, by default `auto.arima` supports maximum orders of 5, 5 and 2 for a non-seasonal ARIMA model and 2, 2 and 1 for a seasonal ARIMA model respectively. The numbers shown in Table 9 correspond only to the highest length/number of series in each experimental scenario as per Table 5. For more results analysis details, data availability experiment results as well as further details on the Hochberg's post-hoc procedures of different scenarios, refer to Appendix G of the Online Appendix.

6.1.1. AR(3) data generating process

Fig. 7 illustrates how the performance of the forecasting models under the AR(3) DGP setting evolves with increasing lengths and number of series, except for the local models in the MS-Hom-Short scenario where we only increase the number of series in the dataset.

In the AR(3) DGP case, we have the model closest to the true DGP, AR(3) as a forecasting technique. In Table 9, under the SS scenario we see that with sufficient lengths of the series, the AR(3) is the best, as expected. However, when looking at the SS scenario in Fig. 7, we can see that at the beginning lengths, the AR(2) model is better than the AR(3). For the SS scenario of the AR(3) DGP, the Friedman test gives an overall p-value of 1.21×10^{-10} in terms of the mean SMAPE values, indicating the significance of the differences of the models. In the MS-Hom-Short case, the local models become worse due to reduced lengths of series. This affects more the models with higher complexity such as the AR(10). The PR(3) model close to the true DGP performs best also since it learns across series. In the MS-Hom-Short case too, AR(2) is better than AR(3) across all the dataset sizes, complying with the findings in the SS setting. At smaller dataset sizes, the global models are also poor in performance, but gradually improve as the number of series is increased.

In the MS-Hom-Long setting, local models become competitive again as series lengths are increased. Thus, AR(2) and AR(3) models are much closer in performance to the PR models. Yet, this also depends on the complexity of the local model as reflected by the performance of the AR(10) model. The PR(3) model still outperforms local models in terms of Mean SMAPE since the series are all homogeneous. According to the MS-Hom-Long setting in Fig. 7, we can see that AR(10) is the worst performing model in the beginning lengths similar to the observations from the MS-Hom-Short scenario. In the MS-Het scenario, the local models again emerge as the best models, due to the heterogeneity among the series. Yet, complex global models such as RNN are also competitive in the heterogeneous setting. At shorter lengths, both RNN models and even the linear global model PR(10) are better than the local AR(10). This demonstrates that although local models can be competitive in a heterogeneous setting, for that they require sufficient lengths in the series. Friedman test for the MS-Het scenario gives an overall p-value of 2.59×10^{-10} in terms of the mean SMAPE values which means that the differences are statistically highly significant.

6.1.2. SAR(1) data generating process

In the SAR(1) DGP too, we experiment with the model closest to the true DGP; i.e., the SAR(1) model which is a simple model with just one coefficient. As seen in Table 9, the SAR(1) model is the best on the SS scenario. The AR(3) model which is a misspecified model with respect to the SAR(1), is the worst. The AR(12) in the SS setting is equivalent in performance to SAR(1) in terms of Mean MASE, since AR(12) nests the true DGP SAR(1). In the MS-Hom-Short scenario, the local AR(12) becomes clearly worse due to the shorter lengths of series. But SAR(1) remains competitive due to its simplicity. However, complex global models such as RNN(12) and FFNN(12) are also quite competitive here.

In the MS-Hom-Long scenario, once again the SAR(1) model surpasses all the other models. However, with increased lengths, the AR(12) has also improved. In the MS-Het scenario, the SAR(1) model is still the best, due to its local modelling, complying with the observation in the AR(3) DGP. The second best model is the AR(12) model, although it shows poor performance in the previous homogeneous scenarios. Among the global models, the PR(12) is the best, which is close to the true DGP. The RNN(12) is also quite competitive and is exactly equal in performance to PR(12) with respect to Mean MASE. The observations related to data availability are the same as with the AR(3) DGP, i.e., the performance of all the models improves in general as the size of the dataset increases, although not quite consistent across the individual lengths and number of series in some scenarios.

6.1.3. Chaotic Logistic Map data generating process

In the Chaotic Logistic Map DGP experiments, we do not have the model closest to the true DGP as a forecasting method. As shown in Table 9, the non-linearity of the patterns generated by the Chaotic Logistic Map DGP are perceived by the non-linear forecasting models LGBM(15), RNN(15) and FFNN(15) outperforming the linear ones in all the experimental scenarios. In both MS-Hom-Short and MS-Hom-Long scenarios, the linear global PR(15) outperforms the AR(15) due to the homogeneity of the series, Even though the individual series patterns are complex. For the MS-Hom-Short and MS-Hom-Long scenarios, the Friedman test of statistical significance gives overall p-values of 3.12×10^{-10} and $<$

Table 9
Results and percentage differences of each model from the best model under each experimental scenario.

Model	SS				MS-Hom-Short				MS-Hom-Long				MS-Het			
	SMAPE		MASE		SMAPE		MASE		SMAPE		MASE		SMAPE		MASE	
	Error	Diff	Error	Diff	Error	Diff	Error	Diff	Error	Diff	Error	Diff	Error	Diff	Error	Diff
AR(3) DGP																
RNN(3)	18.04	3.26	0.55	3.77	18.12	2.32	0.61	1.67	21.37	1.42	0.54	1.89	21.78	9.28	0.86	13.16
RNN(10)	-	-	-	-	-	-	-	-	21.22	0.71	0.53	0.00	20.88	4.77	0.81	6.58
FFNN(3)	19.11	9.39	0.58	9.43	18.91	6.78	0.64	6.67	23.06	9.44	0.58	9.43	23.35	17.16	0.91	19.74
FFNN(10)	-	-	-	-	-	-	-	-	21.84	3.65	0.55	3.77	22.21	11.44	0.87	14.47
LGBM(3)	19.25	10.19	0.59	11.32	19.25	8.70	0.65	8.33	23.07	9.49	0.58	9.43	23.29	16.86	0.90	18.42
LGBM(10)	-	-	-	-	-	-	-	-	21.72	3.08	0.55	3.77	21.85	9.63	0.85	11.84
PR(3)	-	-	-	-	**17.71**	**0.00**	**0.60**	**0.00**	**21.07**	**0.00**	**0.53**	**0.00**	21.72	8.98	0.86	13.16
PR(10)	-	-	-	-	-	-	-	-	21.08	0.05	**0.53**	**0.00**	21.72	8.98	0.86	13.16
AR(2)	17.48	0.06	**0.53**	**0.00**	18.79	6.10	0.63	5.00	21.17	0.47	**0.53**	**0.00**	**19.93**	**0.00**	**0.76**	**0.00**
AR(3)	**17.47**	**0.00**	**0.53**	**0.00**	19.53	10.28	0.66	10.00	21.22	0.71	**0.53**	**0.00**	19.95	0.10	**0.76**	**0.00**
AR(10)	17.55	0.46	0.54	1.89	32.92	85.88	1.03	71.67	21.63	2.66	0.54	1.89	20.34	2.06	0.78	2.63
SAR(1) DGP																
RNN(12)	4.91	3.37	0.97	3.19	4.90	0.82	1.04	0.97	14.03	6.45	1.04	7.22	22.15	4.48	0.74	5.71
FFNN(12)	4.94	4.00	0.98	4.26	**4.86**	**0.00**	**1.03**	**0.00**	13.22	0.30	**0.97**	**0.00**	22.98	8.40	0.77	10.00
LGBM(12)	5.00	5.26	0.99	5.32	5.19	6.79	1.10	6.80	13.29	0.83	0.98	1.03	22.84	7.74	0.76	8.57
PR(12)	-	-	-	-	4.93	1.44	1.05	1.94	13.19	0.08	**0.97**	**0.00**	22.02	3.87	0.74	5.71
AR(12)	4.77	0.42	**0.94**	**0.00**	7.69	58.23	1.59	54.37	13.49	2.35	0.99	2.06	21.65	2.12	0.72	2.86
AR(3)	8.86	86.53	1.76	87.23	8.88	82.72	1.90	84.47	23.40	77.54	1.79	84.54	22.11	4.29	0.74	5.71
SAR(1)	**4.75**	**0.00**	**0.94**	**0.00**	4.90	0.82	1.04	0.97	**13.18**	**0.00**	**0.97**	**0.00**	**21.20**	**0.00**	**0.70**	**0.00**
Chaotic Logistic Map DGP																
RNN(15)	51.63	5.52	0.78	5.41	**49.21**	**0.00**	**0.75**	**0.00**	48.12	2.82	0.73	2.82	27.54	0.33	**0.88**	**0.00**
FFNN(15)	53.27	8.87	0.81	9.46	50.15	1.91	0.77	2.67	47.50	1.50	0.72	1.41	28.18	2.66	0.90	2.27
LGBM(15)	**48.93**	**0.00**	**0.74**	**0.00**	50.43	2.48	0.77	2.67	**46.80**	**0.00**	**0.71**	**0.00**	**27.45**	**0.00**	**0.88**	**0.00**
PR(15)	-	-	-	-	53.37	8.45	0.81	8.00	52.46	12.09	0.78	9.86	27.63	0.66	**0.88**	**0.00**
AR(15)	52.91	8.13	0.79	6.76	57.67	17.19	0.89	18.67	52.65	12.50	0.79	11.27	27.56	0.40	**0.88**	**0.00**
SETAR DGP																
RNN(15)	23.44	14.23	0.47	14.63	**20.57**	**0.00**	**0.41**	**0.00**	**23.68**	**0.00**	**0.41**	**0.00**	25.93	0.43	**0.45**	**0.00**
FFNN(15)	20.78	1.27	0.42	2.44	20.69	0.58	0.42	2.44	23.84	0.68	0.42	2.44	26.00	0.70	0.46	2.22
LGBM(15)	20.63	0.54	**0.41**	**0.00**	21.09	2.53	0.43	4.88	23.80	0.51	0.42	2.44	**25.82**	**0.00**	**0.45**	**0.00**
PR(15)	-	-	-	-	21.21	3.11	0.43	4.88	24.50	3.46	0.43	4.88	26.67	3.29	0.47	4.44
AR(15)	21.19	3.27	0.43	4.88	23.89	16.14	0.48	17.07	25.10	6.00	0.44	7.32	27.16	5.19	0.48	6.67
SETAR	**20.52**	**0.00**	**0.41**	**0.00**	22.26	8.22	0.44	7.32	25.61	8.15	0.44	7.32	29.10	12.70	0.51	13.33
Mackey-Glass Equation DGP																
RNN(15)	3.21	444.07	0.48	433.33	1.47	65.17	0.22	69.23	1.20	166.67	0.18	157.14	**6.76**	**0.00**	**1.11**	**0.00**
FFNN(15)	5.45	823.73	0.82	811.11	1.61	80.90	0.24	84.62	2.59	475.56	0.39	457.14	14.05	107.84	2.35	111.71
LGBM(15)	**0.59**	**0.00**	**0.09**	**0.00**	**0.89**	**0.00**	**0.13**	**0.00**	**0.45**	**0.00**	**0.07**	**0.00**	9.59	41.86	1.59	43.24
PR(15)	-	-	-	-	7.13	701.12	1.06	715.38	7.77	1626.67	1.16	1557.14	11.95	76.78	1.97	77.48
AR(15)	6.81	1054.24	1.02	1033.33	9.22	935.96	1.36	946.15	7.99	1675.56	1.19	1600.00	10.87	60.80	1.82	63.96
Fourier Terms DGP																
RNN(60)	-	-	-	-	-	-	-	-	7.63	423.34	0.32	426.33	14.85	922.94	0.64	959.41
FFNN(60)	-	-	-	-	-	-	-	-	8.38	474.95	0.35	480.09	13.83	853.05	0.60	884.78
LGBM(60)	-	-	-	-	-	-	-	-	4.99	241.88	0.21	254.72	6.68	360.24	0.30	386.47
PR(60)	-	-	-	-	-	-	-	-	12.43	752.36	0.53	789.88	13.63	838.73	0.59	874.41
DHR-ARIMA	-	-	-	-	-	-	-	-	**1.46**	**0.00**	**0.06**	**0.00**	**1.45**	**0.00**	**0.06**	**0.00**

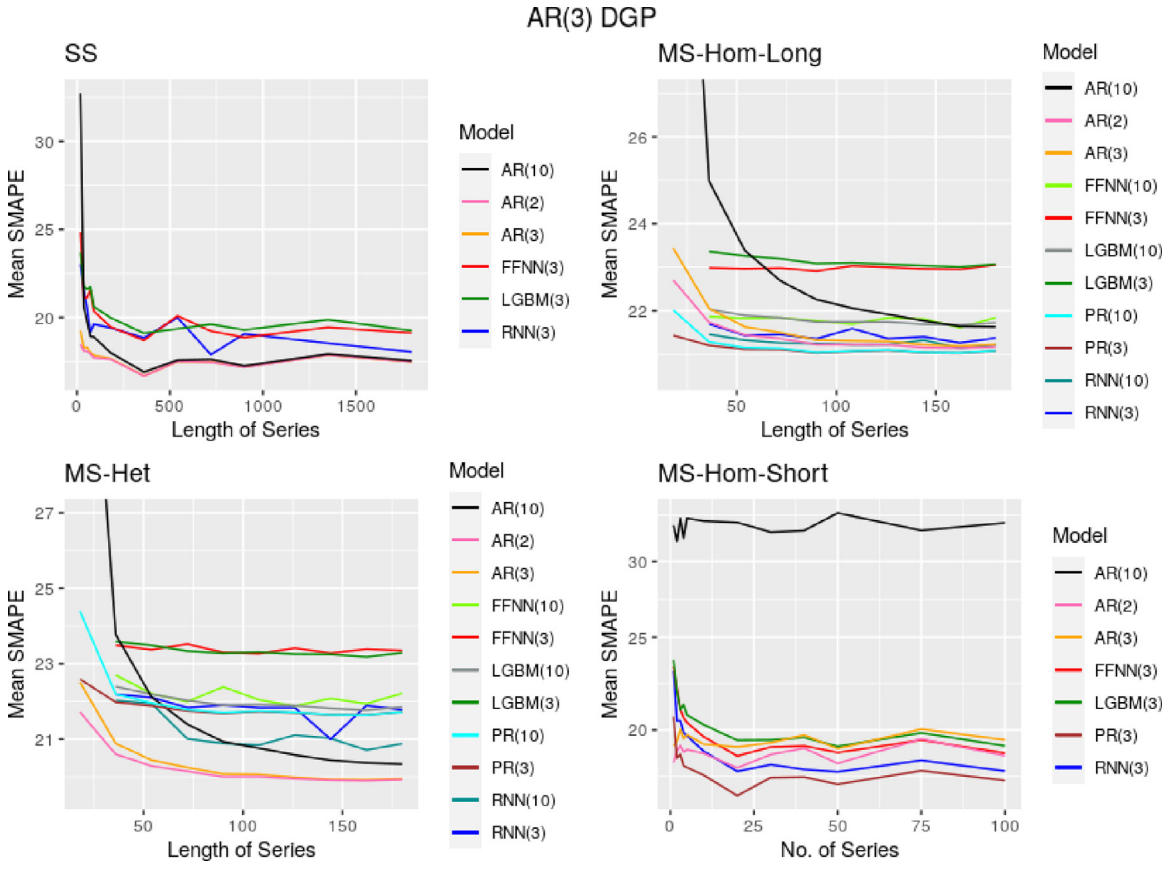

Fig. 7. Visualisation of the change of errors of the models across different amounts of data in the AR(3) DGP scenarios.

10^{-30} respectively, indicating the significance of the differences between the models. When it comes to the MS-Het scenario, similar to the previous DGPs, the local AR(15) outperforms the PR(15). We can observe that the models improve with increased amounts of data in general, in the Chaotic Logistic Map DGP too. Based on the conclusions obtained regarding data availability for forecasting from all the DGPs explained thus far, we eliminate the data availability experiments from the rest of the DGPs explained next.

6.1.4. SETAR data generating process

With the SETAR DGP, we experiment with the model closest to the true DGP; i.e., the SETAR model. Similar to the Chaotic Logistic Map DGP, the complexity of the patterns generated are evident here too, with the complex forecasting techniques being competitive. In the SS scenario the SETAR model is the best as expected, but not quite good with any of the other scenarios. This indicates that the lengths of the series are not sufficient for the SETAR model even though it is the closest to the true DGP. Although the global PR is better than the local models due to the homogeneity of the series in the MS-Hom-Short and MS-Hom-Long scenarios, it is not adequate to capture the complex patterns. In the heterogeneous scenario, different from the previous DGPs, the PR model still performs better than both the AR and SETAR models. With the AR model, this is due to the regime-switching nature of the SETAR generated series which makes one AR model trained for the whole series not competitive.

6.1.5. Mackey-Glass data generating process

The observations from this DGP too are very similar to SETAR and Chaotic Logistic Map DGPs. Due to the complexity of the patterns generated, non-linear models RNN(15), FFNN(15), and LGBM(15) outperform the linear models in most of the scenarios. Here too, we do not have the model closest to the true DGP as a forecasting technique. The Friedman test of statistical significance for the SS scenario, gives an overall p-value of 1.08×10^{-10}, suggesting that the results are statistically highly significant. The PR(15) model is better than the AR(15) model in both MS-Hom-Short and MS-Hom-Long scenarios, due to the homogeneity among the series. In the MS-Het scenario, the local AR(15) is better than the global PR(15) due to the heterogeneity of the datasets. For this case, the Friedman test gives an overall p-value of 1.74×10^{-10}, again indicating the high significance of the performance differences.

6.1.6. Fourier Terms data generating process

In the Fourier Terms DGP scenarios, we experiment with DHR-ARIMA as the model closest to the true DGP. Therefore, in both the MS-Hom-Long and MS-Het scenarios, the DHR-ARIMA performs the best as expected. A key observation here is that, in the MS-Het scenario, different from the previous DGPs, the performance of the complex NN models have deteriorated quite sharply, even worse than the linear PR models. This implies that the globally built NN models are very poor in modelling heterogeneous seasonal patterns across series. This follows the findings by Bandara et al. [3] and Hewamalage et al. [20] that NN models struggle to model seasonality on their own when the series contain different seasonal patterns, even though the frequencies are the same.

6.1.7. Results of the cluster setup experiments

The horizontal line in Table 10 separates the GFM.Cluster setup results with their corresponding GFM.All results along with the local AR model results. The “Cluster” suffix indicates that the GFM.Cluster training paradigm explained in Section 2.5 is used on

Table 10
Mean SMAPE, mean MASE values and complexities of the models for the GFM.Cluster setup. The model complexities are computed at a global level by considering the whole dataset of many series.

Model	Errors		Complexity (MB)
	SMAPE	MASE	
RNN(15)	37.12	0.86	0.226
FFNN(15)	38.49	0.89	0.025
LGBM(15)	36.27	0.84	3.100
PR(15)	38.78	0.87	0.003
AR(15)	39.20	0.88	23.000
RNN(15)-Cluster	36.24	**0.83**	1.003
FFNN(15)-Cluster	36.61	0.85	0.073
LGBM(15)-Cluster	**35.78**	**0.83**	4.000
PR(15)-Cluster	38.65	0.86	0.020

the respective models. As seen in Table 10, the GFM.Cluster setup improves the accuracy of the GFM.All setup in all techniques. All the non-linear Cluster and base models hold a better standing over the linear models in this scenario, while the LGBM(15)-Cluster performs the best. The Friedman test of statistical significance for the Cluster scenario gives an overall p-value of 2.89×10^{-10}, expressing that the differences are statistically highly significant.

The comparison of the complexities of the different models measured as explained in Section 2.5 are also available in Table 10. The model complexities are measured at the global scale, considering all the series in the dataset. All the complexities are reported in terms of the size of the compressed models in Mega Bytes (MB). Table 11 gives details of the objects included within each of the different types of the models saved on disk. As seen in Table 10, the Cluster versions of all the models possess more complexity than their base model versions. The collection of AR(15) models across all the series has the highest complexity of 23MB disk space. Thus, a collection of local AR models built for many series can be remarkably complex, although a single AR model built for a particular series may possess a very small complexity (around 0.002 MB in this case), as their combined complexity is proportional to the size of the dataset. The LGBM models also contain relatively higher complexity since according to our setup one model is built per each step in the horizon. This is followed by the complexities of the NN models; the RNN with more complexity than the FFNN. The model with the lowest complexity is the PR model; however this complexity has increased notably in the cluster based setup.

6.1.8. Results on the real-world experimental scenarios

The results from the experiments involving real-world datasets are as reported in Table 12. On all three datasets the LGBM has become the best model. In general, the non-linear global models LGBM, FFNN and RNN are more competitive across all the datasets than the local model ARIMA and the global linear PR model. Moreover, on both NN5 and WWT datasets (which are considered to be homogeneous in this case), the global PR model has performed better than the local ARIMA model, reaffirming the benefit of cross-learning in homogeneous scenarios. In the Mixture scenario where series are considered heterogeneous, the local ARIMA model is more competitive over the global linear PR model, similar to the aforementioned simulation scenarios. However, due to the complex non-linear modelling of the other global models, they have managed to outperform the ARIMA model even in the heterogeneous scenario.

Since the error values reported in Table 12 cannot be compared across datasets, we also include Table 13 comparing the errors only across the common 100 series from each of the individual datasets and the Mixture dataset. From Table 13, it can be seen that for the common NN5 series (left sub-table), the PR model has faced a huge decrease in accuracy when it learns across heterogeneous series as opposed to the homogeneous series. In contrast, the other non-linear global models are relatively more robust when moving to the Mixture scenario. For the common WWT series (right sub-table), the PR model's accuracy is approximately the same in both the homogeneous and heterogeneous datasets (the errors have even dropped slightly in the heterogeneous case). Yet, the PR model is a poorly performing model in both the heterogeneous and homogeneous scenarios compared to the other models. For all the other global non-linear models, the errors have marginally increased when moving from the homogenous case to the heterogeneous case.

6.2. Analysis on the impact of different levels of relatedness on the global model performance

The results from the experiments combining multiple DGPs together are as reported in Table 14. In the first sub-table in Table 14, we show the Mean SMAPE and Mean MASE values by considering all 1000 series in each dataset. Since the series in each of the datasets having 1 DGP, 2 DGPs, 3 DGPs etc. are different from each other, in this sub-table it is only possible to compare models within a single dataset (i.e within a single column). We also include the second sub-table where we report the errors only across the series common to each of the datasets, which is 200 series from the AR(3) DGP, making the values in this sub-table comparable both vertically (within the column) and horizontally (within the row). We change model complexity here using two techniques; 1) increasing the number of coefficients of the same model 2) switching to a different complex model family (LGBM having tree-based nature). To make it easy to see the small differences between errors, all the errors are reported up to the 6th decimal place. The best model in each column is indicated in boldface and the best out of the PR models is indicated in italics.

According to the first sub-table of Table 14, it is clear that the higher the heterogeneity of the data becomes, the higher the complexity of the models need to be. With 1 DGP, the PR(3) which is the closest to the true DGP is the best model as expected. For the LGBM(200) model, although it is worse than the PR(3), shows slightly better performance than the PR(200), implying that switching to a different model family here is better than training 200 separate coefficients. As heterogeneity is introduced, the LGBM model constantly remains the best. For the PR models both less complex and highly complex models perform more or less poorly and there seems to exist an optimal complexity between these two extremes where the model performs best. This is because high complexity (such as 200 coefficients), is unnecessarily high for the data patterns where as with a low number of lags (such as 3), the complexity of the model is not enough to model the heterogeneous patterns. Thus, as the heterogeneity increases, this optimal complexity of the PR model also seems to increase.

As seen from the second sub-table of Table 14, for the common AR(3) series in each dataset, the PR(3) model which is the closest to the actual DGP has remained the best across all datasets. The LGBM model which is the best model for all the heterogeneous datasets as seen from the first sub-table, has become the worst or the second worst model of each column. Among the PR models too, the performance has always dropped the more complex the model becomes. In the fully homogeneous case this is due to the over-complexity of the models. But, in the heterogeneous settings, this is due to the increased capacity of these models to generalise well over all the different series in the dataset at the cost of performing poorly on the AR(3) series alone. Horizontal comparisons on the second sub-table reaffirm this conclusion, given that apart from the PR(3) model, all the other models consistently become worse on the common AR(3) series as the heterogeneity is increased.

Table 11
Objects included in the different types of saved models for model complexity calculations.

Model	Included Objects
RNN(15)	Overall Model (Optimizer, Metrics, Losses, Input & output shapes, Layers), For each layer (Layer name, Data type, Input & output shapes, Kernel, Whether trainable, Whether stateful, Whether dynamic, Bias, Weights, Trainable & non-trainable weights)
FFNN(15)	Same as for RNN
LGBM(15)	Overall model (Feature names, Tree sizes, Objective function, Features importances, Parameters such as boosting, no. iterations, learning rate, bagging frequency etc.), For each tree (Split features, Split gains, Thresholds, No. leaves, Left & right children, decision types for each split, Leaf weights, values and counts, Internal weights, values and counts)
PR(15)	Optimal coefficients, AIC, Function call, Formula, Error distribution, No. of IWLS iterations, Rank of the fitted linear model, QR Decomposition
AR(15)	Optimal coefficients, AIC, AIC_c, BIC, Log likelihood, Kalman filter for fitting, Function call, Variance matrix of the coefficients

Table 12
Results on the real-world datasets.

Model	NN5		WWT		Mixture	
	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE
ARIMA	24.95	0.97	50.36	1.48	41.56	1.26
PR(70)	22.52	0.89	49.30	1.23	45.20	1.44
LGBM(70)	**21.11**	**0.83**	**44.87**	**1.17**	**36.97**	**1.14**
FFNN(70)	21.99	0.87	45.03	**1.17**	38.03	1.16
RNN(70)	22.05	0.87	44.94	**1.17**	37.79	1.16

6.3. Comparison of computational complexities

Under the computational resources provided under Appendix F, we provide a comparison of the computational times of the different forecasting models. For this, we select the MS-Hom-Long scenario of the SETAR DGP. In Table 15, we record the time taken for one of the 100 datasets. Data preprocessing is not relevant for the AR(15) and SETAR models. Since RNN, FFNN, and LGBM all use the data preprocessed in the same manner, their data preprocessing times are the same. As observed from Table 15, the NN-based techniques spend comparatively higher computational times, with the RNN taking the highest and the FFNN the second highest time. The PR models in general are the most efficient, while the SETAR and the LGBM models are the second and third most efficient models respectively. Compared to LGBM and PR, which are global models, the local AR model in this case with its 15 coefficients also takes a considerable amount of time due to building one model per each series.

6.4. Overall summary of results

Overall, with respect to the data availability experiments, we can confirm the expected behaviour that the performance of all models, both local and global variants, improve as the individual series get longer. Also as expected, this is not the case with local models when we increase the number of series in the dataset, while keeping the lengths constant. Considering the experiments related to single and multiple series (SS and MS-Hom-Short scenarios), we see that for global models it does not make much difference whether the data remains on one long series or spread across multiple different series. Several exceptions to this conclusion are also mentioned under Appendix G.2.

With known DGPs and local models closest to or nesting the true DGP, it is the lengths of the individual series vs the complexity of the model that matter for their performance. This is why on the SAR(1) DGP the SAR(1) model having just one coefficient is always a competitive model in our experiments with both short and long series. On the other hand with the SETAR DGP, the lengths of the series are not sufficient for the SETAR model even in the MS-Hom-Long setting. However, with a clearly misspecified model, such as the AR(3) in the SAR(1) DGP case, the model is always bad despite the lengths of the series. When the series are made more complex, more complex forecasting techniques can be quite competitive over simpler linear ones.

With global models as well, the ones closest to the true underlying DGP can win. The more complex the patterns in the individual series, the more complex global models can win, as seen with the MS-Hom-Short and MS-Hom-Long scenarios of the Chaotic Logistic Map, SETAR and Mackey-Glass Equation DGPs. If the series in the dataset are all homogeneous, the global models which learn across series are competitive over local ones, especially with short series lengths. This holds even for linear global models with complex patterns of the data as seen under the MS-Hom-Short and MS-Hom-Long scenarios of the Chaotic Logistic Map, SETAR and Mackey-Glass-Equation DGPs. With heterogeneous data, the local models are competitive over the simple, linear global models, but not so at shorter series lengths. The complex, non-linear global models are very competitive in heterogeneous settings at both short and long series lengths over both simple, linear global models and local models. This is seen in the AR(3), Fourier Terms, Chaotic Logistic Map, SETAR and Mackey-Glass Equation DGPs. However, specifically on the Fourier Terms based series, while with homogeneous seasonal patterns the NN models are competitive, they become quite worse in performance when the series in the dataset have heterogeneous seasonal patterns. The experiments on the real-world datasets further support these general findings. Complex non-linear global models can be relatively more robust under both homogeneous and heterogeneous scenarios where the

Table 13
Results comparing the models on the common series of the real-world datasets.

Model	Comparison Among NN5 Series				Comparison Among WWT Series			
	NN5		Mixture		WWT		Mixture	
	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE
ARIMA	25.13	0.98	25.13	0.98	58.00	1.54	58.00	1.54
PR(70)	22.63	0.9	32.11	1.37	58.87	1.52	58.29	1.51
LGBM(70)	20.87	0.83	21.17	0.84	52.75	1.44	52.76	1.44
FFNN(70)	21.78	0.86	22.04	0.87	53.21	1.43	54.02	1.44
RNN(70)	21.78	0.86	22.02	0.87	53.2	1.43	53.55	1.44

Table 14
Comparison of the performance of GFMs along with the scale of relatedness of the data.

Model	1 DGP		2 DGP		3 DGP		4 DGP		5 DGP	
	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE
All Series										
LGBM(200)	0.274441	0.562742	**0.375393**	**0.649302**	**0.365885**	**0.579527**	**0.296379**	**0.527399**	**0.286531**	**0.540331**
PR(200)	0.275784	0.565966	0.397804	0.685633	0.386555	0.611864	0.369200	0.801435	0.348077	0.766652
PR(150)	0.274308	0.562577	0.395286	0.678823	0.384318	0.606618	0.366534	*0.792804*	0.345199	*0.757799*
PR(50)	0.273774	0.561347	0.394713	0.676582	0.383768	0.604974	*0.366476*	0.793520	*0.345013*	0.757828
PR(15)	0.273717	0.561210	0.394656	0.676296	*0.383705*	*0.604764*	0.371547	0.820922	0.352633	0.789357
PR(3)	***0.273704***	***0.561179***	*0.394631*	*0.676212*	0.385608	0.606894	0.380206	0.869874	0.366181	0.841929
Common AR(3) Series										
LGBM(200)	0.275149	0.564708	0.278396	0.573422	0.279357	0.575325	0.282533	0.582618	0.288811	0.597804
PR(200)	0.276367	0.567692	0.278554	0.572297	0.279610	0.574877	0.282338	0.581038	0.285870	0.589121
PR(150)	0.274933	0.564377	0.277143	0.568967	0.278091	0.571409	0.280675	0.577215	0.284014	0.584927
PR(50)	0.274412	0.563164	0.276650	0.567832	0.277557	0.570213	0.280158	0.576054	0.283798	0.584449
PR(15)	0.274351	0.563029	0.276585	0.567688	0.277482	0.570053	0.279926	0.575535	0.283241	0.583187
PR(3)	**0.274342**	**0.563008**	**0.276582**	**0.567679**	**0.276396**	**0.567536**	**0.278724**	**0.572693**	**0.277597**	**0.570090**

Table 15
Computational times comparison of the forecasting models (in seconds).

Model	Data Preprocessing	Model Training & Testing	Total
RNN(15)	5.36	10788.62	10793.98
FFNN(15)	5.36	649.73	655.09
LGBM(15)	5.36	4.58	9.94
AR(15)	-	111.08	111.08
SETAR	-	3.03	3.03
PR(15)	0.05	0.40	0.45

true underlying DGP is not known. Using the GFM.Cluster setup to incorporate information of the pre-known clusters within the heterogeneous data can further increase the model complexity and improve model performance. For local models too, their model complexity at the global level for the whole dataset can be considerably high compared to a single global model.

In summary, the complexity of a competitive GFM on a particular dataset depends on the scale of heterogeneity of the data, the amount of data available as well as the complexity of the data patterns. More complexity of the model is needed the more complex or heterogeneous the data becomes. However, while linear models such as PR, AR work best only with known linear patterns in the data, non-linear models such as LGBM, RNN can perform reasonably well irrespective of the exact DGP, heterogeneity in the data, or lengths of the series. Nevertheless, with RNNs, this superior performance comes at the cost of more computational time. LGBMs are catching up with RNNs in terms of their performance, while being extremely fast in model training. To increase model complexity with limited data, switching to a different complex model family can be a better option than increasing the number of coefficients of a linear model.

7. Conclusions

The recent work by Montero-Manso and Hyndman [39] has shown that any local method applied on a dataset of many series can be approximated by a global model with sufficient complexity, irrespective of the relatedness of the underlying series. Therefore, for global models, it is about finding the right amount of complexity to outperform local methods. However, in practice there are complex trade-offs to be made between model complexity and capabilities on the one hand, and factors such as the availability of data, the complexity of DGPs, and the heterogeneity of the underlying data on the other hand. In this work, we have explored in an extensive experimental study some of the poorly understood factors that contribute to GFM performance under these trade-offs.

We have focused on characteristics common in real-world forecasting problems and their related challenges such as series with short history and heterogeneity of the series. Our take on relatedness of series as used in this work is a model based one, meaning that a dataset generated from the same DGP and same coefficients is a homogeneous/related dataset, whereas a dataset with series generated from different DGPs or same DGP with different coefficients is a heterogeneous/non-related one. Through extensive empirical evaluations carried out within such controlled setting of simulated datasets, we have demonstrated the interplay between the different aspects that affect GFM performance. Out of these factors, for model complexity, a quantitative approximation is also provided by using the concept of Kolmogorov complexity.

In terms of the methodology, we start by simulating data using the arguably simplest DGPs available and then making them more complex. We start with linear AR(3), SAR(1), Fourier Terms based DGPs and then move onto more complex, non-linear DGPs with the Chaotic Logistic Map, SETAR, and Mackey-Glass Equations. We simulate both homogeneous and heterogeneous scenarios. In the homogeneous setup, all time series in the dataset are simulated using the same DGP, whereas in the heterogeneous case, time series from different DGPs are mixed together within the same dataset. The availability of data for the experiments is controlled by changing the lengths and the number of series. For each scenario, 100 or 1000 datasets are simulated using different random seeds, to achieve reliable and significant experimental results. Similar to the complexity of the DGPs, the complexity of the forecasting techniques is regulated by experimenting using a number of forecasting techniques with different modelling capabilities. We have used linear AR, ARIMA, DHR-ARIMA, PR models as well as more complex SETAR, RNNs, FFNNs, and LGBMs as forecasting techniques. The complexity of global models is further varied by introducing two model setups, GFM.All and GFM.Cluster. To see how the conclusions derived from the simulated settings hold in the real-world, we also perform several experiments using seemingly homogeneous and heterogeneous real-worlds datasets close to our simulated scenarios in terms of *tsfeatures*.

Our study has confirmed that the model-based definition of relatedness is similar to defining relatedness with respect to *tsfeatures*, as done in many of the previous work. The results from the experiments first demonstrate that local linear models such as AR work best with known linear patterns in the data, with sufficient lengths in the series. The linear global models such as PR can cope well under multiple short series, but they too have the linearity as-

sumptions. When the patterns are made more complex, the complex non-linear global models such as RNNs, LGBMs are superior to simple linear global models. With heterogeneity existing across series, complex non-linear global models are competitive over linear global models irrespective of the simplicity or complexity of the patterns. Thus, non-linear, non-parametric models such as RNNs and LGBMs are in general quite competitive models across a variety of uncertain situations, where we have little prior knowledge about the data. However, there are also exceptions to this conclusion such as the case with the heterogeneous seasonal patterns where RNNs are clearly worse over all the models. The LGBM models hold another advantage of being computationally efficient compared to RNNs. These overall conclusions are attested by the results from the real-world datasets as well. The results and the model complexities reported for the experiments that involve the cluster setup prove that the complexity of GFMs can be further improved by developing GFMs per each cluster of series segregated based on pre-known heterogeneity.

With respect to data availability, unsurprisingly, all global models gradually improve as the lengths and the number of series in the dataset increase. For local models, this improvement is understandably only seen with the length of the individual series. The model complexity of local models grows proportional to the number of series in the dataset, potentially even higher than the constant complexity of a global model built on the same dataset. This is why, even though fitting a single local model on just one series may take very little time, fitting many of them on a whole dataset of series takes a considerable amount of time. As a whole, the trade-offs between the different factors that influence the GFM performance are identified from this study. With all related series, the complexity of the GFMs need not be very high, but if the patterns of the individual series are made complex, the models too need to be more complex proportionately. For linear models, this can be done by increasing the number of lags whereas non-linear, non-parametric models can adapt better due to built-in higher complexity than linear ones. When the datasets are made more heterogeneous as well, the model complexity has to increase proportional to the scale of heterogeneity of the data. However, this model complexity is constrained by the amount of data available for training the model. Therefore, in a limited data scenario, increasing the complexity of the model by shifting to other model families having non-linear, non-parametric characteristics (such as LGBM) can be more useful, in contrast to constantly increasing the number of coefficients in a linear global model. A potential future work of our study is to more extensively investigate how these factors of model complexity, complexity of data patterns and data availability interact with each other when the heterogeneity (different DGPs) exists along the same series (as is also the case with the SETAR series) as opposed to multiple series.

Declaration of Competing Interest

- There are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.
- The manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.
- We have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.
- We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). She is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from Hansika.Hewamalage@monash.edu

Acknowledgement

This research was supported by the Australian Research Council under grant DE190100045, a Facebook Statistics for Improving Insights and Decisions research award, Monash University Graduate Research funding and the MASSIVE - High performance computing facility, Australia.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2021.108441.

References

[1] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems, 2015,. Software available from tensorflow.org, https://www.tensorflow.org/.
[2] K. Bandara, C. Bergmeir, S. Campbell, D. Scott, D. Lubman, Towards accurate predictions and causal 'What-if' analyses for planning and policy-making: a case study in emergency medical services demand, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–10, doi:10.1109/IJCNN48605.2020.9206787.
[3] K. Bandara, C. Bergmeir, H. Hewamalage, LSTM-MSNet: leveraging forecasts on sets of related time series with multiple seasonal patterns, IEEE Trans. Neural Netw. Learn. Syst. 32 (4) (2021) 1586–1599, doi:10.1109/TNNLS.2020.2985720.
[4] K. Bandara, C. Bergmeir, S. Smyl, Forecasting across time series databases using recurrent neural networks on groups of similar series: a clustering approach, Expert Syst. Appl. 140 (2020) 112896.
[5] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, B. Seaman, Sales demand forecast in e-commerce using a long short-term memory neural network methodology, in: T. Gedeon, K.W. Wong, M. Lee (Eds.), Neural Information Processing, Springer International Publishing, Cham, 2019, pp. 462–474.
[6] C. Bergmeir, J.M. Benítez, On the use of cross-validation for time series predictor evaluation, Inf. Sci. 191 (2012) 192–213.
[7] C. Bergmeir, R.J. Hyndman, B. Koo, A note on the validity of cross-validation for evaluating autoregressive time series prediction, Comput. Stat. Data Anal. 120 (2018) 70–83.
[8] C.S. Bojer, J.P. Meldgaard, Kaggle forecasting competitions: an overlooked learning opportunity, Int. J. Forecast. 37 (2) (2021) 587–603.
[9] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time series analysis: forecasting and control, Forecasting and Control Series, Prentice Hall, 1994.
[10] R. Cilibrasi, P.M.B. Vitanyi, Clustering by compression, IEEE Trans. Inf. Theory 51 (4) (2005) 1523–1545, doi:10.1109/TIT.2005.844059.
[11] O. Claveria, S. Torra, Forecasting tourism demand to catalonia: neural networks vs. time series models, Econ. Model. 36 (2014) 220–228.
[12] S.F. Crone, NN5 competition, 2008,. http://www.neural-forecasting-competition.com/NN5/.
[13] A. Fabio Di Narzo, J. Luis Aznarte, M. Stigler, tsDyn: Nonlinear Time Series Models with Regime Switching, R package version 0.9-48.1, 2019. https://CRAN.R-project.org/package=tsDyn
[14] T. Fischer, C. Krauss, A. Treichel, Machine Learning for Time Series Forecasting - A Simulation Study, FAU Discussion Papers in Economics 02/2018, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics, 2018.
[15] J. Franklin, The elements of statistical learning: data mining, inference and prediction, Math. Intell. 27 (2) (2005) 83–85, doi:10.1007/bf02985802.
[16] B.D. Fulcher, N.S. Jones, hctsa: A computational framework for automated time-series phenotyping using massive feature extraction, Cell Syst. 5 (5) (2017). 527–531.e3
[17] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064. Special Issue on Intelligent Distributed Information Systems
[18] Google, Web Traffic Time Series Forecasting, 2017.https://www.kaggle.com/c/web-traffic-time-series-forecasting.
[19] A.C. Harvey, N. Shephard, 10 Structural time series models, in: Econometrics, in: Handbook of Statistics, vol. 11, Elsevier, 1993, pp. 261–302.

[20] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: current status and future directions, Int. J. Forecast. 37 (1) (2021) 388–427.
[21] R. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, Forecast: Forecasting Functions for Time Series and Linear Models, R package version 8.11, 2020. http://pkg.robjhyndman.com/forecast
[22] R. Hyndman, Y. Kang, T. Talagala, E. Wang, Y. Yang, tsfeatures: Time Series Feature Extraction, R package version 1.0.0, 2019. https://pkg.robjhyndman.com/tsfeatures/
[23] R.J. Hyndman, A.B. Koehler, J.K. Ord, R.D. Snyder, Forecasting with Exponential Smoothing: The State Space Approach, Springer, 2008.
[24] R.J. Hyndman, A.B. Koehler, Another look at measures of forecast accuracy, Int. J. Forecast. 22 (4) (2006) 679–688.
[25] T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, L. Callot, Criteria for classifying forecasting methods, Int. J. Forecast. 36 (1) (2020) 167–177.
[26] I.T. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.
[27] Y. Kang, R.J. Hyndman, K. Smith-Miles, Visualising forecasting algorithm performance using time series instance spaces, Int. J. Forecast. 33 (2) (2017) 345–358.
[28] Y. Kang, F. Li, R.J. Hyndman, GRATIS: GeneRAting TIme Series with diverse and controllable characteristics, Stat. Anal. Data Min. 13 (4) (2020) 354–376, doi:10.1002/sam.11461.
[29] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: a highly efficient gradient boosting decision tree, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 3146–3154.
[30] A.N. Kolmogorov, Three approaches to the quantitative definition of information, Int. J. Comput. Math. 2 (1–4) (1968) 157–168.
[31] D. Kwiatkowski, P.C.B. Phillips, P. Schmidt, Y. Shin, Testing the null hypothesis of stationarity against the alternative of a unit root: how sure are we that economic time series have a unit root? J. Econom. 54 (1) (1992) 159–178.
[32] K.W. Lau, Q.H. Wu, Local prediction of non-linear time series using support vector regression, Pattern Recognit. 41 (5) (2008) 1539–1547.
[33] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (4300) (1977) 287–289.
[34] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M4 competition: 100,000 time series and 61 forecasting methods, Int. J. Forecast. 36 (1) (2020) 54–74.
[35] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M5 accuracy competition: results, findings and conclusions, 2020b,
[36] P. Mandal, T. Senjyu, N. Urasaki, T. Funabashi, A neural network based several-hour-ahead electric load forecasting using similar days approach, Int. J. Electr. Power Energy Syst. 28 (6) (2006) 367–373.
[37] M. Mannattil, nolitsa, 2017, (https://github.com/manu-mannattil/nolitsa).
[38] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (5560) (1976) 459–467.
[39] P. Montero-Manso, R.J. Hyndman, Principles and algorithms for forecasting groups of time series: locality and globality, Int. J. Forecast. 37 (4) (2021) 1632–1653.
[40] R.C. Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2020. https://www.R-project.org/
[41] S.S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang, T. Januschowski, Deep State space models for time series forecasting, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, in: NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 7796–7805.
[42] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 36 (3) (2020) 1181–1191.
[43] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, Int. J. Forecast. 36 (1) (2020) 75–85.
[44] J. Sun, Y. Yang, Y. Liu, C. Chen, W. Rao, Y. Bai, Univariate time series classification using information geometry, Pattern Recognit. 95 (2019) 24–35.
[45] I. Svetunkov, smooth: Forecasting Using State Space Models, R package version 2.5.3, 2019. https://CRAN.R-project.org/package=smooth
[46] T. Teräsvirta, C.-F. Lin, C.W.J. Granger, Power of the neural network linearity test, J. Time Ser. Anal. 14 (2) (1993) 209–220.
[47] H. Tong, On a threshold model, in: Pattern Recognition and Signal Processing, Springer Netherlands, 1978, pp. 575–586.
[48] H. Tong, K.S. Lim, Threshold autoregression, limit cycles and cyclical data, J. R. Stat. Soc. Ser. B 42 (3) (1980) 245–268.
[49] N.D. Vanli, M.O. Sayin, M. Mohaghegh N, H. Ozkan, S.S. Kozat, Nonlinear regression via incremental decision trees, Pattern Recognit. 86 (2019) 1–13.
[50] C.S. Wallace, D.L. Dowe, Minimum message length and kolmogorov complexity, Comput. J. 42 (4) (1999) 270–283, doi:10.1093/comjnl/42.4.270.
[51] Y. Wang, A. Smola, D. Maddix, J. Gasthaus, D. Foster, T. Januschowski, Deep factors for forecasting, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 6607–6617.
[52] R. Wen, K. Torkkola, B. Narayanaswamy, D. Madeka, A multi-horizon quantile recurrent forecaster, Time Series Workshop, 2017.
[53] R. Ye, Q. Dai, Implementing transfer learning across different datasets for time series forecasting, Pattern Recognit. 109 (2021) 107617.
[54] G.P. Zhang, M. Qi, Neural network forecasting for seasonal and trend time series, Eur. J. Oper. Res. 160 (2) (2005) 501–514.
[55] J. Zhao, L. Itti, shapeDTW: shape dynamic time warping, Pattern Recognit. 74 (2018) 171–184.
[56] L. Zhu, N. Laptev, Deep and confident prediction for time series at uber, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), 2017.
[57] A.K. Zvonkin, L.A. Levin, The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms, Russ. Math. Surv. 25 (6) (1970) 83–124, doi:10.1070/rm1970v025n06abeh001269.

Christoph Bergmeir is a Senior Lecturer and a 2019 ARC DECRA Fellow in the Department of Data Science and Artificial Intelligence at Monash University. His fellowship is on the development of efficient and effective analytics for real-world time series forecasting." He works as a Data Scientist in a variety of projects with external partners in diverse sectors, e.g. in healthcare or infrastructure maintenance. Christoph holds a PhD in Computer Science from the University of Granada, Spain, and an M.Sc. degree in Computer Science from the University of Ulm, Germany. He has published on time series prediction using Machine Learning methods, recurrent neural networks and long short-term memory neural networks (LSTM), time series predictor evaluation, as well as on medical applications and software packages in the R programming language.

Hansika Hewamalage received the B.Sc. honours degree in Computer Science & Engineering from University of Moratuwa, Sri-Lanka, in 2016. She is currently pursing a Ph.D. degree in Computer Science at the Faculty of Information Technology, Monash University, Melbourne, Australia. Her research interests include Big Data, deep neural networks and time series forecasting. She has previously published in journals such as International Journal of Forecasting and IEEE Transactions on Neural Networks and Learning Systems.

Kasun Bandara received the B.Sc. honours degree in Computer Science from the University of Colombo School of Computing, Sri-Lanka, in 2015 and the Ph.D. degree in Computer Science from Monash University, Australia in 2020. He is currently working as a Postdoctoral Research Fellow at the Melbourne Centre for Data Science at The University of Melbourne, Australia. His research interests include Big Data, deep neural networks and time series forecasting.