Monash Time Series Forecasting Archive

Rakshitha Godahewa
Monash University
Melbourne, Australia
rakshitha.godahewa@monash.edu

Christoph Bergmeir
Monash University
Melbourne, Australia
christoph.bergmeir@monash.edu

Geoffrey I. Webb
Monash University
Melbourne, Australia
geoff.webb@monash.edu

Rob J. Hyndman
Monash University
Melbourne, Australia
rob.hyndman@monash.edu

Pablo Montero-Manso
University of Sydney
Australia
pmontm@gmail.com

Abstract

Many businesses nowadays rely on large quantities of time series data making time series forecasting an important research area. Global forecasting models and multivariate models that are trained across sets of time series have shown huge potential in providing accurate forecasts compared with the traditional univariate forecasting models that work on isolated series. However, there are currently no comprehensive time series forecasting archives that contain datasets of time series from similar sources available for researchers to evaluate the performance of new global or multivariate forecasting algorithms over varied datasets. In this paper, we present such a comprehensive forecasting archive containing 20 publicly available time series datasets from varied domains, with different characteristics in terms of frequency, series lengths, and inclusion of missing values. We also characterise the datasets, and identify similarities and differences among them, by conducting a feature analysis. Furthermore, we present the performance of a set of standard baseline forecasting methods over all datasets across ten error metrics, for the benefit of researchers using the archive to benchmark their forecasting algorithms.

1 Introduction

Accurate time series forecasting is important for many businesses and industries to make decisions, and consequently, time series forecasting is a popular research area. The field of forecasting has traditionally been advanced by influential forecasting competitions. The most popular forecasting competition series is the M-competition series [1–5]. Other well-known forecasting competitions include the NN3 and NN5 Neural Network competitions [6], and Kaggle competitions such as the Wikipedia web traffic competition [7].

The winning approaches of many of the most recent competitions such as the winning method of the M4 by Smyl [8] and the winning method of the M5 forecasting competition [5], consist of global forecasting models [9] which train a single model across all series that need to be forecast. Compared with local models, global forecasting models have the ability to learn cross-series information during model training and can control model complexity and overfitting on a global level [10].

This can be seen as a paradigm shift in forecasting. Over decades, single time series were seen as a dataset that should be studied and modelled in isolation. Nowadays, we are oftentimes interested in models built on sets of series from similar sources, such as series which are all product sales from a particular store, or series which are all smart meter readings in a particular city. Here, time series

Submitted to the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. Do not distribute.

are seen as an instance in a dataset of many time series, to be studied and modelled together. Global (univariate) models and (local) multivariate models are the methods of choice here, the difference being that global models train across series, but predict each series in isolation, with no need for the time series to have the same length or to be aligned in time, whereas multivariate models train and test over time series that are all the same length and all aligned in time, so that dependencies at certain time points and for the forecasts can be modelled. Thus, global models are applicable more broadly than multivariate models.

Both global and multivariate models get attention lately in machine learning (especially deep learning), with Li et al. [11], Rangapuram et al. [12], Wen et al. [13] presenting global models and Salinas et al. [14], Sen et al. [15], Yu et al. [16], Zhou et al. [17] discussing novel approaches for multivariate modelling. However, when it comes to benchmarking, these recent works use a mere two [13] to seven [11] datasets to evaluate the performance of the new algorithms and the chosen datasets are different in each work. The datasets mainly belong to the energy, transport, and sales domains, and they do not include datasets from other domains such as banking, healthcare, or nature.

In contrast, other areas of machine learning, such as general classification and regression, or time series classification, have greatly benefitted from established benchmark dataset archives, which allow a much broader and more standardised evaluation. The University of California Irvine (UCI) repository [18] is the most common and well-known benchmarking archive used in general machine learning, with currently 507 datasets from various domains. In time series classification, the dataset archives from the University of California Riverside (UCR) [19] and from the University of East Anglia (UEA) [20], contain 128 sets of univariate time series, and 30 datasets with multivariate time series, respectively, allowing routinely for much broader and more standardised evaluations of the methods, and therewith enabling more streamlined, robust, and reliable progress in the field.

The time series classification datasets, though they contain time series, do usually not resemble meaningful forecasting problems, so they cannot be used for the evaluation of forecasting methods. Also in the time series forecasting space there are a number of benchmarking archives, but they follow the paradigm of single series as datasets, and consequently contain mostly unrelated single time series such as the Time Series Data Library [21] and ForeDeCk [22].

There are currently no comprehensive time series forecasting benchmarking archives, to the best of our knowledge, that focus on sets of time series to evaluate the performance of global and multivariate forecasting algorithms. We introduce such an archive, available at `https://forecastingdata.org/`. The archive contains 20 publicly available time series datasets covering varied domains, with both equal and variable lengths time series. Many datasets have different versions based on the frequency and the inclusion of missing values, resulting in a total of 50 dataset variations.

We also introduce a new format to store time series data, based on the Weka ARFF file format [23], to overcome some of the shortcomings we observe in the .ts format used in the sktime time series repository [24]. We use a .tsf extension for this new format. This format stores the meta-information about a particular time series dataset such as dataset name, frequency, and inclusion of missing values as well as the series specific information such as starting timestamps, in a non-redundant way. The format is very flexible and capable of including any other attributes related to time series as preferred by the users.

Furthermore, we analyse the characteristics of different series to identify the similarities and differences among them. For that, we conduct a feature analysis using tsfeatures [25] and catch22 features [26] extracted from all series of all datasets. The extracted features are publicly available for further research use. The performance of a set of baseline forecasting models including both traditional univariate forecasting models and global forecasting models are also evaluated over all datasets across ten error metrics. The forecasts and evaluation results of the baseline methods are publicly available for the benefits of researchers that use the repository to benchmark their forecasting algorithms.

2 Data records

This section details the datasets in our time series forecasting archive. The current archive contains 20 time series datasets. Furthermore, the archive contains in addition 6 single very long time series. As a large amount of data oftentimes renders machine learning methods feasible compared with traditional statistical modelling, and we are not aware of good and systematic benchmark data in this space either, these series are included in our repository as well. A summary of all primary datasets included in the repository is shown in Table 1.

A total of 50 datasets have been derived from these 26 primary datasets. Nine datasets contain time series belonging to different frequencies and the archive contains a separate dataset per each frequency. Seven of the datasets have series with missing values. The archive contains 2 versions of each of these, one with and one without missing values. In the latter case, the missing values have been replaced by using an appropriate imputation technique.

Table 1: Datasets in the current time series forecasting archive

	Dataset	Domain	No: of Series	Min. Length	Max. Length	No: of Freq.	Missing	Competition
1	M1	Multiple	1001	15	150	3	No	Yes
2	M3	Multiple	3003	20	144	4	No	Yes
3	M4	Multiple	100000	19	9933	6	No	Yes
4	Tourism	Tourism	1311	11	333	3	No	Yes
5	NN5	Banking	111	791	791	2	Yes	Yes
6	CIF 2016	Banking	72	34	120	1	No	Yes
7	Web Traffic	Web	145063	803	803	2	Yes	Yes
8	Solar	Energy	137	52560	52560	2	No	No
9	Electricity	Energy	321	26304	26304	2	No	No
10	London Smart Meters	Energy	5560	288	39648	1	Yes	No
11	Wind Farms	Energy	339	6345	527040	1	Yes	No
12	Car Parts	Sales	2674	51	51	1	Yes	No
13	Dominick	Sales	115704	28	393	1	No	No
14	FRED-MD	Economic	107	728	728	1	No	No
15	San Francisco Traffic	Transport	862	17544	17544	2	No	No
16	Pedestrian Counts	Transport	66	576	96424	1	No	No
17	Hospital	Health	767	84	84	1	No	No
18	COVID Deaths	Nature	266	212	212	1	No	No
19	KDD Cup	Nature	270	9504	10920	1	Yes	Yes
20	Weather	Nature	3010	1332	65981	1	No	No
21	Sunspot	Nature	1	73931	73931	1	Yes	No
22	Saugeen River Flow	Nature	1	23741	23741	1	No	No
23	US Births	Nature	1	7305	7305	1	No	No
24	Electricity Demand	Energy	1	17520	17520	1	No	No
25	Solar Power	Energy	1	7397222	7397222	1	No	No
26	Wind Power	Energy	1	7397147	7397147	1	No	No

Out of the 26 datasets, 8 originate from competition platforms, 3 from research conducted by Lai et al. [27], 6 are taken from R packages, 1 is from the Kaggle platform [28], and 1 is taken from a Johns Hopkins repository [29]. The other datasets have been extracted from corresponding domain specific platforms. The datasets mainly belong to 9 different domains: tourism, banking, web, energy, sales, economics, transportation, health, and nature. Three datasets, the M1 [1], M3 [2], and M4 [3, 4] datasets, contain series belonging to multiple domains. All datasets are explained in detail in the Appendix (supplementary materials).

2.1 Data format

We introduce a new format to store time series data, based on the Weka ARFF file format [23]. We use the file extension .tsf and it is comparable with the .ts format used in the sktime time series repository [24], but we deem it more streamlined and more flexible. The basic idea of the file format is that each data file can contain 1) attributes that are constant throughout the whole dataset (e.g., the forecasting horizon, whether the dataset contains missing values or not), 2) attributes that are constant throughout a time series (e.g., its name, its position in a hierarchy, product information for product sales time series), and 3) attributes that are particular to each data point (the value of the series, or timestamps for non-equally spaced series). An example of series in this format is shown in Figure 1.

The original Weka ARFF file format already deals well with the first two types of such attributes. Using this file format, in our format, each time series file contains tags describing the meta-information

```
# Dataset Information
# This dataset was used in the NN5 forecasting competition.
# It contains 111 daily time series from the banking domain.
# The goal is predicting the daily cash withdrawals from ATMs in UK.
#
# For more details, please refer to
# Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A., 2012.
# A review and comparison of strategies for multi-step ahead time series forecasting based on
# the nn5 forecasting competition. Expert Systems with Applications 39(8), 7067 - 7083
#
# Neural Forecasting Competitions, 2008.
# NN5 forecasting competition for artificial neural networks and computational intelligence.
# Accessed: 2020-05-10. URL http://www.neural-forecasting-competition.com/NN5/
#
@relation NN5
@attribute series_name string
@attribute start_timestamp date
@frequency daily
@horizon 56
@missing true
@equallength true
@data
T1:1996-03-18 00-00-00:13.4070294784581,14.7250566893424,20.5640589569161,34.7080498866213,26
T2:1996-03-18 00-00-00:11.5504535147392,13.5912698412698,15.0368480725624,21.5702947845805,19
T3:1996-03-18 00-00-00:5.640589569161,14.3990929705215,24.4189342403628,28.7840136054422,20.6
T4:1996-03-18 00-00-00:13.1802721088435,8.44671201814059,19.515306122449,28.8832199546485,19.
T5:1996-03-18 00-00-00:9.77891156462585,10.8134920634921,21.6128117913832,38.5204081632653,24
T6:1996-03-18 00-00-00:9.24036281179138,11.6354875283447,12.1031746031746,21.4143990929705,24
T7:1996-03-18 00-00-00:14.937641723356,16.2840136054422,16.6666666666667,23.5685941043084,26.
T8:1996-03-18 00-00-00:2.89115646258503,12.3582766439909,16.3832199546485,30.1587301587302,31
T9:1996-03-18 00-00-00:7.34126984126984,9.15532879818594,10.5867346938776,12.5,7.157029478458
T10:1996-03-18 00-00-00:10.2891156462585,12.7125850340136,14.4416099773243,19.4019274376417,2
```

Figure 1: An example of the file format for the NN5 daily dataset.

of the corresponding dataset such as *@frequency* (seasonality), *@horizon* (expected forecast horizon), *@missing* (whether the series contain missing values) and *@equallength* (whether the series have equal lengths). We note that in principle these attributes can be freely defined by the user and the file format does not need any of these values to be defined in a certain way, though the file readers reading the files may rely on existence of attributes with certain names and assume certain meanings. Next, there are attributes in each dataset which describe series-wise properties, where the tag *@attribute* is followed by the name and type. Examples are *series_name* (the unique identifier of a given series) and *start_ timestamp* (the start timestamp of a given series). Again, the format has the flexibility to include any additional series-wise attributes as preferred by users.

Following the ARFF file format, the data are then listed under the *@data* tag after defining attributes and meta-headers, and attribute values are separated by colons. The only extension that our format has compared with the original ARFF file format, is that the time series then are appended to their attribute vector as a comma-separated variable-length vector. As this vector can have a different length for each instance, this cannot be represented in the original ARFF file format. In particular, a time series with m number of attributes and n number of values can be shown as:

$$< attribute_1 >:< attribute_2 >: ... :< attribute_m >:< s_1, s_2, ..., s_n > \tag{1}$$

The missing values in the series are indicated using the “?” symbol. Code to load datasets in this format into R and Python is available in our github repository at `https://github.com/rakshitha123/TSForecasting`.

3 Methods

This section details the feature analysis and baseline evaluation we conducted on the datasets in our repository.

3.1 Feature analysis

We characterise the datasets in our archive to analyse the similarities and differences between them, to gain a better understanding on where gaps in the repository may be and what type of data are prevalent in applications. This may also help to select suitable forecasting methods for different types of datasets. We analyse the characteristics of the datasets using the tsfeatures [25] and catch22 [26] feature extraction methods. All extracted features are publicly available in our website `https://forecastingdata.org/` for further research use. Due to their large size, we have not been able to extract features from the London smart meters, wind farms, solar power, and wind power datasets, which is why we exclude them from this analysis.

We extract 42 features using the *tsfeatures* function in the R package *tsfeatures* [25] including mean, variance, autocorrelation features, seasonal features, entropy, crossing points, flat spots, lumpiness, non-linearity, stability, Holt-parameters, and features related to the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test [30] and the Phillips–Perron (PP) test [31]. For all series that have a frequency greater than daily, we consider multi-seasonal frequencies when computing features. Therefore, the amount of features extracted is higher for multi-seasonal datasets as the seasonal features are individually calculated for each season presented in the series. Furthermore, if a series is short and does not contain two full seasonal cycles, we calculate the features assuming a non-seasonal series (i.e., setting its frequency to "one" for the feature extraction). We use the *catch22_all* function in the R package *catch22* [32] to extract the catch22 features from a given time series. The features are a subset of 22 features from the *hctsa* package [33] which includes the implementations of over 7000 time series features. The computational cost of the catch22 features is low compared with all features implemented in the hctsa package.

For the feature analysis, we consider 5 features, as suggested by Bojer and Meldgaard [34]: first order autocorrelation (ACF1), trend, entropy, seasonal strength, and the Box-Cox transformation parameter, lambda. The *BoxCox.lambda* function in the R package *forecast* [35] is used to extract the Box-Cox transformation parameter from each series, with default parameters. The other 4 features are extracted using *tsfeatures*. Since this feature space contains 5 dimensions, to compare and visualise the features across multiple datasets, we reduce the feature dimensionality to 2 using Principal Component Analysis [PCA, 36].

The numbers of series in each dataset are significantly different, e.g., the CIF 2016 monthly dataset and M4 monthly dataset contain 72 and 48,000 series, respectively. Hence, if all series were considered to calculate the PCA components, those components would be dominated by datasets that have large amounts of series. Therefore, for datasets that contain more than 300 series, we randomly take a sample of 300 series, before constructing the PCA components across all datasets. Once the components are calculated, we map all series of all datasets into the resulting PCA feature space. We note that we use PCA for dimensionality reduction over other advanced dimensionality reduction algorithms such as t-Distributed Stochastic Neighbor Embedding [t-SNE, 37] due to this capability of constructing the basis of the feature space with a reduced sample of series with the possibility to then map all series into the space afterwards.

3.2 Baseline forecasting models

In the forecasting space, benchmarking against simple benchmarks is vital [38] as even simple benchmarks can oftentimes be surprisingly competitive. However, many works in the machine learning space are notoriously weak when it comes to proper benchmarking for time series forecasting [39]. To fill this gap, we evaluate the performance of 11 different baseline forecasting models over the datasets in our repository using a fixed origin evaluation scheme, so that researchers that use the data in our repository can directly benchmark their forecasting algorithms against these baselines. The baseline models include 6 traditional univariate forecasting models: Exponential Smoothing [ETS, 40], Auto-Regressive Integrated Moving Average [ARIMA, 41], Simple Exponential Smoothing (SES), Theta [42], Trigonometric Box-Cox ARMA Trend Seasonal [TBATS, 43] and Dynamic Harmonic Regression ARIMA [DHR-ARIMA, 44], and 5 global forecasting models: a linear Pooled Regression model [PR, 45], Feed-Forward Neural Network [FFNN, 46], CatBoost [47], DeepAR [48] and N-BEATS [49].

Again, we do not consider the London smart meters, wind farms, solar power, and wind power datasets for both univariate and global model evaluations, and the Kaggle web traffic daily dataset for the global model evaluations, as the computational cost of running these models was not feasible in our experimental environment.

We use the R packages *forecast* [50], *glmnet* [51], *catboost* [47] and *nnet* [52] to implement the 6 traditional univariate forecasting methods, the globally trained PR method, CatBoost and FFNN, respectively. We use the implementations of DeepAR and N-BEATS available from the Python package `GluonTS` [53] with their default hyperparameters.

The Theta, SES, and PR methods are evaluated for all datasets. ETS and ARIMA are evaluated for yearly, quarterly, monthly, and daily datasets. We consider the datasets with small frequencies, namely, 10 minutely, half hourly, and hourly as multi-seasonal and hence, TBATS and DHR-ARIMA are evaluated for those datasets instead of ETS and ARIMA due to their capability of dealing with multiple seasonalities [54]. TBATS and DHR-ARIMA are also evaluated for weekly datasets due to their capability of dealing with long non-integer seasonal cycles present in weekly data [55].

Forecast horizons are chosen for each dataset to evaluate the model performance. For all competition datasets, we use the forecast horizons originally employed in the competitions. For the remaining datasets, 12 months ahead forecasts are obtained for monthly datasets, 8 weeks ahead forecasts are obtained for weekly datasets, except the solar weekly dataset, and 30 days ahead forecasts are obtained for daily datasets. For the solar weekly dataset, we use a horizon of 5 as the series in this dataset are relatively short compared with other weekly datasets. For half-hourly, hourly and other low frequency datasets, we set the forecasting horizon to one week, e.g., 168 is used as the horizon for hourly datasets.

The number of lagged values used in the PR models are determined based on a heuristic suggested in prior work [56]. Generally, the number of lagged values is chosen as the seasonality multiplied with 1.25. If the datasets contain short series and it is impossible to use the above defined number of lags, for example in the Dominick and solar weekly datasets, then the number of lagged values is chosen as the forecast horizon multiplied with 1.25, assuming that the horizon is not arbitrarily chosen but based on certain characteristics of the time series structure. When defining the number of lagged values for multi-seasonal datasets, we consider the corresponding weekly seasonality value, e.g., 168 for hourly datasets. If it is impossible to use the number of lagged values obtained with the weekly seasonality due to high memory and computational requirements, for example with the traffic hourly and electricity hourly datasets, then we use the corresponding daily seasonality value to define the number of lags, e.g., 24 for hourly datasets. In particular, due to high memory and computational requirements, the number of lagged values is chosen as 50 for the solar 10 minutely dataset which is less than the above mentioned heuristics based on seasonality and forecasting horizon suggest.

4 Results & discussion

This section details the results of feature analysis and baseline evaluation together with a discussion of the results.

4.1 Feature analysis results

Figure 2 shows hexbin plots of the normalised density values of the low-dimensional feature space generated by PCA across ACF1, trend, entropy, seasonal strength and Box-Cox lambda for 20 datasets. The figure highlights the characteristics among different datasets. For the M competition datasets, the feature space is highly populated on the left-hand side and hence, denoting high trend and ACF1 levels in the series. The tourism yearly dataset also shows high trend and ACF1 levels. In contrast, the car parts, hospital, and Kaggle web traffic datasets show high density levels towards the right-hand side, indicating a higher degree of entropy. The presence of intermittent series can be considered as the major reason for the higher degree of entropy in the Kaggle web traffic and car parts datasets. The plots confirm the claims of prior similar studies [34, 57] that the M competition datasets are significantly different from the Kaggle web traffic dataset.

The monthly datasets generally show high seasonal strengths compared with datasets of other frequencies. Quarterly datasets also demonstrate high seasonal strengths except for the M4 quarterly dataset. In contrast, the datasets with high frequencies such as weekly, daily, and hourly show low seasonal strengths except for the NN5 weekly and NN5 daily datasets.

Related to the shapes of the feature space, the 3 yearly datasets: M3, M4, and tourism show very similar shapes and density populations indicating they have similar characteristics. The M4 quarterly dataset also shows a similar shape as the yearly datasets, even though it has a different frequency. The

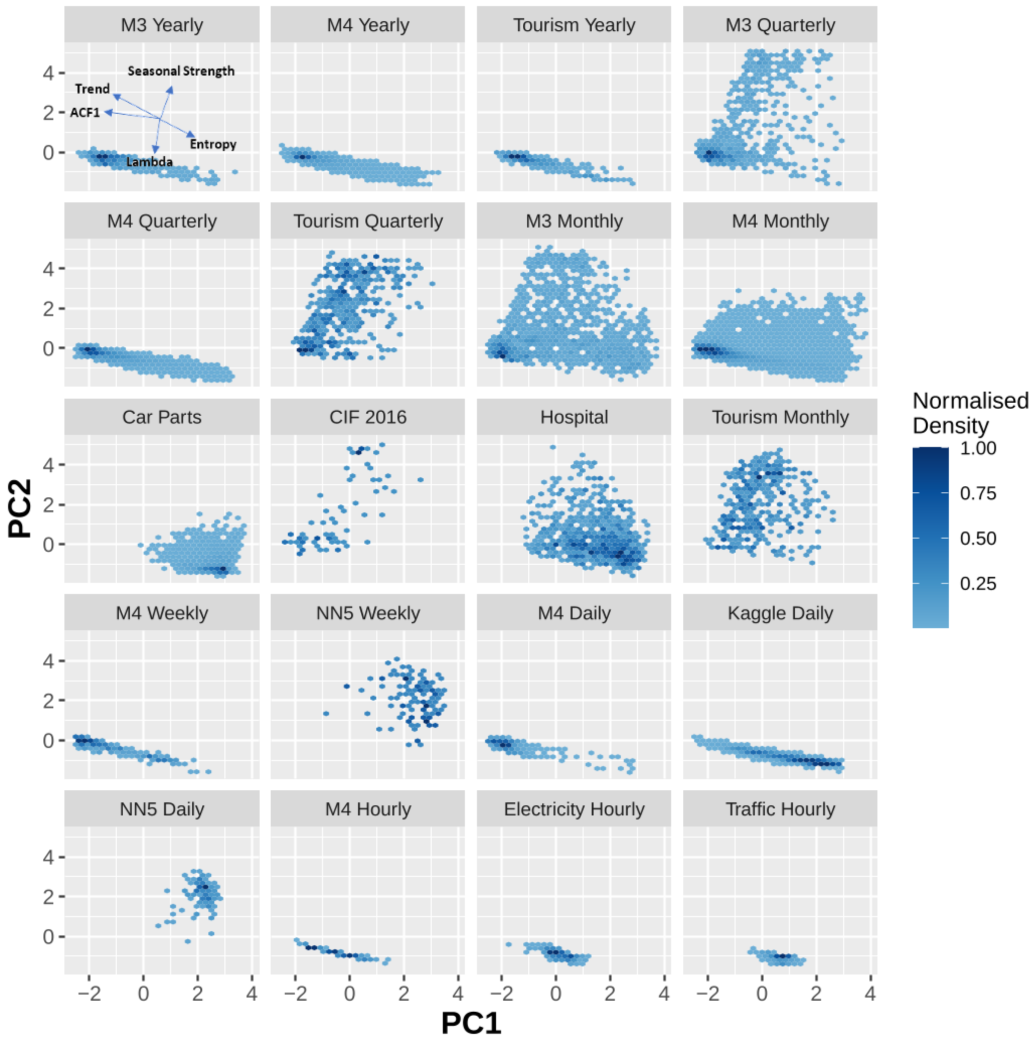

Figure 2: Hexbin plots showing the normalised density values of the low-dimensional feature space generated by PCA across ACF1, trend, entropy, seasonal strength, and Box-Cox lambda for 20 datasets. The dark and light hexbins denote the high and low density areas, respectively. The M3 Yearly facet shows the directions of the 5 features, which are the same across all facets.

other 2 quarterly datasets M3 and tourism are different, but similar to each other. The M3 and M4 monthly datasets are similar to each other in terms of both shape and density population. Furthermore, the electricity hourly and traffic hourly datasets have similar shapes and density populations, whereas the M4 hourly dataset has a slightly different shape compared with them. The daily datasets show different shapes and density populations, where the NN5 daily dataset is considerably different from the other 2 daily datasets: M4 and Kaggle web traffic, in terms of shape and all 3 daily datasets are considerably different from each other in terms of density population. The weekly datasets also show different shapes and density populations compared with each other.

PCA plots showing the normalized density values of all datasets corresponding with both tsfeatures and catch22 features are available in the Appendix (supplementary materials).

4.2 Baseline evaluation results

It is very difficult to define error measures for forecasting that perform well under all situations [58], in the sense that it is difficult to define a scale-free measure that works for any type of nonstationarity in the time series. Thus, how to best evaluate forecasts is still an active area of research, and (especially in the machine learning area) researchers often use ad-hoc, non-adequate measures. For example, usage of the Mean Absolute Percentage Error (MAPE) for normalised data between 0 and 1 may result in undefined or heavily skewed measures, or errors using the mean of a series like the Root Relative Squared Error [RSE, 27] will not work properly for series where the mean is essentially meaningless, such as series with steep trends. We use four error metrics that – while having their own problems – are common for evaluation in forecasting, namely the Mean Absolute Scaled Error [MASE, 59], symmetric MAPE (sMAPE), Mean Absolute Error [MAE, 60], and Root Mean Squared Error (RMSE) to evaluate the performance of the seven baseline forecasting models explained in Section 3.2. For datasets containing zeros, calculating the sMAPE error measure may lead to divisions by zero. Hence, we also consider the variant of the sMAPE proposed by Suilin [61] which overcomes the problems with small values and divisions by zero of the original sMAPE. We report the original sMAPE only for datasets where divisions by zero do not occur. Equations 2, 3, 4, 5, and 6, respectively, show the formulas of MASE, sMAPE, modified sMAPE, MAE, and RMSE, where M is the number of data points in the training series, S is the seasonality of the dataset, h is the forecast horizon, F_k are the generated forecasts and Y_k are the actual values. We set the parameter ϵ in Equation 4 to its proposed default of 0.1.

$$MASE = \frac{\sum_{k=M+1}^{M+h} |F_k - Y_k|}{\frac{h}{M-S} \sum_{k=S+1}^{M} |Y_k - Y_{k-S}|} \tag{2}$$

$$sMAPE = \frac{100\%}{h} \sum_{k=1}^{h} \frac{|F_k - Y_k|}{(|Y_k| + |F_k|)/2} \tag{3}$$

$$msMAPE = \frac{100\%}{h} \sum_{k=1}^{h} \frac{|F_k - Y_k|}{max(|Y_k| + |F_k| + \epsilon, 0.5 + \epsilon)/2} \tag{4}$$

$$MAE = \frac{\sum_{k=1}^{h} |F_k - Y_k|}{h} \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^{h} |F_k - Y_k|^2}{h}} \tag{6}$$

The MASE measures the performance of a model compared with the in-sample average performance of a one-step-ahead naïve or seasonal naïve (snaïve) benchmark. For multi-seasonal datasets, we use the length of the shortest seasonality to calculate the MASE. For the datasets where all series contain at least one full seasonal cycle of data points, we consider the series to be seasonal and calculate MASE values using the snaïve benchmark. Otherwise, we calculate the MASE using the naïve benchmark, effectively treating the series as non-seasonal.

The error metrics are defined for each series individually. We further calculate the mean and median values of the error metrics over the datasets to evaluate the model performance and hence, each model is evaluated using 10 error metrics for a particular dataset: mean MASE, median MASE, mean sMAPE, median sMAPE, mean msMAPE, median msMAPE, mean MAE, median MAE, mean RMSE and median RMSE. Table 2 shows the mean MASE of the SES, Theta, ETS, ARIMA, TBATS, DHR-ARIMA, and PR models on the same 20 datasets we considered for the feature analysis. The results of all baselines across all datasets on all 10 error metrics are available in the Appendix.

Overall, SES shows the worst performance and Theta shows the second-worst performance across all error metrics. ETS and ARIMA show a mixed performance on the yearly, monthly, quarterly, and daily datasets but both outperform SES and Theta. TBATS generally shows a better performance than DHR-ARIMA on the high frequency datasets. For our experiments, we always set the maximum order of Fourier terms used with DHR-ARIMA to $k = 1$. Based on the characteristics of the datasets, k can be tuned as a hyperparameter and it may lead to better results compared with our results. Compared with SES and Theta, both TBATS and DHR-ARIMA show superior performance.

Table 2: Mean MASE results. The best model across each dataset is highlighted in boldface.

Dataset	SES	Theta	ETS	ARIMA	TBATS	DHR-ARIMA	PR	CatBoost	FFNN	DeepAR	N-BEATS
NN5 Daily	1.521	0.885	**0.865**	1.013	-	-	1.263	0.973	1.409		
NN5 Weekly	0.903	0.885	-	-	0.872	0.887	0.854	**0.853**	1.009		
CIF 2016	1.291	0.997	**0.841**	0.929	-	-	1.019	1.175	2.434		
Kaggle Daily	0.924	0.928	1.231	**0.890**	-	-	-	-	-	-	-
Tourism Yearly	3.253	**3.015**	3.395	3.775	-	-	3.516	3.553	7.352		
Tourism Quarterly	3.210	1.661	**1.592**	1.782	-	-	1.643	1.793	6.424		
Tourism Monthly	3.306	1.649	**1.526**	1.589	-	-	1.678	1.699	5.159		
Traffic Hourly	1.922	1.922	-	-	2.482	2.535	**1.281**				
Electricity Hourly	4.544	4.545	-	-	3.690	4.602	**2.912**				
M3 Yearly	3.167	**2.774**	2.860	3.417	-	-	3.223	3.788	7.938		
M3 Quarterly	1.417	**1.117**	1.170	1.240	-	-	1.248	1.441	4.212		
M3 Monthly	1.091	**0.864**	0.865	0.873	-	-	1.010	1.065	2.215		
M4 Yearly	3.981	**3.375**	3.444	3.876	-	-	3.625				
M4 Quarterly	1.417	1.231	**1.161**	1.228	-	-	1.316				
M4 Monthly	1.150	0.970	**0.948**	0.962	-	-	1.080				
M4 Weekly	0.587	0.546	-	-	0.504	0.550	**0.481**	0.615	5.266		
M4 Daily	1.154	**1.153**	1.239	1.179	-	-	1.162	1.593			
M4 Hourly	11.607	11.524	-	-	2.663	13.557	**1.662**	1.771			
Carparts	0.897	0.914	0.925	0.926	-	-	**0.755**				
Hospital	0.813	**0.761**	0.765	0.787	-	-	0.782	0.798	0.986		

The globally trained PR models show a mixed performance compared with the traditional univariate forecasting models. The performance of the PR models is considerably affected by the number of lags used during model training, performing better as the number of lags is increased. The number of lags we use during model training is quite high with the high-frequency datasets such as hourly, compared with the other datasets and hence, PR models generally show a better performance than the traditional univariate forecasting models on all error metrics across those datasets. But on the other hand, the memory and computational requirements are also increased when training PR models with larger numbers of lags. Furthermore, the PR models show better performance across intermittent datasets such as car parts, compared with the traditional univariate forecasting models.

We note that the MASE values of the baselines are generally high on multi-seasonal datasets. For multi-seasonal datasets, we consider longer forecasting horizons corresponding to one week unless they are competition datasets. As benchmark in the MASE calculations, we use a seasonal naïve forecast for the daily seasonality. As therewith the MASE compares the forecasts of longer horizons (up to one week) with the in-sample snaïve forecasts obtained with shorter horizons (one day), the MASE values of multi-seasonal datasets are considerably greater than one across all baselines. Furthermore, the error measures are not directly comparable across datasets as we consider different forecasting horizons with different datasets.

5 Conclusion

Recently, global forecasting models and multivariate models have shown huge potential in providing accurate forecasts for collections of time series compared with the traditional univariate benchmarks. However, there are currently no comprehensive time series forecasting benchmark data archives available that contain datasets to facilitate the evaluation of these new forecasting algorithms. In this paper, we have presented the details of an archive that contains 20 publicly available time series datasets with different frequencies from varied domains. We have also characterised the datasets and have identified the similarities and differences among them by conducting a feature analysis exercise using tsfeatures and catch22 features extracted from each series. Finally, we have evaluated the performance of seven baseline forecasting models over all datasets across ten error metrics to enable other researchers to benchmark their own forecasting algorithms directly against those.

Acknowledgements

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.

References

[1] S. Makridakis, A. Andersen, R. F. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. L. Winkler. The accuracy of extrapolation (time series) methods: results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153, 1982.

[2] S. Makridakis and M. Hibon. The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451–476, 2000.

[3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The M4 competition: results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

[4] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54 – 74, 2020. ISSN 0169-2070.

[5] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The M5 accuracy competition: results, findings and conclusions, 2020.

[6] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067 – 7083, 2012.

[7] Google. Web traffic time series forecasting, 2017. URL `https://www.kaggle.com/c/web-traffic-time-series-forecasting`.

[8] S. Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

[9] T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, and L. Callot. Criteria for classifying forecasting methods. *International Journal of Forecasting*, 36 (1):167–177, 2020.

[10] P. Montero-Manso and R. J. Hyndman. Principles and algorithms for forecasting groups of time series: locality and globality. *International Journal of Forecasting*, 2021. URL `https://arxiv.org/abs/2008.00444`. to appear.

[11] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems*, page 5243–5253, 2019.

[12] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, page 7785–7794, 2018.

[13] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka. A multi-horizon quantile recurrent forecaster. In *31st Conference on Neural Information Processing Systems, Time Series Workshop*, 2017.

[14] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus. High-dimensional multivariate forecasting with low-rank Gaussian copula processes. In *Advances in Neural Information Processing Systems*, page 6827–6837, 2019.

[15] R. Sen, H.-F. Yu, and I. S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. In *Advances in Neural Information Processing Systems*, page 4837–4846, 2019.

[16] H.-F. Yu, N. Rao, and I. S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in Neural Information Processing Systems*, page 847–855, 2016.

[17] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, 2020.

[18] D. Dua and C. Graff. UCI machine learning repository. `https://archive.ics.uci.edu/`, 2017.

[19] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6): 1293–1305, 2019.

[20] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The UEA multivariate time series classification archive, 2018, 2018.

[21] R. J. Hyndman and Y. Yang. tsdl: time series data library. v0.1.0. `https://pkg.yangzhuoranyang./tsdl/`, 2018.

[22] Forecasting & Strategy Unit. Foredeck. `http://fsudataset.com/`, 2019.

[23] G. Paynter, L. Trigg, and E. Frank. Attribute-relation file format (arff). `https://www.cs.waikato.ac.nz/ml/weka/arff.html`, 2008.

[24] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király. sktime: a unified interface for machine learning with time series. In *Workshop on Systems for ML at NeurIPS 2019*, 2019.

[25] R. J. Hyndman, Y. Kang, P. Montero-Manso, T. Talagala, E. Wang, Y. Yang, and M. O'Hara-Wild. *tsfeatures: time Series Feature Extraction*, 2020. URL `https://pkg.robjhyndman.com/tsfeatures/`.

[26] Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. catch22: Canonical time-series characteristics. *Data Mining and Knowledge Discovery*, 33(6): 1821–1852, 2019.

[27] G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 95–104, New York, NY, USA, 2018.

[28] Kaggle. `https://www.kaggle.com/`, 2019.

[29] Center for Systems Science and Engineering at Johns Hopkins University. COVID-19 data repository. `https://github.com/CSSEGISandData/COVID-19`, 2020.

[30] C. Baum. KPSS: stata module to compute Kwiatkowski-Phillips-Schmidt-Shin test for stationarity, 2018. URL `https://EconPapers.repec.org/RePEc:boc:bocode:s410401`.

[31] P. Phillips and P. Perron. Testing for a unit root in time series regression. *Cowles Foundation, Yale University, Cowles Foundation Discussion Papers*, 75, January 1986.

[32] C. H. Lubba. *catch22: subset of hctsa-features*, 2018.

[33] B. D. Fulcher and N. S. Jones. hctsa: a computational framework for automated time-series phenotyping using massive feature extraction. *Cell Sys.*, 5:527, 2017.

[34] C. S. Bojer and J. P. Meldgaard. Kaggle forecasting competitions: an overlooked learning opportunity. *International Journal of Forecasting*, 2020. ISSN 0169-2070.

[35] R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008. URL `http://www.jstatsoft.org/v27/i03`.

[36] I. Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer, 2011. ISBN 978-3-642-04898-2.

[37] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[38] J. S. Armstrong. *Evaluating Forecasting Methods*, pages 443–472. Springer US, Boston, MA, 2001. ISBN 978-0-306-47630-3.

[39] S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and machine learning forecasting methods: concerns and ways forward. *PLOS ONE*, 13(3):1–26, March 2018.

[40] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer, 2008.

[41] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. HoldenDay Inc., 1990.

[42] V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16(4):521–530, 2000.

[43] A. M. De Livera, R. J. Hyndman, and R. D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.

[44] R. J. Hyndman. *fpp2: data for "Forecasting: Principles and Practice" (2nd Edition)*, 2018. URL `https://CRAN.R-project.org/package=fpp2`.

[45] J. R. Trapero, N. Kourentzes, and R. Fildes. On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, 66(2):299 – 307, 2015.

[46] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[47] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[48] V. Flunkert, D. Salinas, and J. Gasthaus. DeepAR: probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2017.

[49] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. `https://arxiv.org/abs/1905.10437`, 2019.

[50] R. J. Hyndman, G. Athanasopoulos, C. Bergmeir, G. Caceres, L. Chhay, M. O'Hara-Wild, F. Petropoulos, S. Razbash, E. Wang, F. Yasmeen, R Core Team, R. Ihaka, D. Reid, D. Shaub, Y. Tang, and Z. Zhou. forecast: Forecasting Functions for Time Series and Linear Models, 2021. URL `https://CRAN.R-project.org/package=forecast`. R package version 8.14.

[51] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

[52] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4`.

[53] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, L. Stella, A. C. TÃ¼rkmen, and Y. Wang. GluonTS: probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020.

[54] K. Bandara, C. Bergmeir, and H. Hewamalage. LSTM-MSNet: leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[55] R. Godahewa, C. Bergmeir, G. I. Webb, and P. Montero-Manso. A strong baseline for weekly time series forecasting. https://arxiv.org/abs/2010.08158, 2020.

[56] H. Hewamalage, C. Bergmeir, and K. Bandara. Recurrent neural networks for time series forecasting: current status and future directions. *International Journal of Forecasting*, 2021.

[57] C. Fry and M. Brundage. The M4 forecasting competition - a practitioner's view. *International Journal of Forecasting*, 36(1):156 – 160, 2020. ISSN 0169-2070.

[58] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 3rd edition, 2021. URL `http://OTexts.com/fpp3`.

[59] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[60] Mean absolute error. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 652–652. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8.

[61] A. Suilin. kaggle-web-traffic. `https://github.com/Arturus/kaggle-web-traffic`, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Sections 2, 3 and 4.
   (b) Did you describe the limitations of your work? [Yes] See Sections 3 and 4. We mention that we could not consider all datasets for feature analysis and benchmark evaluation due to the high computational requirements of some datasets.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Not applicable to our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Not applicable to our work.
   (b) Did you include complete proofs of all theoretical results? [N/A] Not applicable to our work.
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix A of supplementary materials.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4.2 and Appendix C of supplementary materials.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D of supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix A.6 and A.7 of supplementary materials.
   (b) Did you mention the license of the assets? [Yes] See Appendix A.3 and A.5 of supplementary materials.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix A of supplementary materials.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix A.5 of supplementary materials.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Not applicable to our work.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] Not applicable to our work.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not applicable to our work.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not applicable to our work.