Forecast Evaluation Techniques for I4.0 Systems

Andrey Davydenko, Cuong Sai, and Maxim Shcherbakov

Abstract We focus on forecast evaluation techniques that comply with the design principles of Industry 4.0 (or I4.0). The I4.0 concept refers to trends and principles attributed to the 4th industrial revolution and, in particular, assumes the following capabilities of automated systems: interoperability, decentralization, real-time processing, and service-orientation. Generally, effective forecast evaluation requires us to store both actuals and forecasts. We look at how to handle rolling-origin forecasts produced for many series over multiple horizons. This setup is met both in research (e.g., in forecasting competitions or when proposing a new method) and in practice (when tracking/reporting forecasting performance). We show how to ensure access to all the variables required for exploratory analysis and performance measurement. We propose flexible yet simple and effective data schemas allowing the storage and exchange of actuals, forecasts, and any additional relevant info. We show how to construct various tools for forecast exploration and evaluation using the schemas proposed. In particular, we present our implementation of a prediction-realization diagram showing forecasts from different methods on one plot. We propose special tools for measuring the quality of point and interval rolling-origin predictions across many time series and over multiple horizons. The workflow for using techniques proposed is illustrated using R codes.

Keywords Forecasting · Forecast evaluation · Forecasting accuracy · Forecast bias · Interval predictions · Data visualization · R · I4.0

A. Davydenko
JSC CSBI, Saint-Petersburg, Russia
e-mail: andrey@live.co.uk

C. Sai (✉) · M. Shcherbakov
Computer-aided Design Department, Volgograd State Technical University, Volgograd, Russia
e-mail: svcuonghvktqs@gmail.com

A. G. Kravets et al. (eds.), *Cyber-Physical Systems: Modelling and Intelligent Control*, Studies in Systems, Decision and Control 338,
https://doi.org/10.1007/978-3-030-66077-2_7

1 Introduction

Forecasting methods are used in various fields ranging from weather forecasting to supply chain management. It is important for companies to produce good forecasts in order to operate efficiently. In order to know how good a forecasting method is, we need to compare forecasts against corresponding actuals being obtained. In other words, we need empirical evaluation to assess forecast performance. A number of forecasting competitions have been held to empirically evaluate alternative methods (e.g., [1] tested 24 methods using 3003 series and recently [2] tested well-known machine learning and parametric methods using 100,000 series). These competitions have had a huge influence on the field of forecasting by focusing on forecasting performance, rather than on models design. Choosing a good metric to compare alternative forecasts is itself a challenging and controversial task (for an overview of existing approaches see, e.g., [3]).

This chapter addresses the following questions: (1) How to store forecast data in order to ensure effective forecast evaluation? (2) What tools can be used to report forecast performance in an informative and interpretable way? In particular, how to measure accuracy of point forecasts, how to measure forecast bias, and how to measure the quality of interval predictions? Some preliminary results of our research were earlier presented by us at CyberPhy-2020 conference, see [4].

We show that formats used in well-known datasets and packages do not provide some important capabilities (for example, by not allowing the storage of rolling-origin forecasts and not ensuring cross-platform and real-time capabilities). After suggesting a more suitable format we illustrate how it can be used as a basis for building forecast evaluation tools.

Generally, the solution we propose in this chapter can be seen as a statistical framework that involves the elements suggested in [5]: (i) a setup describing the tasks and requirements, (ii) principles used to ensure effective implementation, (iii) the description of tools and algorithms, (iv) codes, and (v) a workflow describing step-by-step instructions to solve the tasks identified in the setup.

The next section outlines the setup where we summarize typical settings of obtaining and evaluating forecasts and outline the principles we adopt in our solution. Then we present our view of how forecast evaluation workflow should look like. We then propose data schemas that meeting the requirements stated. Then we demonstrate how the schemas can be used to implement tools for forecast exploratory analysis and performance measurement. Data formats and tools presented can be used regardless of a scripting language or database, but our examples use R and plain csv-files. We conclude by providing a summary of our recommendations for using the tools presented.

2 Forecast Evaluation Setup, Principles, and Terminology

We propose the following forecast evaluation setup summarizing (i) what kinds of data we want to work with, (ii) what we want to obtain as a result of the analysis, (iii) under what requirements and what principles.

1. Suppose we have a set of time series. The set can contain from one to a relatively large number of series (say, millions or hundreds of thousands).
2. For each series we want to store both actuals and forecasts. In particular, we need to store out-of-sample forecasts produced from different origins (we will call them rolling-origin forecasts) over different horizons. Forecasts can be produced using alternative methods. In addition to point forecasts we may want to store prediction intervals (PIs), density forecasts, and additional information related to forecasting process (such as model structure, model coefficients, reasons for judgmental adjustments, etc.).
3. Both actuals and forecasts can be frequently and asynchronously updated as new data becomes available (this scenario is described, e.g., in [6]).

Being able to store each forecast for each horizon and each origin of interest is important because it is often not easy to reproduce forecasts for evaluation purposes. E.g., computing a single forecast can take a substantial time when using computationally intensive methods, such as MCMC, to generate posterior densities [6]. Moreover, judgmental forecasts and judgmental adjustments cannot be reproduced at all.

Given the above settings and considerations, we need convenient means to store and access (and, perhaps, to distribute or exchange) forecast data including actuals, forecasts, and other relevant info. Our aim is to find a solution that would be fast in operation (applicable in industrial settings), cross-platform, and easy to learn and to implement. In particular, we need data structures to implement a reliable and informative forecast cross-validation and a credible comparison of alternative methods. In the case of forecasting competitions, a well-defined approach to store forecasting data enables independent and objective out-of-sample evaluation of forecasting accuracy.

In our solution we aim to comply with the latest trends defined by the Industry 4.0 vision [7]. In particular, this involves interoperability, real-time capabilities, decentralization, service-orientation, and modularity. We also aim to ensure effective forecast evaluation by the use of the most appropriate error metrics and tools. We propose a forecast evaluation framework based on the following rules: (1) Use unified cross-platform data formats to store actuals and forecasts (this is needed to ensure interoperability and modularity, cross-platform data formats make it easier to exchange forecast data). (2) Use separate tables to store forecasts and actuals (this ensures effective updating of data, real-time capabilities, and objective evaluation by third parties). (3) Use well-defined and well-grounded algorithms for measuring forecasting performance (this ensures objective and reproducible forecast evaluation, see [8] for a discussion on the requirements for constructing appropriate error measurement algorithms).

We focus on numeric univariate time series, but the framework proposed is also applicable to multivariate series and to panel or longitudinal studies where observations related to various objects are collected over time. Although the chapter focuses on numeric series, the data structures proposed are capable of storing categorical data.

Some important terms we use are clarified below.

- *Forecast origin*—the most recent historical period for which data is used to obtain a forecast.
- *Forecast horizon*—the number of periods from the forecast origin to the end of the time period being forecast [9].
- *Prediction intervals (PIs)*—the bounds within which future observed values are expected to fall, given a specified level of confidence [9].
- *Prediction interval width (PIW)*—the difference between the upper and the lower bounds of a given PI.
- *Rolling-origin forecast*—a forecasting process in which forecast origin rolls forward in time [10].
- *Density forecast*—a prediction of the distribution of any given statistical object of interest [11].
- *Coverage probability*—the empirical probability that PIs contain actuals.
- *Nominal coverage probability*—the confidence level of PIs.

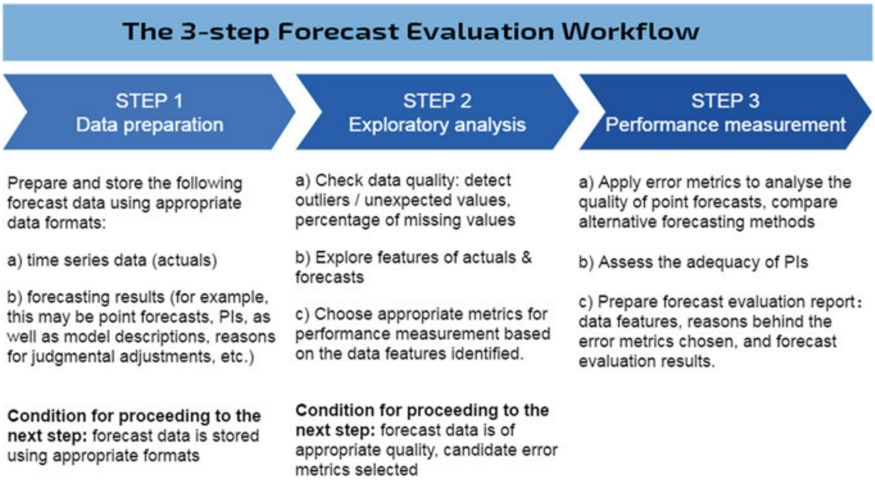
3 Forecast Evaluation General Workflow

Data science analysis involves not only applying a specific algorithm of interest, but also data preparation and data quality checks. Our approach is based on including these steps into a forecast evaluation framework allowing it to comply with more general methodologies, such as CRISP-DM or SEMMA.

We propose the general workflow for forecast evaluation shown on Fig. 1.

The topic of how to obtain forecasts is out of the scope of this chapter. Instead, we look at how forecast data should be stored in order to allow access to previous forecasts, to match them with actuals, and to track forecasting performance. Forecasts can be produced not only by statistical or machine learning methods but also judgmentally. It is important for a forecasting system to have the capabilities of keeping relevant information related to the reasons associated with judgmental estimates. The approach proposed in the next section is capable of keeping this kind of information as well.

The next section proposes data structures allowing the implementation of the framework shown on Fig. 1. In subsequent sections we use these structures for the exploratory analysis and performance measurement steps of the workflow.



The 3-step Forecast Evaluation Workflow

STEP 1
Data preparation

Prepare and store the following forecast data using appropriate data formats:

a) time series data (actuals)

b) forecasting results (for example, this may be point forecasts, PIs, as well as model descriptions, reasons for judgmental adjustments, etc.)

Condition for proceeding to the next step: forecast data is stored using appropriate formats

STEP 2
Exploratory analysis

a) Check data quality: detect outliers / unexpected values, percentage of missing values

b) Explore features of actuals & forecasts

c) Choose appropriate metrics for performance measurement based on the data features identified.

Condition for proceeding to the next step: forecast data is of appropriate quality, candidate error metrics selected

STEP 3
Performance measurement

a) Apply error metrics to analyse the quality of point forecasts, compare alternative forecasting methods

b) Assess the adequacy of PIs

c) Prepare forecast evaluation report: data features, reasons behind the error metrics chosen, and forecast evaluation results.

Fig. 1 Forecast evaluation workflow

4 New Data Formats

4.1 Existing Packages with Forecast Data

Some R packages contain data for forecasting competitions:

- Mcomp: Data from the M-competition and M3-competition [12];
- Tcomp: Data from the Kaggle tourism competition [13];
- tscompdata: Data from the NN3 and NN5 competitions [14];
- M4comp2018: Data from the M4-competition [15].

The above packages use objects to store forecasts and actuals. The downside of this approach is that we need to use R to access data. In other words, the approach is not cross-platform. Besides, there is no unified approach to store rolling-origin forecasts and interval forecasts.

We propose a general format based on special table schemas. The schemas we propose can be implemented in any environment, e.g., using .csv files or a SQL database.

4.2 New Approach and Its Capabilities

This section presents our approach to store forecast data in accordance with what was said in Sect. 2. More specifically, we aim to have a specification with the following capabilities (in order to comply with the I4.0 principles):

- Rolling-origin cross-validation.
- Storage of any type of forecasting results (not only point forecasts, but also interval forecasts, density forecasts, model parameters, reasons for judgmental forecasts, etc.).
- Cross-platform usage & portability.
- Ability to work with data collected for any frequency (hours, minutes, seconds, etc.) and any number of time series.
- Ease of updating actuals and forecasts, ability to store actuals separately from forecasting results, and to store forecasting results for each forecasting method separately.
- Fast access to data.
- Ease of use, understanding, and implementation.

We propose to store forecast data in plain tables (as opposed to the use of environment-specific objects, as in the above packages). This allows using a relational database (RDB) or portable files (e.g., .csv). As RDBs are widely used, companies usually have an IT infrastructure optimized to work with tabular data. DB engines and SQL statements allow accessing such data instantly ensuring real-time capabilities.

We propose the use of the following two major table schemas to store forecasts and actuals:

- Time Series Table Schema (TSTS) to store time series actuals;
- Forecast Table Schema (FTS) to store forecasting results including point forecasts, prediction intervals, and other variables of interest.

According to our approach, forecasts and actuals are stored in separate tables. To slice-and-dice forecast data easier, we may need a table containing both actuals and forecasts. To do this we propose the Actual and Forecast Table Schema (AFTS).

4.3 Time Series Table Schema (TSTS)

In this table schema each actual is stored as a separate single row (Table 1).

We may have additional columns or an additional table specifying time series features (e.g., series description, units, frequency, category, etc.). However, the schema specified by Table 1 includes the columns that are always necessary for forecast evaluation across many series. This specification does not impose any restrictions on data types, but we advise that timestamps be stored as strings. We recommend to use the ISO 8601 standard for timestamps as it allows adequate sorting of rows and correct comparison of strings containing timestamps. For example, “1997-01-20” is less than “1998-01-19” but in case of using another format this may not be the case, e.g.: “20.01.1997” > “19.01.1998”.

Table 1 Time series table schema (TSTS)

Column	Description	Example
series_id*	Time series identifier—a unique name identifying a time series	"Y1"
timestamp*	Any representation of the period to which the observation relates. We recommend the use of the ISO 8601 standard	"1997" in case of yearly data, "1997-01-20" in case of daily data, "1997-11" in case of monthly data, "1997-W03" in case of weekly data, "2018-Q2" in case of quarterly data
value	The value observed	100

*These columns form the composite key for this table schema. A composite key is a combination attributes that must not be duplicated. For this table schema it is <series_id, timestamp>. In other words, we cannot have two (or more) records relating to the same time series and the same period of observation (timestamp)

Here is how the M3-Competition data can look like in the TSTS format:

series_id	value	timestamp
Y1	3103.96	1984
Y1	3360.27	1985
Y1	3807.63	1986
Y1	4387.88	1987
Y1	4936.99	1988
Y1	5379.75	1989

In our examples below we use numeric series, but for categorical data the "value" column can store category identifiers instead of numeric values.

Panel and multivariate series can also be stored using the above schemas. To store panel data the "series_id" column can be replaced with two columns: "panel_id" and "var_id". Alternatively, the "series_id" can be a column containing both a panel identifier and a variable identifier in one string. In the latter case columns "panel_id" and "var_id" can still be added in order to easier query data. The same relates to multivariate series.

For missing observations the corresponding rows can be omitted or corresponding values can be coded as NA's. Sometimes it is necessary to store indications of censored data or out-of-stock events, etc. This should be specified by special rules, which we will not address in this chapter. The purpose of the above schemas is only to set out the general approach.

Table 2 Forecast table schema (FTS)

Column	Description	Example
series_id*	Time series identifier for which the forecast was calculated	"Y1"
timestamp*	Any representation of the period to which the observation relates. We recommend the use of the ISO 8601 standard	"1997" in case of yearly data, "1997-01-20" in case of daily data, "1997-11" in case of monthly data, "1997-W03" in case of weekly data, "2018-Q2" in case of quarterly data
origin_timestamp*	Origin of the forecast (provided in the same format as the timestamp)	"2000" in case of yearly data, "1997-01-23" in case of daily data, etc
horizon*	Forecast horizon	3
method_id*	Method identifier—a unique name that identifies a method by which the forecasting result was produced	"ARIMA"
forecast	Point estimate	234
lo95	The lower limit for the 95% prediction interval	178
hi95	The upper limit for the 95% prediction interval	273
lo90	The lower limit for the 90% prediction interval	162
hi90	The upper limit for the 90% prediction interval	283
…	…	…

*the composite key for this table schema is < series_id, method_id, timestamp, origin_timestamp, horizon>

4.4 Forecast Table Schema (FTS)

This schema is needed to store forecasting results. Each row contains forecasting results relating to a given time series produced using a given method for a given horizon at a given origin. Table 2 specifies the columns required. Columns containing PIs can be added or excluded.

The FTS table may be extended by adding columns to represent additional types of forecasting results (e.g., this may be textual info to store reasons for judgmental forecasts or arrays containing density forecasts). Also, if needed, we can have columns to store structured info using JSON or XML formats. Table 2 specifies typical results used in forecast evaluation.

Here is how forecasts for the M3-Competition data can look like in the FTS format:

series_id	method_id	forecast	horizon	timestamp	origin_timestamp
Y1	NAIVE2	4936.99	1	1989	1988
Y1	NAIVE2	4936.99	2	1990	1988
Y1	NAIVE2	4936.99	3	1991	1988
Y1	NAIVE2	4936.99	4	1992	1988
Y1	NAIVE2	4936.99	5	1993	1988
Y1	NAIVE2	4936.99	6	1994	1988

4.5 Actual and Forecast Table Schema (AFTS)

It is often convenient to work with a table having both actuals and forecasts in one row. By joining columns from TSTS and FTS tables based on the TSTS key fields, the Actual and Forecast Table Schema (AFTS) is obtained.

4.6 Data Preparation Process and Scenario Examples

Since the formats proposed above (TSTS and FTS) assume a table structure, the most efficient way (in terms of updating the data and accessing relevant slices) is to store forecast data within a RDBMS. When TSTS and FTS tables are stored in a RDBMS, relevant pieces of data are obtained through SQL queries. Tables containing both actuals and forecasts (formatted using the AFTS) then can be obtained using a simple INNER JOIN SQL query.

Another scenario is to use .csv files. Then actuals are stored separately from forecasts. Moreover, it is possible to store forecasts from each method in a separate file and then merge the tables for further analysis.

Of course, forecast data can be stored inside an R-package, but the idea behind our approach is still to store data as a table and not as a list of language specific data structures.

5 The Forvision Package

The forvision package for R [16] implements tools to facilitate the workflow shown on Fig. 1. The tools are implemented assuming that forecast data is stored in the TSTS and FTS formats. Our further illustrations will be based on this package, but similar functionality and API design can be implemented in other environments (e.g., in Python or Julia).

5.1 Downloading and Installation

To install and use the package, run this code:

```
install.package(devtools)
devtools::install_github("forvis/forvision", build_vignettes = TRUE)
library(forvision)
```

5.2 Data Used for Illustrations

The package has several example datasets available as data frames. For further illustrations, we will use the following datasets:

- *m3_yearly_ts*—yearly actuals from the M3-competition (format: TSTS),
- *m3_yearly_fc*—yearly forecasts from the M3-competition (format: FTS),
- *m3_quaterly_ts*—quarterly actuals from the M3-competition formatted (format: TSTS),
- *m3_quaterly_fc_pis*—quarterly forecasts containing prediction intervals calculated for the M3-competition data (format: FTS).

The package has a special function to obtain a data frame containing both actuals and forecasts (a table in the AFTS format):

```
m3_yearly_af <- createAFTS(m3_yearly_ts, m3_yearly_fc)
```

5.3 Visual Tools Implementation

The forvision package contains functions implementing visual tools for exploratory analysis and performance measurement. These functions produce objects created with the use of ‘ggplot2’ and ‘dygraphs’ packages. This allows the aesthetics of graphs to be adjusted with high flexibility.

Graphical output can be improved by using rules for choosing most suitable colors and markers for producing cross-sectional plots (see [5]), but here we use default output with no special adjustments.

6 Exploratory Analysis of Forecast Data

When forecast data is stored in the appropriate format, we can define APIs and algorithms to query, slice-and-dice, and visualise forecast data.

This section presents some exploratory tools defined using the data structures we proposed above.

6.1 Prediction-Realization Diagram

The prediction-realization diagram is a scatterplot showing how forecasts correlate with actuals [17]. Here we propose plotting point forecasts and actuals relating to different series, methods, origins, and horizons on the same graph. We use different colors and symbols to denote different methods. The objective is to explore the distribution of forecast errors, to identify outliers and biases, to compare alternative forecasts.

To plot the diagram we need forecast data in the AFTS format. We can use any subset of the initial data set if needed (for example, we can use only forecasts for a specified horizon). This R-code shows the function call for constructing the diagram, Fig. 2 shows the result:

```
plotPRD(m3_yearly_af)
```

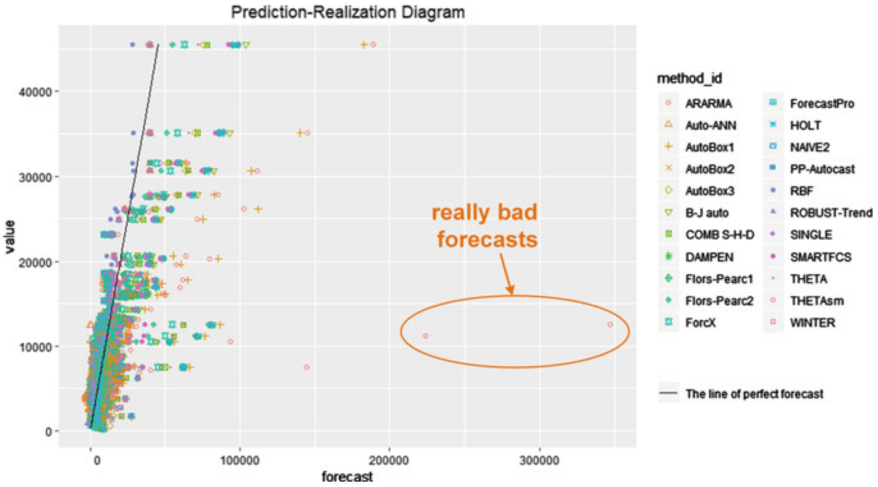


Fig. 2 Prediction-realization diagram for M3 yearly data. Different colors and marks are used to show forecasts relating to different forecasting methods. The Y = X line represents perfect (zero error) forecasts

The graph on Fig. 2 spots some really unwanted cases (when forecast seriously overestimated actuals). For example, having a forecast close to 350,000 units we had actual of only about 11,000 units. We also observe some negative forecasts, but actuals are always non-negative. The distribution is skewed and in some areas points largely overlap in the bottom left corner of the graph. Using a log scale for this diagram is therefore sometimes useful, but here we will work with the raw data to keep things more simple.

Let's take a closer look at the cases spotted. With the AFTS format we can query forecast data in order to get a table with details:

```
subset(m3_yearly_af, forecast > 100000)
```

Query results indicated that the unwanted cases related to series id = "Y113". Below we illustrate how to show these forecasts on a time series graph.

6.2 *Fixed Origin and Fixed Horizon Graphs*

The fixed origin graph shows point forecasts produced for the same time series from the same origin, but for different horizons. It is possible to show forecasts from several methods on the same graph using different colors and symbols.

When actuals table is given in the TSTS format and forecasts table in the FTS format, we can define the following algorithm for producing the fixed origin graph. Firstly, we need to select (from the forecasts table) all point forecasts produced by the methods of interest and relating to the specified origin and specified time series. Then plot a time series graph using actuals stored in the actuals table, then plot the forecasts selected, use different colors for different methods.

Figure 3 shows the cases of poor forecasts we identified based on the prediction-realization diagram (these cases relate to series Y113 of the M3 yearly data set).

Code used to obtain the graph:

```
# Firstly, in order to use the ‘dygraphs’ package
# we must prepare appropriate time-based object timestamps:
library(zoo)
m3_yearly_ts$timestamp_dbo <- as.yearmon(m3_yearly_ts$timestamp, format =
'%Y')
m3_yearly_fc$timestamp_dbo <- as.yearmon(m3_yearly_fc$timestamp, format =
'%Y')

# plot fixed origin graph
plotFixedOrigin(m3_yearly_ts, m3_yearly_fc, "Y113",  1988, c("ARARMA", "HOLT",
"NAIVE2"))
```

Method "ARARMA" sometimes performs badly as it tends to extrapolate trends that do not hold. Thus, in this example the prediction-realization diagram together with the fixed origin graph helped identify the risks of using ARARMA.

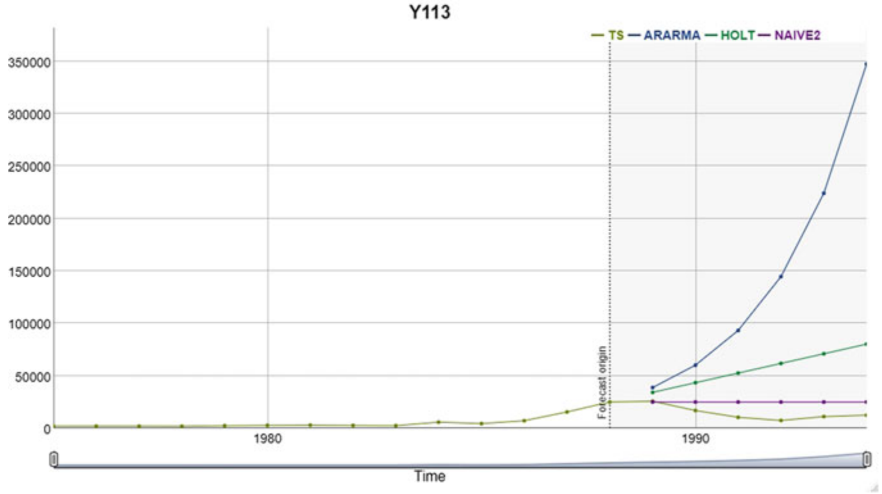


Fig. 3 Fixed origin graph

If we have a data set containing rolling-origin forecasts, the fixed horizon graph can be constructed as well. The fixed horizon graph shows point forecasts produced for the same time series, with the same horizon, but from various rolling origins. The graph is constructed in a similar manner to what we described above.

6.3 *Fancharts*

Fan chart shows point forecasts and prediction intervals produced for the same time series from some given fixed origin, with different horizons. Fan chart relates shows forecasts produced by only one selected method. The objectives is to visually explore PIs, to identify outliers/unexpected results, to assess the uncertainty around forecasts and adequacy of the PIs. Examples of unacceptable PIs: too wide for the practical settings, lower limit is below zero for non-negative time series, the actual coverage does not correspond to the nominal coverage, etc.

Similarly to the point forecasts graphs presented above, fan charts can be constructed given data in the TSTS and FTS formats. Figure 4 shows a fan chart produced using the forvision package.

The code below illustrates the API design based the use of the TSTS and FTS formats.

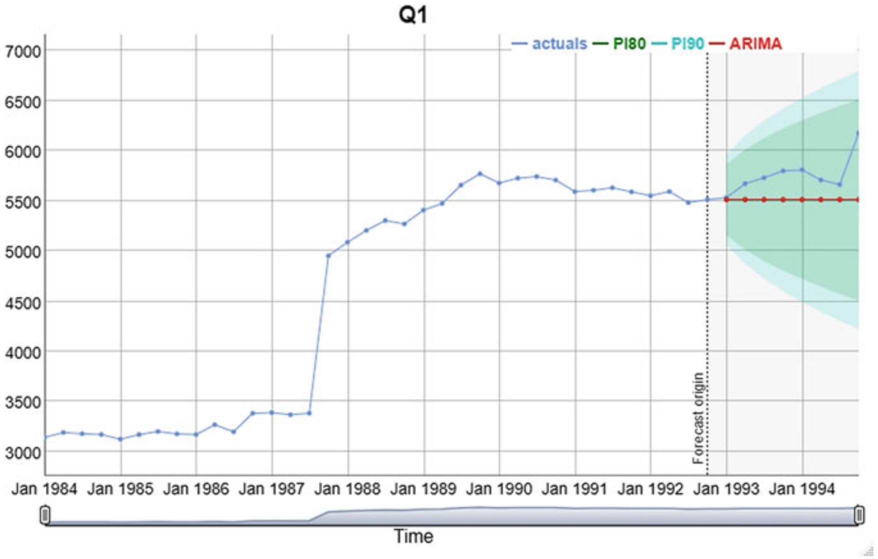


Fig. 4 Fan chart

```
# prepare appropriate time-based object timestamp columns
# for the correct use of the ‘dygraphs’ package
library(zoo)
m3_quarterly_ts$timestamp_dbo <- as.yearqtr(m3_quarterly_ts$timestamp, format
= '%Y-Q%q')
m3_quarterly_fc_pis$timestamp_dbo <- as.yearqtr(m3_quarterly_fc_pis$timestamp,
format = '%Y-Q%q')

# plot a fan chart
plotFan(m3_quarterly_ts, m3_quarterly_fc_pis,  "Q1", "1992-Q4", "ARIMA")
```

Figure 4 shows forecasts produced from fixed origin ‘1992-Q4′ with horizons from 1 to 8. In order to show the dynamics of updating PIs, it is useful to show fan charts for different origins (one example can be found in [6, p. 17]).

By analogy with the fixed horizon graph for point forecasts, for a particular method we also can construct a graph depicting PIs for rolling-origin forecasts, but with some fixed horizon.

7 Measuring Forecasting Performance

When measuring forecast performance we can look at the accuracy and bias of point forecasts and assess the adequacy of prediction intervals. In this section we present effective tools to accomplish these tasks.

7.1 Assessing the Accuracy of Point Forecasts

In the exploratory analysis section we explored the distribution of actuals and forecasts for the M3 yearly data. We spotted zero actuals and negative forecasts, which makes some popular metrics such as MAPE (mean absolute percentage error, [18]) inapplicable. But even we have only positive actuals and forecasts, MAPE can be unreliable or even misleading [8]. Nonetheless, MAPE remains very popular. So we start with an example based on MAPE.

Having input data in the AFTS format lets us construct well-known accuracy versus horizon graphs and tables. Figure 5 shows the "MAPE versus horizon" graph for the M3 early data plotted using the following code:

```
# Exclude non-positive cases
m3_yearly_af2 <- subset(m3_yearly_af, value > 0 & forecast > 0)
# calculate MAPEs and show the "accuracy vs horizon" graph
acc <- calculateMAPE(m3_yearly_af2)
acc$plot
```

As noted above, MAPE has many flaws. It is therefore desirable to use alternatives. However, choosing the most appropriate metric remains a controversial task [19]. Sometimes researchers and practitioners are facing difficulties as the use of different metrics leads to different rankings of methods and the results become difficult to interpret [3]. Formal statistical tests for accuracy comparisons are also not always straightforward to implement.

One important property of an error measure (formulated in [8, p. 240]) is that the criteria used for optimisation of predictions must correspond to the criteria used for

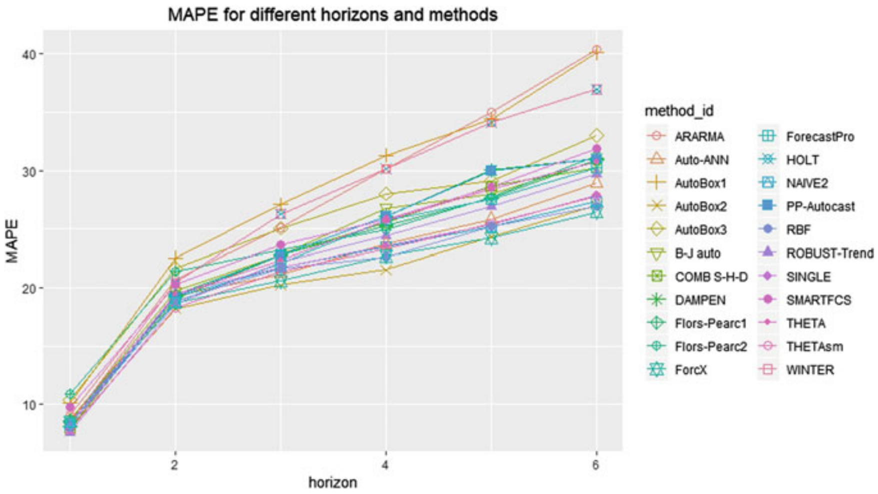


Fig. 5 Accuracy versus horizon graph

their evaluation. It is well-known [20, p. 30] that if a density forecast is available, point forecast in terms of MSE will correspond to its mean and optimal forecast in terms of MAE will correspond to its median. Usually, if density forecast is non-symmetric, original series are log-transformed, predicted, and then predictions are returned to the original scale. In this case we aim to obtain forecasts optimal in terms of MAE [8, p. 240]. If density forecast is symmetric, both MAE and MSE are suitable, but MAE is less affected by outliers.

It therefore makes sense to use out-of-sample MAE for accuracy evaluation. But this approach is applicable if we have only one time series. Importantly, metrics based on percentage errors (e.g., MAPE) are not suitable to represent accuracy in terms of linear or quadratic loss even for the case of only one time-series [19, pp. 49–53], [8, p. 241].

When it comes to measuring accuracy across series, scale-independent measures are required in order to avoid a so called "oranges versus apples comparison" [18]. Given that percentage errors are not advisable [8, 18, 19], one option is the mean scaled absolute error (MASE, [18]) where errors are scaled by MAE of the naive method. MASE, however, has some limitations [8, pp. 244–245]: it may not show relative accuracy reliably due to biases of the arithmetic mean and structural breaks in time series.

Instead, it was proposed in [3] to use the average relative mean absolute error, AvgRelMAE. This metric is based on the geometric mean of relative MAEs (RelMAEs). Suppose we have N time series, M methods, T origins from which each method produced forecasts (and corresponding outcomes are already known) with horizons from *1* to H. We denote forecast error as

$$e_{t+h,i,j} = Y_{t+h,i,j} - F_{t+h,i,j}, \tag{1}$$

where t—forecast origin ($t = 1 \ldots T$), h—forecast horizon ($h = 1 \ldots H$), i—method ($i = 1 \ldots M$), j—time series ($j = 1 \ldots N$), $Y_{t+h,i,j}$—actual, $F_{t+h,i,j}$—forecast.

MAE for a given combination of h, i, and j is

$$MAE_{h,i,j} = \frac{1}{T}\sum_{t=1}^{T}\left|e_{t+h,i,j}\right|. \tag{2}$$

To assess relative performances, we need to use a benchmark method (we recommend the naive method, but it can be any other method). Let B denote the index of the benchmark method. Then relative MAE (RelMAE) is:

$$RelMAE_{h,i,j} = MAE_{h,i,j}/MAE_{h,B,j}. \tag{3}$$

For a given series the RelMAE has the following interpretation: RelMAE < 1 means method i is better than method B in terms of the linear loss function, RelMAE > 1 meaning the opposite. In order to aggregate RelMAEs across series we can use various options such as the arithmetic mean or the median, but, as explained below, the geometric mean is a better option.

The AvgRelMAE for a specified method i and horizon h is found as the geometric mean of RelMAEs:

$$AvgRelMAE_{h,i} = \left(\prod_{j=1}^{N} RelMAE_{h,i,j} \right)^{1/N} \tag{4}$$

Different series lengths require the use of the weighted geometric mean, as proposed in [21]. The weight of RelMAE is then the number of origins used to calculate RelMAE.

The geometric mean has the following important advantages over alternatives (such as the median or the arithmetic mean) when averaging relative forecast performances: (i) it gives equal weight to reciprocal relative changes [21, p. 61] and (ii) the resulting rankings are invariant to the choice of the benchmark [21, p. 66]. It may occur, however, that MAE becomes zero for some cases and then the above formula cannot be used directly. In these cases a special trimming procedure should be used, as proposed in [19, p. 62].

Studies of the statistical properties of the AvgRelMAE have shown that it gives a better indication of relative accuracy (compared with alternative metrics) in terms of the linear loss [19]. One important advantage of this metric is the ease of interpretation. For example, obtaining an AvgRelMAE value of 0.95 for a particular method means that this method is likely to reduce the MAE of the benchmark by 5%.

Also (as noted in [22, p. 53]) the AvgRelMAE is applicable if one wants to implement the so-called ‘forecast value added’ (FVA) concept proposed by the SAS Institute [23]. The FVA is defined as ‘the change in a forecasting performance metric that can be attributed to a particular step or participant in the forecasting process.’ [23]. So we can say that obtaining AvgRelMAE = 0.95 corresponds to the FVA of 5%.

Given its advantages, we recommend the AvgRelMAE for accuracy visualisations. By replacing MAPE with AvgRelMAE on the graph shown on Fig. 5, obtaining the “AvgRelMAE versus horizon” graph is straightforward.

A corresponding “AvgRelMAE vs horizon” table is also required for reporting accuracy. To make table output effective, we recommend these rules: (i) use 2 or 3 digits after the decimal point and use bold font to show best results, (ii) indicate the benchmark method used or show ranks; (iv) (optional) use asterisk (*) to indicate statistical significance of changes in accuracy. Good examples of the “AvgRelMAE vs horizon” table can be found in [24, p. 253] and [25, p. 465].

When summarizing AvgRelMAE values across horizons, some researchers used MAEs containing errors for different horizons [25, p. 465]. We think this method is not desirable as accuracy becomes over-influenced by forecasts with higher horizons. We propose the following formula:

$$AvgRelMAE_i = \left(\prod_{h=1}^{H} AvgRelMAE_{h,i} \right)^{1/H} \tag{5}$$

By analogy to the AvgRelMAE we can define the AvgRelMSE, which indicates relative accuracy under quadratic loss [21, pp. 62–63]. Generally, if point forecasts were obtained as medians of forecast densities, it makes sense to use the AvgRelMAE [21]. And if forecast densities are non-symmetric and point forecasts were obtained as means of forecast densities, it makes sense to use the AvgRelMSE. Both the AvgRelMAE and AvgRelMSE converge to the geometric mean of relative absolute errors when T gets close to 1, which makes them biased indicators of relative accuracy [3, p. 518]. Our experiments, however, show that the AvgRelMSE is more biased compared to the AvgRelMAE, which makes the AvgRelMAE more preferable. Examples of using the AvgRelMAE and the AvgRelMSE metrics can be found in a number of recent studies (e.g., [24–27]).

The abbreviation prefix “AvgRel” to indicate averaging relative forecasting performances across time series using the geometric mean was proposed in [21] and since then this abbreviation has been used in many studies (e.g., [24–26]). Some authors used abbreviations ARMAE and ARMSE instead of the AvgRelMAE and AvgRelMSE (e.g., [27]). We do not recommend changing “AvgRel” to “ar” or “AR” in order to avoid confusion with some well-known measures for multitarget regression that are based on arithmetic mean and have the ‘ar’ prefix, such as arMAE. We propose that, if a compact notation for the “AvgRel” is needed, the “AvgRel” prefix can be replaced with the ‘Ø’ symbol. However, keeping the original “AvgRel” prefix makes measures more recognizable across studies.

When using the AvgRelMAE, one powerful tool to explore the underlying distribution is the boxplot of RelMAEs on a log scale (see Fig. 6).

This visual tool was proposed in [3, p. 520]. Examples of boxplots of log(RelMAE) can, e.g., be found in [19, p. 65] and [28, p. 18]. These plots help spot outliers and heavy tails to see if trimming is needed for a more reliable analysis. Besides, for skewed distributions additional precautions should be taken into account when applying statistical tests. For more details we refer our readers to [19].

In our package we implemented the following API for obtaining the “AvgRelMAE versus horizon” graph, table, and the log(RelMAE) boxplot:

```
# Prepare results for AvgRelMAE
acc <- calculateAvgRelMAE(m3_yearly_af2)
acc$plot     # show “AvgRelMAE vs horizon” plot
acc$accuracy # show “AvgRelMAE vs horizon” table
acc$boxplot  # show “RelMAE boxplot” (log scale)
```

The above listing demonstrates that the schemas proposed earlier allow the effective construction of graphical and tabular output for accuracy evaluation. Any subsets of the input dataset for the evaluation can be easily constructed by standard query functions (e.g., using the’subset()’ function in R).

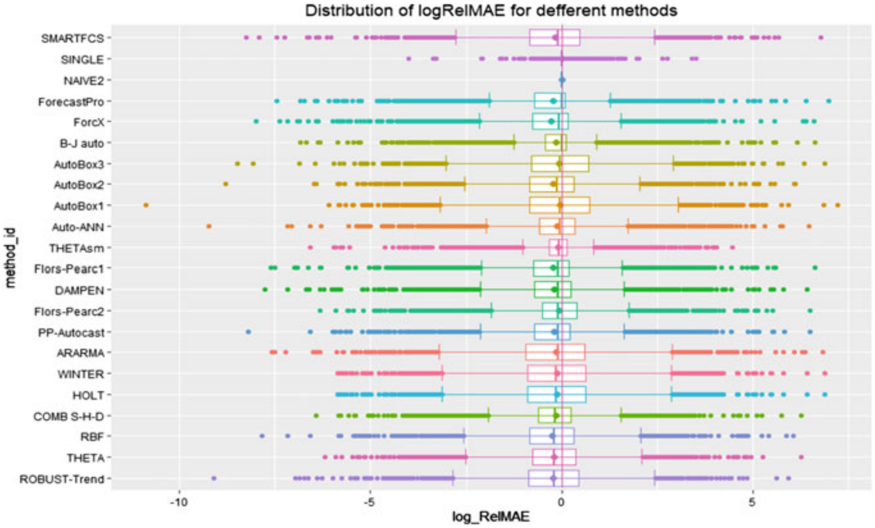


Fig. 6 Boxplots of log(RelMAEs) for M3 yearly data for different methods. Benchmark method: NAIVE2. Each boxplot shows cases for all time series and for all horizons available

7.2 *Assessing the Bias of Point Forecasts*

Apart from the accuracy of point forecast is often useful to see is forecasts systematically over-estimate or under-estimate actuals. The prediction-realization diagram described above is one tool to quickly explore the forecast bias. However, a more rigorous analysis requires the use of special metrics.

Usually, the term bias is used to see if the mean forecast error significantly differs from zero. However, as noted above, optimal forecasts for the linear loss correspond to the median of the forecast density. Thus, it does not make sense to evaluate forecast bias if densities are skewed and point forecasts were optimized for the linear loss. In these setting optimal forecasts will inevitably be biased [21] and reducing bias will reduce accuracy. Bias evaluation only makes sense if forecast densities are symmetric or forecasts were optimised for the quadratic loss (in this case we need to use the AvgRelMSE for accuracy evaluation).

For a single series the mean error (ME) is a good indicator of forecast bias. In order to obtain a scale-independent metric, some researchers use the mean percentage error (MPE) calculated by analogy to MAPE. One clear disadvantage of MPE is that it is vulnerable to outliers and has other limitations of MAPE. There are examples of using AvgRelMAE in combination with MPE [24, p. 253], but we do not recommend this approach due to the above reasons.

One alternative is to use the average relative absolute mean error (AvgRelAME), this metric was proposed in [21, p. 64]. Keeping the previous notation, the AvgRelAME is

$$AvgRelAME_{h,i} = \left(\prod_{j=1}^{N} RelAME_{h,i,j} \right)^{1/N}, \tag{6}$$

where

$$RelAME_{h,i,j} = AME_{h,i,j} / AME_{h,B,j} \tag{7}$$

and

$$AME_{h,i,j} = \left| \frac{1}{T} \sum_{t=1}^{T} e_{t+h,i,j} \right|. \tag{8}$$

When AvgRelAMEs for each method and each horizon are obtained, by analogy to the AvgRelMAE we can construct a “AvgRelAME versus horizon” graph and table. Averaging AvgRelAMEs across horizons is performed using the geometric mean, by analogy to formula (5).

7.3 Measuring the Quality of Prediction Intervals

Assessing the adequacy of PIs is based on calculating coverage probabilities and comparing them with the nominal probability. If confidence limits for the coverage probability are too wide, we can conclude there’s not enough observations to draw conclusions about the validity of PIs. Ideally, the confidence limits should be relatively narrow and include the nominal coverage. Surprisingly, very little research has been conducted in the area of validating PIs. Perhaps, most well-known attempt is [29], but it still did not provide evidence on confidence bounds for empirical coverage. Here we propose a visual tool showing both empirical coverage and corresponding error bounds.

Using the AFTS format we can implement the coverage chart shown on Fig. 7. The chart shows the coverage for ARIMA method for different forecasting horizons. The corresponding code is:

```
# show coverage chart for ARIMA forecast method
m3_quarterly_af <- createAFTS(m3_quarterly_ts, m3_quarterly_fc_pis)
plotCoverage(m3_quarterly_af, pi = 90, methods = "ARIMA")
```

By looking at the coverage chart on Fig. 7, it can be seen that the method (we used auto.arima function from the ‘forecast’ R-package to obtain PIs) tends to underestimate the uncertainty associated with the forecasts produced. Confidence limits for the actual coverage probability were obtained using the standard ‘binom.test’ function in R. If methods systematically underestimate or overestimate the uncertainty,

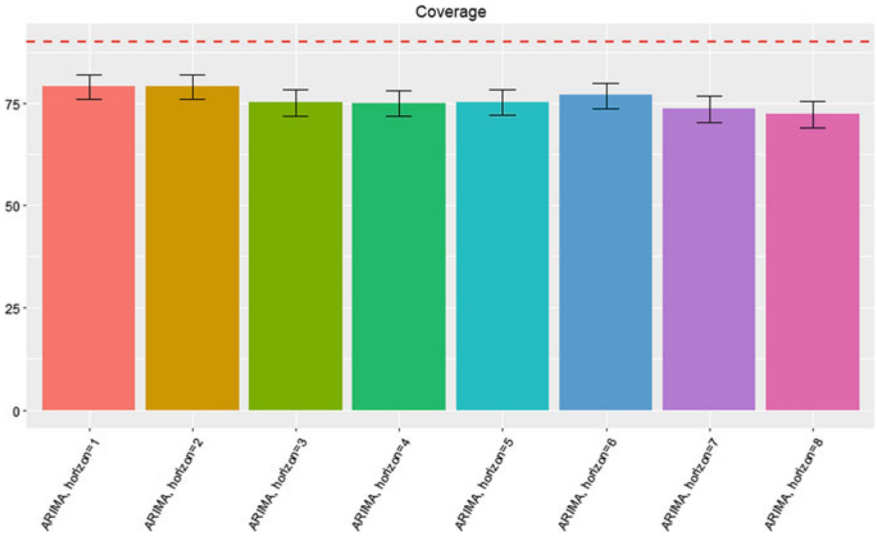


Fig. 7 Coverage chart. Error bars indicate 90% confidence intervals for coverage probabilities

this indicates the need to calibrate forecasts or apply data transformations prior to forecasting.

The coverage chart only assesses the adequacy of PIs. To compare the average width of PIs we propose the use the Average Relative Prediction Interval Width (AvgRelPIW) metric. For a specified confidence level, the AvgRelPIW is calculated by analogy to the AvgRelMAE by averaging relative PIWs across series using the geometric mean. If two methods have adequate PIs (this can be confirmed using the coverage chart) but different AvgRelPIW, the method with lower AvgRelPIW is more preferable. For example, if the task is to obtain adequate and accurate PIs for a real-time electricity consumption tracking system, we need to (1) ensure that PIs are adequate, (2) choose a model producing the lowest AvgRelPIW.

8 Recommendations Summary

In the light of the above discussion, we recommend the following algorithm for implementing the forecast evaluation workflow:

(1) Prepare the evaluation dataset: store forecasts and actuals using the FTS and TSTS formats, respectively.
(2) Before accuracy evaluation: produce summary of available cases, explore missing and NA values (both for forecasts and actuals), validate data integrity.
(3) Use the prediction-realization diagram (PRD) to identify potential data issues (use log scale if needed), explore the distribution of forecasts and actuals.

(4) If needed, use the fixed origin/fixed horizon graphs to explore time series and forecast features. Use fan charts to explore the adequacy of PIs.

(5) Perform forecast accuracy evaluation:

    (5.1) Choose the benchmark method (we recommend the naive method).

    (5.2) Use the log(RelMAE) boxplot to explore the distribution of RelMAEs.

    (5.3) If no evident data flaws can be found, use the AvgRelMAE metric (here we assume forecasts are optimised for the linear loss, use the AvgRelMSE or AvgRelRMSE for the quadratic loss).

    (5.4) Produce the "AvgRelMAE vs horizon" graph and table. Perform statistical tests to detect changes in accuracy (one possible test is described in [8, p. 62].

    (5.5) If needed, aggregate AvgRelMAEs across horizons.

(6) If forecast densities are symmetric (this can be assessed using the the PRD), measure forecast bias using the AvgRelAME metric. If forecast densities are skewed, but forecasts were optimised for the quadratic loss and the AvgRelMSE was used, we can still use the AvgRelAME metric. But if forecast densities are skewed and the AvgRelMAE was used to report accuracy, measuring bias will not be informative for forecast evaluation.

(7) Given a set of pre-defined confidence levels (say, 80, 90, and 95%) validate PIs using the coverage chart. Compare PIWs using the AvgRelPIW metric.

9 Conclusions

Having forecast data stored in a well-defined way is crucial for monitoring and evaluating forecast accuracy. In spite of the fact that a number of large-scale forecasting competitions have been conducted, at present there is no unified approach of how to store forecast data. In this chapter we proposed data schemas suitable for keeping forecast data in tables as a part of an RDB or as a portable file.

We also showed how to implement forecast evaluation based on the data structures proposed. We provided our examples in R, but, analogously, any other language (e.g., Python) can be used to implement the same approach.

The target audience for the techniques proposed involves both academics/researchers and practitioners. The framework can be applied in various scenarios. In particular, it can be used to create forecast-value-added (FVA) reports recommended by the SAS guidelines [23]. Also, separating forecast data from the evaluation algorithms and tools allows third parties to perform the forecast evaluation process, which is important to ensure objective evaluation. Finally, we showed that the framework complies with I4.0 principles [7]. We proposed RDBMS-oriented, well-defined and unified data structures, API, and visual tools ensuring the following capabilities of the framework: interoperability, decentralization, real-time operation, service-orientation, and modularity.

Acknowledgements The reported study was supported by RFBR research projects 19-47-340010 r_a.

References

1. Makridakis, S., Hibon, M.: The M3-Competition: results, conclusions and implications. Int. J. Forecast. **16**(4), 451–476 (2000)
2. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M4 competition: results, findings, conclusion and way forward. Int. J. Forecast. **34**(4), 802–808 (2018)
3. Davydenko, A., Fildes, R.: Measuring forecasting accuracy: the case of judgmental adjustments to SKU-level demand forecasts. Int. J. Forecast. **29**(3), 510–522 (2013)
4. Davydenko, A., Sai, C., & Shcherbakov, M. (2020, September 14–16). Data formats and visual tools for forecast evaluation in cyber-physical system design [conference presentation]. *International scientific multiconference "Cyber-physical systems design and modelling" CyberPhy-2020*, Kazan, Russia. https://doi.org/10.6084/m9.figshare.12981329.
5. Davydenko, A., Charith, K. (2020, July 29–30).: A Visual Framework for Longitudinal and Panel Studies (with Examples in R) [ePoster]. IRCUWU2020. https://doi.org/10.6084/m9.figshare.12749432
6. Davydenko, A., Fildes, R.: A joint Bayesian forecasting model of judgment and observed data (LUMS Working Paper 2012:4). Lancaster University, The Department of Management Science (2012). https://www.researchgate.net/publication/282136270_A_joint_Bayesian_forecasting_model_of_judgment_and_observed_data
7. Industry 4.0: Definition, Design Principles, Challenges, and the Future of Employment. https://www.cleverism.com/industry-4-0/. Accessed 14 Jun 2020.
8. Davydenko, A., Fildes, R.: Forecast error measures: critical review and practical recommendations. In: Business forecasting: practical problems and solutions. Wiley, Hoboken, NJ (2016)
9. Armstrong, J.S.: Principles of Forecasting: A Handbook for Researchers and Practitioners. Kluwer Academic, Boston, MA (2001)
10. Hyndman, R.: Cross-validation for time series [Blog post] (2016, December 5). Retrieved from https://robjhyndman.com/hyndsight/tscv/
11. Glossary: Density forecast. https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Density_forecast. Accessed 14 Jul 2020.
12. Hyndman, R.: Mcomp: Data from the M-Competitions. R package version 2.8 (2018). CRAN.R-project.org/package=Mcomp
13. Ellis, P: Tcomp: Data from the 2010 Tourism Forecasting Competition. R package version 1.0.1 (2018). CRAN.R-project.org/package=Tcomp
14. Hyndman, R.: tscompdata: Time series data from various forecasting competitions. R package version 0.0.1 (2018). github.com/robjhyndman/tscompdata
15. Montero-Manso, P., Netto, C., Talagala, T.: M4comp2018: Data from the M4-Competition. R package version 0.1.0 (2018). github.com/carlanetto/M4comp2018
16. Sai, C., Davydenko, A., Shcherbakov, M.: Forvision: Tools for forecast visualisation and evaluation. R package version 0.0.1 (2019). github.com/forvis/forvision
17. Theil, H.: Applied economic forecasting. North-Holland, Amsterdam (1996)
18. Hyndman, R., Koehler, A.: Another look at measures of forecast accuracy. Int. J. Forecast. **22**(4), 679–688 (2006)
19. Davydenko, A., Fildes, R.: Measuring forecasting accuracy: problems and recommendations (by the example of SKU-level judgmental adjustments). In: Intelligent Fashion Forecasting Systems: Models and Applications, pp. 43–70. Springer, Berlin Heidelberg (2014).

20. Zellner, A.: An Introduction to Bayesian Inference in Econometrics. Wiley Classics Library. Wiley, New York (1996)
21. Davydenko, A.: Integration of judgmental and statistical approaches for demand forecasting: Models and methods. PhD Thesis. Lancaster University, UK (2012). https://www.researchgate.net/publication/338885739_Integration_of_judgmental_and_statistical_approaches_for_demand_forecasting_Models_and_methods
22. Goodwin, P.: Profit from your Forecasting Software: A Best Practice Guide for Sales Forecasters. Wiley and SAS Business Series, Wiley, Hoboken, New Jersey (2018)
23. Gilliland, M.: Forecast value added analysis: Step-by-step. SAS Institute whitepaper (2008)
24. Ma, S., Fildes, R., Huang, T.: Demand forecasting with high dimensional data: The case of SKU retailsales forecasting with intra- and inter-category promotional information. Eur. J. Oper. Res. **249**(1), 245–257 (2016)
25. Huang, T., Fildes, R., Soopramanien, D.: Forecasting retailer product sales in the presence of structural change. Eur. J. Oper. Res. **279**(2), 459–470 (2019)
26. Fildes, R., & Goodwin, P. (2020). Stability in the inefficient use of forecasting systems: A case study in a supply chain company. *International Journal of Forecasting*. https://doi.org/10.1016/j.ijforecast.2020.11.004.
27. Spiliotis, E., Petropoulos, F., Kourentzes, N., & Assimakopoulos, V. (2020). Cross-temporal aggregation: Improving the forecast accuracy of hierarchical electricity consumption. *Applied Energy, 261*, 114339. https://doi.org/10.1016/j.apenergy.2019.114339.
28. Chen C., Twycross J., Garibaldi J.M.: A new accuracy measure based on bounded relative error for time series forecasting. PloS ONE 12(3), e0174202 (2017). https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174202
29. Athanasopoulos, G., Hyndman, R.J., Song, H., Wu, D.C.: The tourism forecasting competition. Int. J. Forecast. **27**(3), 822–844 (2011)