Forecasting for Data Scientists

A random walk through the random forest

Christoph Bergmeir

November, 2020

Monash University, Melbourne, Australia
https://www.cbergmeir.com

Introduction

My background

- 2008 Graduated in Computer Science, University of Ulm, Germany
- 2008-2013 Ph.D. in Computer Science and IT, University of Granada, Spain
  - Topic: Time Series Forecasting
  - Forecast Evaluation
  - Cross-validation for forecasting
  - Research stay at Dept of EBS, Monash University with RJ Hyndman
  - Forecasting for fault detection in wind turbines
  - Forecasting for pest outbreaks in green houses
  - Predictive maintenance

My background (2)

- 2013-2014 Postdoc, Univ. Granada: Predictive Maintenance for railways
- 2014-current Research Fellow, Lecturer, Senior Research Fellow at Faculty of IT, Monash University
  - Currently 3-year Research Fellowship to focus on time series forecasting
  - Renewable energy production forecasting (wind/solar)
  - Energy demand forecasting
  - Energy price forecasting
  - Retail sales/demand forecasting

What I forecast

- Sales forecasting in the supply chain, retail
- Forecasting in Energy
  - Demand
  - Prices
  - Renewable energy production
  - building efficiency
- Predictive maintenance (road, railway, ...mining)
- ...

What people think I forecast

When I go to a party, and I tell people that my job is forecasting, usually one of two things happens:

- ...like, weather forecasting?
- ...so, can you predict the stock market and we all get rich?

Weather forecasting

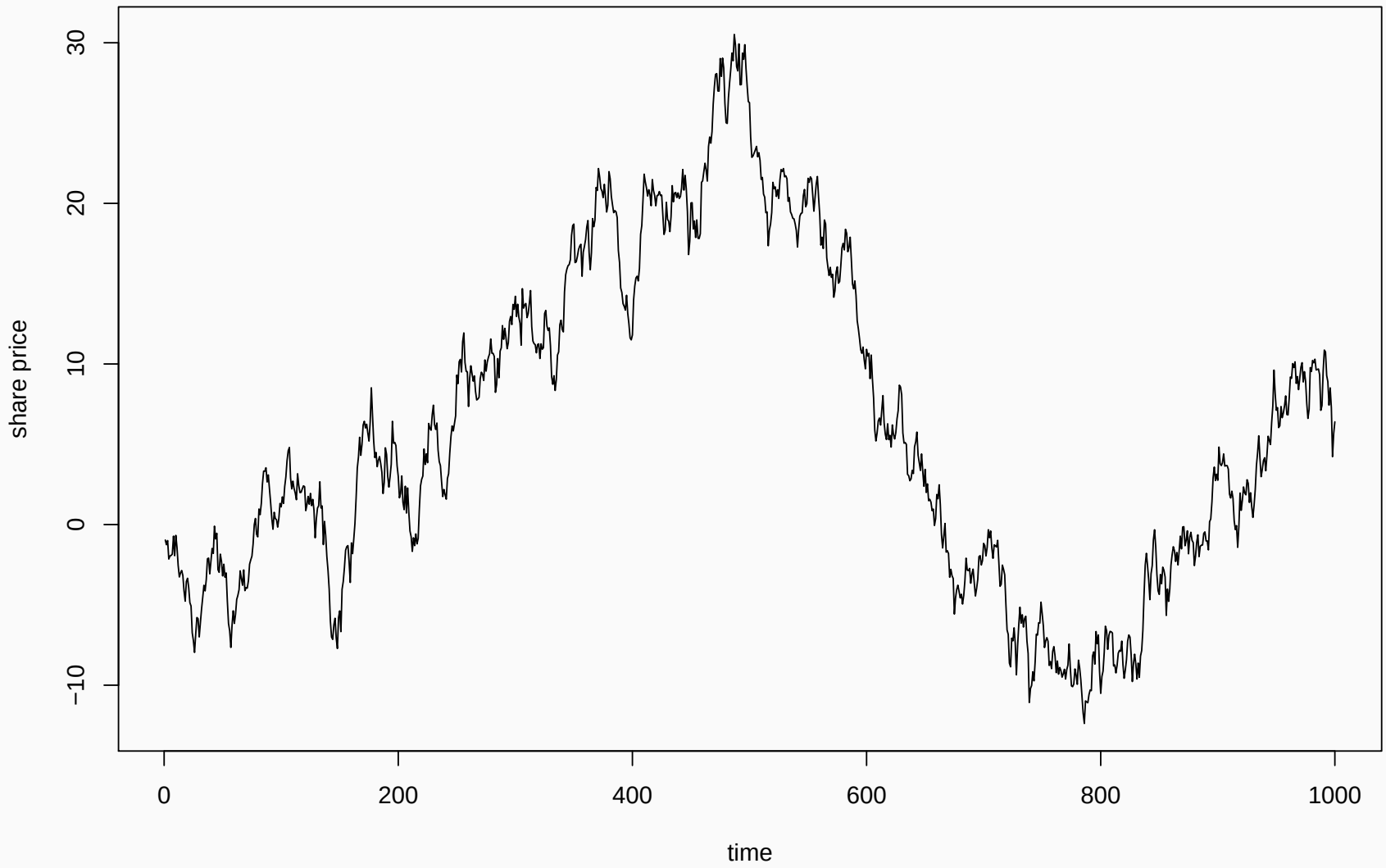
- Lots of domain knowledge and specialised models exist
- We leave it to meteorologists

Stock market forecasting

I'll tell you how, and we're all going to be rich!

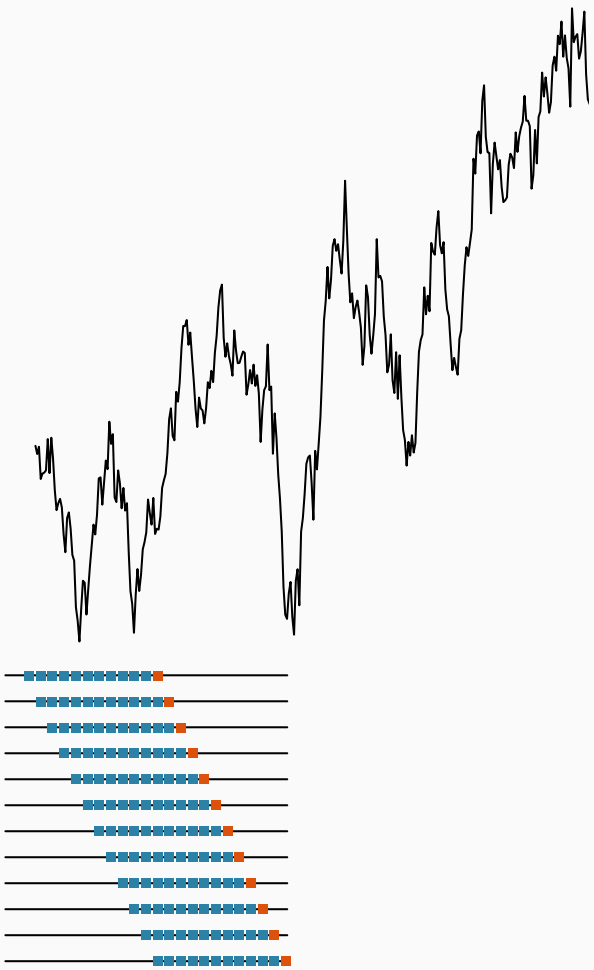
Source: https://www.flickr.com/photos/143999785@N05/36570061752

My favorite share

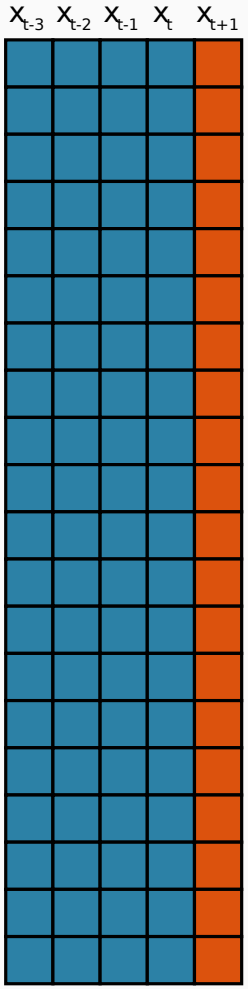


Autoregression (regression on itself)

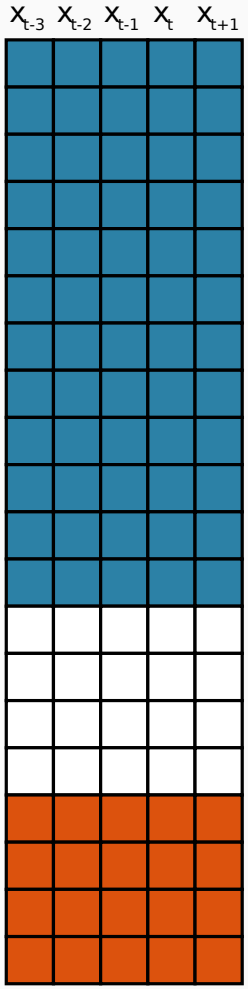
Sliding Window



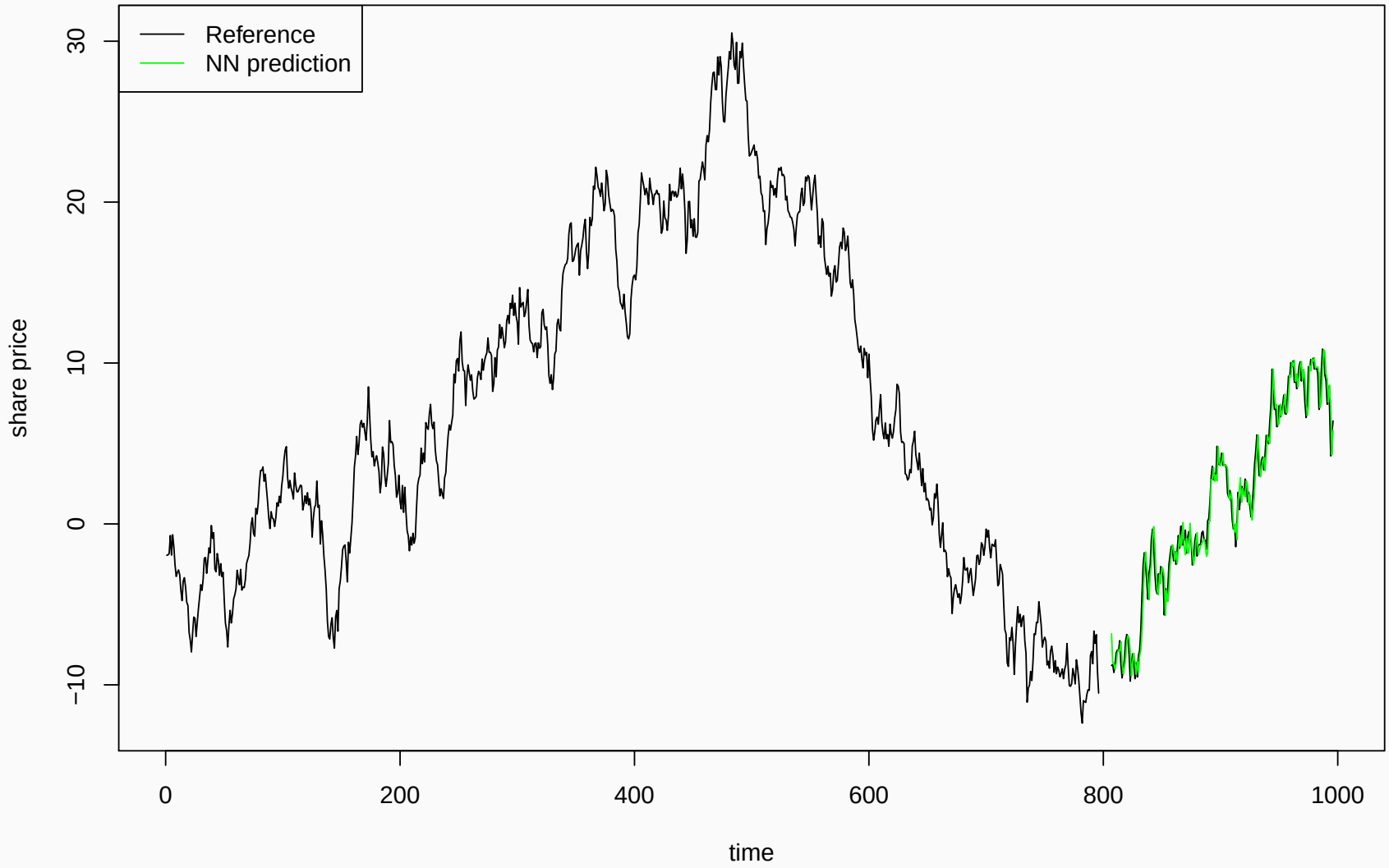
Embedded Time Series



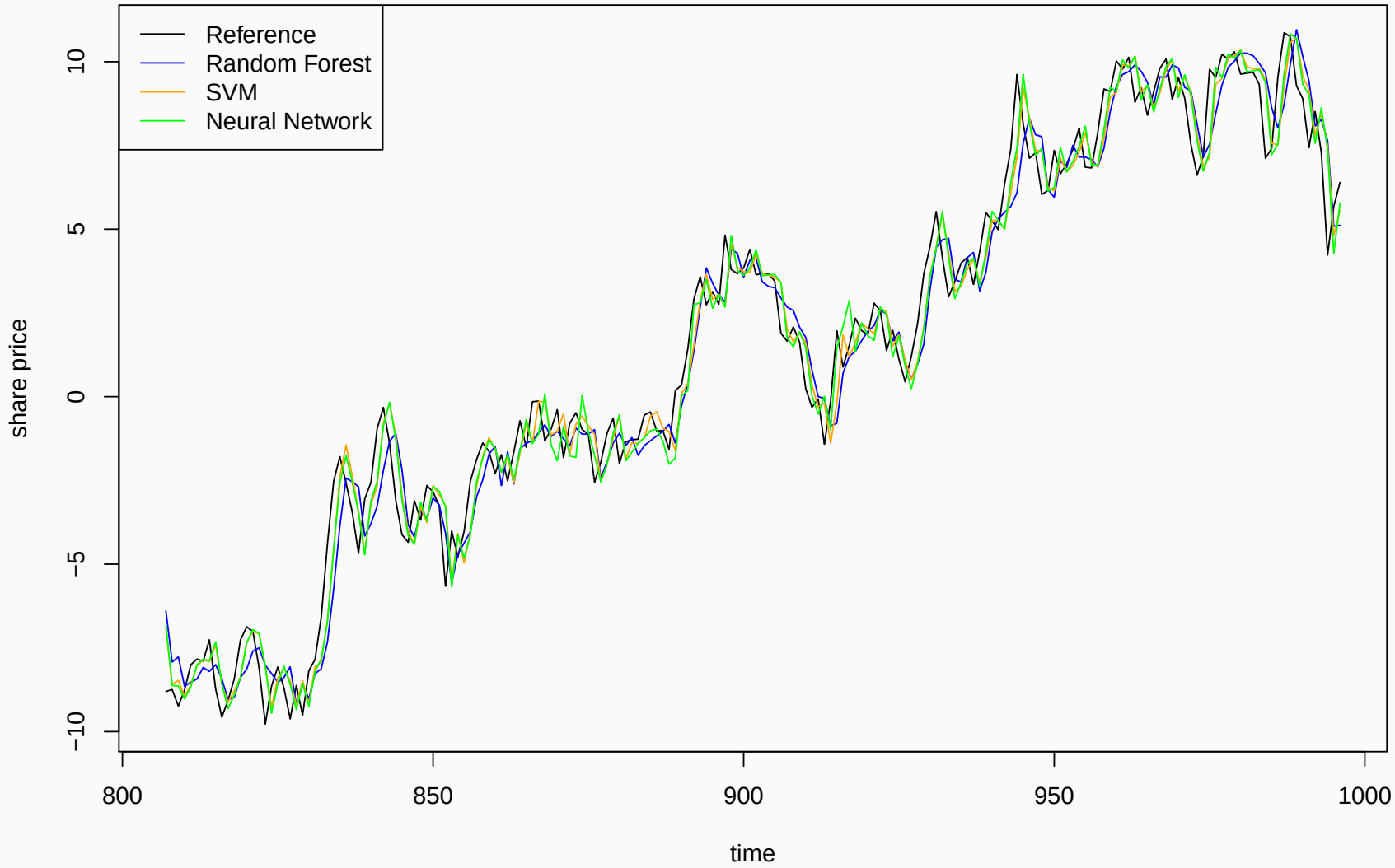
Training and test set



Build your favorite machine learning model



Build some more models



Calculate some errors

	SVM	RF	NN
MAE	0.816	0.883	0.829
RMSE	1.015	1.123	1.009

Done, write a paper...

C. Bergmeir et al., Novel robust SVM method for stock market prediction.

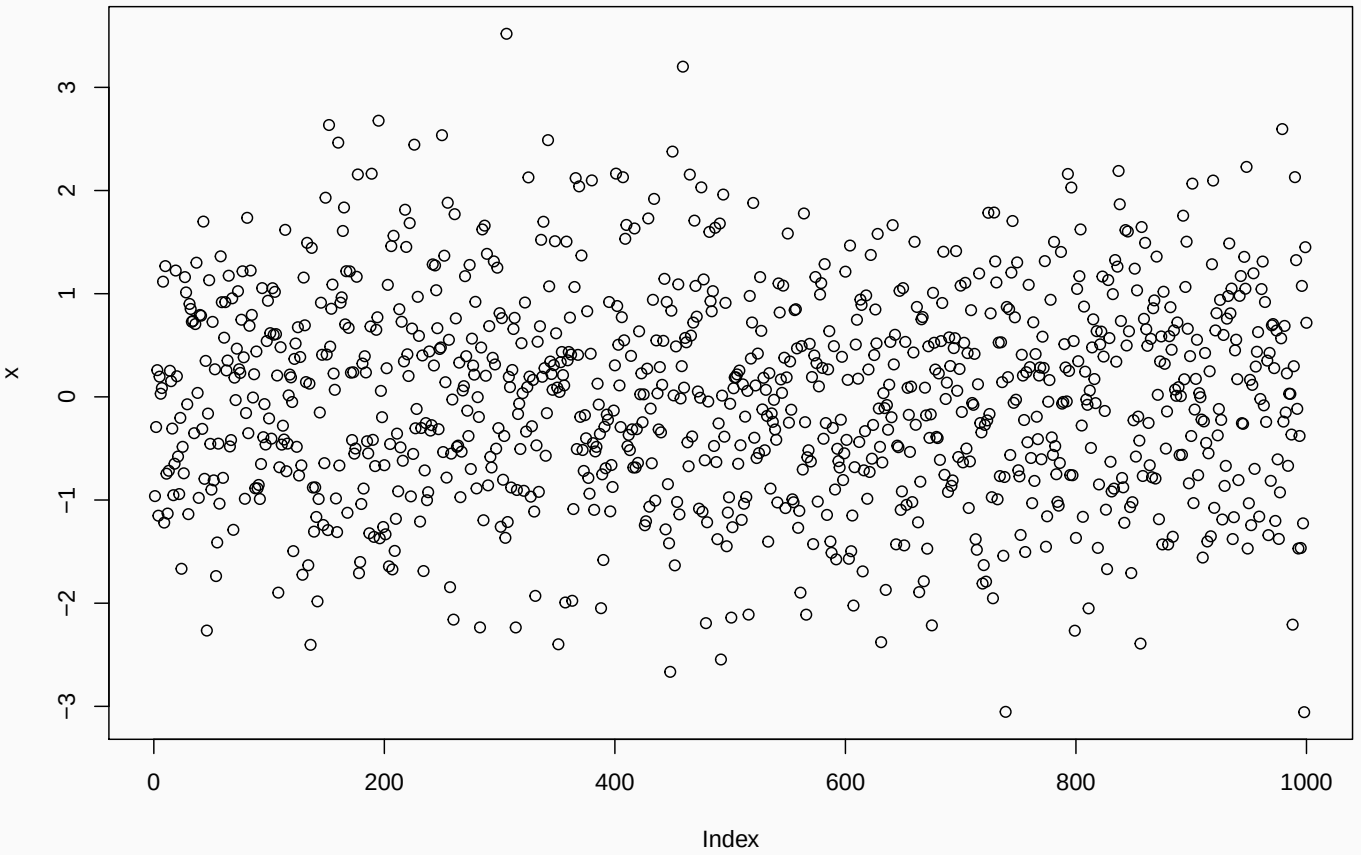
...but hold on

Regretfully:

- I reject papers like this one at least once a month
- And we are not all getting rich :(

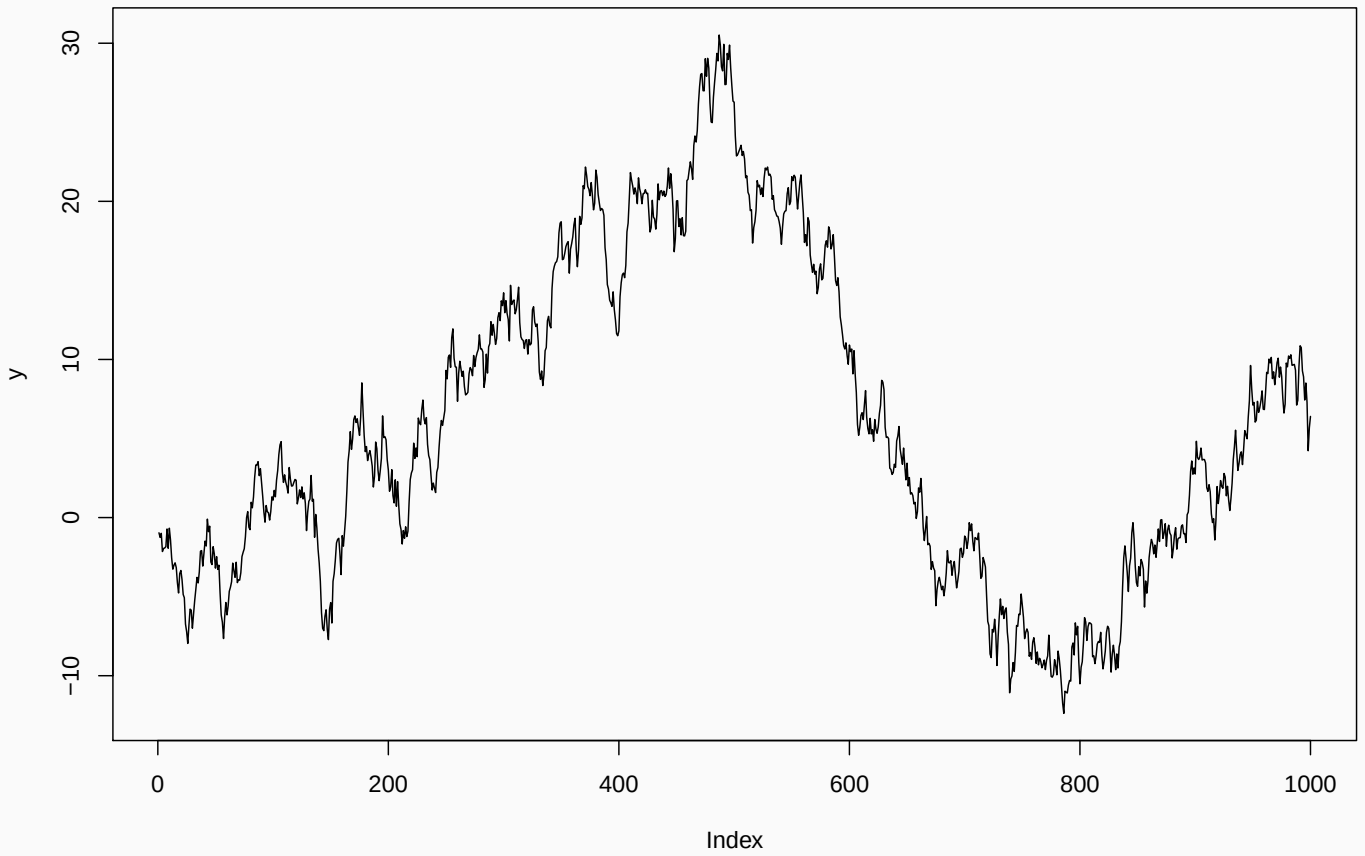
Some random numbers

```
set.seed(3)
x <- rnorm(1000)
plot(x)
```



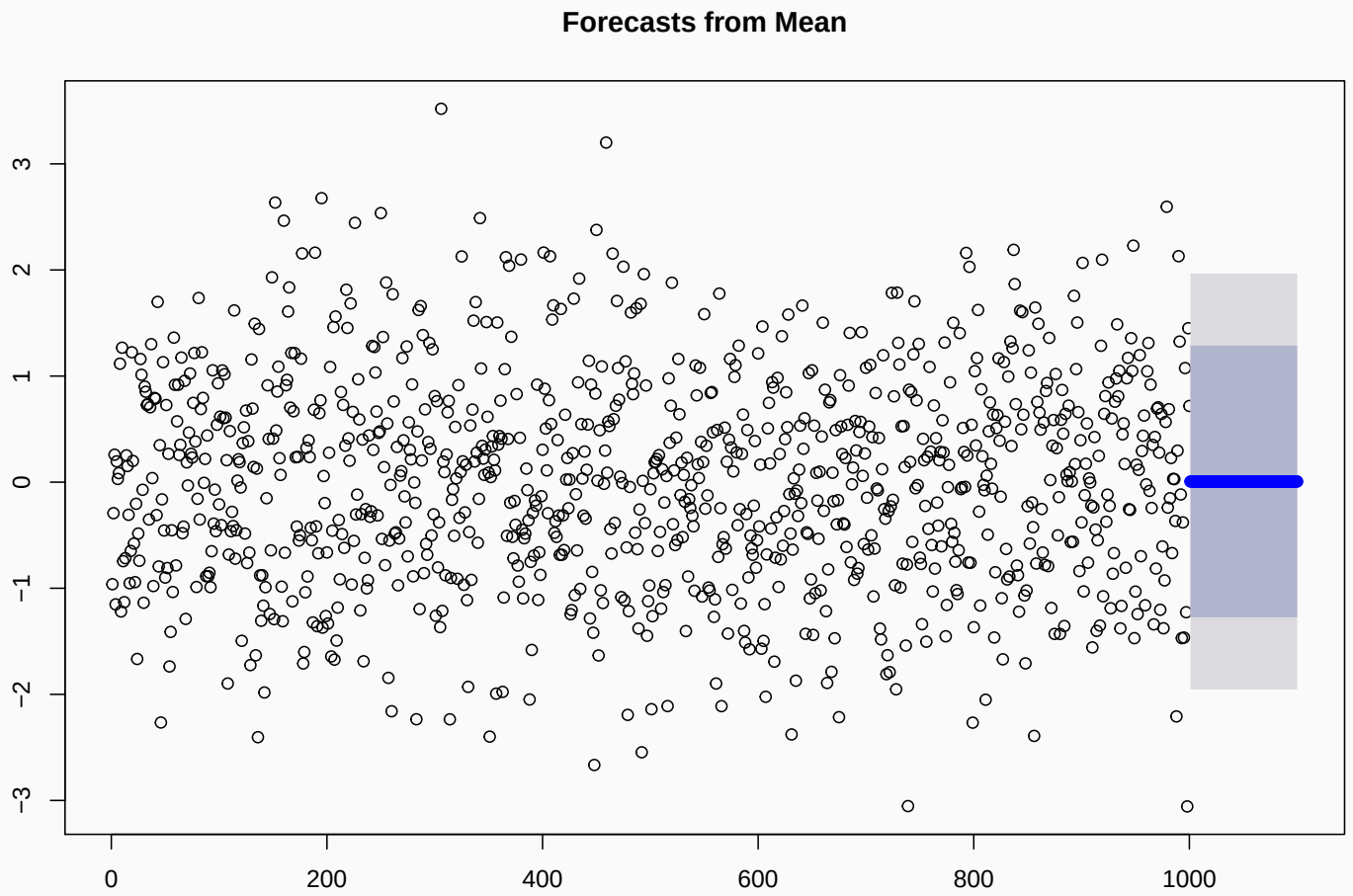
A random walk

```
y <- cumsum(x)
plot(y, type="l")
```



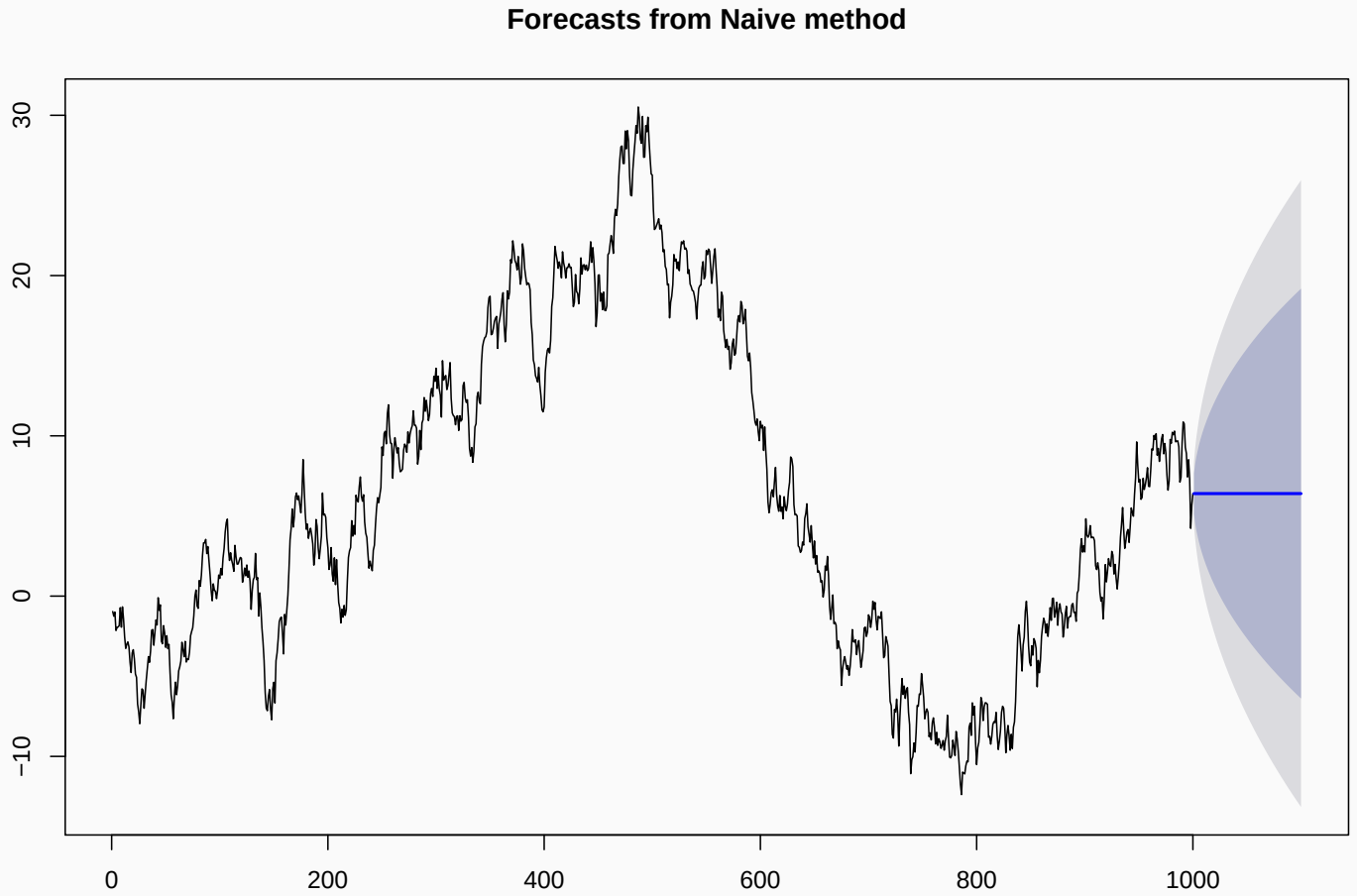
Mean forecast

```
plot(meanf(x, h=100), type="l")
```

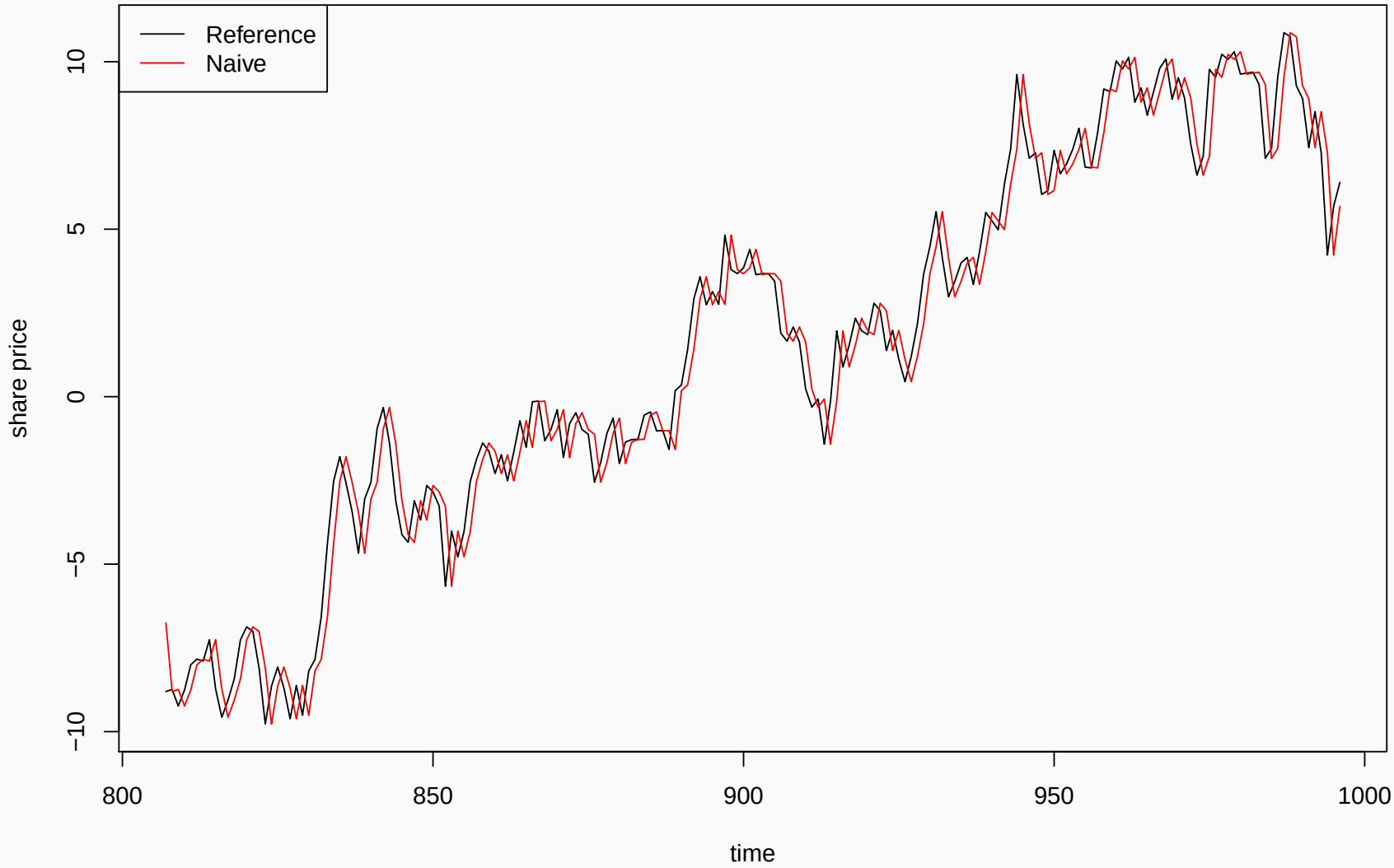


Naive (persistence/random walk) forecast

```
library(forecast)
plot(naive(y, h=100))
```



Naive forecast (2)



Naive forecast (3)

	SVM	RF	NN	Naive
MAE	0.816	0.883	0.829	0.827
RMSE	1.015	1.123	1.009	1.007

Stock market forecasting (pro)

Pros:

- Lots of freely available high-frequency data
- Clear motivation and use case (making money)

Stock market forecasting (contra)

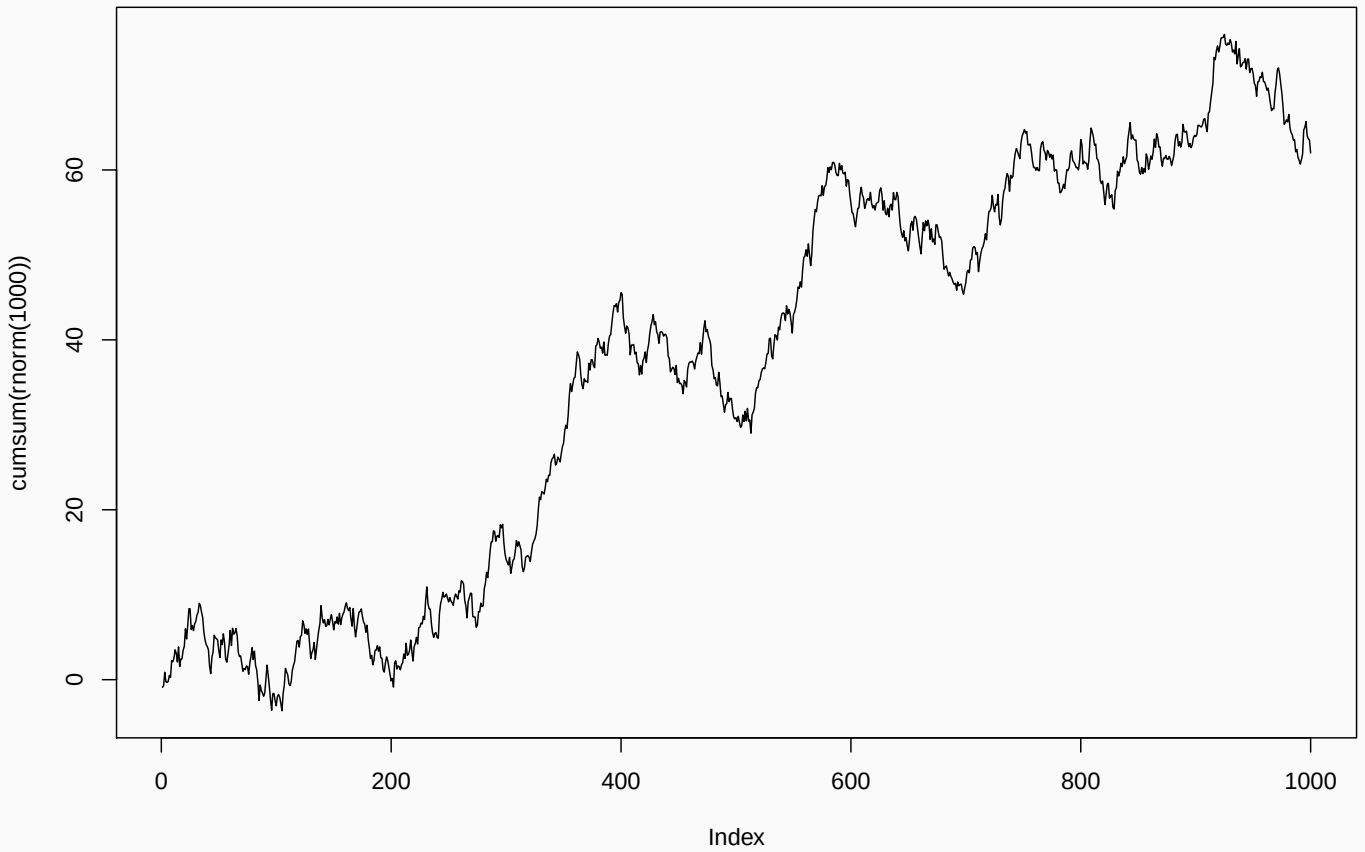
Cons:

- Shareprice not a function of its own past, but of its anticipated future
- Low signal to noise ratio
- Efficient market hypothesis makes your life difficult
- R. Engle: got the Nobel prize in Economics in 2003 "for methods of analyzing economic time series with time-varying volatility (ARCH)."
- All about predicting risk, not returns (i.e., predicting the tails of the distributions)

–> Predicting stock market returns from just a couple of lags won't work, no matter what method you are using.

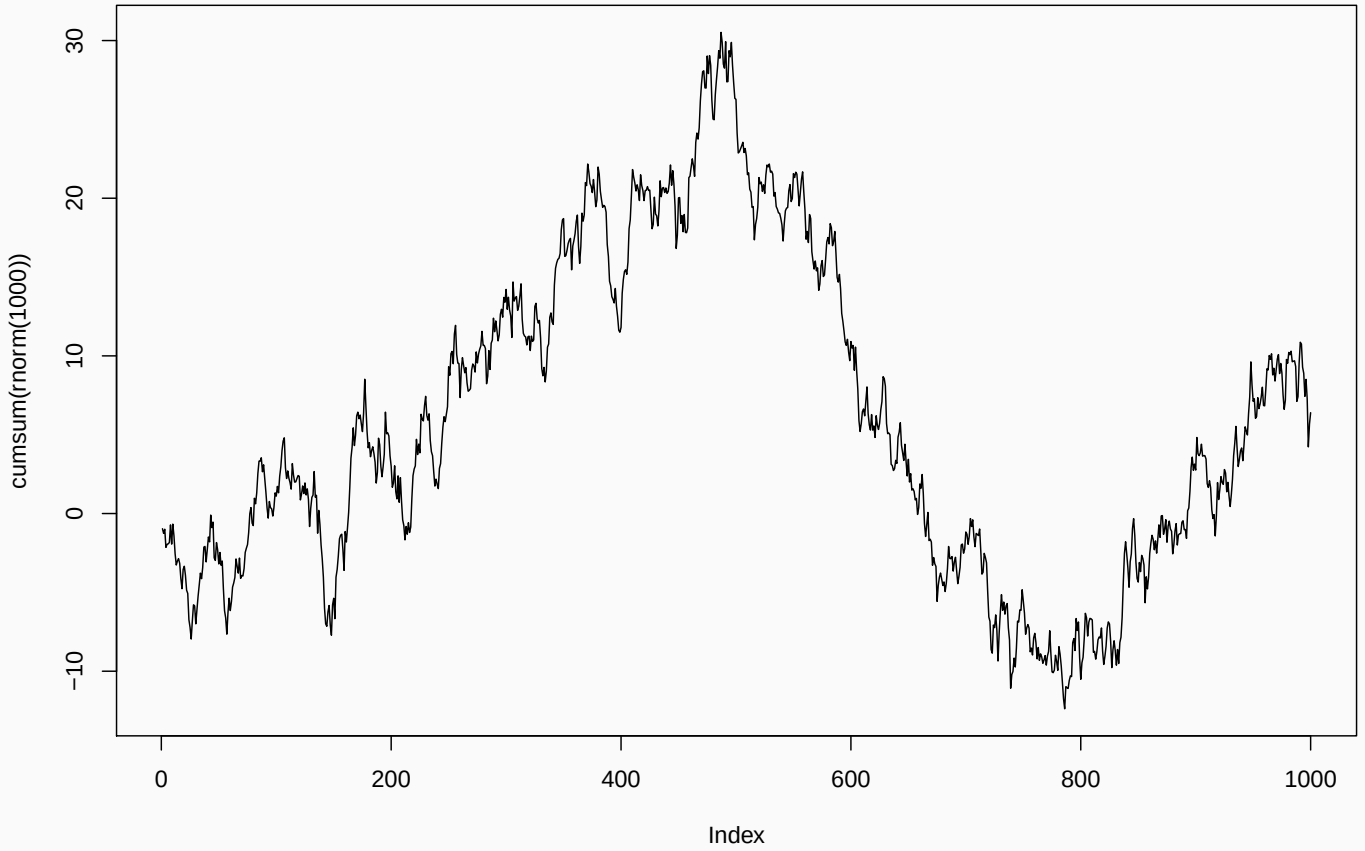
Random walk (3)

```
set.seed(2)
plot(cumsum(rnorm(1000)), type="l")
```



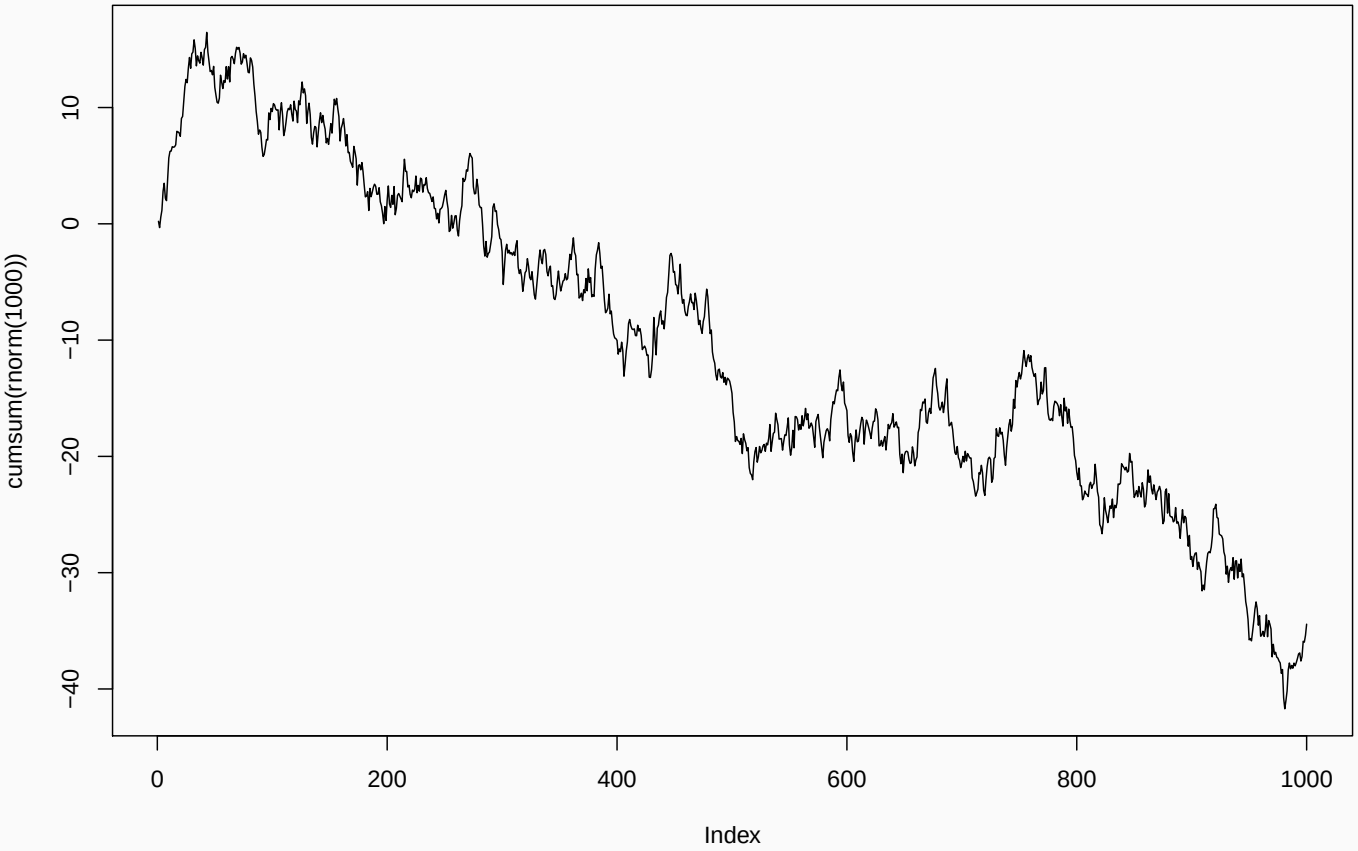
Random walk (4)

```
set.seed(3)
plot(cumsum(rnorm(1000)), type="l")
```



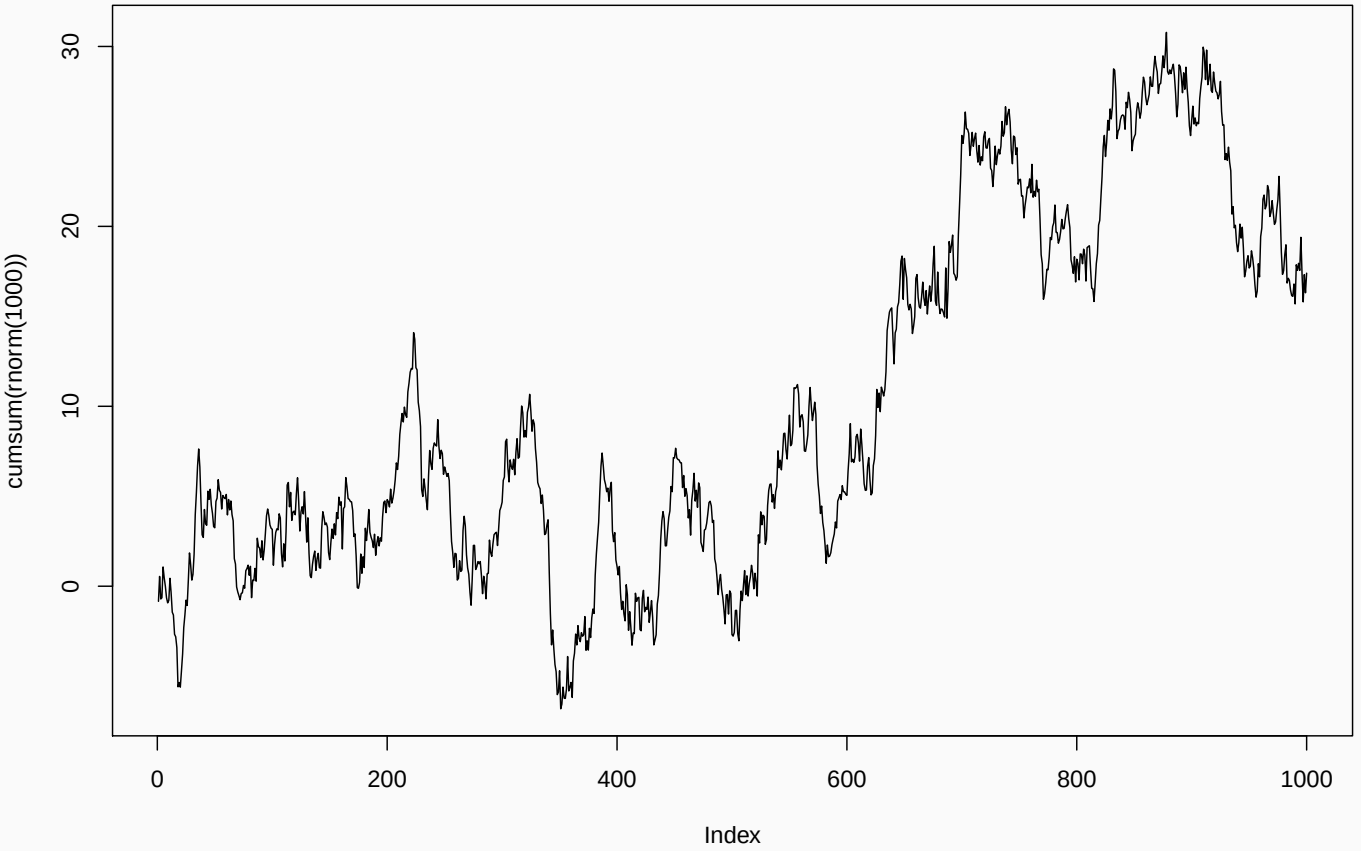
Random walk (5)

```
set.seed(4)
plot(cumsum(rnorm(1000)), type="l")
```



Random walk (6)

```
set.seed(5)
plot(cumsum(rnorm(1000)), type="l")
```



What can we forecast?

- Fundamental assumption: Future depends on the past
- We need past data/experience to extrapolate
- But predictability varies widely
- Some more examples:
  - next Monday's time of sunset
  - the TV schedule in 2 weeks' time
  - next Saturday's lottery numbers
  - financial variables: exchange rates, stock prices, etc.

What can we forecast? (2)

Important factors for predictability are:

- How well do we understand the underlying phenomena?
- How much data do we have?
- Will the forecast affect the future?

Forecasting with mathematical models:

- Model adequately repetitive patterns
- Do not let you distract by noise and unique, unforeseeable events

–> Separate the time series into a part with predictive value and a part of noise

What can we forecast? (3)

- Forecasts are always wrong
- Use prediction intervals to quantify uncertainty
- If you don't have to forecast, then don't forecast!
  - forecasting usually for decision making
  - can the decision be made without a forecast?
  - focus on resilience of the decision-making
  - Bad idea: "If I had a perfect forecast, I could make an optimal decision"
- Forecasting will give you the biggest wins in situations where you are already using a (bad) ad-hoc forecast in your decision making

Traditional (statistical univariate) forecasting techniques

A bit of forecasting terminology

- model fitting, parameter estimation: model training
- in-sample: training set
- out-of-sample: test set
- forecast horizon: target variable
- forecast origin: from where you do the forecasting from, the last known observation
- rolling origin: the forecast origin changes (for every point in the test set)
- fixed origin: the forecast origin is fixed

A bit of forecasting terminology (2)

- forecast combination: ensembling
- lags, independent variables, regressors, covariates, predictors: features, inputs
- dummy variable, indicator variable: one-hot encoding
- seasonality, seasonal period: cyclic change in mean of the series. Its length is know a priori and will not change in the future
- trend: (smooth) change in the mean of the series

Trading off the naive and the mean forecast

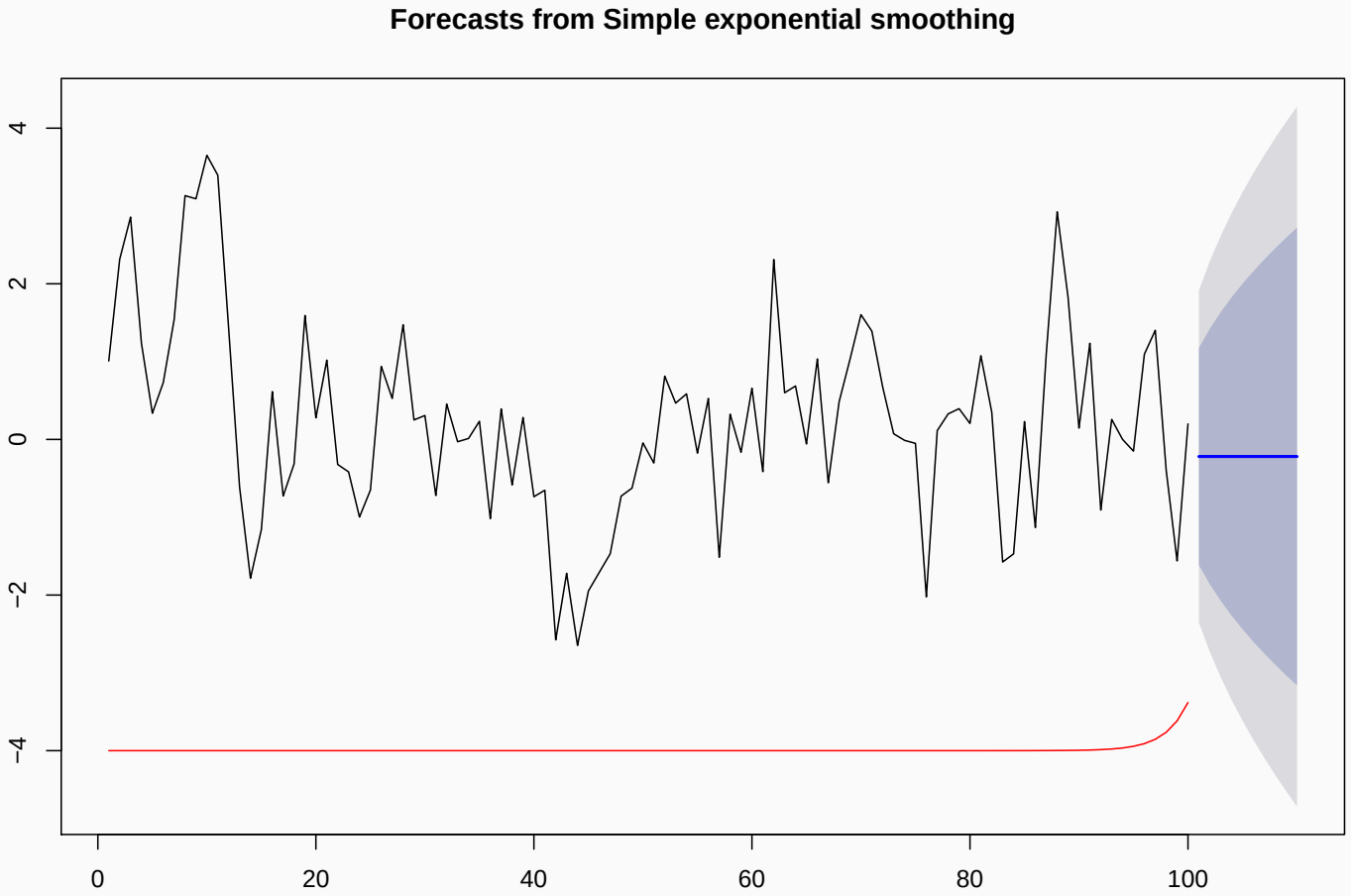
- Naive and mean forecast are two very extreme cases.
- Naive forecast: We only consider the last observation, none of the others
- Mean forecast: All observations are weighted equally

A central assumption in forecasting is often that the more recent past is more important than the more distant past.

Idea: What about weighting the observations with an exponential decay?

–> Exponential smoothing

Simple exponential smoothing (SES)



red line: exponentially decaying weights, $\alpha = 0.62$.

Exponential smoothing

- Exponential smoothing applied to components: level, seasonality, and trend
- Holt (1957), and his student Winters (1960); (see Goodwin, 2010, for an overview)
- Was used for a long time as a relatively ad-hoc method without a theoretical underpinning
- Hyndman et al. (2002) gave it a solid statistical foundation in state-space models
- ETS stands for both ExponenTial Smoothing and Error, Trend, and Seasonality

Box and Jenkins (1970): Linear modelling...

Autoregressive moving average model (ARMA):

$$\hat{x}_{t+1} = c + \phi_1 x_t + \cdots + \phi_p x_{t-p+1} + \theta_1 \epsilon_t + \cdots + \theta_q \epsilon_{t-q+1}$$

- Model is linear in the lags and linear in the errors
- When fitting the model, where do the errors come from?
- Need to estimate some initial conditions and step through the whole series (in ETS as well)
- No closed-form solution
- Need a non-linear fitting procedure. By default in R: BFGS
- fitting can be slow in long time series

ARIMA

Integrated:

- See the random walk from before, do we want to model the values directly or the change to the last value?
- addresses non-stationarity
- Preprocessing step: Do differencing of the series before we do the ARMA
- Pro: Hopefully makes the series stationary
- Contra: We loose some information about the scale in the preprocessing

ARIMA (cont'd)

- Any (stationary) AR(1) model has an equivalent MA(∞) model, and any (invertible) MA(1) model has an equivalent AR(∞) model (Hyndman and Athanasopoulos, 2018)
- Thus, we can approximate the MA part of the ARMA model with a higher order AR part
- In practice, this "higher order" often is not very high to get satisfactory results (in Econometrics, 5 is often a "high order" already)

–> Can be seen as a re-parametrisation, to get fewer parameters and a smaller input window, at the cost of more complex model fitting

ets() and auto.arima()

- standard implementations, from the "forecast" package in R (Hyndman and Khandakar, 2008)
- Wrappers for automatic model selection of ETS and ARIMA
- `ets()`: 15 models with different Error, Trend, and Seasonality
- `auto.arima()`: by default up to order 5 for p and q, and up to 2 differences
- choose the best one with AICc
- widely used standard benchmarks
- often very competitive

A brief history of forecasting competitions

The M competitions

I'm following here mostly Hyndman (2020b)

Makridakis and Hibon (1979). Journal of the Royal Statistical Society: Series A.

- They benchmarked a bunch of time series methods against each other on 111 time series.
- Simple methods (such as seasonal naive or SES) worked often better than ARIMA...which was considered a complex method

The M competitions (cont'd)

- Discussion published alongside the paper.
  "I find it hard to believe that Box-Jenkins, if properly applied, can actually be worse than so many of the simple methods." —Chris Chatfield

- Strong belief at that time that there is a true underlying data model that can be discovered
- And that this underlying model will necessarily give the best forecasts
- So, the discussants questioned the ability of Makridakis and Hibon to use methods like ARIMA properly.
- And they didn't agree with the finding that "forecast combination" works well (we would call this nowadays "ensembling").

M1 competition (1982)

- Results published in Makridakis et al. (1982)
- To answer to the criticism, Makridakis organised a competition where contestants could submit forecasts
- 1001 time series
- results largely confirmed the earlier results
- best method was a seasonal naive combined with SES
- forecast combination worked well
- after this, forecasters focused on OOS accuracy rather than model properties
- Forecasting emerged as an independent field from Time Series Analysis

M3 competition (1998)

- Results published in Makridakis and Hibon (2000)
- Had as a central conclusion (again) that complex methods do not necessarily outperform simple methods
- Again, ETS and ARIMA where the "complex" models, and simple models were models like a random walk with drift
- The only neural network that participated was quite bad
- Won by the "theta" method, which was later shown to be an average of a linear regression and simple exponential smoothing with drift
- **The** benchmark dataset in forecasting for almost 20 years
- A lot of focus (too much?) on this dataset

Controversy of ML vs Statistical methods

Controversy in the forecasting field, whether Machine Learning or Statistical methods work better for forecasting

- Forecasters "knew" that simple methods work best
  - ... because they had won the M3, the NN3, the NN5
- Machine Learners "knew" that Neural Networks work best
  - ... because, hey, it's a NN, it's a universal approximator
- Thousands of papers in Neural Network journals and others
- "A Novel method X for stock market forecasting..."
- Cherry-picked datasets, no proper benchmarking

Controversy of ML vs Statistical methods (cont'd)

S Makridakis, E Spiliotis, V Assimakopoulos (2018), Statistical and Machine Learning forecasting methods: Concerns and ways forward. PloS one 13 (3), e0194889.

- Makridakis et al. (2018b) benchmarked ML methods against statistical benchmarks
- not surprisingly, the ML methods lost
- The paper is published in PLOS ONE because it got rejected at 2-3 Neural Network outlets beforehand.

Controversy of ML vs Statistical methods (cont'd)

- Reasons for rejection were that: there are many ML methods that have "proven to overcome the results provided," though the editors and reviewers didn't name any in particular
- Makridakis was upset over this and did what he seemingly does best when he is upset...organise a competition: the M4
- so that Machine Learners could submit forecasts and show that their methods work well

Controversy of ML vs Statistical methods (cont'd)

- The history explains why the forecasting field got so hung up on this controversy
- Still somewhat surprising as Forecasting emerged from Statistics just in the same way Machine Learning emerged from Statistics

–> Focus on OOS accuracy, not model properties

–> Some things were discovered in parallel in both fields:

- forecast combination (Bates and Granger, 1969) vs ensembling (in ML in the 1990s)
- simple methods vs regularisation

–> Forecasting and Machine Learning should learn from each other!

M4 competition (2018)

- 100k series, across many different domains
- hourly, daily, weekly, monthly, quarterly, yearly series
- prediction intervals

Results

- published in Makridakis et al. (2018a)
- most forecasters expected that a combination approach of statistical methods would win
- however, such methods got 2nd and 3rd
- winner was RNN-ES, a hybrid between RNN and ETS, by Smyl (2020)
- a big surprise to many people that a ML model could win
- ... but not so much for me

Computational Intelligence in Forecasting (CIF) 2016 competition (Štěpnička and Burda, 2016)

- ... I got beaten in a competition by Slawek Smyl 2 years earlier
- 72 monthly series (lengths 22-108)
- I participated with plain ETS and BaggedETS
- the BaggedETS won several sub-categories and was also winning on median sMAPE
- On the competition metric mean sMAPE, BaggedETS got 9th, plain ETS got 3rd
- 1st and 2nd place were LSTMs, from Slawek Smyl.
- Globally trained across series

Global Models

Global models

Name "global models" was introduced by Januschowski et al. (2020)

- Traditionally, *one* time series is seen as a dataset
- One model is built per time series
- low sampling frequencies and non-stationarities like structural breaks make usually that we don't have enough data to fit complex (ML) models
- M3, M4 datasets are put together under this paradigm; very different series from different frequencies, different domains, etc.

Global models (cont'd)

Paradigm shift:

- a *set* of time series is a dataset (e.g., a set of series from retail, smart meters, etc.)
- build a model across the series
- names: global modelling, cross-learning, multi-task learning, pooled regression

–> Now, enough data, due to more series.

–> ML methods are competitive now

Global models (cont'd)

- Local model: typically fitting a model with few ($<$10) parameters to a single series
- if you have 10k series and fit 5 parameters, you end up with 50k parameters

$->$ fit a global model with 5k parameters instead

$->$ Overall complexity of set of local models grows when dataset grows; complexity of global model stays the same

Global models (cont'd)

- Global models can afford to be more complex
- Complexity can be added as:
  - longer memory (longer input windows, more lags)
  - non-linear/non-parametric models (NNs, GBT, ...)
  - data partitioning

Global models are not multivariate models

- Global models learn across series but predict every series in isolation
- they can work on datasets where series have different lengths and/or are not aligned, like the M3, M4 datasets
- they do not take into account interactions between series

History of global models

- Dating back to the early 2000s and earlier (Duncan et al., 2001)
- Pooled regression: Trapero et al. (2015)
- 2016: Smyl and Kuber (2016), CIF competition: Štěpnička and Burda (2016)
- 2017: DeepAR (Salinas et al., 2019b) and other works from Amazon, e.g., Wen et al. (2017); our work in Bandara et al. (2017)
- after 2018: ... many more works

Kaggle competitions

- A good overview give Bojer and Meldgaard (2020)
- The following are relevant forecasting competitions held on Kaggle:
  - Walmart Store Sales Forecasting (2014)
  - Walmart Sales in Stormy Weather (2015)
  - Rossmann Store Sales (2015)
  - Wikipedia Web Traffic Forecasting (2017)
  - Corporación Favorita Grocery Sales Forecasting (2018)
  - Recruit Restaurant Visitor Forecasting (2018)
- All competitions were won by global models, the latter four by either GBT or NN models or ensembles of those.

M5 competition

- held in 2020 on Kaggle
- dominated by LightGBM: Makridakis et al. (2020), DeepAR and NBEATS also successful
- A team of my students won a Kaggle Gold, 17th in the competition out of >5k participants.
- They used an ensemble of LightGBM and pooled (linear) regression.

Global models (cont'd)

- Global models have shown success to a surprising / unreasonable degree
- Idea in recent years was that the series have to be in some way "related/similar" so that we can learn something useful across them
- "Related" in terms of similarity of their DGP (not necessarily mere correlations)

Global models (cont'd)

Montero-Manso and Hyndman (2020):

- Global model can produce the same forecasts as local models, without any assumptions about similarity

–> The series don't have to be "related"

- Instead of fitting one complex pattern across series, a global model even works well to fit many simple patterns that are different in the series
- This result is quite remarkable, similar results, e.g., in multi-task learning, don't exist

Machine Learning methods for forecasting

Machine Learning methods for forecasting

Nowadays often a good option

- global modelling across series
- can apply them to a single time series, if long enough
- longer series due to finer granularities (secondly, minutely, half-hourly series available over years)
- additional metadata

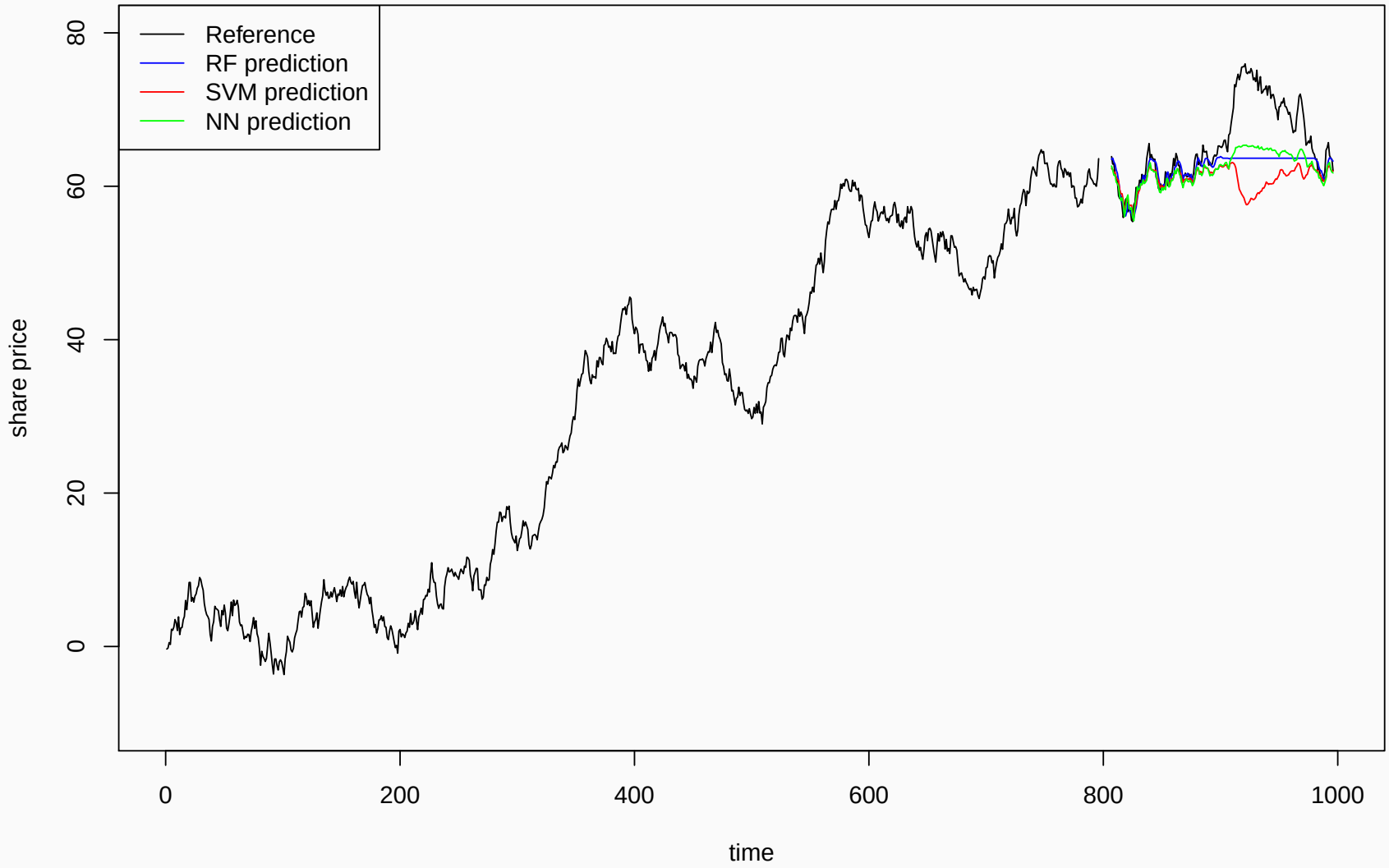
Non-linear autoregression

- basic setup is a (non-linear, non-parametric) autoregressive model
- then, you can use your favourite ML method out of the box
- we've seen that this can approximate an ARMA model, if we choose the input window larger than the ARMA model has it

Problem: Non-stationarity

- Data Distribution changes over time
- Many (most?) real-world problems have a time component and changing distributions
- Think about detecting cars on the street with a dataset from the 1970s
- In time series it is more explicit though and has more impact

Problem: Non-stationarity (2)



Problem: Non-stationarity (3)

> *Dividing the world into linear and non-linear is like dividing the world into bananas and non-bananas.*

Just as there are many different forms of non-linearity, there are also many different forms of non-stationarity

Typical non-stationarities in time series:

- change in mean: seasonality, trend
- change in variance: heteroskedasticity
- Random walks (unit roots)

How to achieve stationarity?

- As in Econometrics and Finance, many series are close to random walks, there, differencing is a common tool to achieve stationarity
- see ARIMA modelling earlier
- Differencing only solves some forms of non-stationarity, not others
- Loosing information about the scale. Can have an additional input as the (log of) the original scale.
- Differencing can help to make ML models more robust

How to model trend?

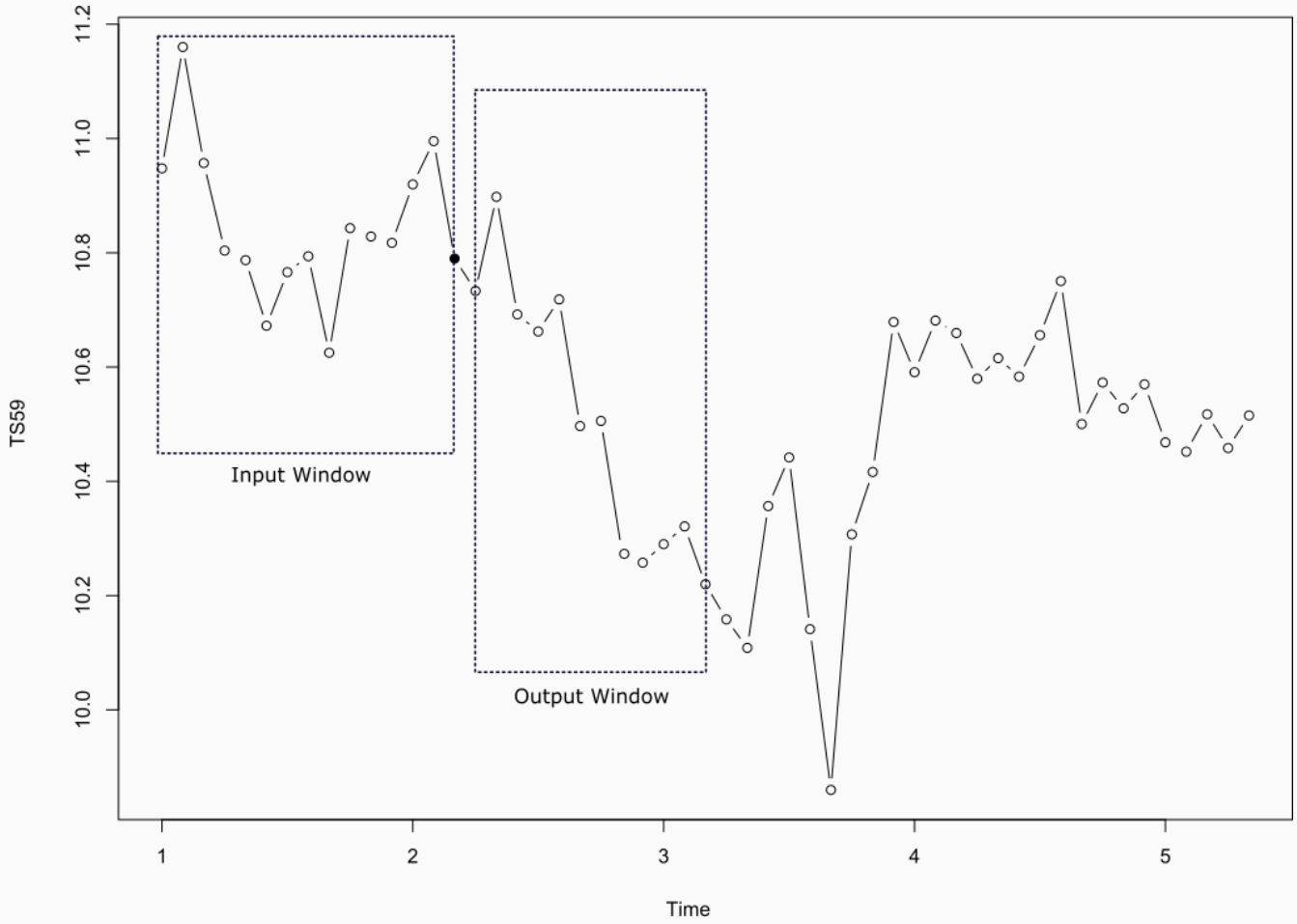
- Detrending
  - Problem: it is not well specified what a trend is
  - Essentially just a smoothed version of the series
  - We still need to forecast the trend then
- Logarithm or Box-Cox transform
  - makes exponential trends linear
  - also stabilises the variance
  - Box-Cox Transformation

$$w_t = \begin{cases} log(y_t) & \text{if } \lambda = 0, \\ (y_t^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \end{cases}$$

  - Choice of optimal value for λ is difficult

Window-wise normalisation

Smyl and Kuber (2016);Bandara et al. (2017)

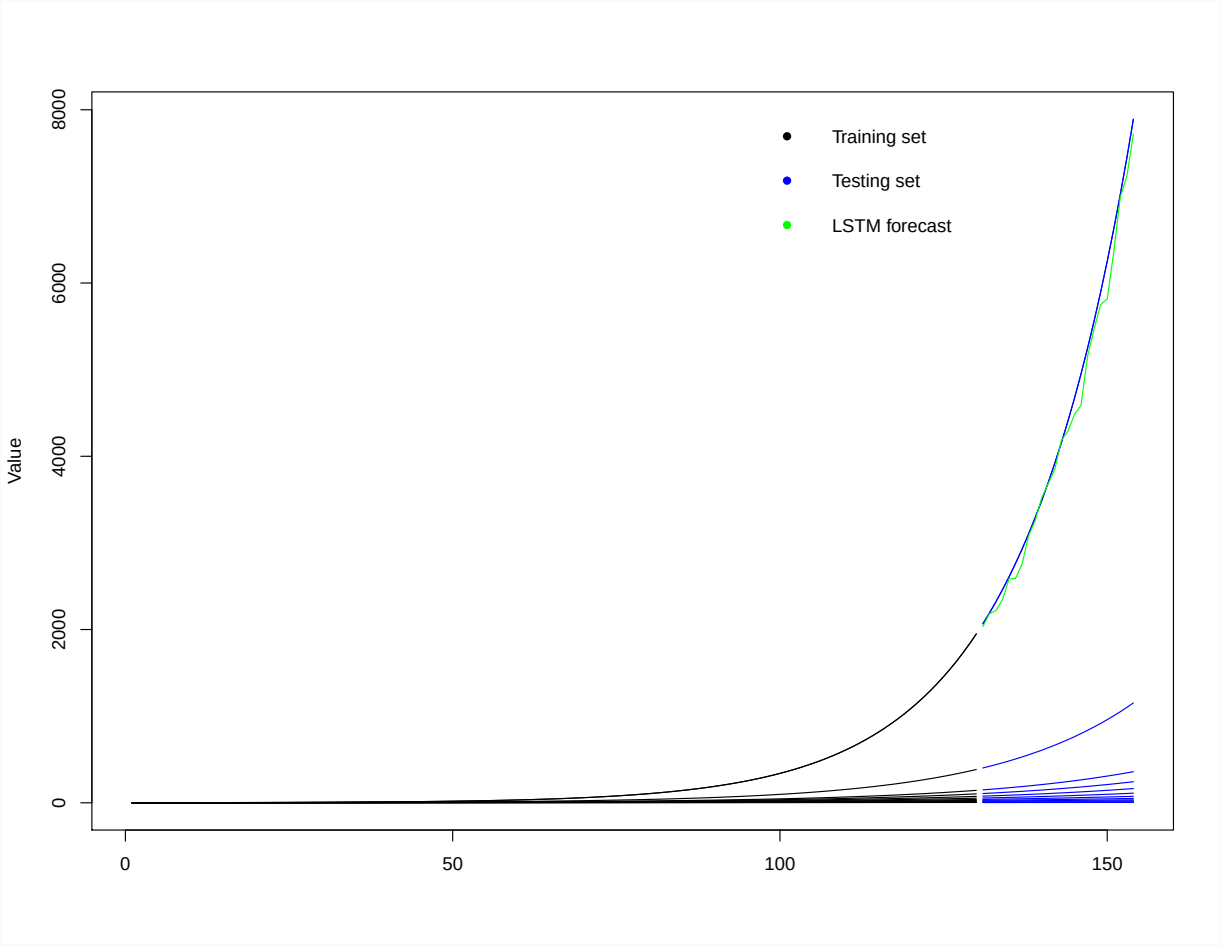


Window-wise normalisation (cont'd)

- To avoid saturation issues of sigmoid, tanh activation functions
- Similar to batch normalization
- Instead of saturating on the absolute values of the training data, it is saturating on the absolute value of the steepness of the trend in the training data
- It is usually a good idea for forecasts to be conservative.

Window-wise normalisation (cont'd)

Such a system is able to predict even steep trends (Bandara et al., 2020b)



Expert knowledge about trends

- be careful with strong trends
- exponential trends will slow eventually
  - how much more can Facebook grow until every person on earth has 5 accounts each?
- even a linear trend is a very bold assumption oftentimes
- damped trends: often not justified from the data, just to be conservative about the forecasting
- forecasts should always be conservative

How to model seasonality?

- some discussion in the literature whether a NN can model seasonality directly or not
- Early works suggest that NNs can model seasonality (Sharda and Patil, 1992; Tang et al., 1991).
- Later, works suggest that deseasonalization is necessary (Claveria and Torra, 2014; Zhang and Qi, 2005; Zhang and Kline, 2007; Nelson et al., 1999).
- Latest findings: Machine Learning models can model seasonality well *if they have enough data* (Bandara et al., 2020a)

–> Assumption in forecasting is usually that we know the seasonality beforehand and that it is valid to extrapolate it infinitly into the future.

Modelling seasonality

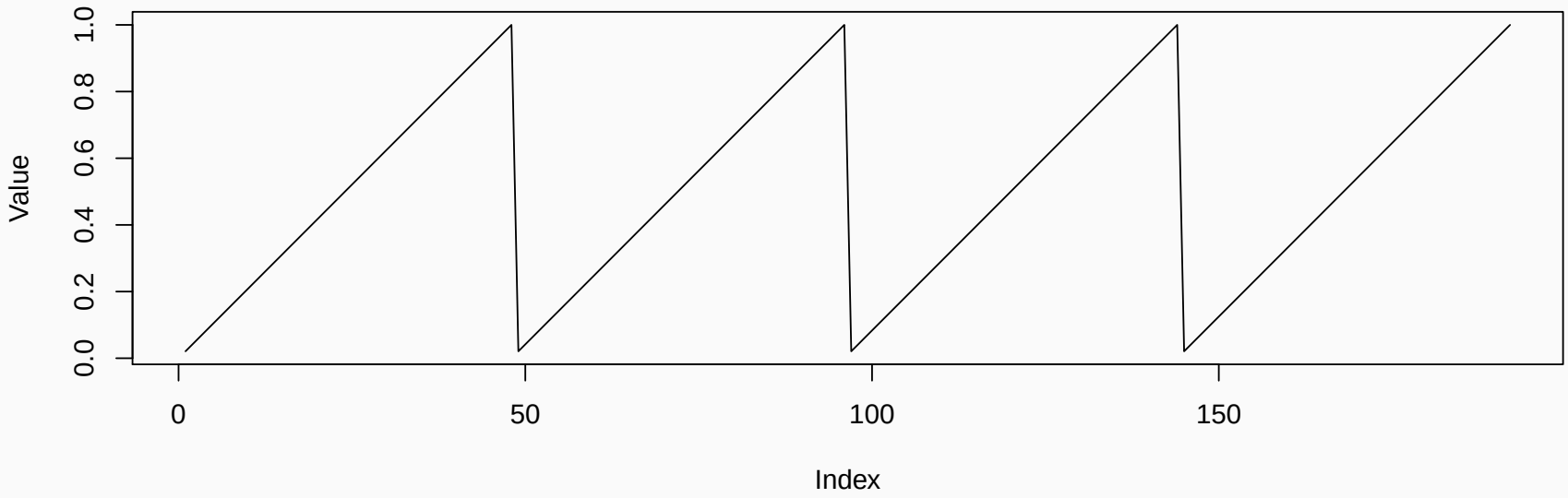
Seasonal indicator variables:

- Categorical variable: Monday, Tuesday, Wednesday, ...
- One-hot encoded version of this variable: "Seasonal dummy"

Problems if there are many seasons.

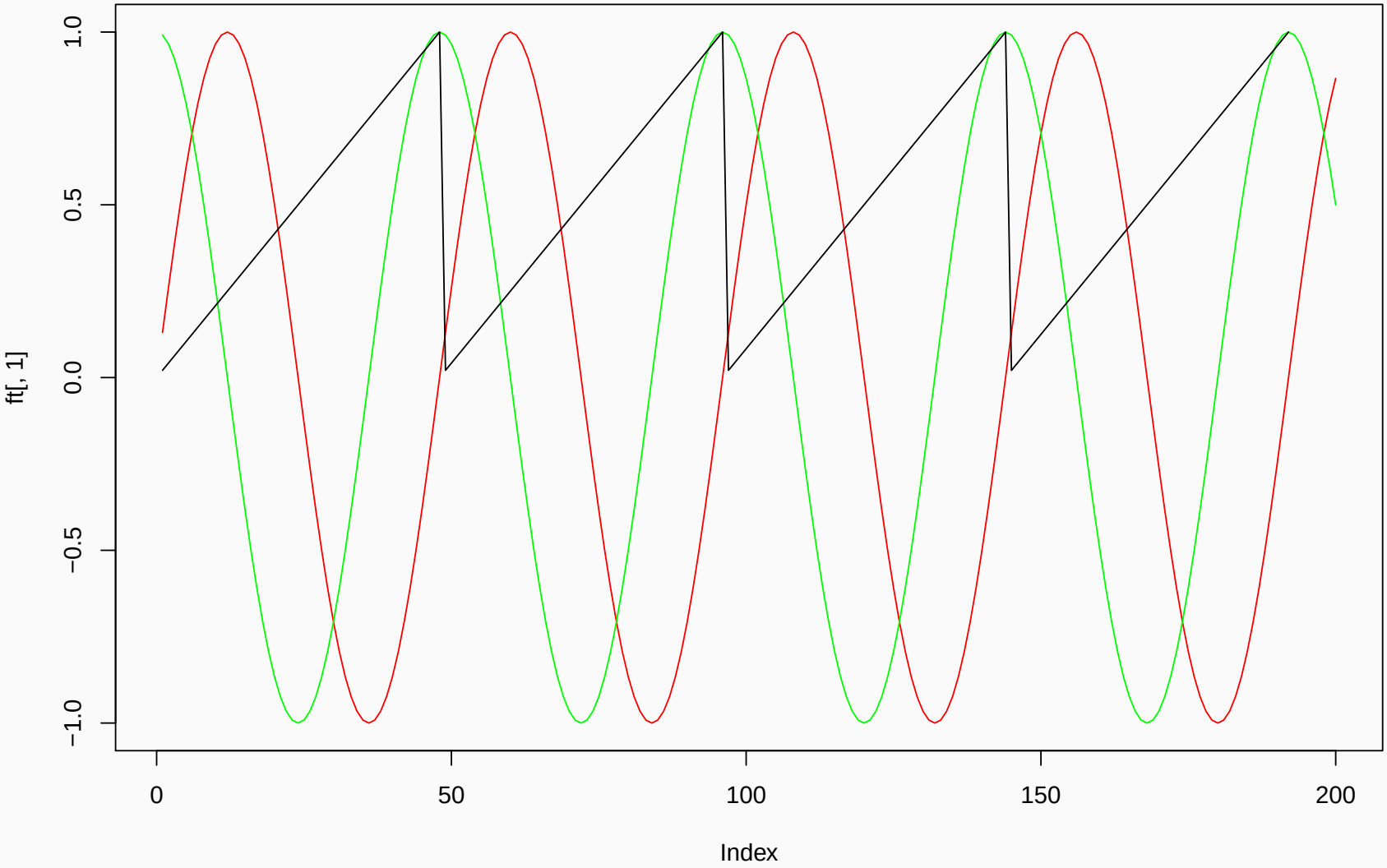
Modelling seasonality (2)

Continuous seasonal indicators



Problem: Abrupt changes. December much more distant from January than, e.g., January from February.

Modelling seasonality (3)



Fourier terms

see, e.g., Hyndman and Athanasopoulos (2018)

$$sin\left(\frac{2\pi kt}{s}\right), cos\left(\frac{2\pi kt}{s}\right)$$

t is the time point

s is the seasonal periodicity of the time series and

k is the number of sine cosine pairs used with the transformation

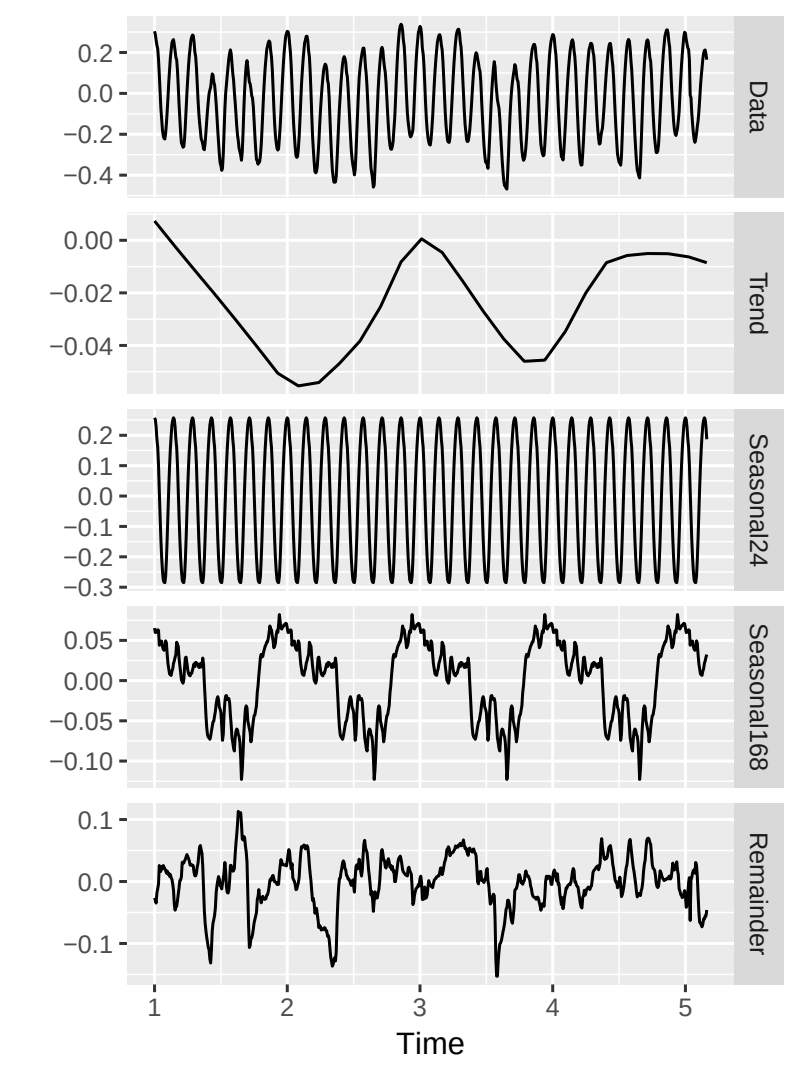
The number of Fourier terms controls the smoothness of the seasonal pattern.

Deseasonalisation

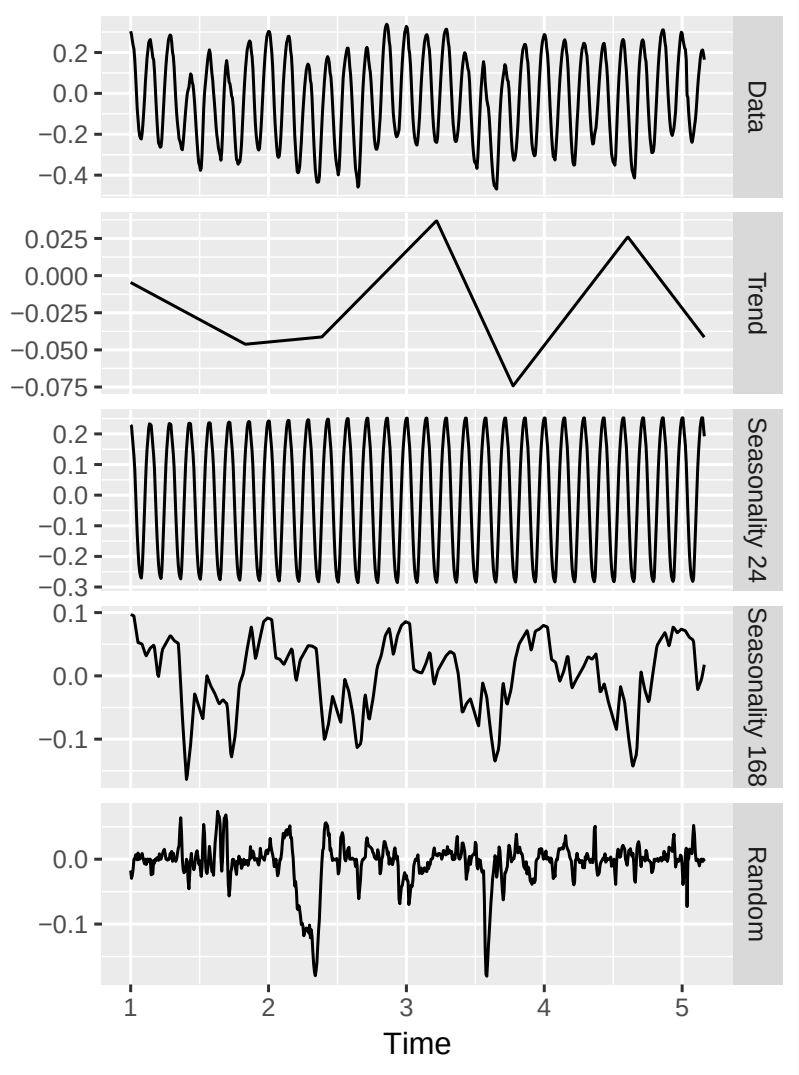
Decomposition methods such as

- STL (Cleveland et al., 1990)
- MSTL(Hyndman and Athanasopoulos, 2018)
- STR (Dokumentov and Hyndman, 2020)

Deseasonalisation (cont'd)



(a) MSTL Decomposition

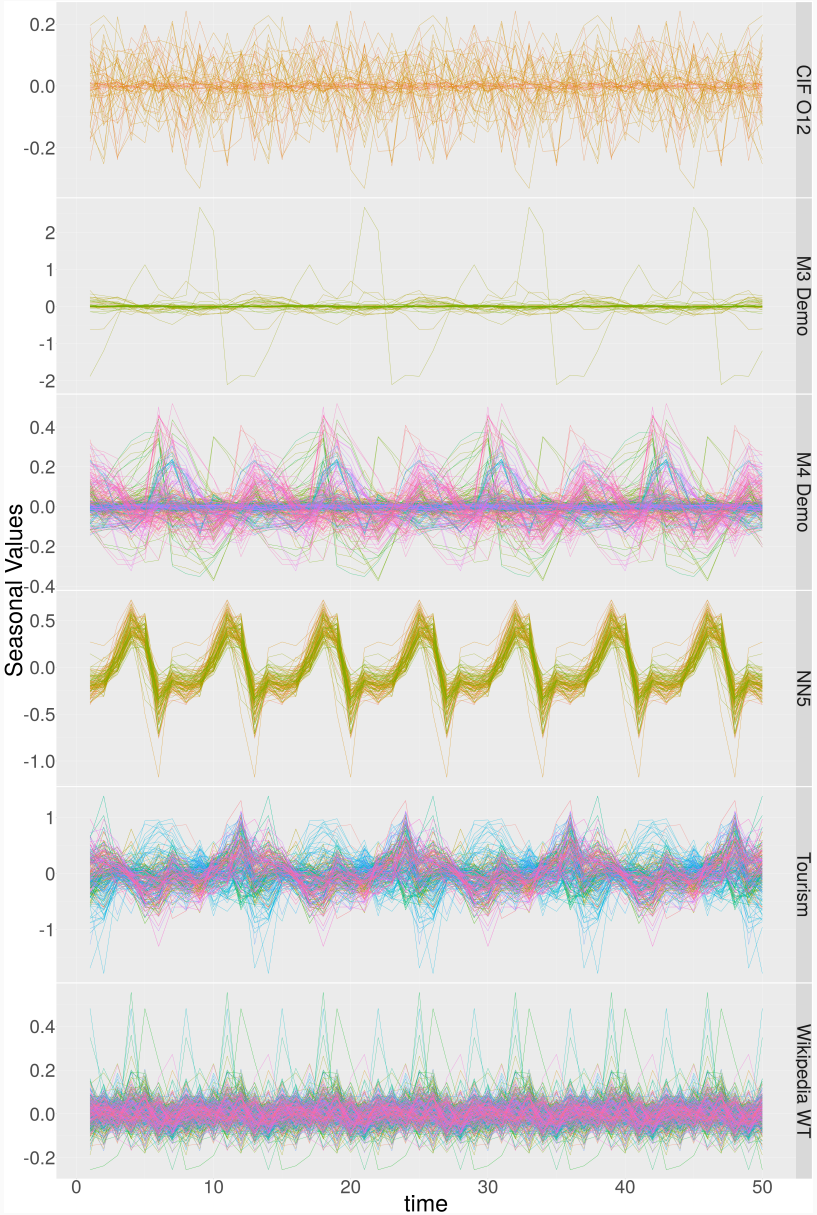


(b) STR Decomposition

Deseasonalisation (cont'd)

- we can deseasonalise and then only feed the trend and remainder component into the ML algorithm
- can be seen as a form of boosting (weak learner to model seasonality, ML model trained on residuals)
- effectively putting expert knowledge into the model
- works well if not enough data to model seasonality directly, or if seasonal components are very different between series

Deseasonalisation (cont'd)



Further considerations about seasonality

- M3, M4 datasets have different types of seasonalities mixed together, and series are not aligned
- in real-world datasets this will normally not be the case
- seasonal indicators and Fourier terms need aligned series

–> In real-world datasets, with enough data, seasonal indicators and Fourier terms will work better than deseasonalisation

Normalisation

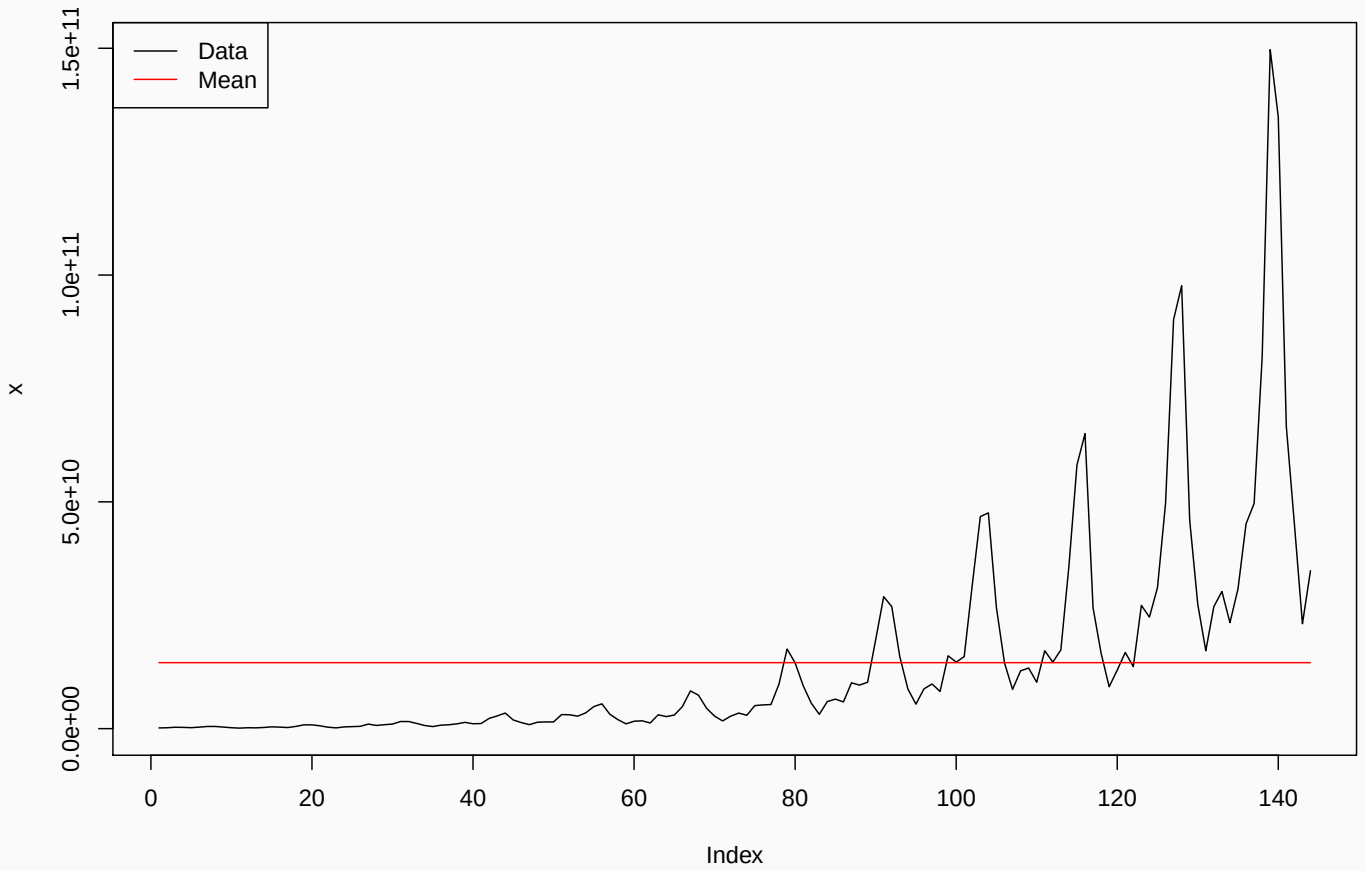
- in some forecasting problems, the forecasts are within a pre-specified domain (wind speed, wind power, electricity price)
- in others, they are not (share prices, web page hits, businesses that grow fast, like ride share applications, social networks, etc.)
- if the domain is limited, the scale has information (if you are at zero, you know the value will not be able to fall more)
- if the level is already high, trends tend to be less steep

–> include information about scale as additional input, if normalising

- error measures (sMAPE, MASE) are also often scale invariant, though this may not reflect the real world

Normalisation (2)

- Mean normalisation, if series don't have a strong trend
- Not a good idea if there are strong trends/level shifts:



Direct output versus iterative 1-step-ahead

- Traditional methods (ETS, ARIMA) optimise 1-step-ahead accuracy
- forecasts further out are obtained by iteration, by feeding back the forecasts as inputs into the model
- this leads to error accumulation
- better to predict directly all horizons needed (Ben Taieb et al., 2012; Wen et al., 2017)
- methods such as NNs can have multiple outputs
  –> output windows
- other methods, such as GBT, can either iterate out or train one model per horizon

Gradient boosted trees (GBT) for forecasting

- Successful in several forecasting competitions (GEFCom 2014: Landry et al. (2016), Kaggle competitions)
- M5 competition dominated by LightGBM
- Catboost (Prokhorenkova et al., 2018) performs *ordered* gradient boosting, especially suitable for time series
- Seasonality and trend handling as discussed earlier
- build one model per horizon
- especially suitable with (diverse) external variables
- engineer features such as rolling means, rolling sds
- use differences as additional inputs (of different lags, e.g., lag1 difference, lag 12 difference)

Further feature engineering

- Holiday effects
  - one-hot encoded
  - as distance maps (days before/after holiday)
- Monthly series: number of trading days in the month
- ...

Ensembling and forecast combination

- Ensembling works in forecasting just as well as in any other area
- Heavily used, e.g., in Kaggle competitions
- Ensembles of GBTs, NNs, pooled regression
- Ensembles of local and global models

Forecast combination

- Seminal paper by Bates and Granger (1969)
- Show that combining forecasts often leads to better accuracy
- Widely accepted and adopted in the forecasting field since then
- Simple average is hard to beat, though many more sophisticated methods exist

Deep learning for forecasting

Overview: Deep learning for forecasting

- Not covered extensively here, only a brief overview
- A recent review is given by Benidis et al. (2020)
- Great recent tutorials by Amazon: Wang et al. (2020)
- Usually, recent NLP research (LSTM, attention, transformers, ...) is adapted to the time series use case
- Main differences to NLP:
  - most recent observations are the most important ones
  - long-term dependencies are relatively simple and stable (only seasonalities)

Beyond autoregression: Recurrent neural networks

- Overview by Hewamalage et al. (2020)
- Have an internal state which allows them to memorize
- They have problems to learn long memory (Pascanu et al., 2013)
- LSTM mitigates that to a certain extent
- They still work better in practice if used with input and output windows (Hewamalage et al., 2020)
  - making them more autoregressive
  - making the state less important
- Recent results suggest that in general RNNs that can be trained and are stable can be well approximated by feed-forward networks (Miller and Hardt, 2018; Miller, 2018)

Convolutional neural networks

- Recent results suggest that CNNs work just as well as RNNs for forecasting, but are a lot faster to train (Borovykh et al., 2018)
- WaveNet (Oord et al., 2016; Sen et al., 2019): causal convolutions, dilations
- CNNs don't have a state, so windowing needs to cover everything

–> Long input windows, especially with dilations

–> RNNs are preferrable if input windows need to be small, e.g., across series of different length in a global model.

Specialised architectures

- DeepAR (Flunkert et al., 2017): Generative RNN model
- Deep state space models (Rangapuram et al., 2018): Parametrizes a linear state-space model with an RNN
- Deep Factors for forecasting (Wang et al., 2019): Combines a local probabilistic model and a global time series that is a linear combination of factors
- NBEATS (Oreshkin et al., 2019): Decomposes series into basis functions, residual stacking
  - State-of-the-art accuracy on M4
  - 2nd place in M5 (as part of an ensemble)
- Transformers for forecasting (Li et al., 2019), Temporal Fusion Transformers (Lim et al., 2019)

Multivariate forecasting

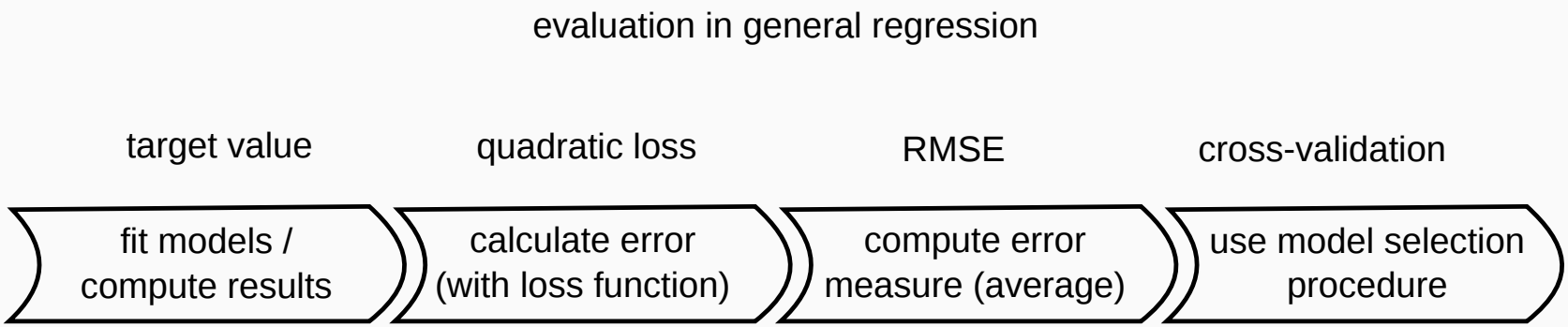
- Not the same as global models: series need to be aligned, and often have the same length
- Very active research area. How to scale methods to thousands of time series?
- Usually sparseness constraints: Each series can interact only with few other series

Multivariate forecasting (cont'd)

- Yu et al. (2016): Matrix factorization with temporal regularization
- Lai et al. (2018), LSTNet: CNN feeding into RNN with skip connections
- Sen et al. (2019): Matrix factorization, causal convolutions and attention
- Salinas et al. (2019a): Gaussian copulas
- Wu et al. (2020): Graph neural networks

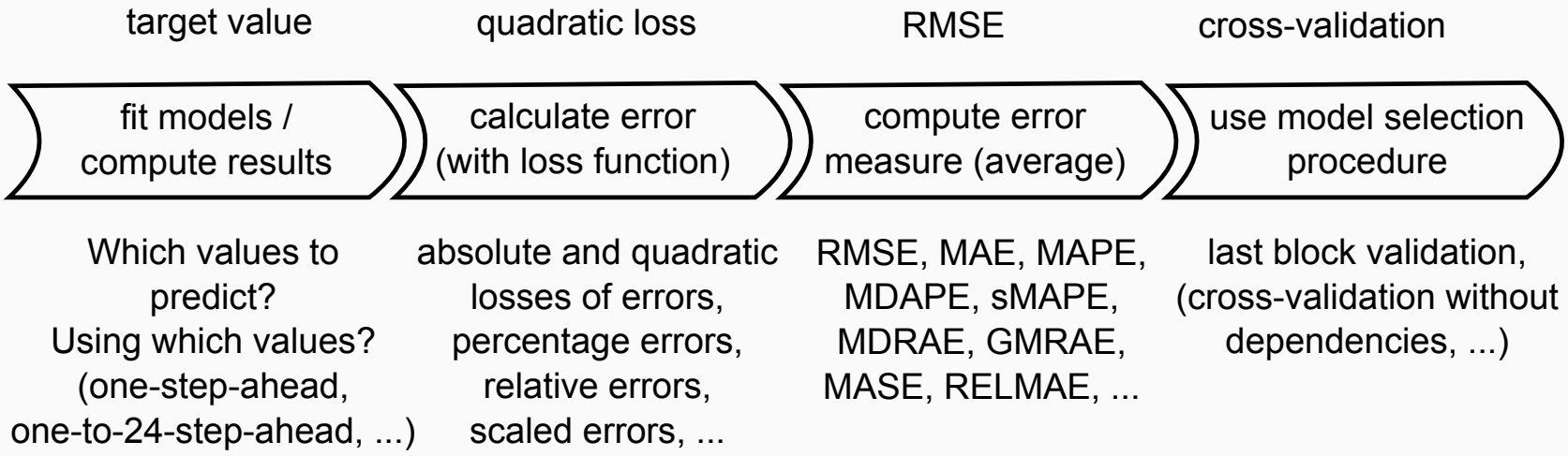
Forecast Evaluation

Evaluation


evaluation in general regression
target value
quadratic loss
RMSE
cross-validation
fit models / compute results
calculate error (with loss function)
compute error measure (average)
use model selection procedure

Evaluation

evaluation in general regression



traditional time series forecast evaluation

Errors and error measures

- Overview by Hyndman and Koehler (2006)
- Still controversial, no solution that always works
- Still papers that come up with their own ad-hoc measure
- Measures often misused

Scale-dependent errors and error measures

Scale-dependent errors, squared error (SE), absolute error (AE):

$$\text{SE}_t = (y_t - \hat{y}_t)^2$$

$$\text{AE}_t = |y_t - \hat{y}_t|$$

Corresponding error measures, e.g., RMSE and MAE:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

Problem: Cannot be used to compare forecasts across series on different scales

–> We need to somehow normalise

Percentage Errors (PE)

$$\mathrm{PE}_t = 100\ \frac{y_t - \hat{y}_t}{y_t}, \qquad \mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|100\ \frac{y_t - \hat{y}_t}{y_t}\right|$$

–> Problem: Cannot be used if y_t is zero and is distorted when y_t is small. Was originally used for inventory count data.

–> Is also not symmetric: exchanging the prediction and the true value changes the result

Solution: symmetric MAPE (sMAPE)

$$\text{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|100\ \frac{y_t - \hat{y}_t}{\frac{|y_t|+|\hat{y}_t|}{2}}\right| = 200\frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{|y_t| + |\hat{y}_t|}\right|$$

- Achieves symmetry from before, but breaks another symmetry: overprediction is penalised less than underprediction (as dividing by the prediction)
- Still has problems with zeros; if both actual and prediction are zero, it is not defined
- If there is a zero in the data and we don't predict an exact zero, the sMAPE is maximal.
    - This has large implications for intermittent data.

modified sMAPE (Suilin, 2017)

$$\text{msMAPE} = 200\frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{max(|y_t| + |\hat{y}_t| + \epsilon, 0.5 + \epsilon)}\right|$$

With a default of $\epsilon = 0.1$

Relative errors and error measures

Use a benchmark method B (usually the naive forecast)

Relative Errors:

$$\text{RE}_t = \frac{y_t - \hat{y}_t}{y_t - \hat{y}_{tB}}, \qquad \text{MRAE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t - \hat{y}_{tB}}\right|.$$

–> Same problems as before, if true values are zero and predictions from the benchmarks are zero, etc.

Relative Error Measures:

$$\text{RelMAE} = \frac{\text{MAE}}{\text{MAE}_B}.$$

Only has benefits if we are evaluating many predictions on the same scale. If series on different scales and only one forecast per series, it has the same problems as before.

Mean Absolute Scaled Error (MASE)

- Proposed by Hyndman and Koehler (2006)
- Defined as the MAE divided by the MAE of the (seasonal) naive forecast over the training part of the time series.

$$\text{MASE} = \frac{\sum_{t=1}^{n} |\hat{y}_t - y_t|}{\frac{n}{m-s} \sum_{k=s+1}^{m} |y_k - y_{k-s}|}$$

The naive forecast works a lot better on certain (smooth) series than on others.

–> High or low MASE is not necessarily equivalent with good or bad performance.

Error measures: Summary

- To get a scale-free measure, we need to divide by a normalising factor
- Due to the potential non-stationarity of the series, it has turned out to be extremely difficult to do this in a way that always works
- Today, the sMAPE and the MASE are standard error measures
  - used, e.g., in the recent M4 competition
  - but watch out for small values, intermittent series, and other problems

Error measures: Summary (2)

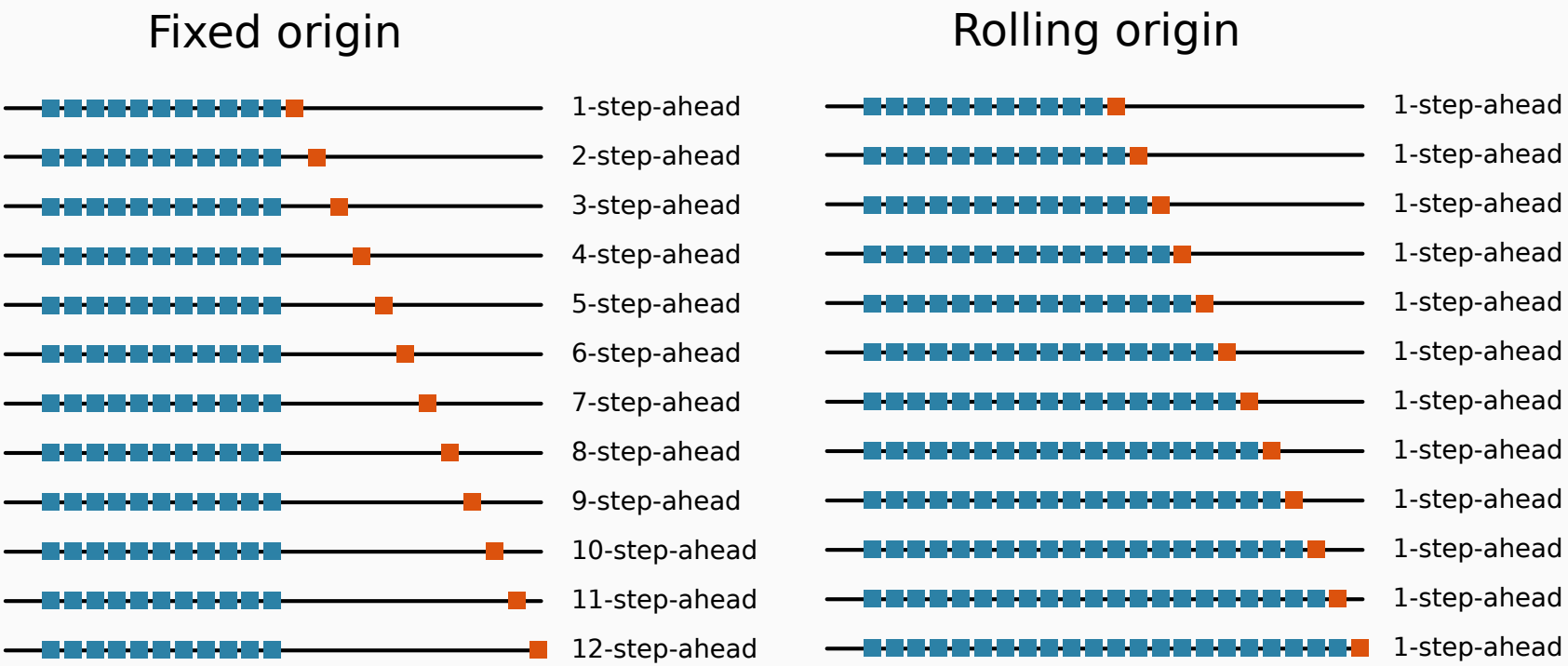
- When building a global model, often the series have meaningful scales (SKUs, dollars, ...)
- Then you probably don't want a percentage error per series, but an error on the scale

–> If you don't need a scale-free measure, better stick to MAE or RMSE

Training and test set splitting

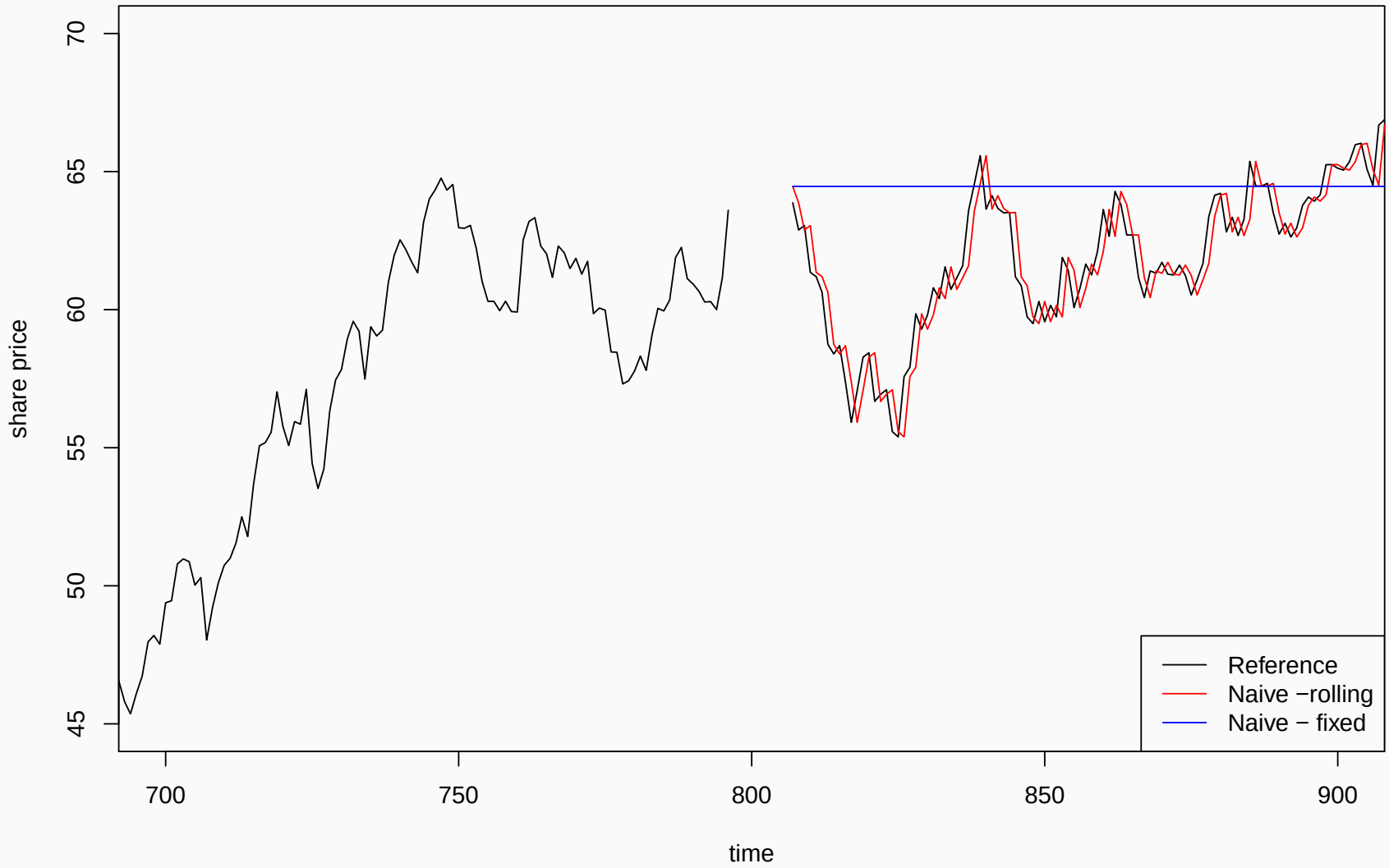
Training and test split:

Fixed and rolling origin evaluation:



Fixed and rolling origin evaluation are very different

Example: Naive forecast



Remember:

- Always benchmark against a naive forecast
- Don't trust the plots!

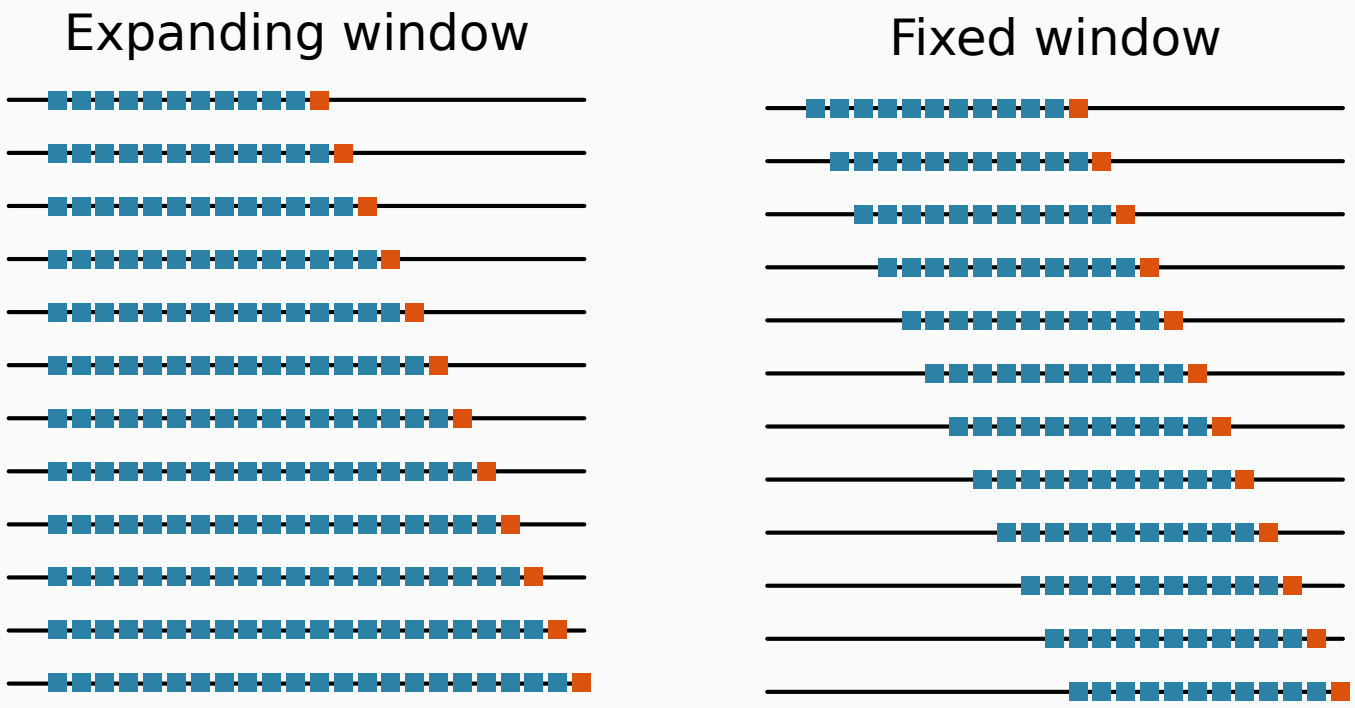
Fixed origin evaluation

- Easier to implement than rolling origin
- Forecasting competitions mostly operate like this, as test set cannot be disclosed
- Drawback: Not usually how we operate in practice (forecasts every week, every 5 minutes, ...)
- Only one forecast can be evaluated per horizon
- Often difficult to capture the full picture: Summer holidays, Christmas, etc.

Rolling origin evaluation

- Can be difficult to implement with some of the traditional methods, as they would usually be re-trained for every forecast
- Much more natural with ML methods, where it is normal to have new data and not re-train the model
- Even more important in a global model where all the series are in sync regarding their timestamps.
- In a competition dataset like the M3 or M4, the series are not aligned and already mixed together wildly, so it is less important.
- Beware of leakage from training to test set!
- Rolling origin evaluation is also called time series cross-validation (TSCV)

Rolling origin evaluation (cont'd)

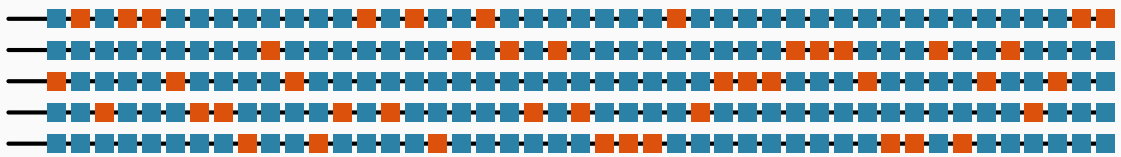


- Expanding window better if less data available, as training set grows (Bell and Smyl, 2018)
- Also can combine the two: start with expanding window, then move to fixed window

Cross-validation

- We have seen TSCV
- Why not just do a normal cross-validation?

Out of sample evaluation



5-fold cross-validation

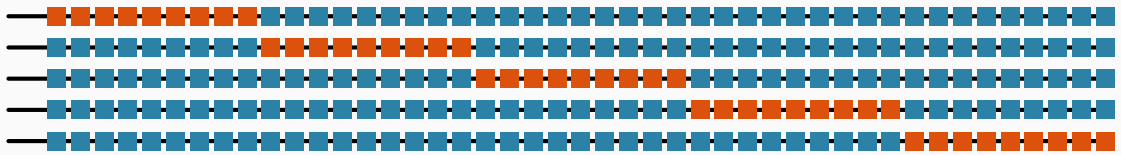
Cross-validation (cont'd)

- Problems of serial correlation between the data
- Problems of non-stationarities
- Using data from the future to predict the past doesn't feel right

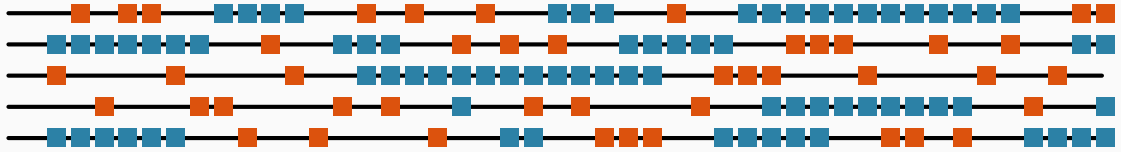
–> fear that CV will grossly underestimate the generalisation error

- Most traditional methods (ETS, ARIMA) and also RNNs cannot deal well with missing data (that is reserved for testing)

Cross-validation (cont'd)



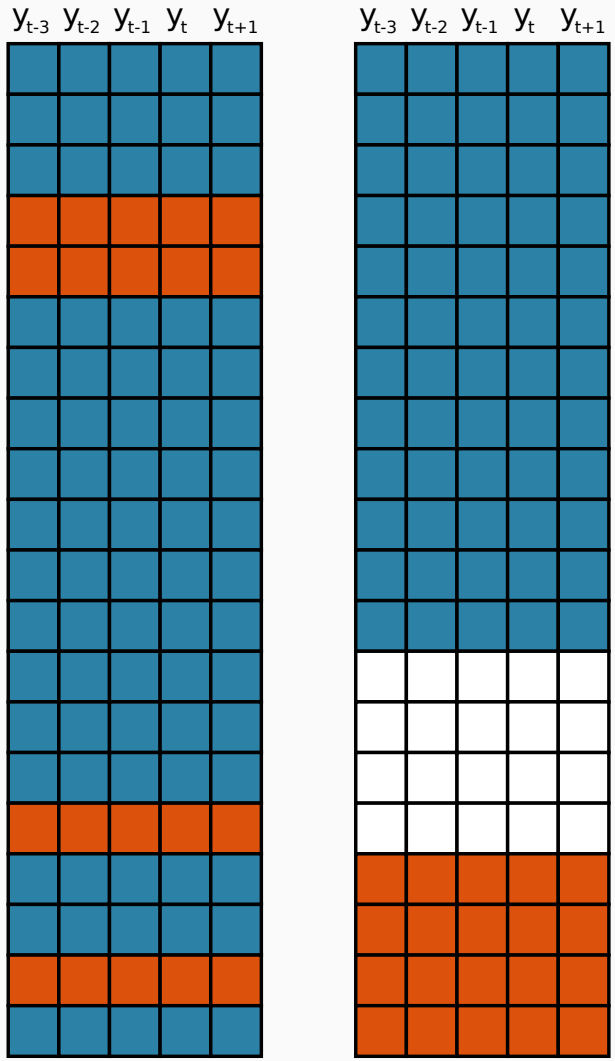
5-fold blocked cross-validation



5-fold non-dep. cross-validation

- Works in the literature to address problems of serial correlation: non-dependent cross-validation or blocked cross-validation (Burman et al., 1994; Racine, 2000; Bergmeir and Benítez, 2012)
- But general problems of non-stationarity remain

Pure AR models and Cross-validation



Pure AR models and Cross-validation (cont'd)

- Theoretical prove that cross-validation performs well in a purely autoregressive setup, as long as models nest or approximate the true model, as then errors are uncorrelated (Bergmeir et al., 2018)
- As seen before, many Machine Learning methods work like this
- Cross-validation can and should be used without modification to detect overfitting then.
- Underfitting can be detected separately, e.g., by a test for serial correlation.
- Implemented in the `CVar` function in the `forecast` package in R (Hyndman and Khandakar, 2008)

Model diagnostics: Ljung-Box test for serial correlation

- Ljung and Box (1978)
- Detects serial correlation in the residuals, which means the model is underfitting

–> Look at the residuals that you are getting. Is there autocorrelation in there? Are there any seasonal patterns in there?

–> If yes, model is still not using all the information available in the data

Probabilistic forecasting

Probabilistic forecasting

Main ways for probabilistic forecasting:

- use analytical prediction intervals
- bootstrapping, MCMC sampling
- forecast the parameters of a distribution
- use quantile regression (pinball loss)
- determine uncertainty empirically through backtesting

Use analytical prediction intervals

- Possible for some (well-understood) models
- ETS, ARIMA do this
- intervals tend to be too narrow

Bootstrapping, MCMC sampling

- slow
- intervals can also be too narrow if, e.g., they only consider parameter uncertainty

Determine uncertainty empirically through backtesting

- often used by companies in practice
- leads to more realistic prediction intervals
- needs a lot of past data and rolling origin forecasts

Forecast the parameters of a distribution

- e.g., assuming a normal distribution: μ, σ
- DeepAR (Salinas et al., 2019b): Normal distribution and negative binomial distribution
- Have to assume a certain distribution
- Good if we have limited amounts of data, or knowledge of the distribution

Quantile regression (pinball loss)

- implemented in, e.g., Wen et al. (2017)
- no distribution assumptions need to be made; therewith better if a lot of data are available
- fast to compute and easy to implement
- only certain quantiles can be obtained, not the full distribution
- in practice, often 5 or 7 quantiles are enough anyway
- with, e.g., a Neural Network, we can fit different quantiles at the same time, by having multiple outputs
- can interpolate between quantiles to get full distribution (Gasthaus et al., 2019)

Pinball loss function

$$
\begin{aligned}
L_\tau(y, \hat{y}) &= (y - \hat{y})\tau && \text{if } y \geq \hat{y} \\
&= (\hat{y} - y)(1 - \tau) && \text{if } \hat{y} > y
\end{aligned}
$$

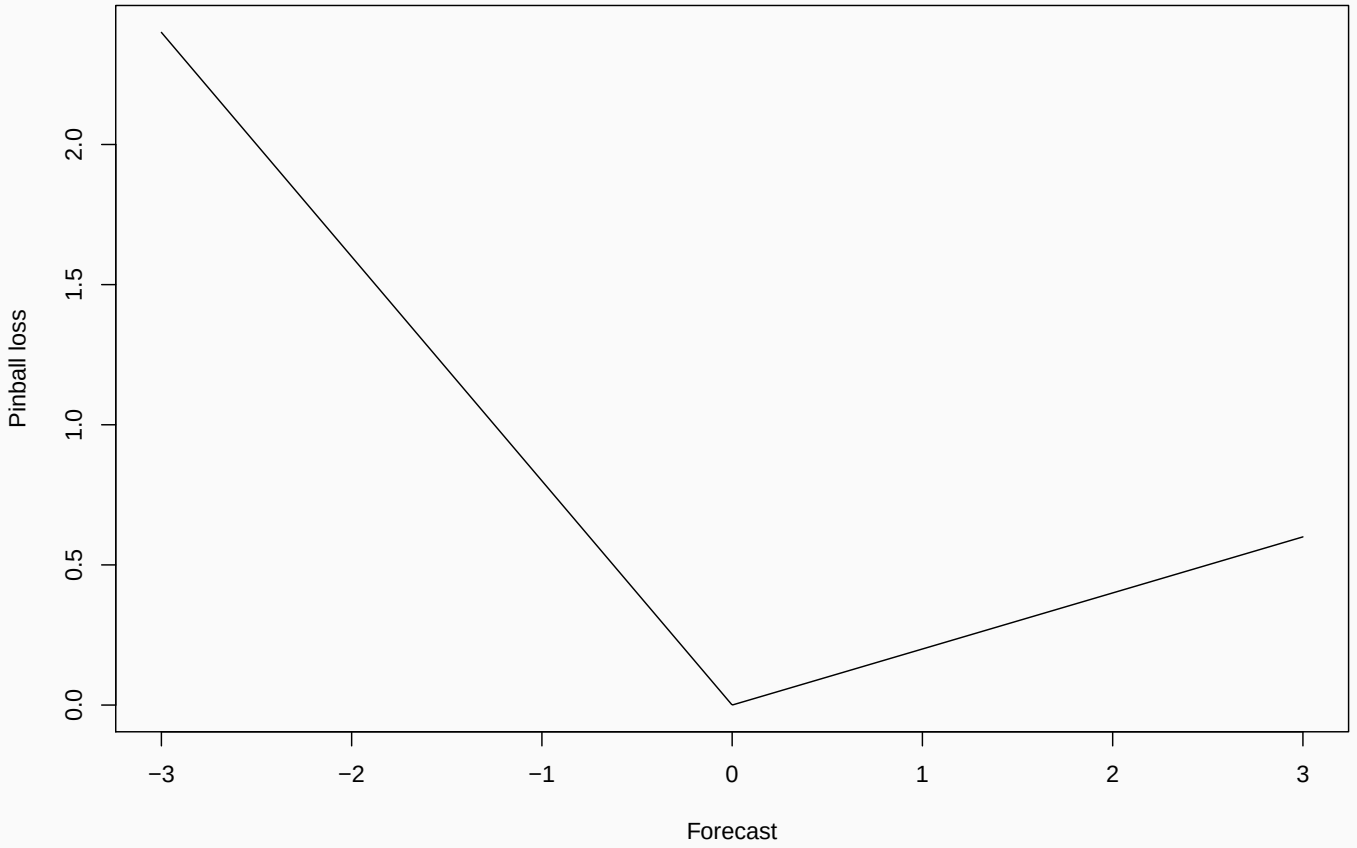
y is the true value

$\hat{y}$ is the forecast

τ is the target quantile

It can be proved that minimizing the pinball loss results in the most accurate quantile

Pinball loss (2)



Evaluation of probabilistic forecasts

- Simple hit/miss rule
  - directly interpretable (e.g., “80% of forecasts are within the interval”)
  - doesn’t consider the magnitude of the error
  - Problem: trivial solutions are possible by grossly over- and underpredicting certain amounts of times

Evaluation of probabilistic forecasts (cont'd)

- Overviews by Gneiting and Raftery (2007);Jordan et al. (2017)
- Mean Scaled Interval Score (MSIS)
  - used in the M4
  - sum over the size of the intervals and the magnitude of error for points that lie outside of the interval
- Continuous Ranked Probability Score (CRPS)
  - integrates over all quantiles
  - generalises the MAE to the probabilistic case
- logarithmic score, energy score, variogram score

Special forecasting problems

Intermittent data

- Zero-inflated models
  - One model that predicts the probability for a zero
  - A second model that predicts a value under the assumption that it is non-zero
  - Croston's method (Croston, 1972): Essentially a zero-inflated model using SES for both these predictions
- Specialised loss functions
  - negative binomial loss function (used in DeepAR)
  - Poisson loss function
  - Tweedie loss function (Zhou et al., 2020)
- It seems that different models work better, depending on the degree of intermittency. Rule of thumb:
  - With $>$90% zeros, better to use a zero-inflated model
  - Otherwise, an adapted loss function

Hierarchical forecasting

- Very common problem in forecasting
- Forecasts need to add up
  - Spatially: across stores, distribution centres, countries, ...
  - Product categories, store departments, ...
  - Temporally: Weekly, monthly, yearly forecasts
- Recent overview by Hyndman (2020a)

Hierarchical forecasting: Classic approaches

- top down
  - predict the series at the top
  - disaggregate
  - Problem: How to disaggregate?
  - According to ratios from the past, train a model to disaggregate, etc.
- bottom up
  - predict series at the bottom
  - aggregate
  - Problem: Bottom-level series often have a lot of noise, no clear patterns of seasonality and trend
- middle out
  - predict series at a middle level
  - aggregate and disaggregate

Hierarchical forecasting: Optimal reconciliation

- Introduced by Hyndman et al. (2011)
- Forecast all series separately
- Make them consistent with a reconciliation step, least squares optimisation
- Leads to more accurate forecasts than the classical approaches
- A lot of research and theoretical insights in the forecasting literature since then (e.g., trace minimisation by Wickramasuriya et al. (2019))
- Geometric view: Forecasts must lie on a coherent subspace where linear constraints hold (Panagiotelis et al., 2020)

Probabilistic hierarchical forecasting

- What does "forecasts add up correctly" mean for a probabilistic forecast?
- Panagiotelis et al. (2020): Multivariate density must lie on a coherent subspace
- Some work in the machine learning community (Ben Taieb et al., 2017, 2020)

Hierarchical forecasting (cont'd)

Interesting research possibilities for the Machine Learning community, especially on methods that directly produce reconciled forecasts in one step, as opposed to the current approach of forecasting and then reconciling

Conclusions

- Forecasting as a field has come a long way
- Great time to do forecasting as a Machine Learner or Data Scientist
- Machine Learning methods become more and more competitive
- Be aware of some of the common pitfalls of forecasting, such as evaluation, benchmarks, data-leakage, non-stationarity
- Happy forecasting!

Acknowledgements

I'd like to thank all my collaborators, in particular my PhD students Kasun Bandara, Hansika Hewamalage, Rakshitha Godahewa, and Dilini Rajapaksha for all their hard work!

Thank You

https://www.cbergmeir.com

christoph.bergmeir@monash.edu

References i

K. Bandara, C. Bergmeir, and S. Smyl. Forecasting across time series databases using long short-term memory networks on groups of similar series. *arXiv preprint arXiv:1710.03222*, 8:805–815, 2017.

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*, (forthcoming), 2020a. URL `https://arxiv.org/pdf/1909.04293`.

K. Bandara, C. Bergmeir, and S. Smyl. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, 140:112896, 2020b.

J. M. Bates and C. W. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4): 451–468, 1969.

F. Bell and S. Smyl. Forecasting at Uber: An introduction, 2018. URL `https://eng.uber.com/forecasting-introduction/`. Accessed 2 September 2020.

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 Syst. Appl.*, 39(8):7067–7083, June 2012.

S. Ben Taieb, J. W. Taylor, and R. J. Hyndman. Coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pages 3348–3357, 2017.

S. Ben Taieb, J. W. Taylor, and R. J. Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, pages 1–17, 2020.

K. Benidis, S. S. Rangapuram, V. Flunkert, B. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, et al. Neural forecasting: Introduction and literature overview. *arXiv preprint arXiv:2004.10240*, 2020.

References ii

C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, 2018.

C. S. Bojer and J. P. Meldgaard. Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, 2020.

A. Borovykh, S. Bohte, and C. W. Oosterlee. Dilated convolutional neural networks for time series forecasting. *Journal of Computational Finance*, 2018. doi: 10.21314/jcf.2019.358. URL `https://doi.org/10.21314/jcf.2019.358`.

G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.

P. Burman, E. Chow, and D. Nolan. A cross-validatory method for dependent data. *Biometrika*, 81(2):351–358, 1994. ISSN 00063444.

O. Claveria and S. Torra. Forecasting tourism demand to catalonia: Neural networks vs. time series models. *Econ. Model.*, 36:220–228, Jan. 2014.

R. B. Cleveland, W. S. Cleveland, J. McRae, and I. Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6:3–73, 1990.

J. D. Croston. Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303, 1972.

A. Dokumentov and R. J. Hyndman. Str: A seasonal-trend decomposition procedure based on regression. *arXiv preprint arXiv:2009.05894*, 2020.

References iii

G. T. Duncan, W. L. Gorr, and J. Szczypula. Forecasting analogous time series. In *Principles of forecasting*, pages 195–213. Springer, 2001.

V. Flunkert, D. Salinas, and J. Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *CoRR*, abs/1704.04110, 2017. URL `http://arxiv.org/abs/1704.04110`.

J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski. Probabilistic forecasting with spline quantile function rnns. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1901–1910, 2019.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

P. Goodwin. The Holt-Winters approach to exponential smoothing: 50 years old and going strong. *Foresight: The International Journal of Applied Forecasting*, 19:30–33, 2010.

H. Hewamalage, C. Bergmeir, and K. Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, (forthcoming), 2020. URL `https://arxiv.org/pdf/1909.00590`.

R. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22, 2008.

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

R. Hyndman, A. Koehler, R. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002.

References iv

R. J. Hyndman. Ten years of forecast reconciliation, 2020a. URL https://robjhyndman.com/seminars/reconciliation_review_talk/. Accessed 10 November 2020.

R. J. Hyndman. A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14, 2020b.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9):2579–2589, 2011.

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.

A. Jordan, F. Krüger, and S. Lerch. Evaluating probabilistic forecasts with scoringrules. *arXiv preprint arXiv:1709.04743*, 2017.

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*, SIGIR '18, pages 95–104, New York, NY, USA, 2018. ACM.

M. Landry, T. P. Erlinger, D. Patschke, and C. Varrichio. Probabilistic gradient boosting machines for gefcom2014 wind forecasting. *International Journal of Forecasting*, 32(3):1061–1066, 2016.

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*, pages 5243–5253, 2019.

B. Lim, S. O. Arik, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *arXiv preprint arXiv:1912.09363*, 2019.

References v

G. M. Ljung and G. E. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

S. Makridakis and M. Hibon. Accuracy of forecasting: An empirical investigation. *Journal of the Royal Statistical Society: Series A (General)*, 142(2):97–125, 1979.

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

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

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018a.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018b.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m5 accuracy competition: Results, findings and conclusions. 10 2020.

J. Miller. When recurrent models don't need to be recurrent, 2018. URL `http://www.offconvex.org/2018/07/27/approximating-recurrent/`. Accessed 10 November 2020.

J. Miller and M. Hardt. Stable recurrent models. *arXiv preprint arXiv:1805.10369*, 2018.

P. Montero-Manso and R. J. Hyndman. Principles and algorithms for forecasting groups of time series: Locality and globality. *arXiv preprint arXiv:2008.00444*, 2020.

References vi

M. Nelson, T. Hill, W. Remus, and M. O'Connor. Time series forecasting using neural networks: should the data be deseasonalized first? *J. Forecast.*, 18(5):359–367, 1999.

A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

A. Panagiotelis, P. Gamakumara, G. Athanasopoulos, R. Hyndman, et al. Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. Technical report, Monash University, Department of Econometrics and Business Statistics, 2020.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648, 2018.

J. Racine. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99(1):39–61, 2000.

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*, pages 7785–7794, 2018.

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*, pages 6827–6837, 2019a.

References vii

D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2019b. ISSN 0169-2070.

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*, pages 4837–4846, 2019.

R. Sharda and R. B. Patil. Connectionist approach to time series prediction: an empirical test. *J. Intell. Manuf.*, 3 (5):317–323, Oct. 1992.

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.

S. Smyl and K. Kuber. Data preprocessing and augmentation for multiple short time series forecasting with recurrent neural networks. In *36th International Symposium on Forecasting*, 2016.

M. Štěpnička and M. Burda. Computational intelligence in forecasting (CIF) 2016 time series forecasting competition. In *IEEE WCCI 2016, JCNN-13 Advances in Computational Intelligence for Applied Time Series Forecasting (ACIATSF)*, 2016.

A. Suilin. kaggle-web-traffic. `https://github.com/Arturus/kaggle-web-traffic/blob/master/how_it_works.md`, 2017. Accessed: 2018-11-19.

Z. Tang, C. de Almeida, and P. A. Fishwick. Time series forecasting using neural networks vs. box- jenkins methodology. *Simulation*, 57(5):303–310, Nov. 1991.

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, Feb 2015. ISSN 1476-9360. doi: 10.1057/jors.2013.174.

References viii

Y. Wang, A. Smola, D. C. Maddix, J. Gasthaus, D. Foster, and T. Januschowski. Deep factors for forecasting. *arXiv preprint arXiv:1905.12417*, 2019.

Y. Wang, C. Faloutsos, V. Flunkert, J. Gasthaus, and T. Januschowski. Forecasting big time series: theory and practice. In *The Web Conference*, 2020. URL `https://www.amazon.science/videos-and-tutorials/forecasting-big-time-series-theory-and-practice`.

R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka. A Multi-Horizon quantile recurrent forecaster. In *Neural Information Processing Systems*, Nov. 2017.

S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526): 804–819, 2019.

Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. *arXiv preprint arXiv:2005.11650*, 2020.

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*, pages 847–855, 2016.

G. P. Zhang and D. M. Kline. Quarterly Time-Series forecasting with neural networks. *IEEE Trans. Neural Netw.*, 18(6):1800–1814, Nov. 2007.

G. P. Zhang and M. Qi. Neural network forecasting for seasonal and trend time series. *Eur. J. Oper. Res.*, 160(2): 501–514, 2005.

H. Zhou, W. Qian, and Y. Yang. Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics-Simulation and Computation*, pages 1–23, 2020.