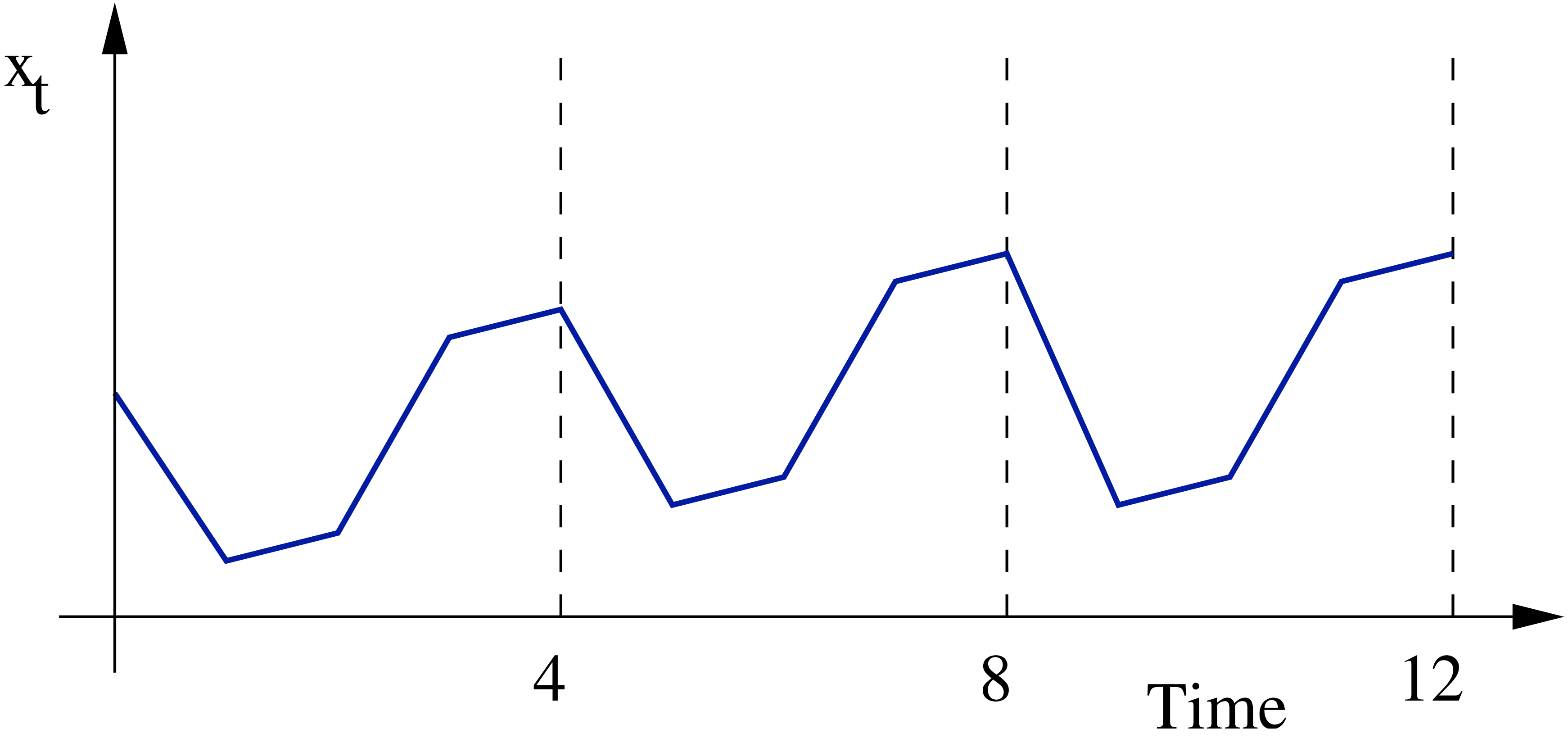
Time Series Forecasting



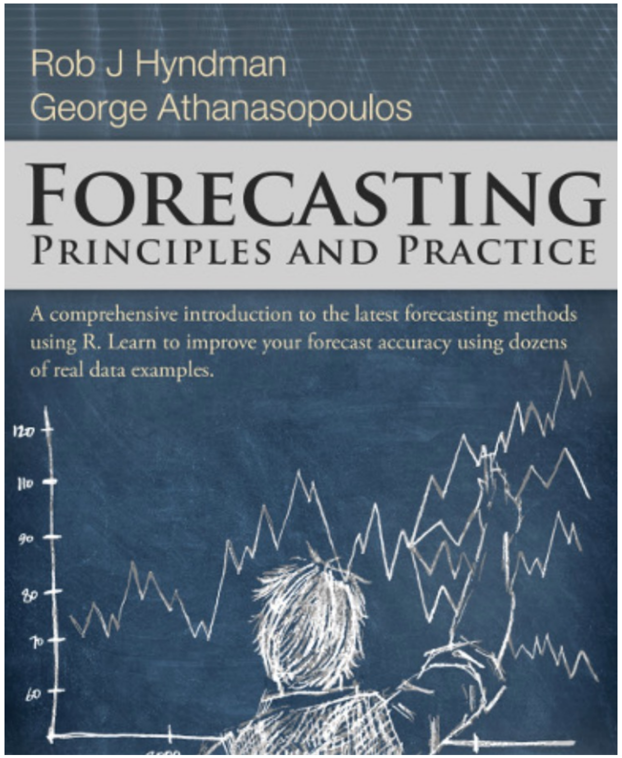
> Bibliography

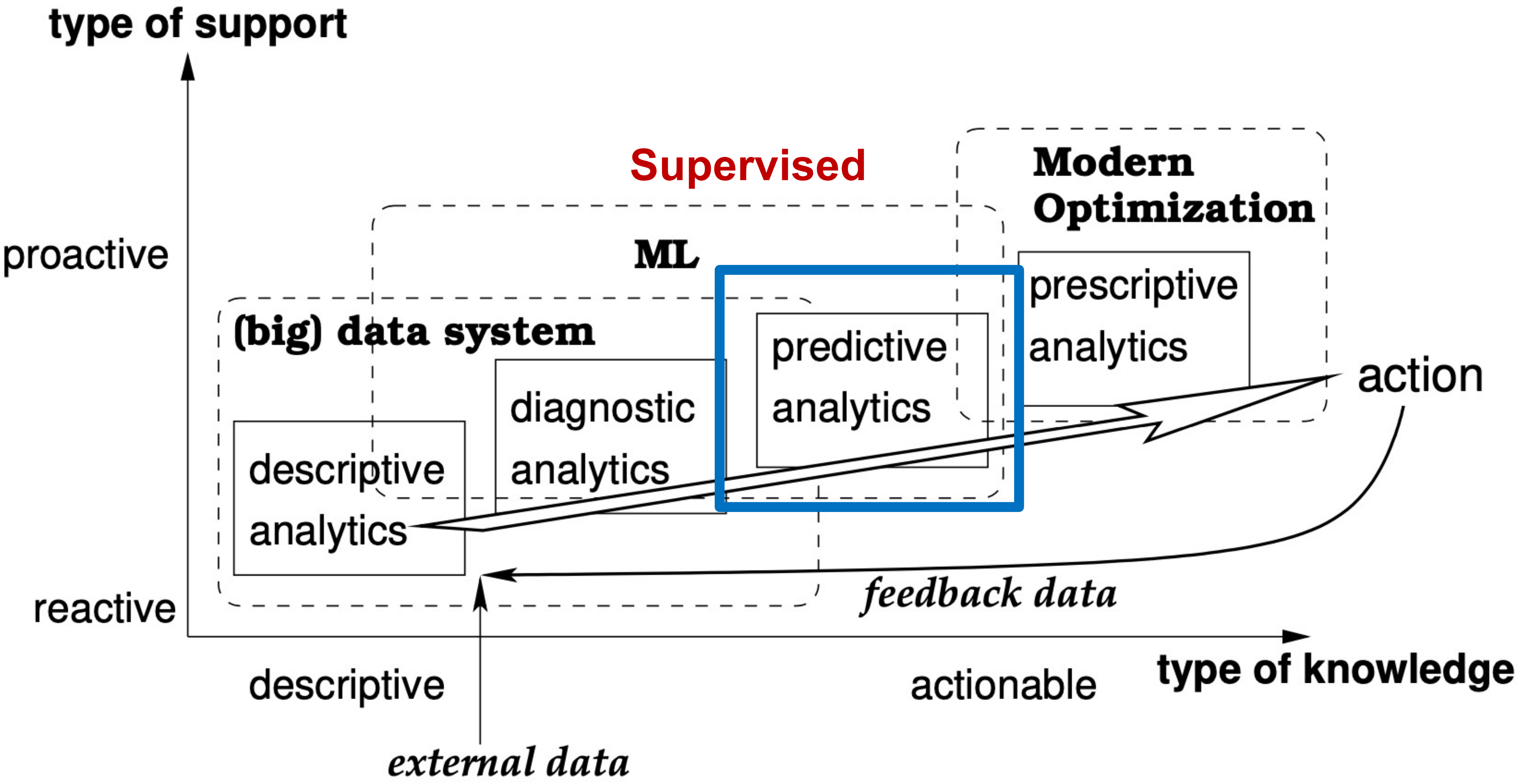
- Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 2nd and 3rd editions, OTexts: Melbourne, Australia:

https://otexts.com/fpp2/

https://otexts.com/fpp3

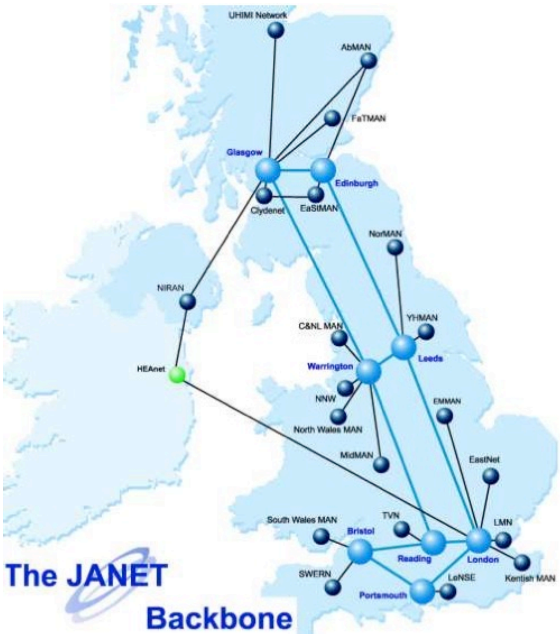
- Other complementary papers

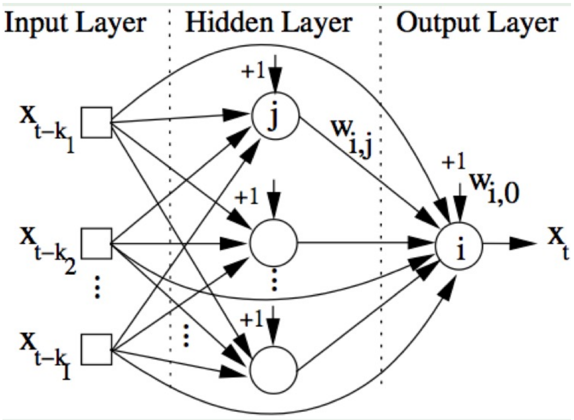



type of support
proactive
reactive
Supervised
Modern
Optimization
ML
(big) data system
descriptive
analytics
diagnostic
analytics
predictive
analytics
prescriptive
analytics
action
feedback data
external data
descriptive
actionable
type of knowledge

Examples

INTERNET TRAFFIC FORECASTING (Time Series Forecasting– NN):





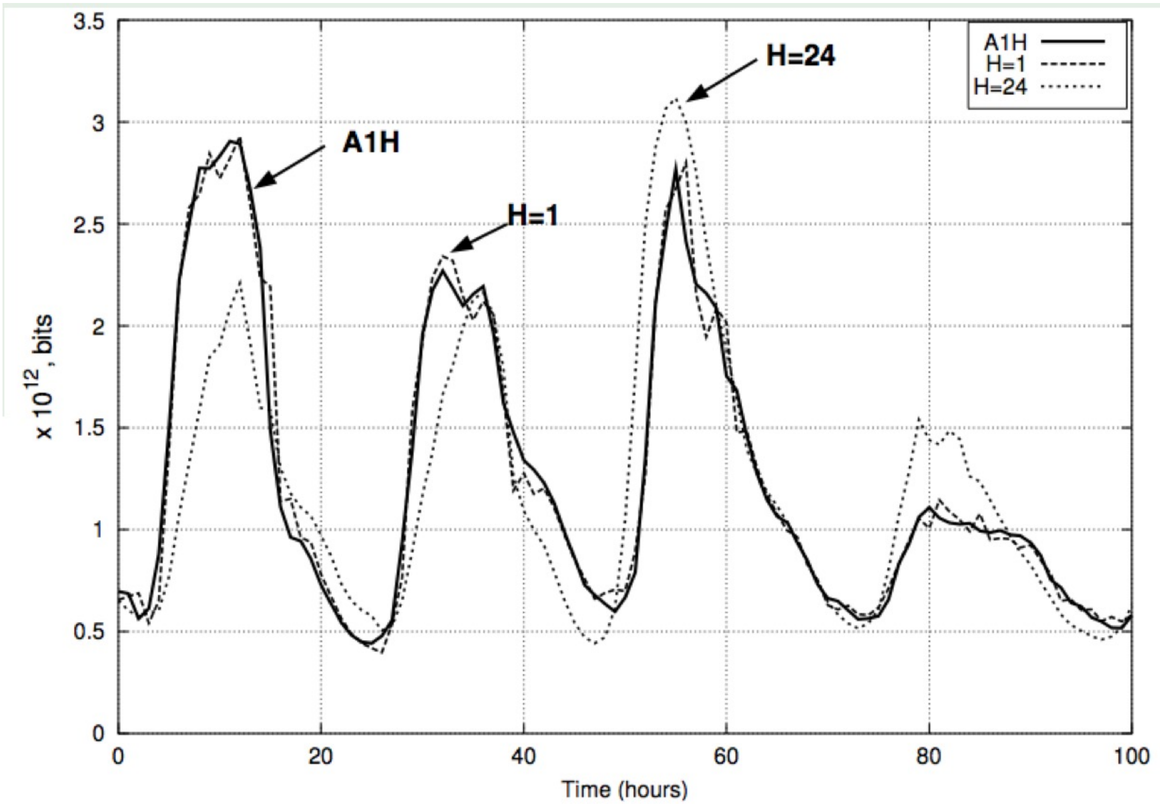
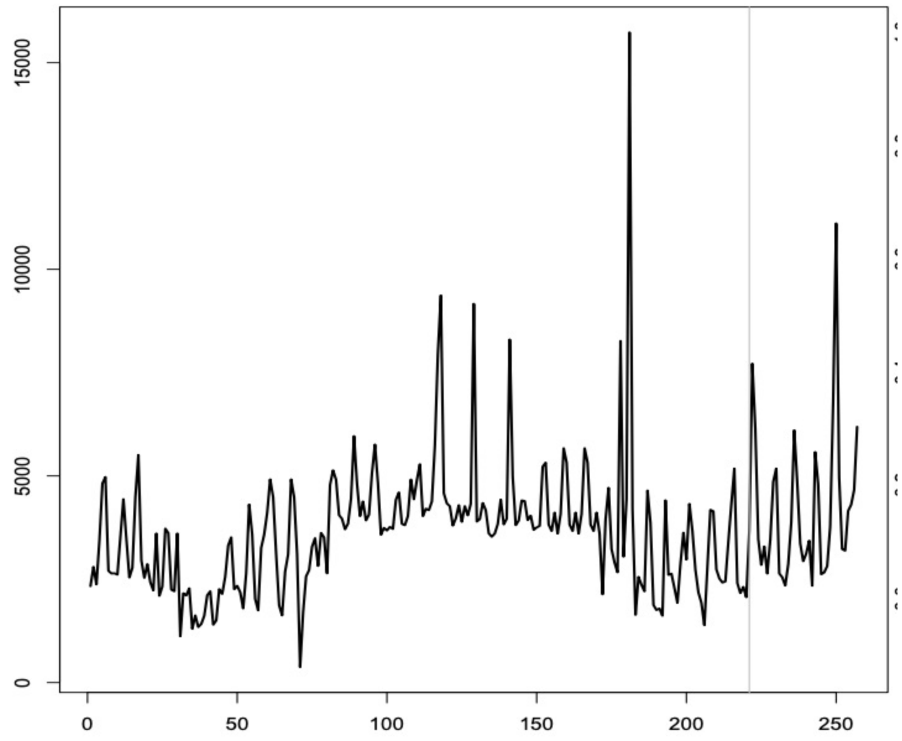


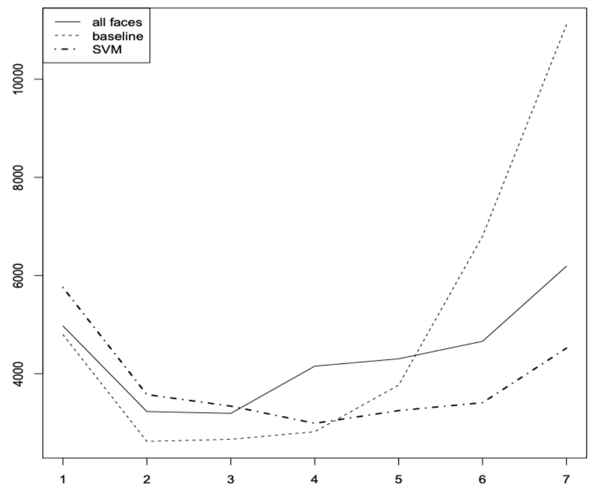
Fig. 5. Example of the Neural Network Ensemble forecasts for series **A1H** and lead times of $h = 1$ and $h = 24$

http://hdl.handle.net/1822/14482

Foot Store Traffic using facial recognition (Time Series Forecasting– NN):



Series	Baseline	HW	ARIMA	SVM	HW-SVM	SVM-SVM
all faces	11.74	9.22	11.42	**8.32**	10.59	9.81
female	11.37	**9.32**	11.25	9.36	10.86	11.53
male	11.25	9.69	11.44	11.15	**9.56**	12.82



Cortez, P., Matos, L. M., Pereira, P. J., Santos, N., & Duque, D. (2016, October). Forecasting store foot traffic using facial recognition, time series and support vector machines. In International Joint Conference SOCO'16-CISIS'16-ICEUTE'16 (pp. 267-276). Springer, Cham..
http://hdl.handle.net/1822/43068

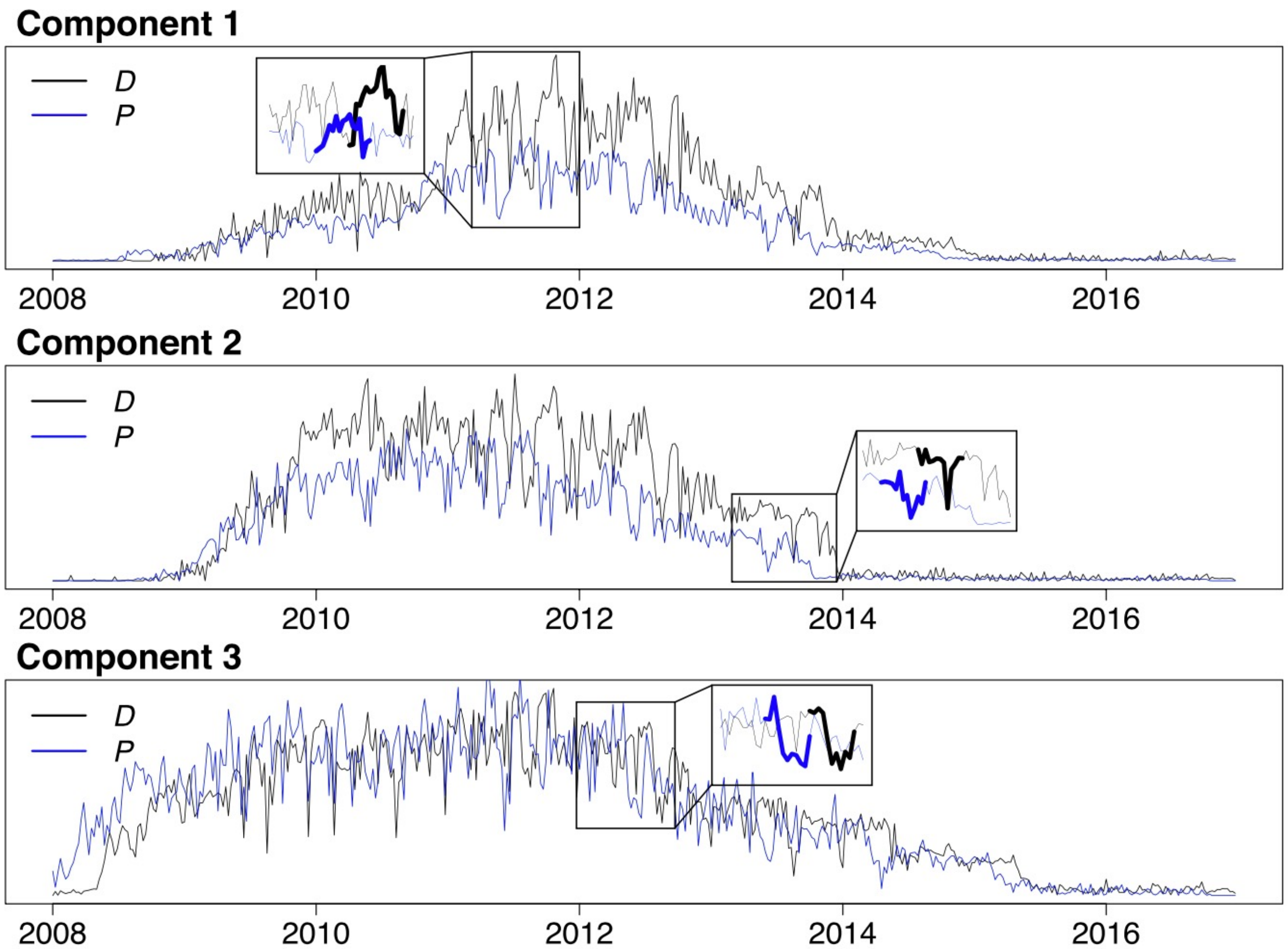
ELSEVIER
Decision Support Systems
Volume 142, March 2021, 113452

A multivariate approach for multi-step demand forecasting in assembly industries: Empirical evidence from an automotive supply chain

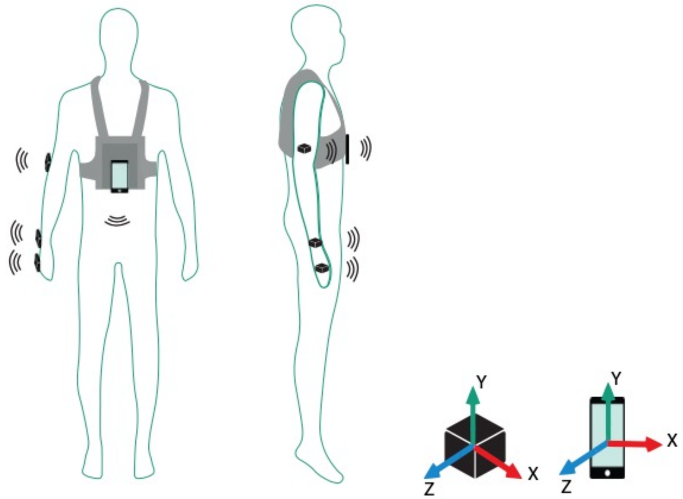
João N.C. Gonçalves [a], Paulo Cortez [b], M. Sameiro Carvalho [a], Nuno M. Frazão [c]

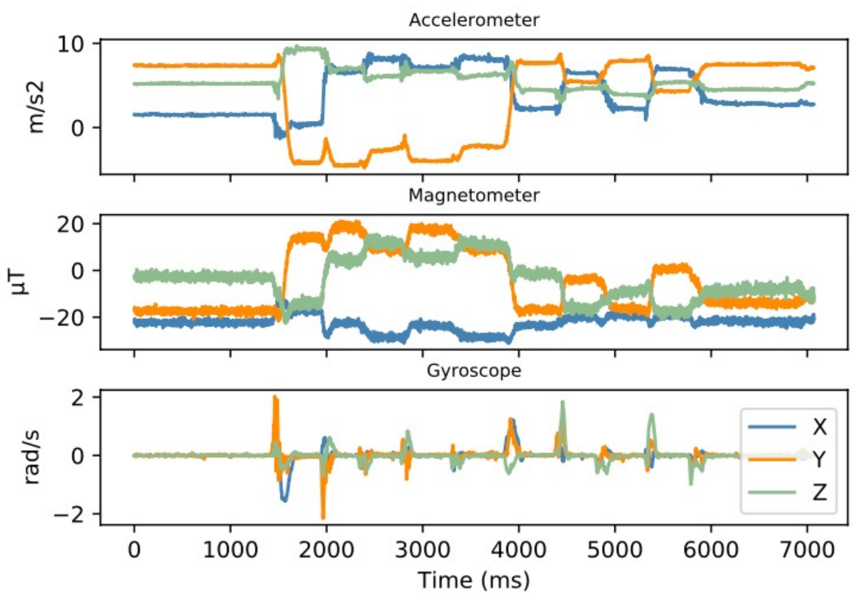
https://doi.org/10.1016/j.dss.2020.113452

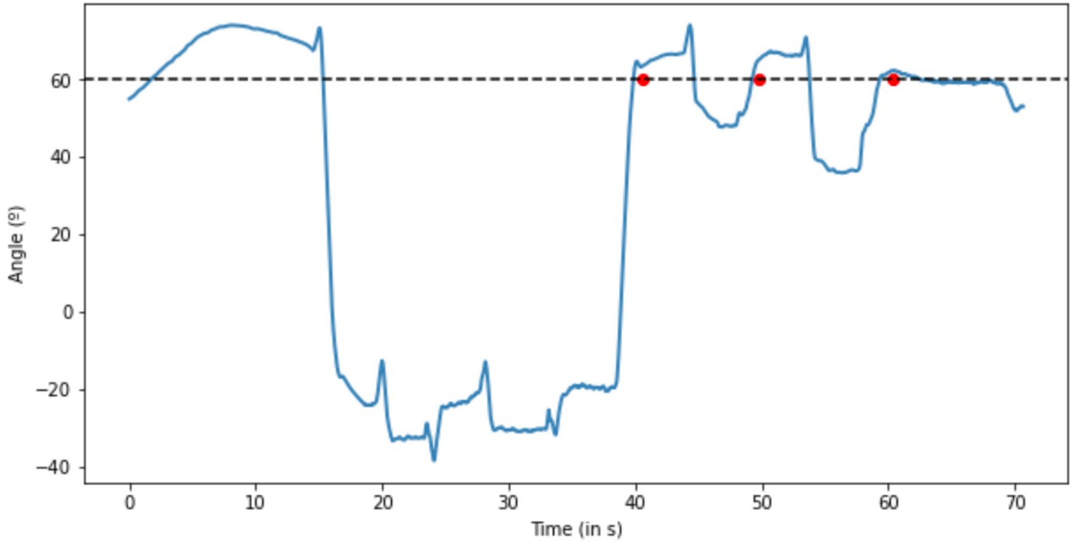
P – planned production orders
D – manufacturer demand



A Deep Learning Approach to Prevent Problematic Movements of Industrial Workers Based on Inertial Sensors







(a) person A

Open your smartphone browser and go to

live.voxvote.com

enter the following 5 numbers

PIN: 78267

~~Voting is anonymous.~~

Screen name or

~~optio~~

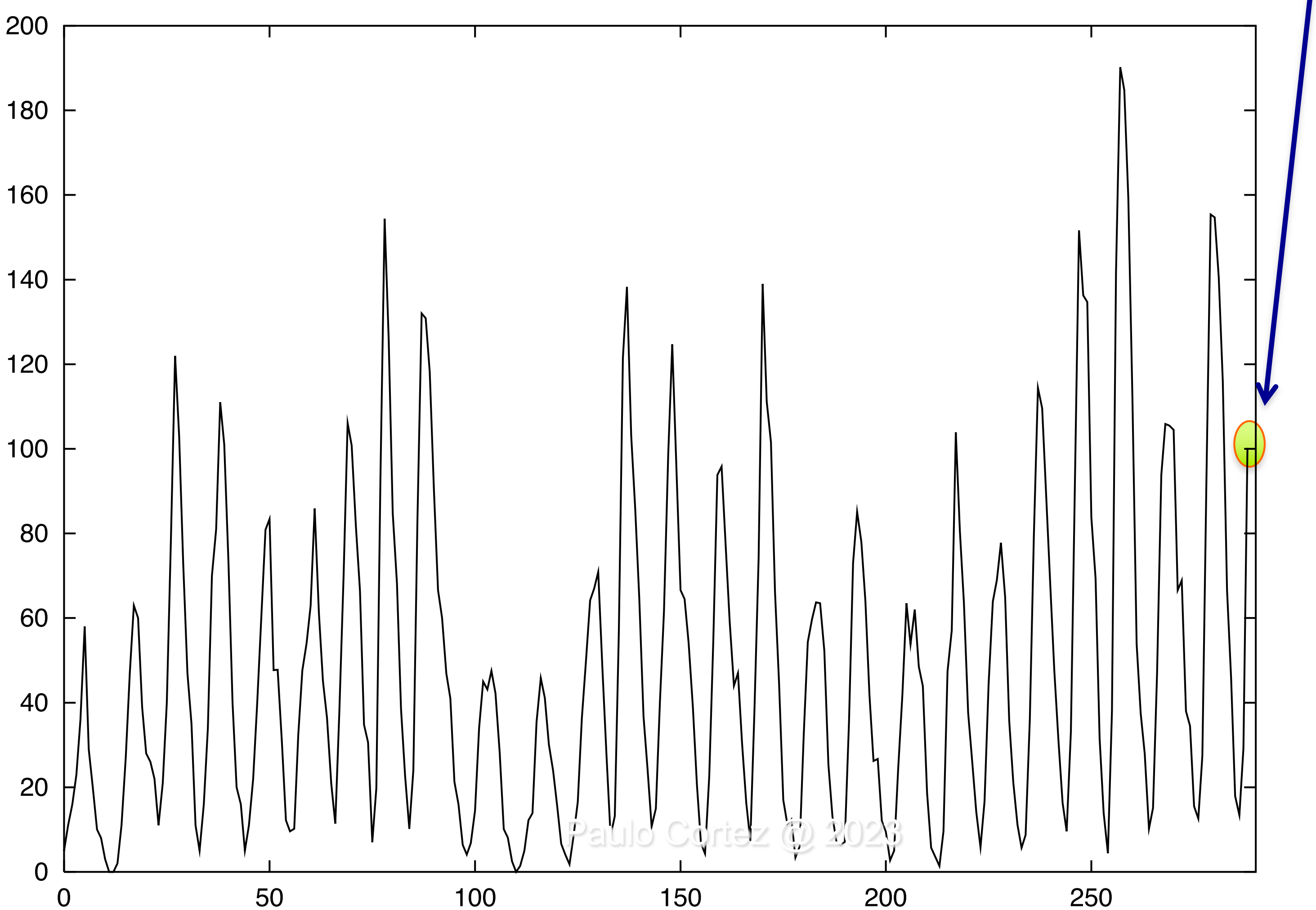
É obrigatório usar o número mecanográfico:

a12345

Q8

> Forecasting

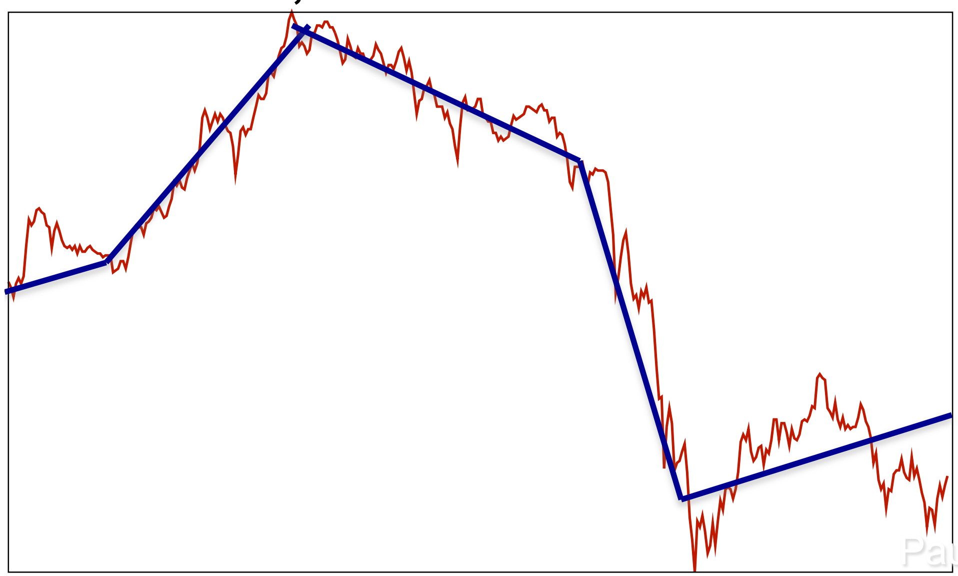
- Sunspots Time Series: 66.6, 45.9, 17.9, 13.4, 29.2,**100.2**

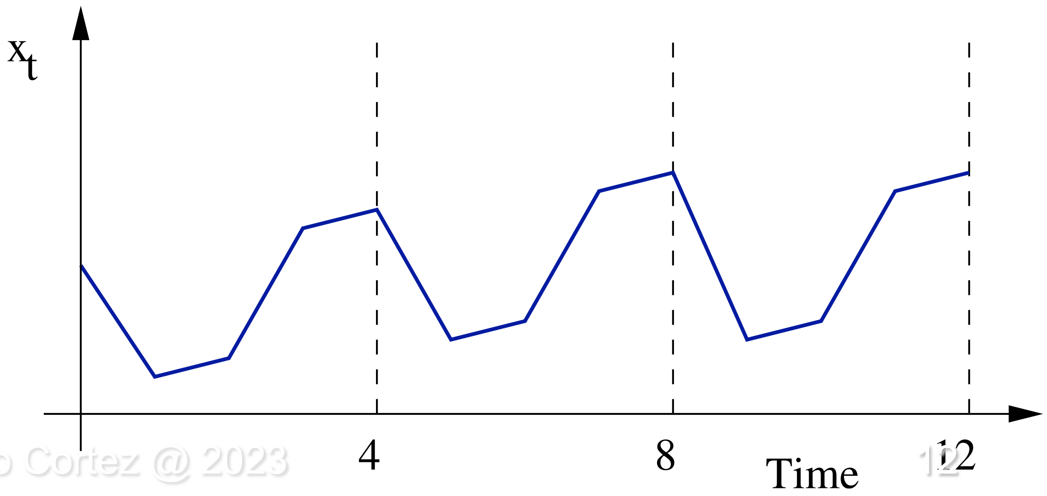


Methods

> Forecasting

- Predict the future based on past patterns;
- **Univariate Time Series**: uses data from a single phenomenon (one time series);
- **Multivariate Time Series**: uses data from several variables (several time series);
- Often, there are trend and seasonal cycles:





> Forecasting Scales:

- **Real-time**, which concerns samples not exceeding a few minutes and requires an on-line forecasting system;
- **Short-term**, from one to several hours, crucial for optimal control or detection of abnormal situations;
- **Middle-term**, typically from one to several days, used to plan resources; and
- **Long-term**, often issued several months/years in advance and needed for strategic decisions, such as financial investments.

> Lead Time (Horizon):

- The number of N-lookahead predictions;
- Example: 1, 2, 3, 4, **5 <- current day**
  - 1-lookahead forecast – 6
  - 2-lookahead forecast – 7
  - 3-lookahead forecast – 8
  - …
- Typically, the error of a N-lookahead forecast is higher than a N-1-looahead prediction… (**chaos effect**!)

> Evaluation:

- **Metrics:**
  - **Any used in regression:** MAE, **NMAE**, RMSE, R2, …;
  - MAPE**: sum(|y-pred|)/y*N, SMAPE, MASE, …**
  - **Theil' sU**: **RMSE/RMSE_naive**

Symmetric Mean Absolute Percentage Error (SMAPE) and Mean Absolute Scaled Error (MASE) [34]:

$$\text{SMAPE} = \frac{1}{h} \sum_{t=T+1}^{T+h} \frac{|e_t|}{(|y_t| + |F_t|)/2} \times 100\%, \tag{10}$$

$$\text{MASE} = \frac{1}{h} \sum_{t=T+1}^{T+h} \frac{|e_t|}{\frac{1}{T-1} \sum_{j=2}^{T} |y_j - y_{j-1}|}, \tag{11}$$

where $e_t = y_t - F_t$ for $t = T+1, \ldots, T+h$, both belong to scale independent error measures, thus can be more easily used to compare methods across different time series.

> Evaluation:

uating the quality of the predictions. In this work, we selected an absolute error based metric, which is a common approach in the forecasting domain (Hyndman & Koehler, 2006). For instance, in (Armstrong, 2001; Armstrong & Collopy, 1992) it is argued that squared error metrics, such as Root Mean Square Error (RMSE), are not reliable due to their sensitivity to outliers and should be replaced by absolute error metrics when comparing across time series. We note that other related works also have adopted absolute error metrics, such as: Mean Absolute Error (MAE) (Deng et al., 2011) and Mean Absolute Percentage Error (MAPE) (Bollen et al., 2011; Deng et al., 2011; Mao et al., 2011).

Using any absolute error measure should lead to the same ranking differences when comparing distinct forecasting models, thus the particular choice of such measure affects mostly its range of values and interpretation. In this paper, we selected the Normalized Mean Absolute Error (NMAE) metric that is calculated as (Goldberg, Roeder, Gupta, & Perkins, 2001):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

$$NMAE = \frac{MAE}{y_H - y_L} \qquad (14)$$

> Evaluation:

When compared with other absolute based metrics, the **NMAE** presents several advantages:

- it is easier to interpret than MAE, since it expresses the error as a percentage of the full target scale. The lower the NMAE values, the better are the forecasts.
- it is scale independent.
- it does not contain the limitations of other scale independent measures (MAPE, SMAPE).

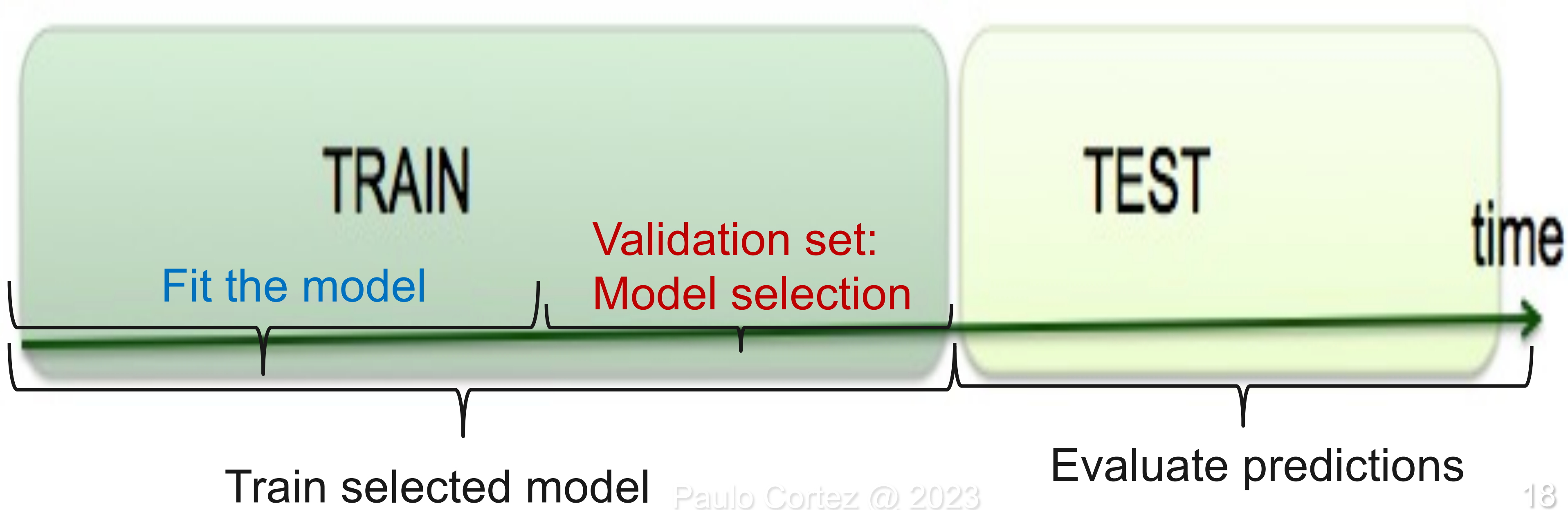
> Evaluation:

Time Ordered Holdout Split

Data split into train and test sets, according to a temporal split (not random)!

2/3 for Train (fit, validation) + 1/3 for Test

70% or 80% or …   30% or 20% or….



> Evaluation:

- Validation method:
  - Incremental retraining (growing windows)

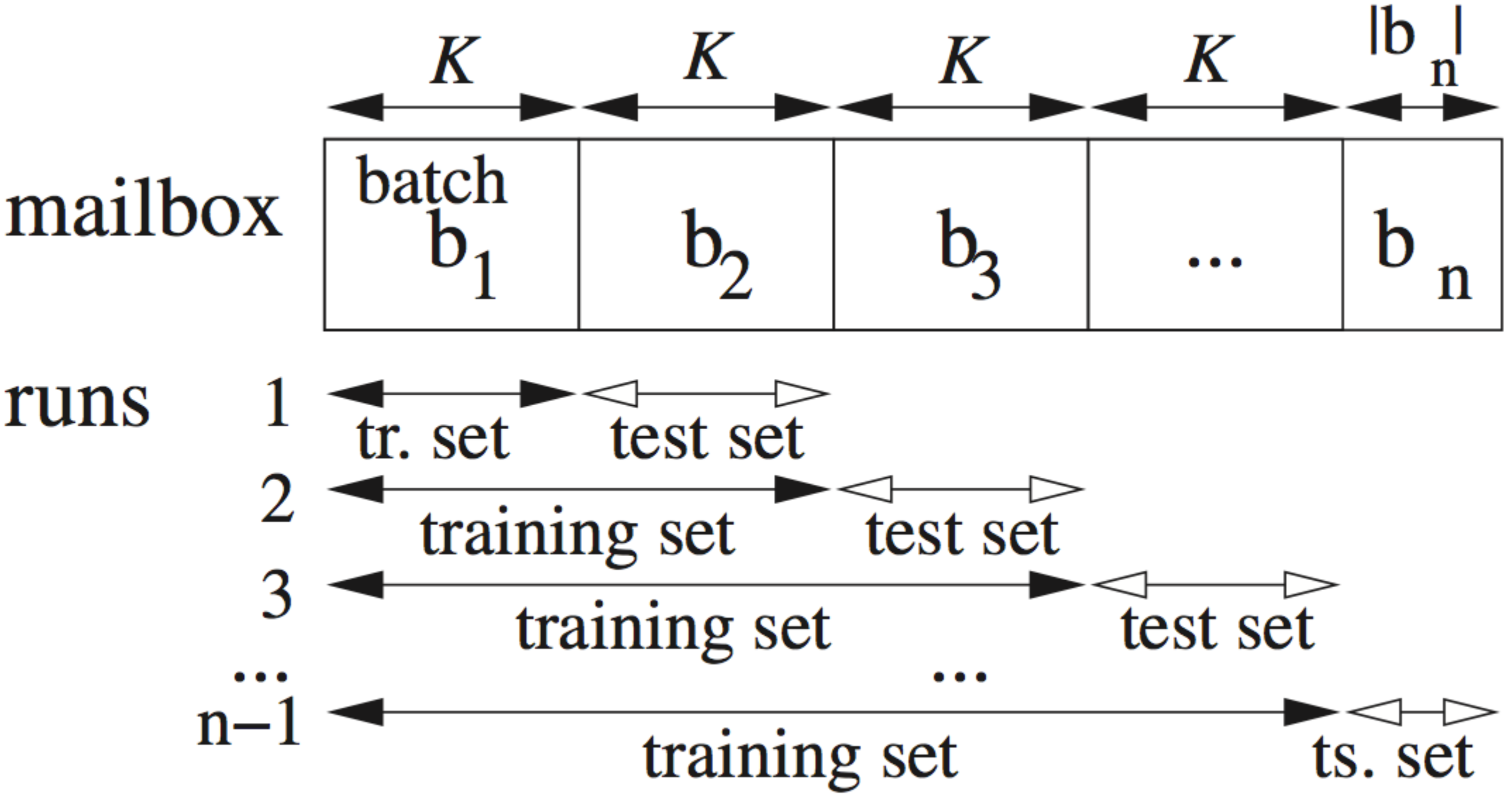
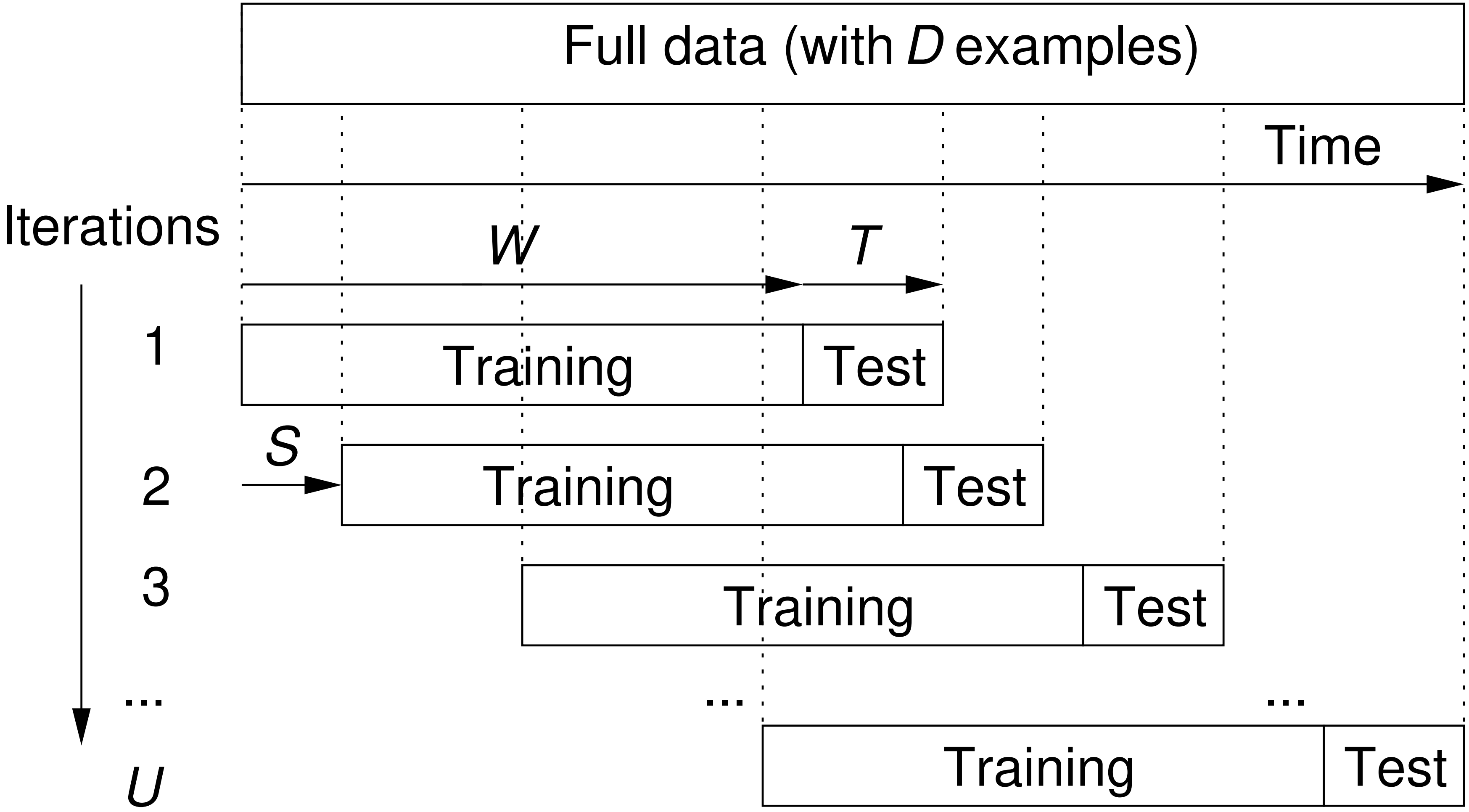


Fig. 4. The incremental retraining procedure.

Rolling Window



$$U = (D - (W + T))/S \text{ model updates}$$

> Simple Univariate Forecasting Methods

Time Series: 66.6, 45.9, 17.9, 13.4, 29.2, **?**

- **Naïve: 29.2**
- **Moving Average:**
    - **M=2: 21.3**
    - **M=5: 34.6**

> Univariate Forecasting Methods

- **Mathematical:**
  - Linear regression;
  - **Time Series:** Exponential Smoothing, ARIMA methodology, …
- **Modern Heuristics and Data Mining:**
  - Decision Trees;
  - Neural Networks;
  - Support Vector Machines;
  - Evolutionary Computing;
  - …

> Exponential Smoothing or Holt-Winters

- **Exponential Smoothing**: extension of the moving average, simple method that encompasses trend and cycle effects

$$
\begin{array}{lll}
\text{Level} & S_t = \alpha \frac{y_t}{D_{t-K_1}} + (1-\alpha)(S_{t-1} + T_{t-1}) & \\
\text{Trend} & T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1} & (4)\\
\text{Seasonality} & D_t = \gamma \frac{y_t}{S_t} + (1-\gamma)D_{t-K_1} & \\
 & \hat{y}_{t+h,t} = (S_t + hT_t) \times D_{t-K_1+h} &
\end{array}
$$

where S_t, T_t and D_t stand for the level, trend and seasonal estimates, K_1 for the seasonal period, and α, β and γ for the model parameters. When there is no seasonal component, the γ is discarded and the D_{t-K_1+h} factor in the last equation is replaced by the unity.

> Exponential Smoothing or Holt-Winters

- **Exponential Smoothing**: there is an updated version (Taylor, 2003) that handles two seasonal cycles (e.g., daily and weekly; weekly and monthly):

$$
\begin{array}{ll}
\text{Level} & S_t = \alpha \frac{y_t}{D_{t-K_1} W_{t-K_2}} + (1-\alpha)(S_{t-1} + T_{t-1}) \\
\text{Trend} & T_t = \beta(S_t - S_{t-1}) + (1-\beta) T_{t-1} \\
\text{Seasonality 1} & D_t = \gamma \frac{y_t}{S_t W_{t-K_2}} + (1-\gamma) D_{t-K_1} \\
\text{Seasonality 2} & W_t = \omega \frac{y_t}{S_t D_{t-K_1}} + (1-\omega) W_{t-K_2} \\
 & \hat{y}_{t+h,t} = (S_t + hT_t) \times D_{t-K_1+h} W_{t-K_2+h}
\end{array} \qquad (5)
$$

> Forecasting Methods

- **ARIMA methodology**: based on ARMA models (linear combination of past values and past errors), more sophisticated, requires several steps, such as model identification, validation,…

The non-seasonal model is denoted by the form $ARIMA(p, d, q)$ and is defined by the equation

$$\phi_p(L)(1-L)^d y_t = \theta_q(L)e_t \qquad (6)$$

where y_t is the series; e_t is the error; L is the lag operator (e.g. $L^3 y_t = y_{t-3}$); $\phi_p = 1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p$ is the AR polynomial of order p; d is the differencing order; and $\theta_p = 1 - \theta_1 L - \theta_2 L^2 - \ldots - \theta_q L^q$ is the MA polynomial of order q.

> Forecasting Methods

Differencing, autoregressive, and moving average components make up a non-seasonal ARIMA model which can be written as a linear equation:

$$Y_t = c + \phi_1 y_{d\ t-1} + \phi_p y_{d\ t-p} + \dots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t$$

where yd is Y differenced d times and c is a constant.

ARIMA(1,0,1) it is simply $Y_t = rY_{(t-1)} + e_t + ae_{(t-1)}$

> Forecasting Methods

A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$\text{ARIMA} \quad \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}} \quad \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part of} \\ \text{of the model}}}$$

where $m =$ number of observations per year. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model.

The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period. For example, an ARIMA(1,1,1)(1,1,1)$_4$ model (without a constant) is for quarterly data ($m = 4$), and can be written as

$$(1 - \phi_1 B)\ (1 - \Phi_1 B^4)(1 - B)(1 - B^4) y_t = (1 + \theta_1 B)\ (1 + \Theta_1 B^4)\varepsilon_t.$$

The additional seasonal terms are simply multiplied by the non-seasonal terms.

> Data Mining Methods

- **Any (supervised learning) regression method**: provided a sliding time window is used to build the training cases (examples).
- Example: series 1,2,3,4,5,6…
  - Sliding window <1,3> (past time lags):

    1,3 -> 4

    2,4 -> 5

- **Problem:** which sliding time window is best?

=> Feature Selection task

Open your smartphone browser
and go to
live.voxvote.com
enter the following 5 numbers
PIN: 78267

~~Voting is anonymous.~~

Screen name or

~~optio~~

É obrigatório usar o número mecanográfico:

a12345

Q9

> Sunspots Example in R

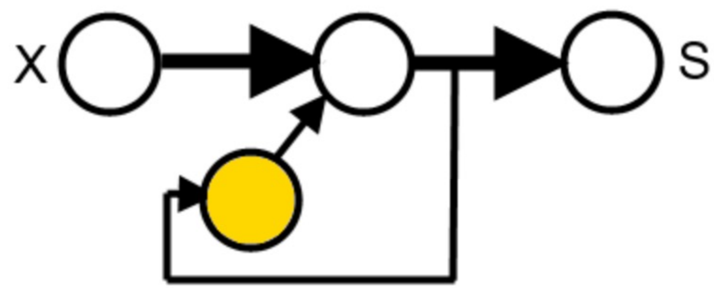
```
library(rminer) # by Paulo Cortez@
S=read.table("sunspots.ts",header=TRUE,sep=";")[,1]
D=CasesSeries(S,c(1:11),1,289) # 11 time lags
NN=fit (y~.,D[1:(nrow(D)-1),],model="mlpe",search=6)
P=predict(NN,D[nrow(D),])
print(P)
```

91.4

> Recurrent Models

No sliding time window is needed if the Machine Learning model has a temporal “memory”:

- **Jordan** and **Elman** Recurrent Neural Networks (RNN): http://software-tecnico-libre.es/en/article-by-topic/all_sections/all-topics/all-articles/recurrent-neural-network-and-time-series
- Reservoir computing
- **LSTM** deep learning model: https://www.r-bloggers.com/2018/04/time-series-deep-learning-forecasting-sunspots-with-keras-stateful-lstm-in-r/
- **Gated Recurrent Unit (GRU)**;
- …



Elman neural network

> multi-step h-ahead forecasting (horizon)

Statistical methods typically handle in a natural way h-ahead forecasting: Exponential Smoothing or Holt-Winters, ARIMA, ….

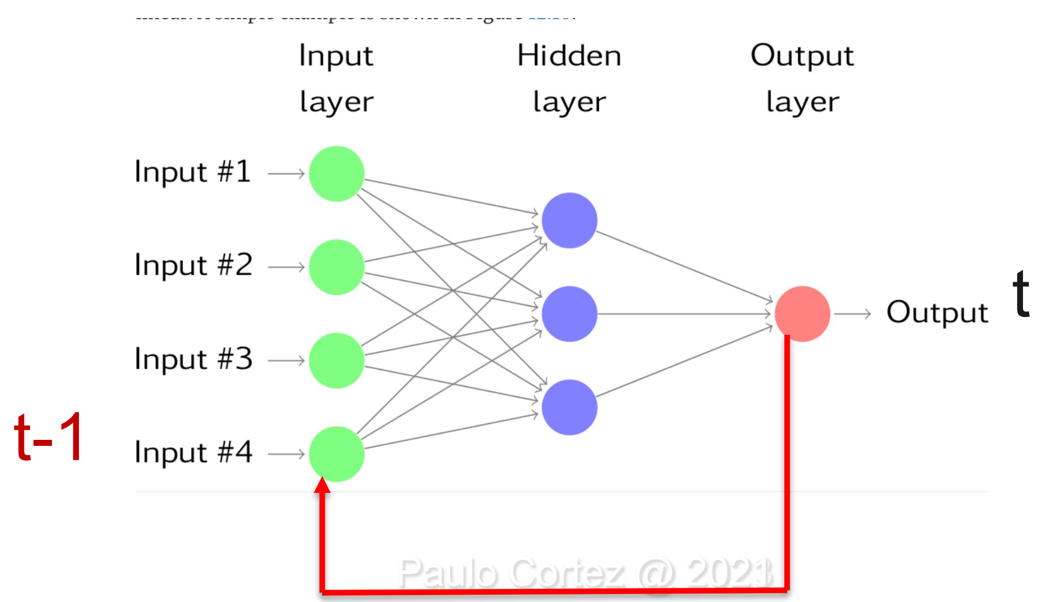
t

> multi-step h-ahead forecasting

Machine Learning methods (non-recurrent Neural Networks and others**):**

1. **if one output and 1 model** then use iterative feedback of previous outputs.

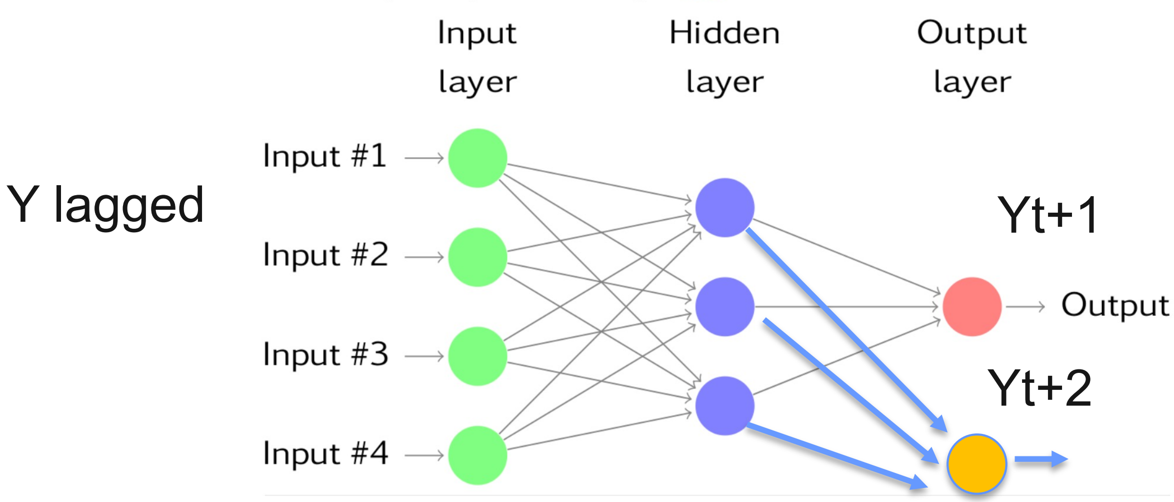
In the rminer package, the function **lforecast** performs this iterative feedback but only works with univariate time series



> multi-step h-ahead forecasting

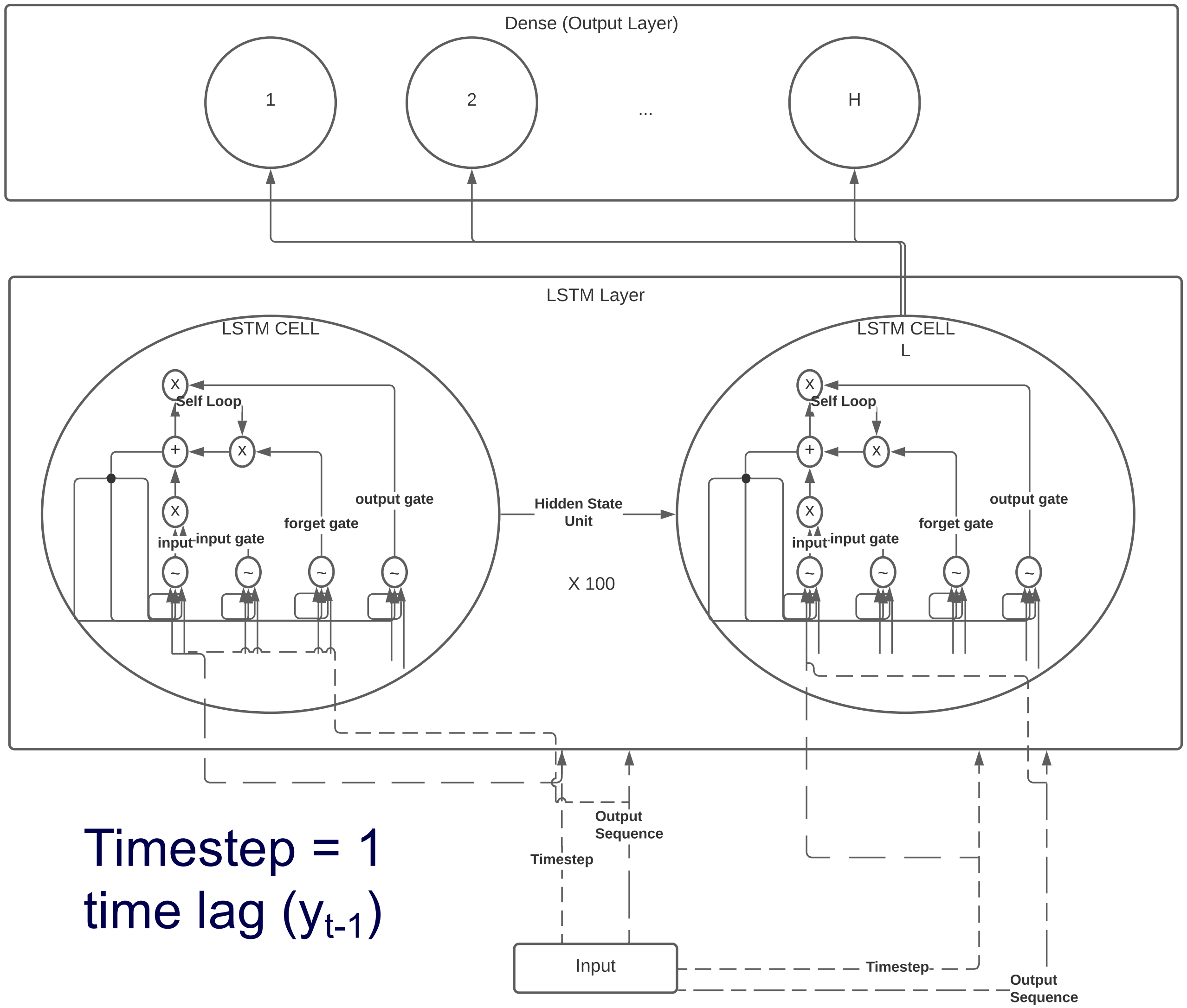
Machine Learning methods (non-recurrent Neural Networks and others**):**

2. One model for each h-ahead forecasting.
3. One model and multiple outputs (Multi-Target Regression).



> h-ahead forecasting

- **LSTM** deep learning model: natural H-ahead forecasting model



L = LTSM cells or units

Timestep = 1
time lag (y_{t-1})

> Multivariate Time Series

- **Vector Autoregressive (VAR):** https://otexts.com/fpp2/VAR.html

A VAR model is a generalisation of the univariate autoregressive model for forecasting a vector of time series.[22] It comprises one equation per variable in the system. The right hand side of each equation includes a constant and lags of all of the variables in the system. To keep it simple, we will consider a two variable VAR with one lag. We write a 2-dimensional VAR(1) as

$$y_{1,t} = c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + e_{1,t} \tag{11.1}$$

$$y_{2,t} = c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + e_{2,t}, \tag{11.2}$$

Estimates a VAR by OLS per equation. The model is of the following form:

$$\boxed{\mathbf{y}_t = A_1 \mathbf{y}_{t-1} + \ldots + A_p \mathbf{y}_{t-p} + CD_t + \mathbf{u}_t}$$

R **vars** package

where $\mathbf{y}_t$ is a $K \times 1$ vector of endogenous variables and u_t assigns a spherical disturbance term of the same dimension. The coefficient matrices $A_1, \ldots, A_p$ are of dimension $K \times K$. In addition, either a constant and/or a trend can be included as deterministic regressors as well as centered seasonal dummy variables and/or exogenous variables (term CD_T, by setting the `type` argument to the corresponding value and/or setting `season` to the desired frequency (integer) and/or providing a matrix object for `exogen`, respectively. The default for `type` is `const` and for `season` and `exogen` the default is set to `NULL`.

> Multivariate Time Series

- **ARIMAX:**

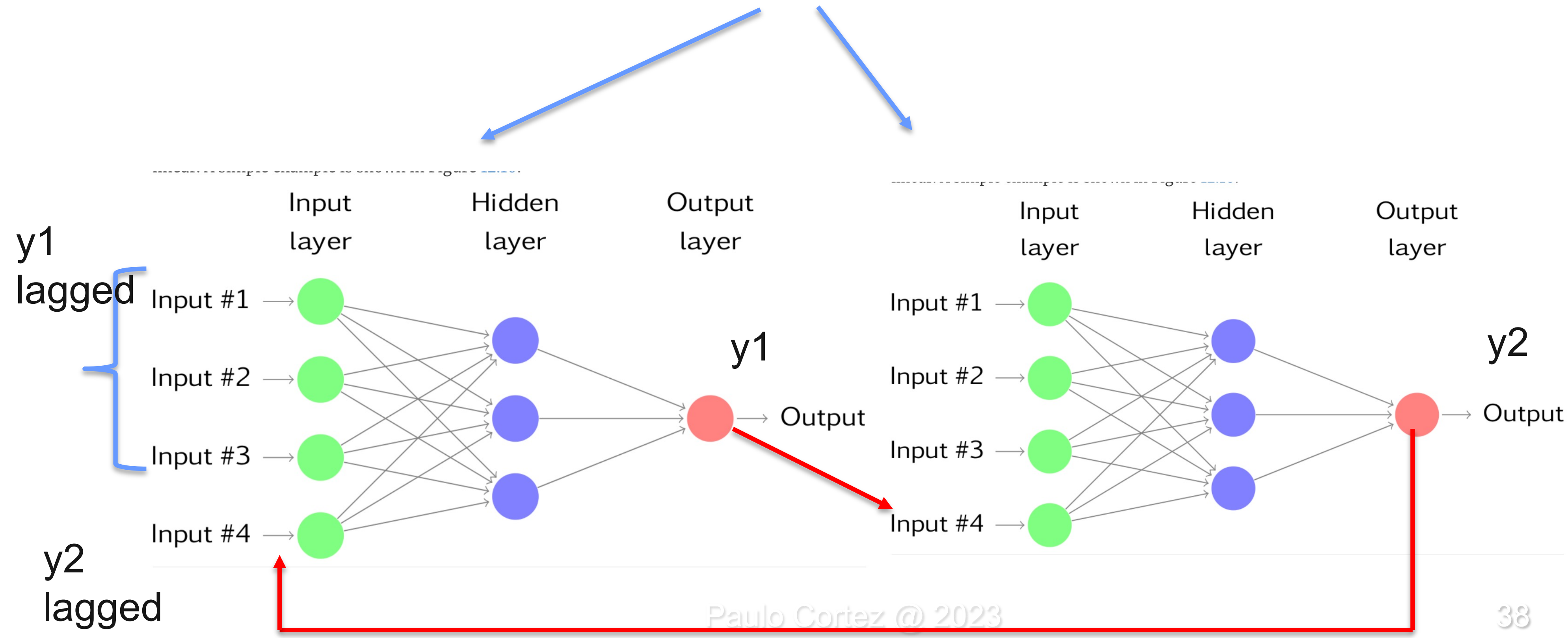
Autoregressive integrated moving average (ARIMAX) models extend ARIMA models through the inclusion of exogenous variables X. We write an $ARIMAX(p, d, q)$ model for some time series data y_t and exogenous data X_t, where p is the number of autoregressive lags, d is the degree of differencing and q is the number of moving average lags as:

$$\Delta^D y_t = \sum_{i=1}^{p} \phi_i \Delta^D y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \boxed{\sum_{m=1}^{M} \beta_m Xm, t} + \epsilon_t$$

$$\epsilon_t \sim N\left(0, \sigma^2\right)$$

> Multivariate Time Series

- **Neural Networks** and other **Machine Learning (ML)** algorithms: **one model per output** variable



> Multivariate Time Series

- **Neural Networks** and other **Machine Learning (ML)** algorithms: one model and multiple outputs.