DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Time Series

Francesco Marcelloni
Artificial Intelligence Group
IT2PAO Coordinator
Department of Information Engineering
Largo Lucio Lazzarino 1
56123 PISA

E-mail: francesco.marcelloni@unipi.it

1

University of Pisa

Francesco Marcelloni

Material

The slides are extracted from the book:
Marco Peixeiro: Time series forecasting in Python
Manning Publications Co.
2022

2

Machine Learning Approaches

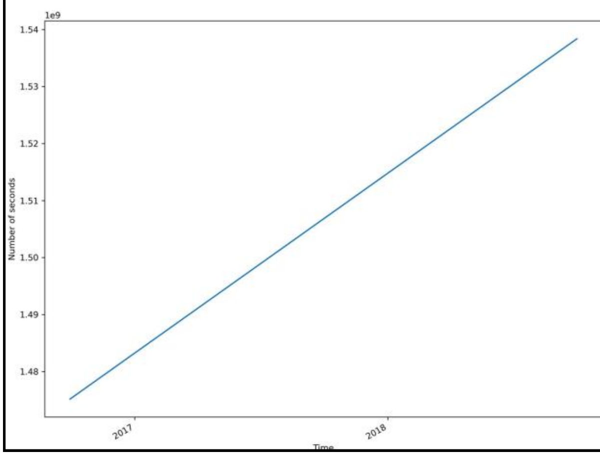
- **Statistical methods work particularly well when you have small datasets (less than 10000 data points). Otherwise, they become very slow**
- **Machine learning** is used either when statistical models take too much time to fit or when they result in correlated residuals that do not approximate white noise.
- Three models:
  - **Single-step model**: outputs a single value representing the prediction for the next timestep. The input can be of any length, but the output remains a single prediction for the next timestep.
  - **Multi-step model:** the output of the model is a sequence of values representing predictions for many timesteps into the future. For example, if the model predicts the next 6 hours, 24 hours, it is a multi-step model.
  - **A multi-output model** generates predictions for more than one target. For example, if we forecast the temperature and wind speed, it is a multi-output model.

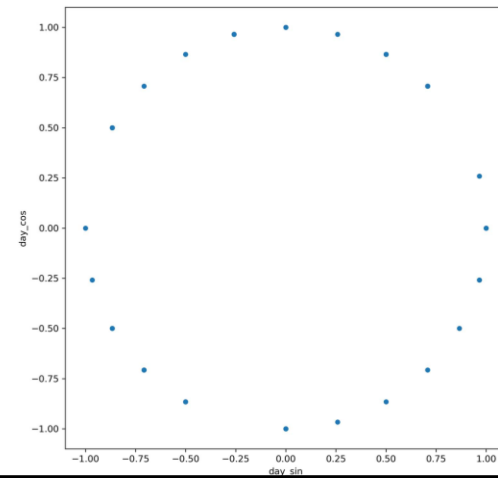
Machine Learning Approaches

- **How can we encode time expressed as YYYY-MM-DD HH:MM:SS?**
  - **Transform date in seconds**
  - **Apply a transformation to recover the cyclical behavior of time**
    - First, we apply a sine transformation
      - day = 24 * 60 * 60
      - df['day_sin'] = (np.sin(timestamp_s * (2*np.pi/day))).values
      - Using this transformation, 12 p.m. is equivalent to 12 a.m.
    - Second, we apply a cosine transformation to avoid the problem
      - df['day_cos'] = (np.cos(timestamp_s * (2*np.pi/day))).values

Machine Learning Approaches

The number of seconds linearly increase with time (left). After transformation, the time is encoded as a numerical value



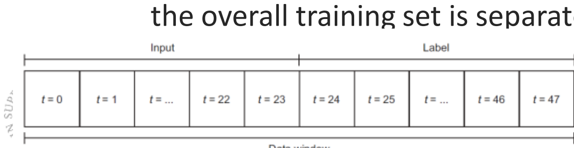


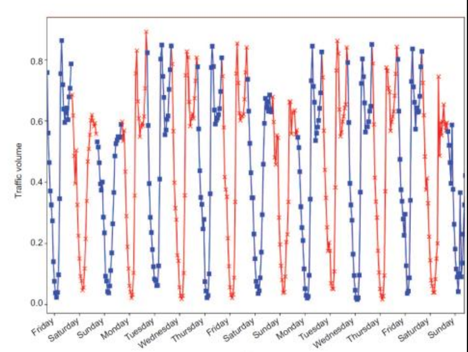
5

Machine Learning Approaches

We use a simple 70:20:10 split for the training, validation and test sets

- We scale all the values between 0 and 1 (this decreases the time for training deep learning models)
- **Data windowing process**: we define a sequence of data points on the time series and define which are inputs and which are labels
  - For example, we can adopt a window of 24 hours to predict the next 24 hours. Obviously, the overall training set is separated into multiple

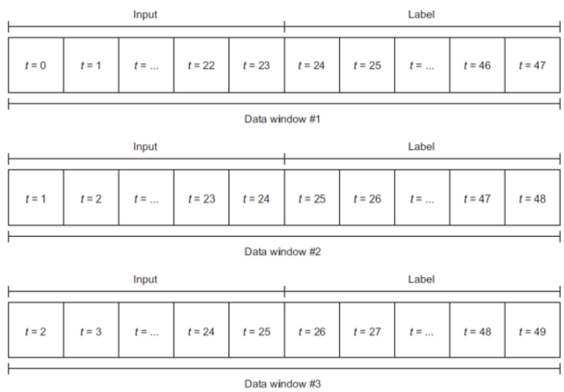




6

Machine Learning Approaches

- **Data windowing process**
  - With the approach presented in the previous slide, we waste a lot of training data. We can exploit moving windows with step 1

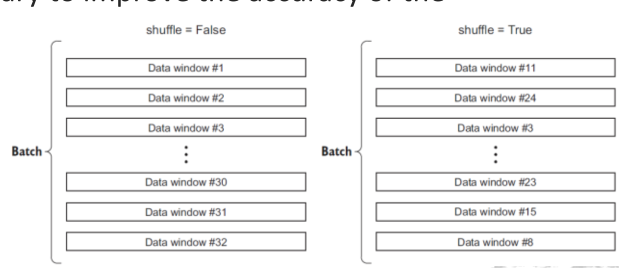


Machine Learning Approaches

- **Data windowing process**
  - Deep learning models are trained with **batches**. A batch is a collection of data windows that are fed to the model for training (update of the parameters after each batch). For example, the batch size is 32.
  - For a training set with 12285 rows, we have 384 batches.
  - Training the model on all the batches is called one epoch.
  - A lot of epochs are normally necessary to improve the accuracy of the predictions
  - **Shuffling** is used at the batch level

Loss function: mean squared error (MSE) (after each batch)

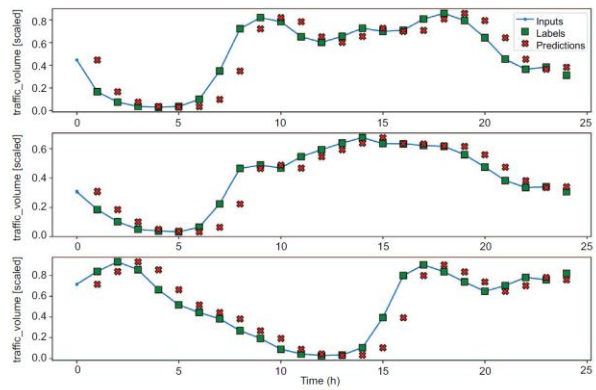
Evaluation metric: mean absolute error (MAE) (after each epoch)



Baseline models

- **Single-step baseline**
  - The input is one timestep and the output is the prediction of the next timestep
  - Window: **input width 1**, **label width 1**, **shift 1**.

Baseline: the simplest prediction we can make is the last observed value. Basically, the prediction is simply the input data point



9

Baseline models

- **Multi-step baseline**
  - For a multi-step of N, the input is N timesteps and the output is the prediction of N timesteps (for instance, N = 24 hours)
  - **Window**: input width N, label width N, shift N.

Two possible baselines

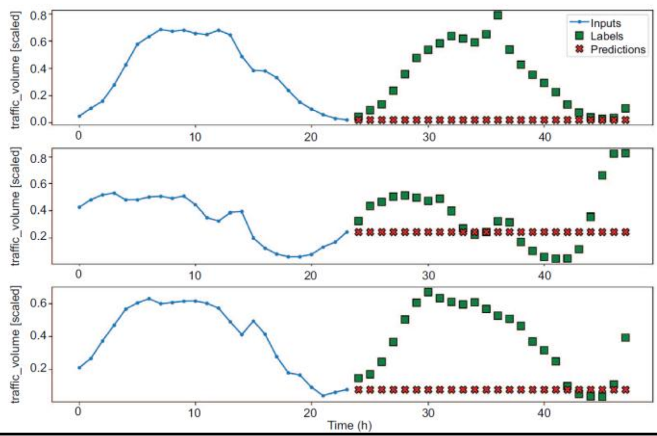
1. **Predict the last known value** for the next 24 timesteps.

2. **Predict the last 24 timesteps** for the next 24 timesteps.

10

Baseline models

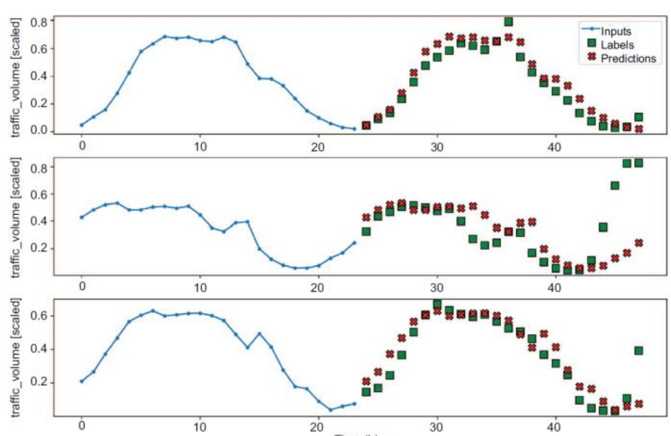
- **Multi-step baseline (predict the last known value)**
  - Simply takes in the input and repeats the last value of the input sequence over N time-steps



11

Baseline models

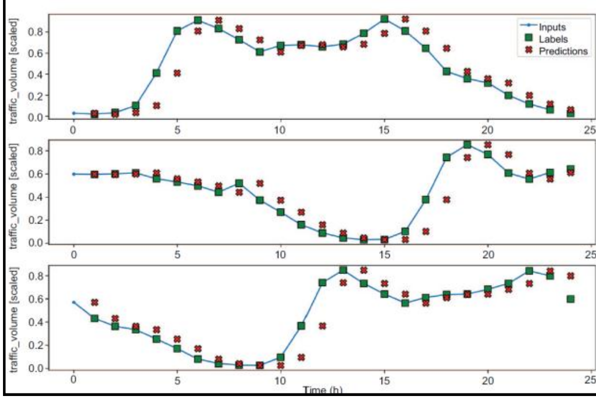
- **Multi-step baseline (repeating the input sequence of N timesteps)**
  - The prediction for the next N timesteps is simply the last known N timesteps of data

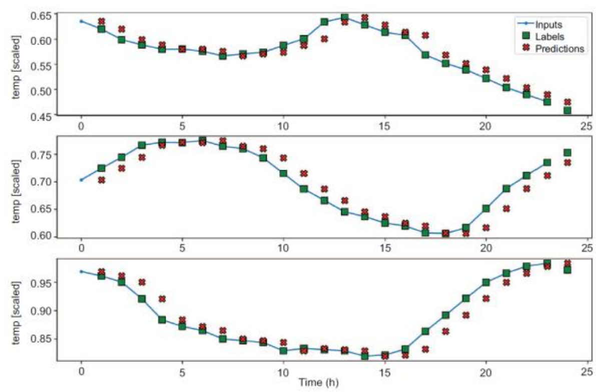


12

Baseline models

- **Multi-output baseline (single-step model for predicting different outputs)**
  - Let us assume that we have two outputs. We apply a single-step model for predicting both the outputs



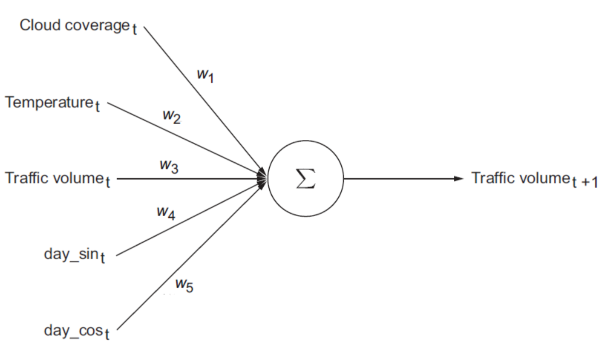


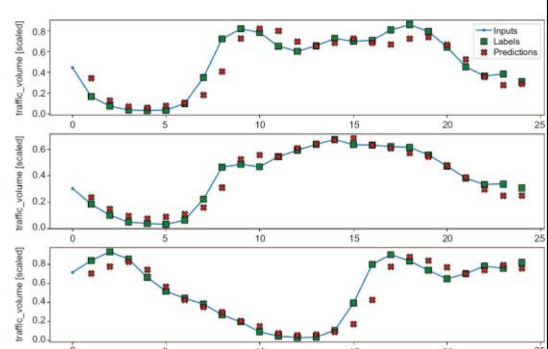
Deep Learning Models

- **Simple linear model**
  - Not actually a deep learning model: A simple multivariate linear regression

$$\text{traffic volume}_{t+1} = w_1 x_{1,t} + w_2 x_{2,t} + w_3 x_{3,t} + w_4 x_{4,t} + w_5 x_{5,t}$$

Single Step Mode





University of Pisa

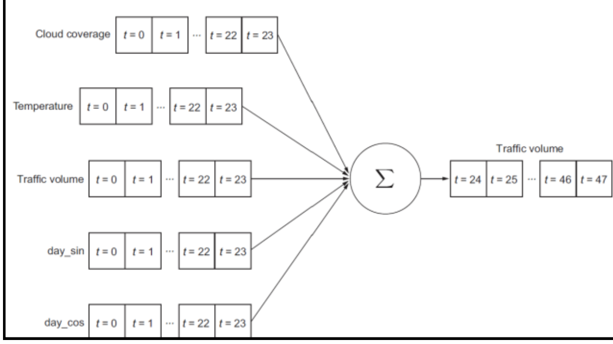
Francesco Marcelloni

Deep Learning Models

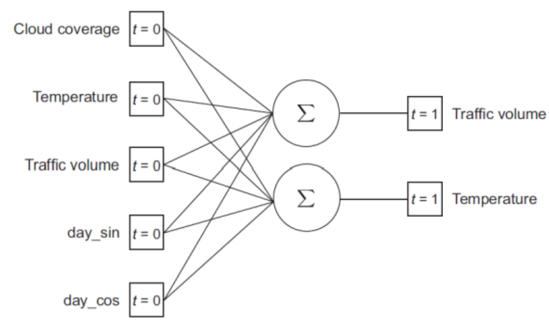
- **Simple linear model**
  - Not actually a deep learning model: A simple multivariate linear regression

$$\text{traffic volume}_{t+1} = w_1 x_{1,t} + w_2 x_{2,t} + w_3 x_{3,t} + w_4 x_{4,t} + w_5 x_{5,t}$$

Multi-Step linear Model



Multi-Output linear Model



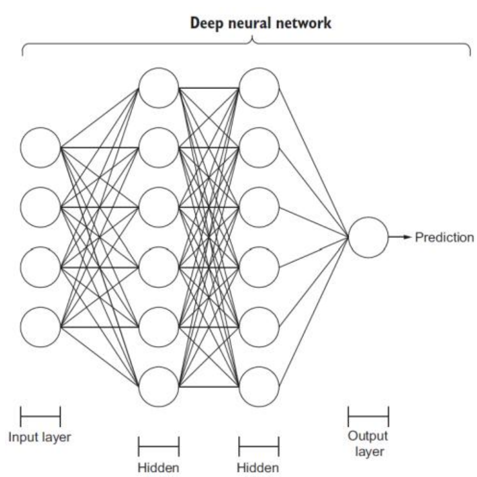
15

University of Pisa

Francesco Marcelloni

Deep Learning Models

- **Deep neural network**



Activation function: in each neuron of the hidden layers and responsible for generating the output.
If a linear activation function is used, the model will only model linear relationships.
Therefore, to model nonlinear relationships in the data, we must use a nonlinear activation function. Example: Rectified Linear Unit

$$f(x) = x^+ = \max(0, x)$$

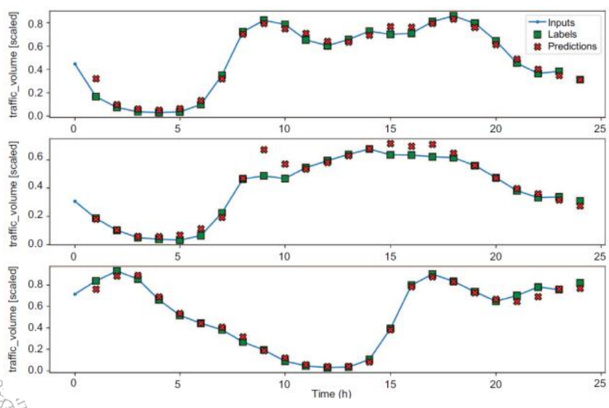
16

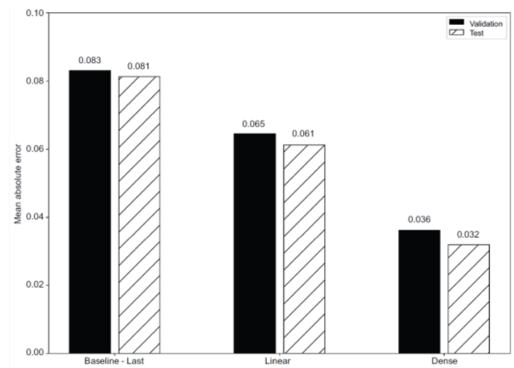
University of Pisa

Francesco Marcelloni

Deep Learning Models

- **Deep neural network (single-step model)**





17

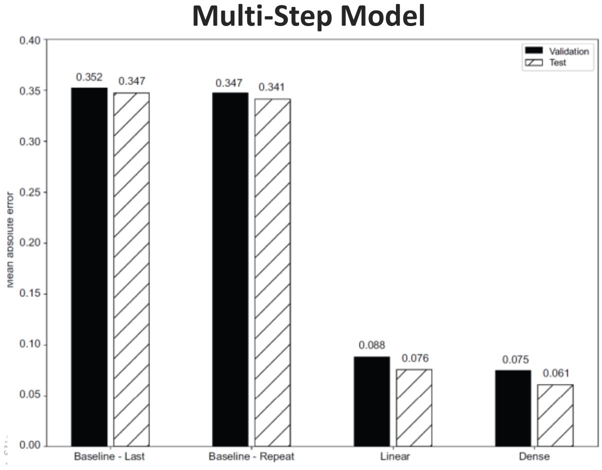
University of Pisa

Francesco Marcelloni

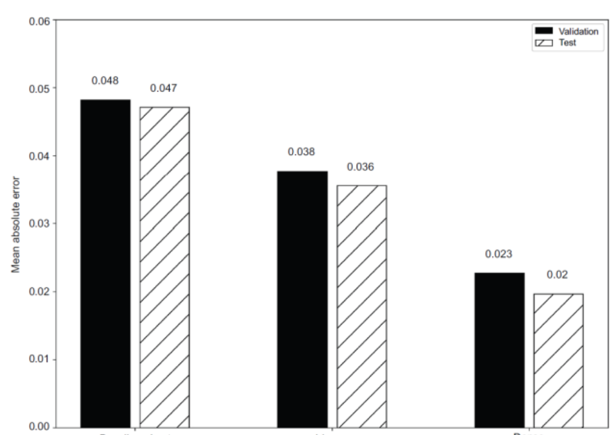
Deep Learning Models

- **Deep neural network**

Multi-Step Model



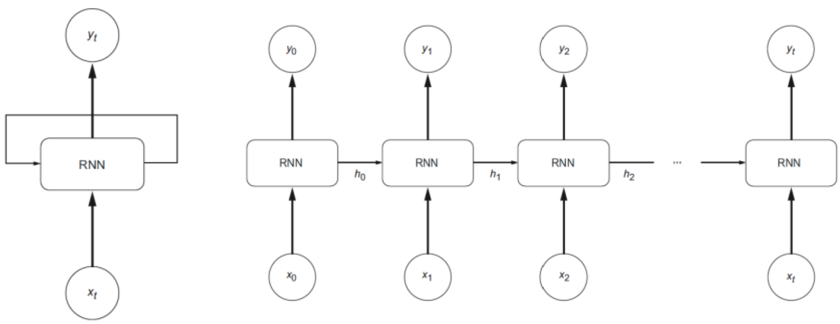
Multi-Output Model



18

Recurrent Neural Networks

- **A recurrent neural network (RNN) is a deep learning architecture especially adapted to processing sequences of data**
  - It uses a hidden state that is fed back into the network so it can use past information as an input when processing the next element of a sequence.
  - This is how it replicates the concept of memory.



RNNs suffer from short-term memory, meaning that information from an early element in the sequence will stop having an impact further into the sequence.

19

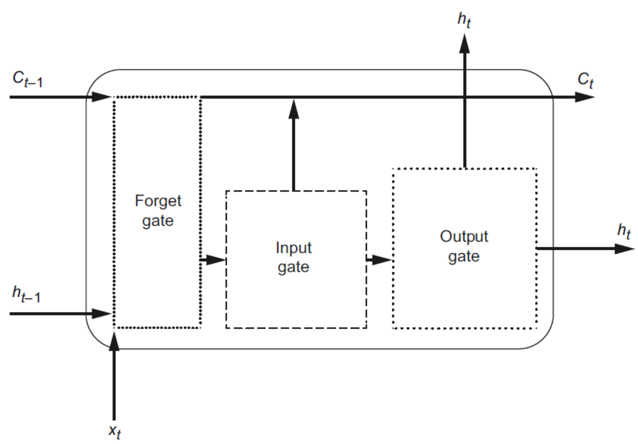
Long short-term memory

- **Long short-term memory**
  - Long short-term memory (LSTM) is a deep learning architecture that is a subtype of RNN. **LSTM addresses the problem of short-term memory by adding the cell state.** This allows for past information to flow through the network for a longer period of time, meaning that the network still carries information from early values in the sequence.
  - The LSTM is made up of three gates:
    - The **forget gate** determines what information from past steps is still relevant.
    - The **input gate** determines what information from the current step is relevant.
    - The **output gate** determines what information is passed on to the next element of the sequence or as a result to the output layer.

20

Long short-term memory

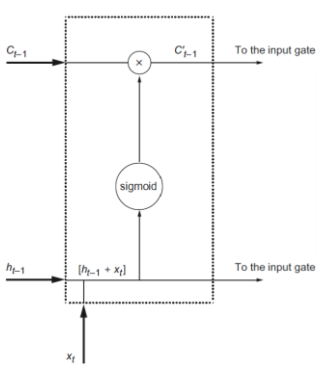
- **Long short-term memory**



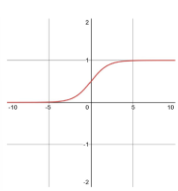
21

Long short-term memory

- **The forget gate**



$$f(x) = \frac{1}{1 - e^{-x}}$$



The present element of a sequence, x_t, and past information, h_{t-1}, are first combined. They are duplicated, and one is sent to the input gate while the other goes through the sigmoid activation function.

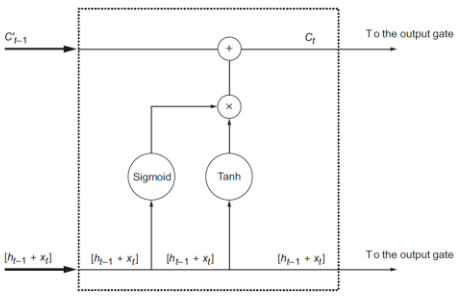
The **sigmoid outputs a value between 0 and 1**, and if **the output is close to 0**, this means that **information must be forgotten**. If **it is close to 1, the information is kept**. The output is then combined with the past cell state using pointwise multiplication, generating an updated cell state C'_{t-1}.

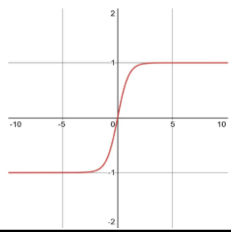
22

Long short-term memory

- **The input gate**

The **past hidden state and current element of the sequence are duplicated** again and sent through a sigmoid activation function and a hyperbolic tangent (tanh) activation function. Again, the sigmoid determines what information is kept or discarded, while the tanh function regulates the network to keep it computationally efficient.





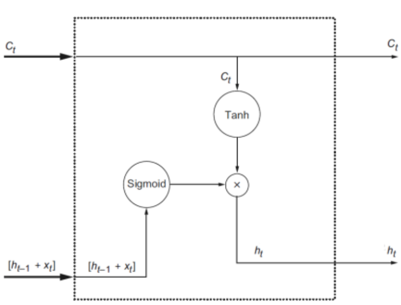
tanh function

23

Long short-term memory

- **The output gate**

The past hidden state and current element of a sequence $[h_{t-1} + x_t]$ are passed through the sigmoid function to determine if information will be kept or discarded. Then **the cell state is passed through the tanh function and combined with the output of the sigmoid using pointwise multiplication**. This is the step where past information is used to process the current element of a sequence. We then output a new hidden state h_t, which is passed to the next LSTM neuron or to the output layer. The cell state is also output.



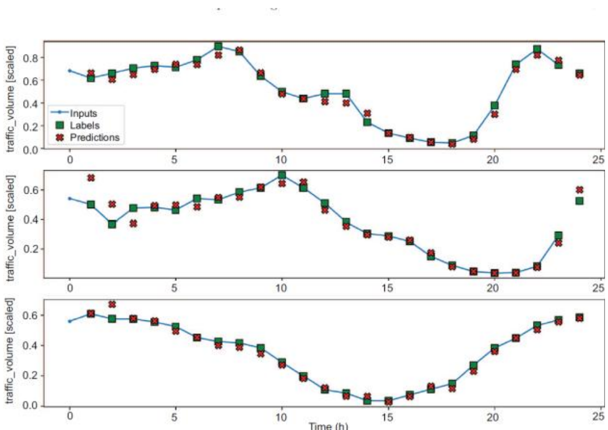
24

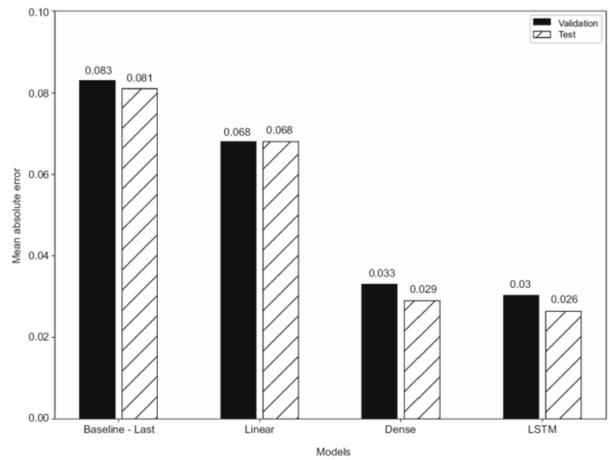
University of Pisa

Francesco Marcelloni

Long short-term memory

- **Single-step model:** we use N timestamps to predict the next timestamp





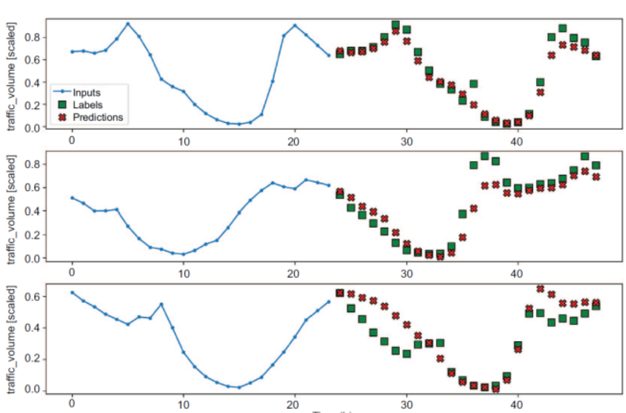
25

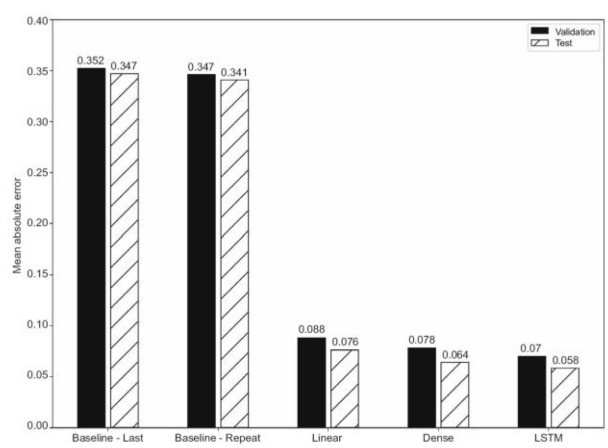
University of Pisa

Francesco Marcelloni

Long short-term memory

- **Multiple-step model:** we use N timestamps to predict N timestamps





26

Long short-term memory

- **Multi-output model:** we use N timestamps to predict one timestamp