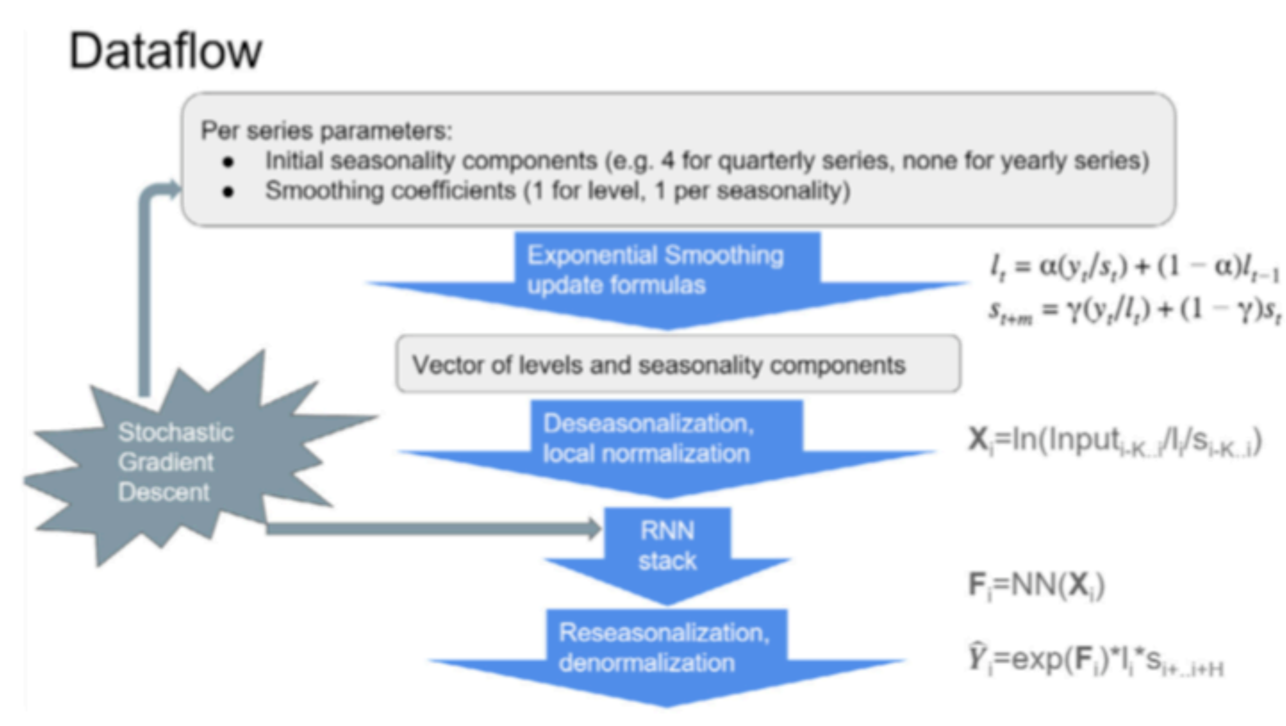
Introduction

- unlike standard statistical time series algorithms, where a separate model is developed for each series
- The strength of ML algorithms, and in fact the requirement for their successful use, is cross–learning, i.e., using many series to train a single model.
- NNs are particularly sensitive in this area

the most important ingredients

- on–the–fly preprocessing
- 预处理的常用做法：处理过程与预测过程相互独立，在 Normalization 和 deseasonality 之后进行预测
- 在 ESRNN 中，预处理参数由 ETS 模型确定，该参数和 NN 的参数一样，通过随机梯度下降进行更新（通过参数更新寻找最能够提高预测精度的预处理方法）

Methodology



三部分参数：

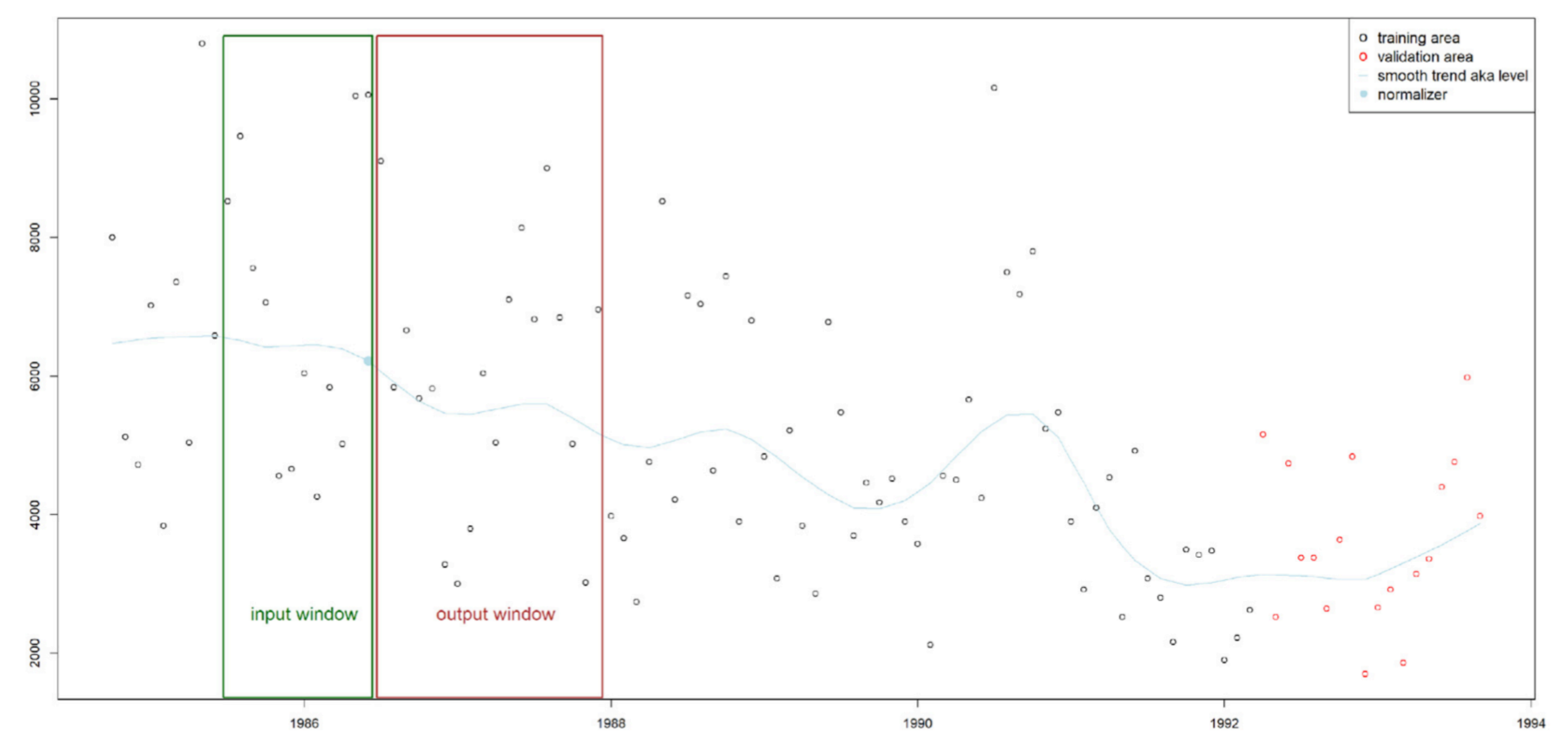
- Local constants：在每个 Epoch 中保持不变的常数，如 ETS 模型的参数
- Local states： 局部状态，随着滑窗改变的，如 level 和 seasonal component 以及 NN state
- Global constants：不随着滑窗而改变的全局参数，如 NN 的权重

ETS and pre–processing

$$l_t = \alpha y_t / s_t + (1-\alpha) l_{t-1}$$

$$s_{t+K} = \beta y_t / l_t + (1-\beta) s_t$$

对于每条时间序列，初始化（第一个 epoch）α 和 β 并计算平滑值，然后以滑窗的方式进行 denormalization 和 deseasonality，并将处理后的数据与其他的 feature(eg. category) 输入 NN，得到输出以后进行 renormalizaion 和 reseasonality

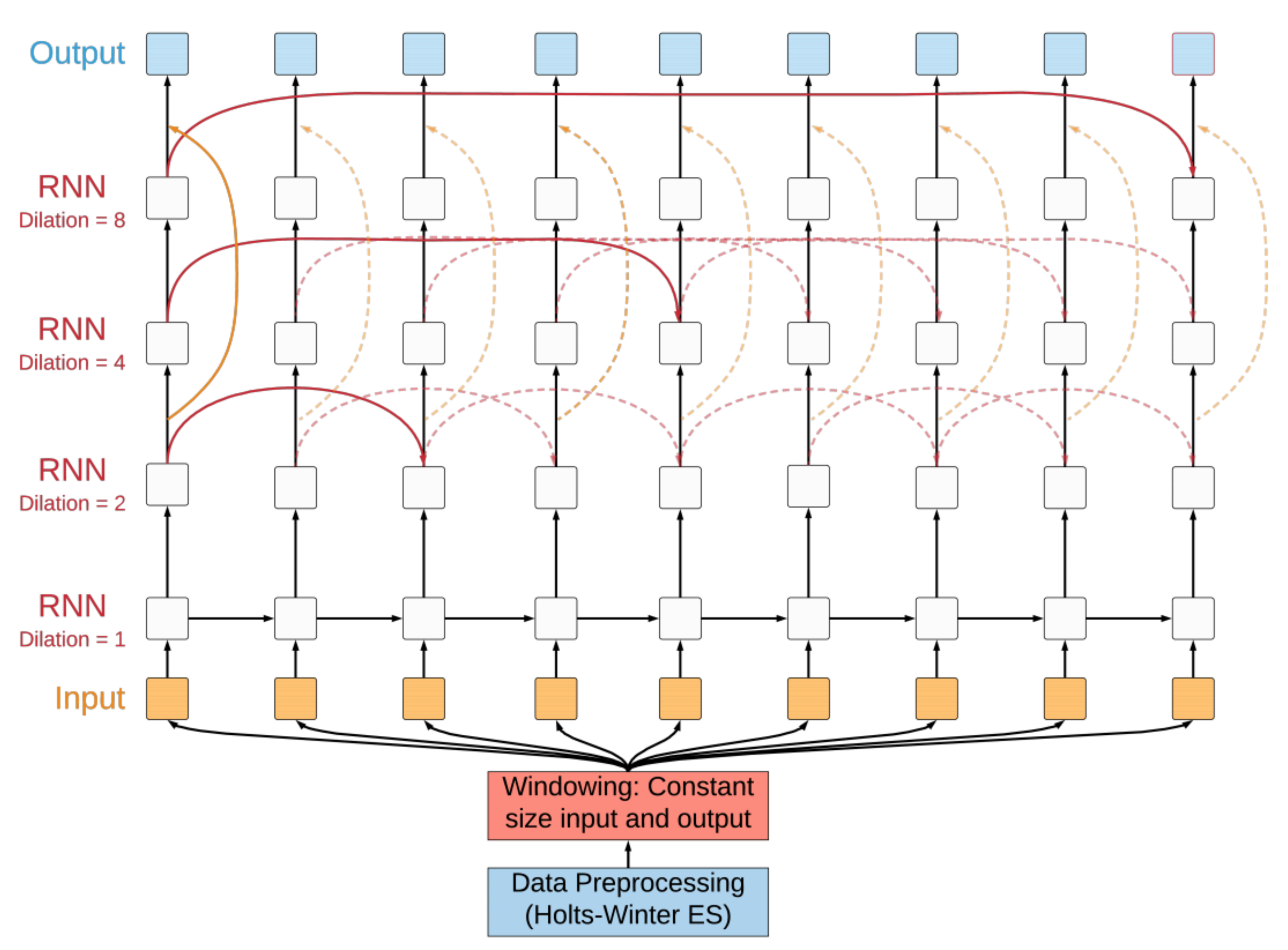


NN Architectures

NN 的输入维度： window_size + features_dim

NN 的输出维度： forecast_horizon

Table 1: Summary of network parameters



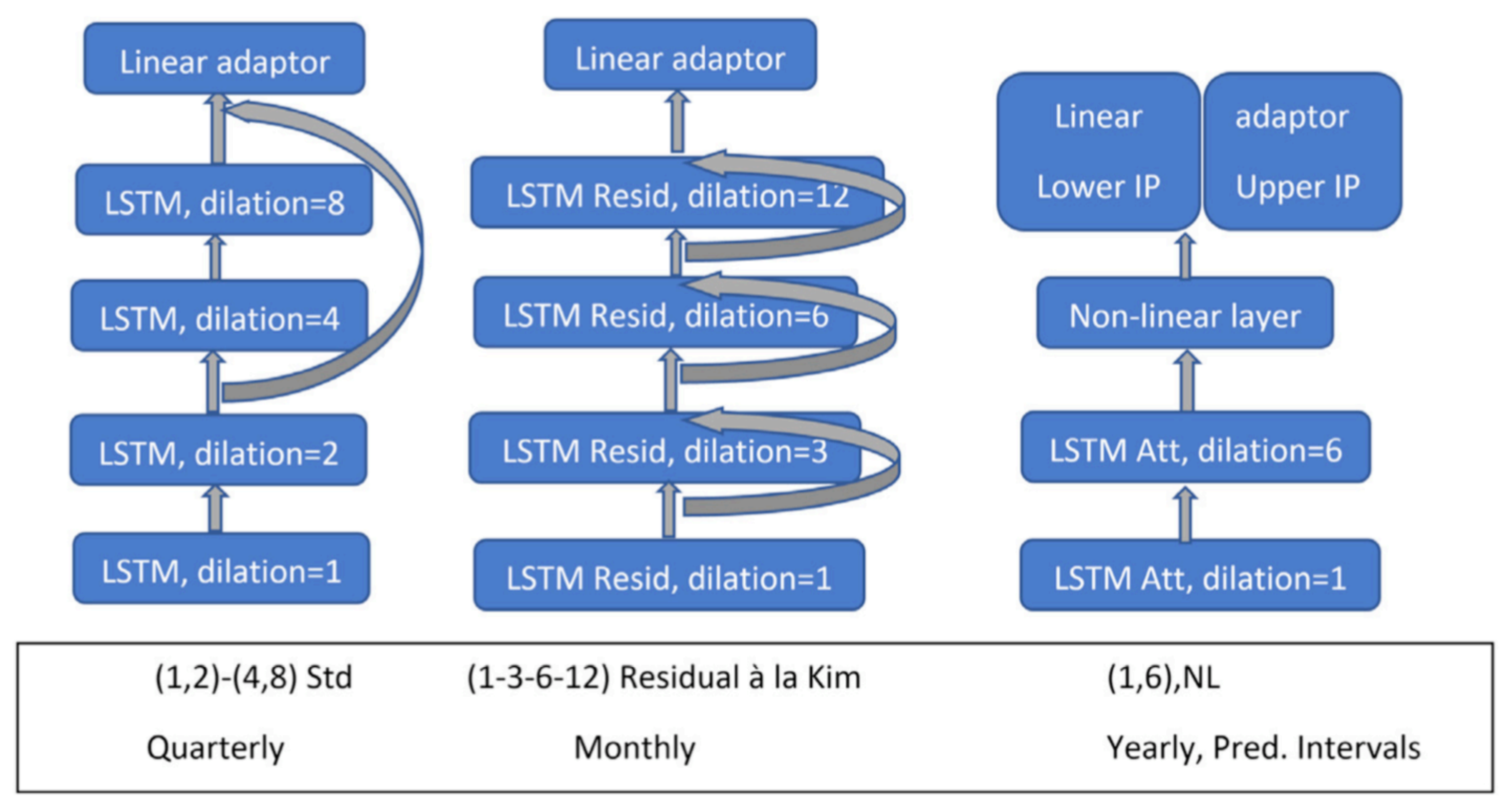


Fig. 3. NN architectures used for generating some of the PFs and PIs.

Other details

Pinball Loss

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

τ 通常介于 0.45–0.49 之间

原因：backtest 显示模型常常会有正偏差，在向 NN 中输入数据之前进行了 Log 变换，训练过程在 log space，但是计算预测误差确实在线性空间，导致预测值常常会有正偏差

Level wiggliness penalty

the smoothness of the level influenced the forecasting accuracy substantially.

It appears that when the input to the NN was smooth, the NN concentrated on predicting the trend, instead of over–fitting on some spurious, seasonality–related patterns.

A smooth level also means that the seasonality components absorbed the seasonality properly

- $d_t = log(y_{t+1}/y_y)$
- $e_t = d_t - d_{t-1}$
- Square and average them for each series.
- This penalty, multiplied by a constant parameter in the range of 50—100, called the level variability penalty (LVP), was added to both PFs and PIs loss function.

Ensembling and data subsetting

1. Create a pool of models (e.g. seven models) and randomly allocate a part (e.g. half of the time series) to each model.
2. For each model:
   (a) Execute a single training on the allocated subset.
   (b) Record the performance for the whole training set (in–sample, average over all points of the training part of a series).
3. Rank the models for each series and then allocate each series to the top N (e.g. two) best models.
4. Repeat steps 2 and 3 until the average error in the validation area starts growing.