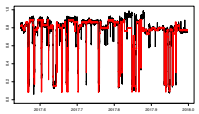
Electricity Consumption Optimization Based on Time-Series Forecasting

Junjie Hu*
Brenda López Cabrera*
Awdesch Melzer°*

*Ladislaus von Bortkiewicz Chair of Statistics
°Institute of Finance
Humboldt-Universität zu Berlin

Motivation

- Electricity load forecasting for chemical products
- Optimizing production schedules based on forecasting
- Better forecasts for day-ahead market reduce costs in short-term balancing demand and supply in spot market



Electricity Load

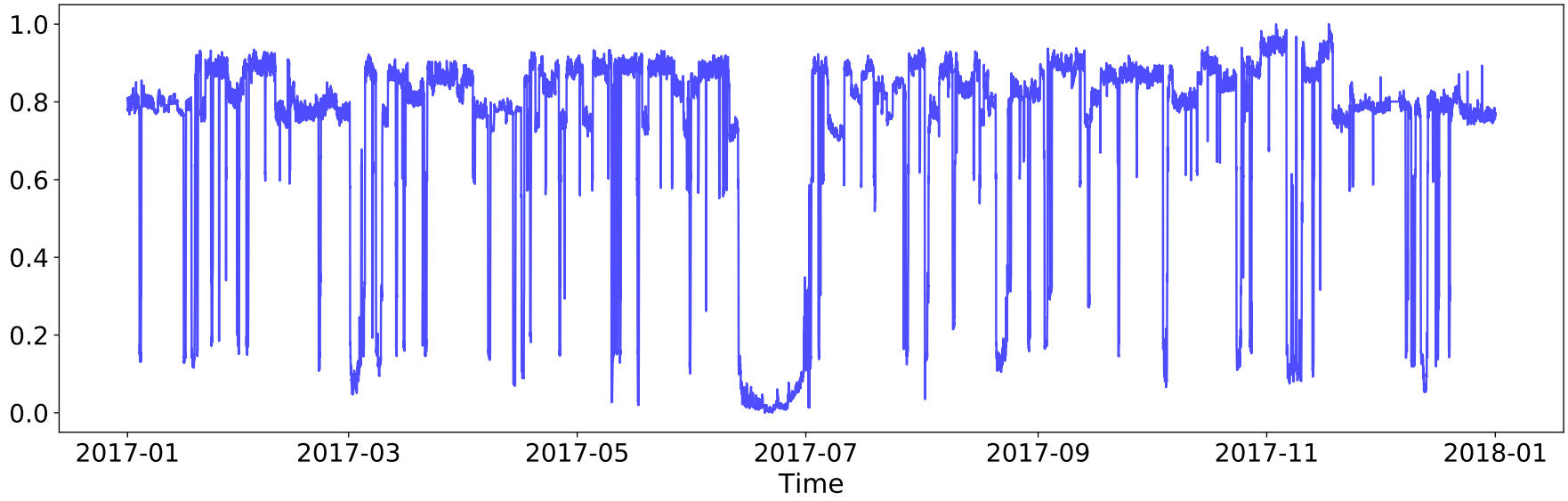
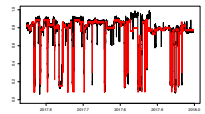


Figure 1: Electricity load curve with 15 min frequency in 2017, scaled into range $[0, 1]$ by transformation

- Electricity load curve for chemical products
- Load curve with jumps
- No periodic phenomenon exists



Production Schedules

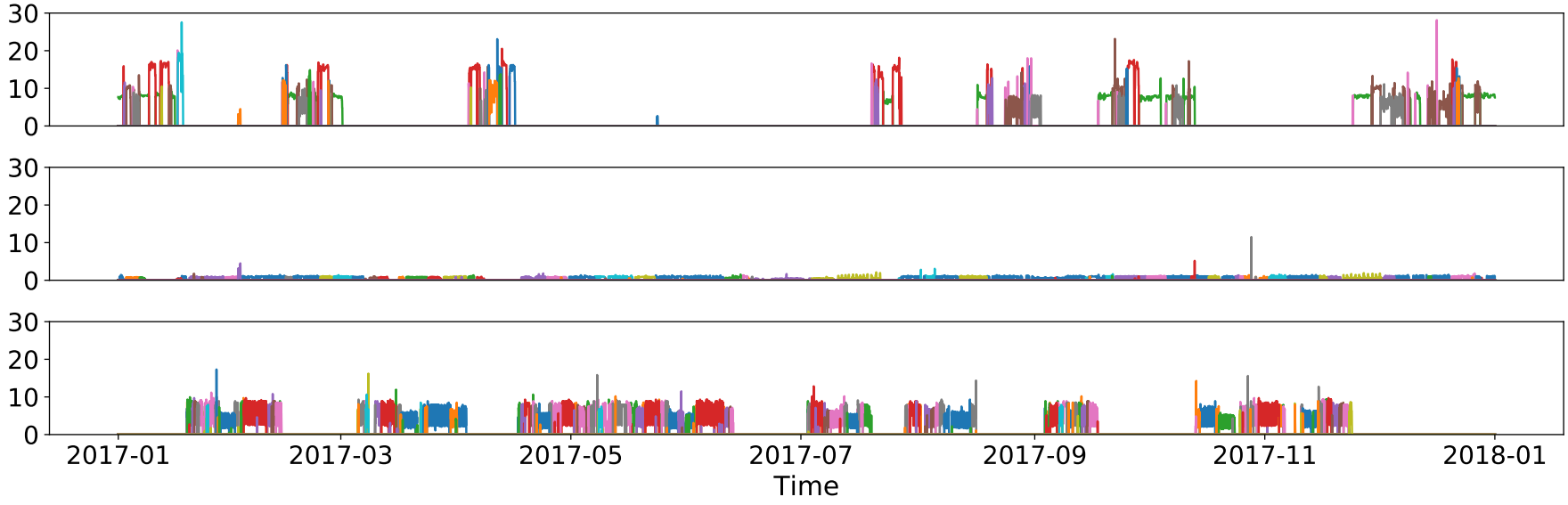
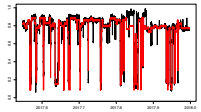


Figure 2: Totally 72 Production schedules variables can be treated as 3 independent groups



Distribution of Load

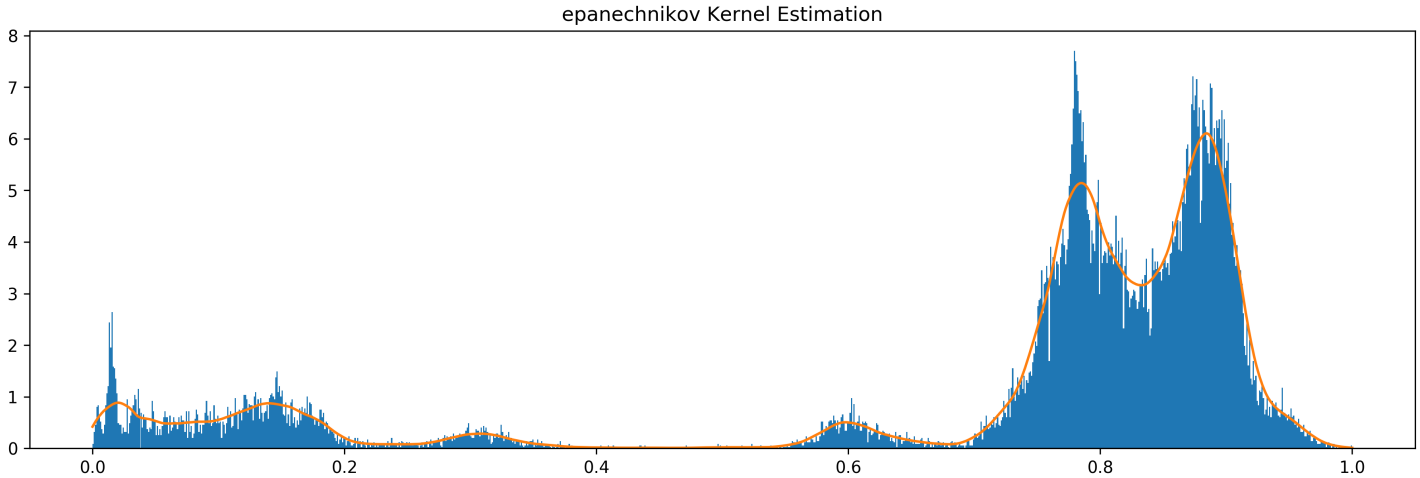
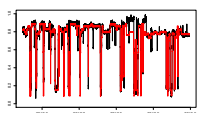


Figure 3: Epanechnikov Kernel Estimation of Load Distribution

- Estimated by Gaussian Mixture Model (GMM)
- Plugging GMM into forecasting model
- Analysis under probabilistic subspace



Analysis Framework

- Distribution of load can be expressed as linear combination of K normal distribution

$$f(y) = \sum_{k=1}^{K} \alpha_k \varphi\left(\mu_k, \sigma_k^2\right) \tag{1}$$

  Where μ_k, σ_k^2 are mean and variance of k_{th} normal distribution, y is load, $\sum_{k=1}^{K} \alpha_k = 1$

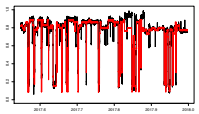
- In general, a forecasting model can be expressed as

$$\widetilde{y_t} = g\left(y_{t-i}, X_{t-j}\right), i = 1, \cdots, p; j = 0, \cdots, q \tag{2}$$

  Where X_{t-j} is the lagged j periods exogenous variables, $\widetilde{y_t}$ is the forecast load value as time t

- k_{th} probabilistic subspace $\varphi\left(\mu_k, \sigma_k^2\right)$ for forecast load value $\widetilde{y_t}$

$$k = \arg\max_{k \in K} \frac{\alpha_k \varphi\left(\widetilde{y_t} | \mu_k, \sigma_k^2\right)}{\sum_{k=1}^{K} \alpha_k \varphi\left(\widetilde{y_t} | \mu_k, \sigma_k^2\right)} \tag{3}$$



EM on GMM

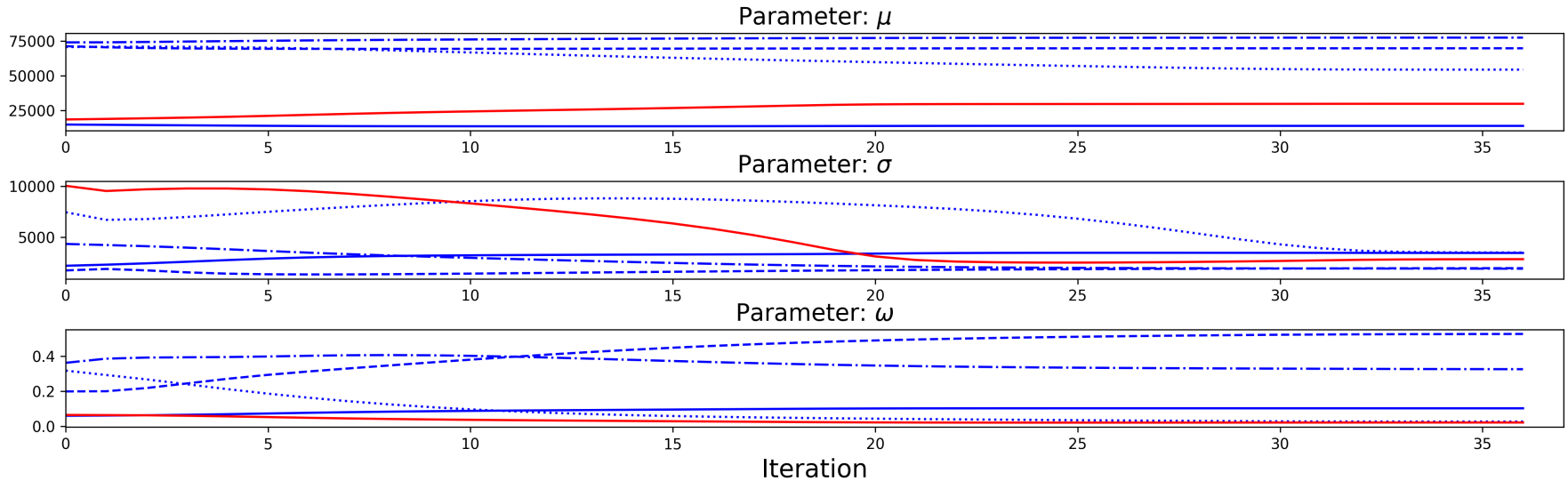
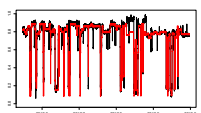


Figure 4: Solving GMM by EM

- Specify the number of distributions
- Maximize the Likelihood function for complete data
- Estimate the μ, σ and ω for each distribution



Load and Production Schedules

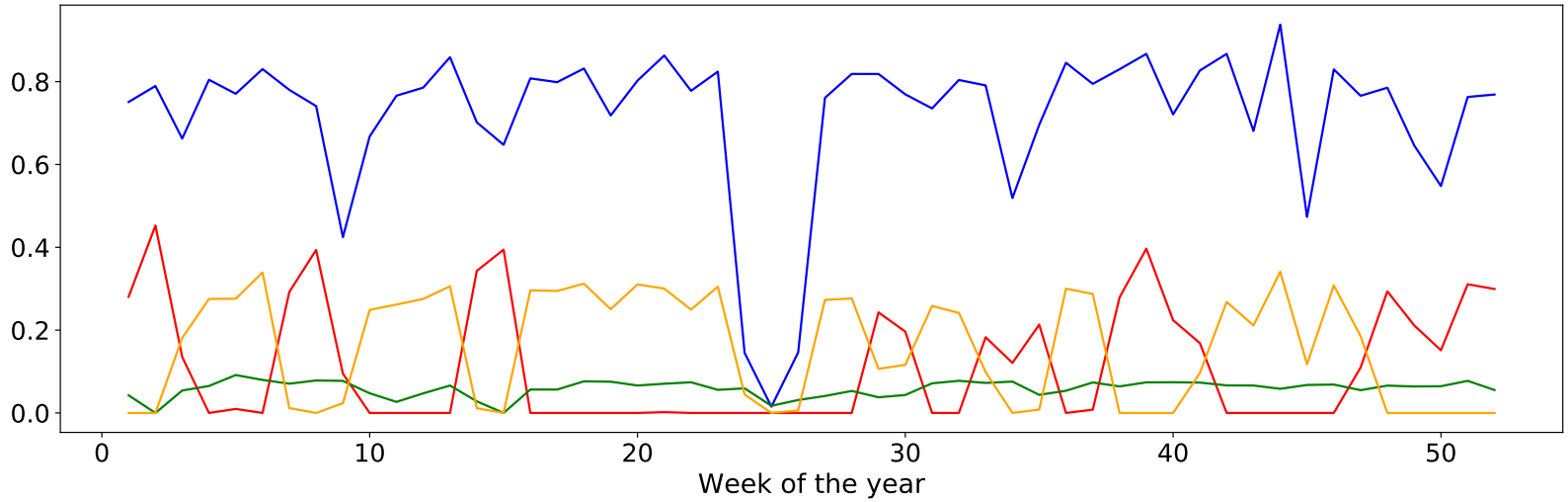
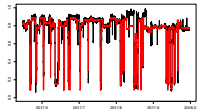


Figure 5: Time series with electricity load and exogenous variables in 3 groups, aggregated in terms of week of the year

- Fluctuation of electricity load mainly affected by group 1 and group 3



ACF - Long Memory Effect of Load Curve

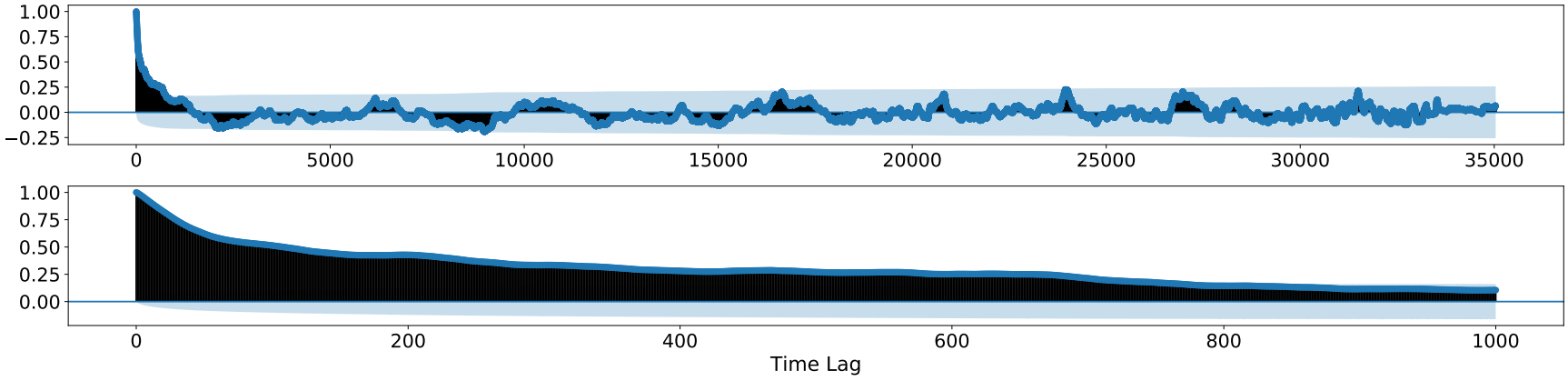
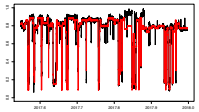


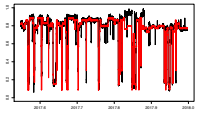
Figure 6: ACF plot of load curve. Upper plot with max lags = 35040, lower plot with max lags = 1000

- Under 95% confidence, the long memory effect lasts about 10 days



Outline

1. Motivation
2. Forecasting models
3. Forcasting Evaluation
4. Outlook
5. Appendix



Recurrent Neural Networks (RNN)

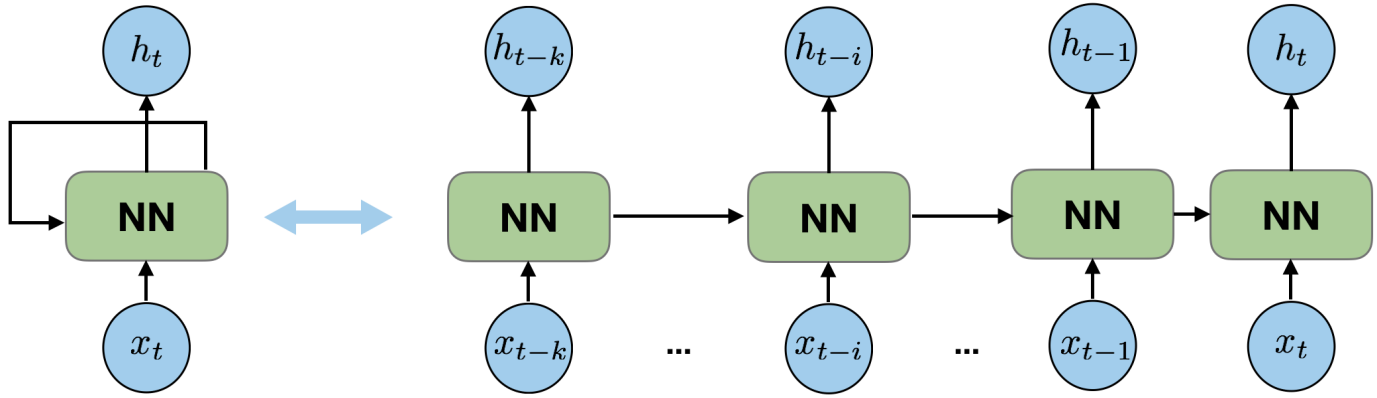
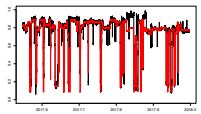


Figure 7: RNN is an architecture which has loops in it that allows the information persist

- Unfolding the RNN, it is constructed by multiple copied of the same neural network
- RNN is structured to handle sequences-related problems, for example, time-series forecasting
- However, RNN suffers from the famous "Gradient Exploding" and "Gradient Vanishing" problems



LSTM model

- Input layer

$$i_t = \sigma\left(\omega_i h_{t-1} + \mu_f x_t + b_i\right) \tag{4}$$

- Update layers

$$f_t = \sigma\left(\omega_f h_{t-1} + \mu_f x_t + b_f\right) \tag{5}$$

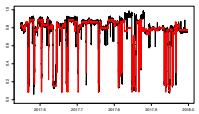
$$\widetilde{S}_t = tanh\left(\omega_s h_{t-1} + \mu_f x_t + b_s\right) \tag{6}$$

$$o_t = \sigma\left(\omega_o h_{t-1} + \mu_o x_t + b_o\right) \tag{7}$$

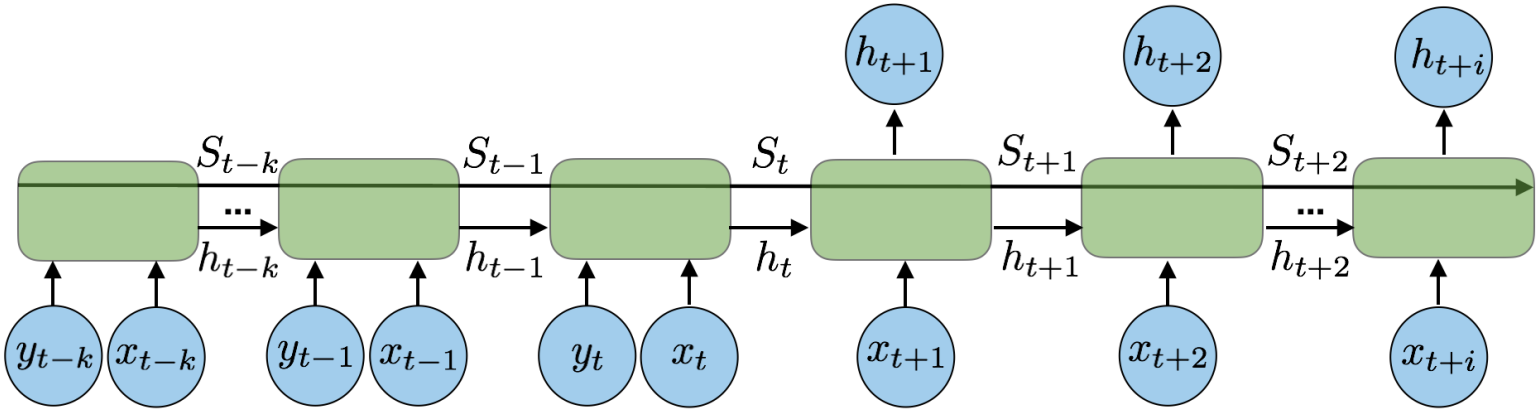
- Output layer

$$S_t = f_t \times S_{t-1} + i_t \times \widetilde{S}_t \tag{8}$$

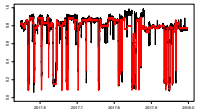
$$h_t = tanh(S_t) \times o_t \tag{9}$$



LSTM on Load Prediction



- Forecast the next 96 quarterly hour (and 1 day) load
- LSTM model memorizes the states along one training step
- Parameters are adjusted by Backpropagation Algorithm



FASTEC construction

- Data: $\{\mathbf{X}_i\}_{i=1}^n \in \mathbb{R}^p, \{\mathbf{Y}_i\}_{i=1}^n \in \mathbb{R}^m$ i.i.d.
- Linear model for τ-expectile curve of Y_j, $j = 1, ..., m, 0 < \tau < 1$:

$$Y_j = e_j(\tau|\mathbf{X}_i) + u_{ij,\tau} = \mathbf{X}_i^\top \Gamma_{*j}(\tau) + u_{ij,\tau}, \tag{10}$$

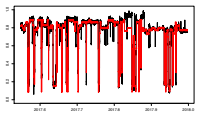
where coefficients for j response: $\Gamma_{*j}(\tau) \in \mathbb{R}^p$

- Sparse factorisation: $f_k^\tau(\mathbf{X_i}) = \varphi_{\mathbf{k}}^\top(\tau)\mathbf{X_i}$ factors

$$e_j(\tau|\mathbf{X}_i) = \sum_{k=1}^{r} \psi_{j,k}(\tau) f_k^\tau(\mathbf{X}_i), \tag{11}$$

where r : number of factors;
$\Gamma_{*j}(\tau) = (\sum_{k=1}^r \psi_{j,k}(\tau)\varphi_{k,1}(\tau), ..., \sum_{k=1}^r \psi_{j,k}(\tau)\varphi_{k,p}(\tau))$

FASTEC_with_Expectiles



MER formulation: penalised loss

$$\widehat{\Gamma}_\lambda(\tau) = \arg\min_{\Gamma \in \mathbb{R}^{p\times m}} \left\{ (mn)^{-1} \sum_{i=1}^{n}\sum_{j=1}^{m} \rho_\tau \left(Y_{ij} - \mathbf{X}_i^\top \Gamma_{*j} \right) + \lambda \|\Gamma\|_* \right\},$$

$\|\Gamma\|_* = \sum_{j=1}^{\min(p,m)} \sigma_j(\Gamma)$ nuclear norm of Γ

$\mathbf{X}_i$: B-splines

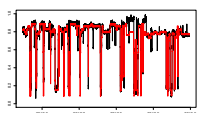
$\mathbf{Y}_i$:

Γ : factor matrix

λ : penalisation parameter (optimality via CV)

Chao et al. (2015), Härdle et al (2016)

▸ FISTA algorithm



Conditional moment based procedure

1. Aggregate exogenous variables: 3 groups X_1, X_2, X_3
2. Estimate conditional moments: certain expectiles as cluster centers
3. Determine affiliation based on t-statistic
4. Perform FASTEC (training)

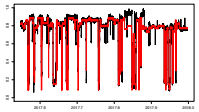
4.1 Select periods at conditional moments with 96 observations Y_τ

4.2 Bootstrap to get larger sample Y_τ^*

4.3 Perform FASTEC on each conditional moment sample

4.4 Predict conditional moments $\widehat{Y}_\tau^*$

5. Find optimal weights vector γ for $\widehat{Y}_\tau^*, X_1, X_2, X_3$ s.t. MAE is minimised
6. Forecast based on $\widehat{\gamma}$, $\widehat{Y}_\tau^*, X_1, X_2$ and X_3

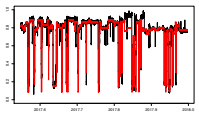


Accuracy measures

$$nrRMSE = \frac{1}{\sqrt{T}\{max(y) - min(y)\}}\sqrt{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}$$

$$nmRMSE = \frac{1}{\bar{y}\sqrt{T}}\sqrt{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}$$

$$niqrRMSE = \frac{1}{\sqrt{T}\{q(y, 0.75) - q(y, 0.25)\}}\sqrt{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}$$

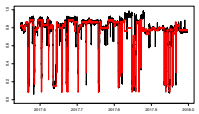


Accuracy measures

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|y_t - \widehat{y}_t|$$

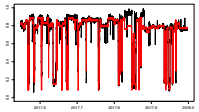
$$q_t = \frac{y_t - \widehat{y}_t}{\frac{1}{T-1}\sum_{s=2}^{T}|y_s - y_{s-1}|}$$

$$MASE = \frac{1}{T}\sum_{t=1}^{T}|q_t|$$



Forecast evaluation: daily aggregate

Measure	VARX	ARX	LSTM
nrRMSE	0.264	0.102	0.084
nmRMSE	0.282	0.109	0.110
niqrRMSE	2.044	0.789	0.557
MAE	0.136	0.056	0.051
MASE	1.294	0.536	0.475

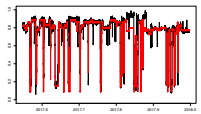


Forecast evaluation: quarter hourly data

Measure	MS FAST	VARX	SCAD	ARX	LSTM	NN(3,6)
nrRMSE	0.086	0.199	0.199	0.164	0.142	0.202
nmRMSE	0.107	0.247	0.247	0.203	0.197	0.250
niqrRMSE	0.791	1.804	1.803	1.497	1.168	1.830
MAE	0.045	0.096	0.091	0.081	0.079	0.093
MASE	0.339	0.715	0.677	0.604	0.599	0.700

Table 1: 1 day ahead forecast.

▸ MS FASTEC VARX ▸ VARX & SCAD ▸ ARX ▸ LSTM ▸ NN



Clustering on Forecast results

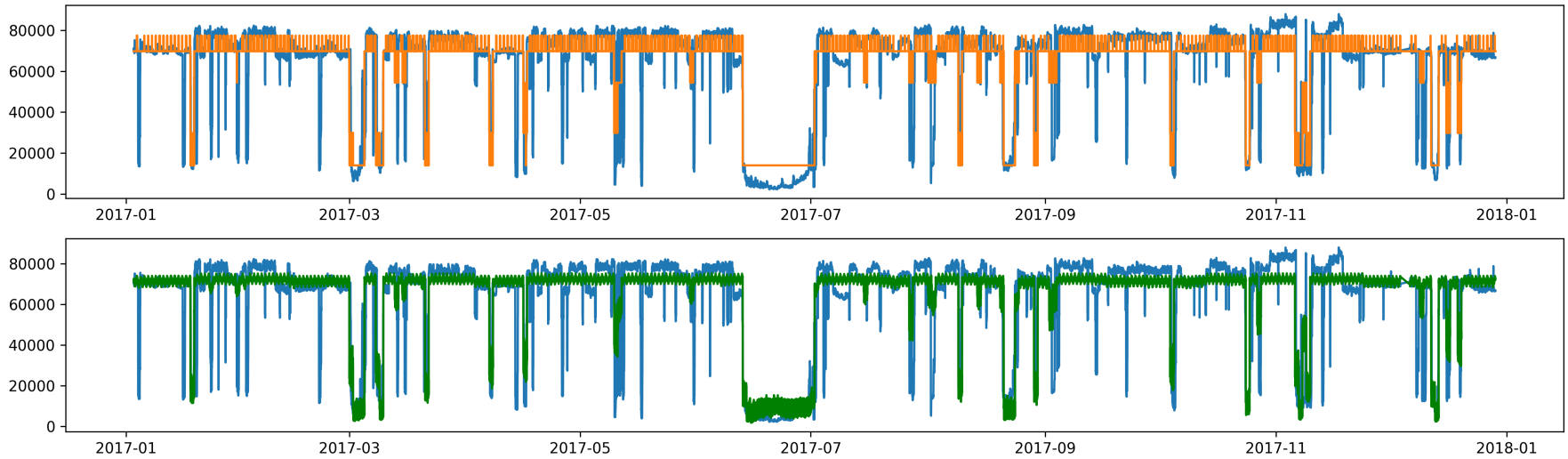
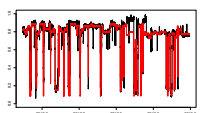
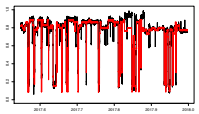


Figure 8: Actual load, μ of forecast value distribution, LSTM forecast load



Outlook

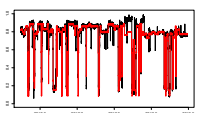
- Model refinements and sensitivity test
  - Test more advanced machine learning models
  - Forecasts up to 3 hours and 12 hours ahead
- Information augmentation
  - what type of variable could enhance forecast?
  - which could be measured by the company?
- Electricity consumption optimization
  - Adjust production schedules by stablizing load curve
  - Reduce electricity cost by trading on energy market



FISTA algorithm

1. Initialise: $\Gamma_0 = 0, \Omega_1 = 0$, step size $\delta_1 = 1$
2. For $t = 1, 2, \ldots, T$
   - $\Gamma_t = \arg\min_\Gamma \left[\frac{g(\Gamma)}{L_{\bigtriangledown g}} + \frac{1}{2} \left\| \Gamma - \left\{ \Omega_t - \frac{1}{L_{\bigtriangledown g}} \bigtriangledown g(\Omega_t) \right\} \right\|^2 \right]$
   - when penalising nuclear norm $\Gamma_t = \mathbf{P} \left(\mathbf{R} - \frac{\lambda}{L_{\bigtriangledown g}} \mathbf{I}_{p \times m} \right) \mathbf{Q}^\top$, and $\Omega_t - \frac{1}{L_{\bigtriangledown g}} \bigtriangledown g(\Omega_t) = \mathbf{PRQ}^\top$ with ALS-SVD (Hastie et al. (2014))
   - $\delta_{t+1} = \frac{1+\sqrt{1+4\delta_t^2}}{2}$
   - $\Omega_{t+1} = \Gamma_t + \frac{\delta_t - 1}{t+1}(\Gamma_t - \Gamma_{t-1})$
3. $\widehat{\Gamma} = \Gamma_T$

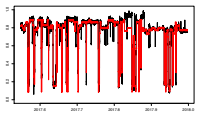
Return



ALS-SVD algorithm

1. Initialise $A = UD$, $U_{m\times r}$ is randomly chosen matrix with orthonormal columns and $D = I_r$
2. Given A, solve for B
   - $\min_B \|X - AB^\top\|_F^\top + \lambda\|B\|_F^2$
   - $\widetilde{B}^\top = (D^2 + \lambda I)^{-1} DU^\top X$
3. Compute SVD $\widetilde{B}D = \widetilde{V}\widetilde{D}^2\widetilde{R}^\top$, let $V \leftarrow \widetilde{V}, D \leftarrow \widetilde{D}, B = VD$
4. Given B, solve for A
   - $\min_A \|X - AB^\top\|_F^\top + \lambda\|A\|_F^2$
   - $\widetilde{A}^\top = XVD(D^2 + \lambda I)^{-1}$
5. Compute SVD $\widetilde{A}D = \widetilde{U}\widetilde{D}^2\widetilde{R}^\top$, let $U \leftarrow \widetilde{U}, D \leftarrow \widetilde{D}, A = UD$
6. Repeat (2)-(5) until convergence of $AB^\top$
7. Compute $M = XV$, its SVD $M = UD_\sigma R^\top$, output: $U, V \leftarrow VR, \mathcal{S}_\lambda(D_\sigma) = diag\{(\sigma_1 - \lambda)+, \ldots, (\sigma_r - \lambda)\}$

Return



Results: MS FASTEC VARX

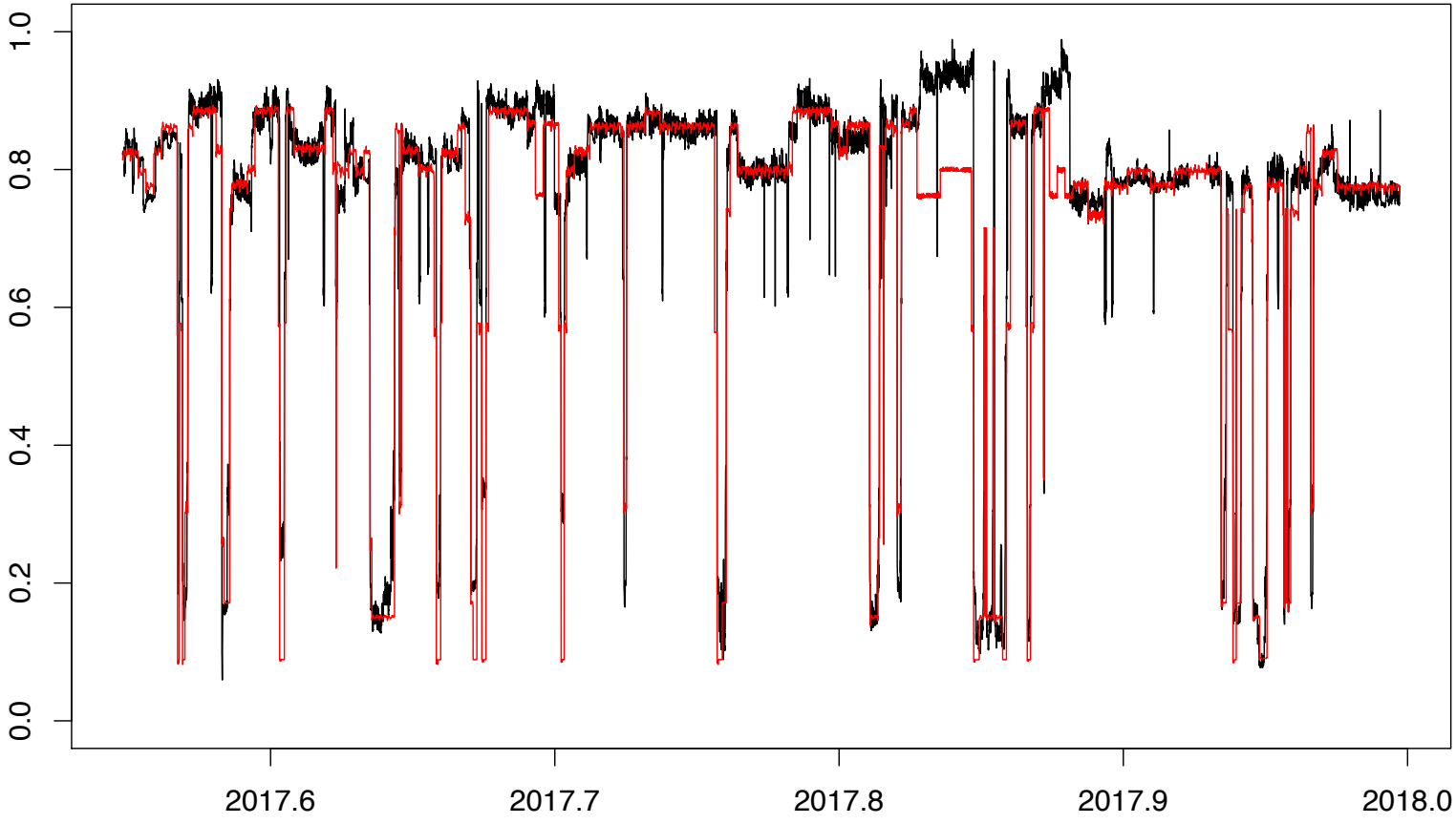


Figure 9: True curve and forecasts at local τ-expectile level.

Results: VARX, SCAD

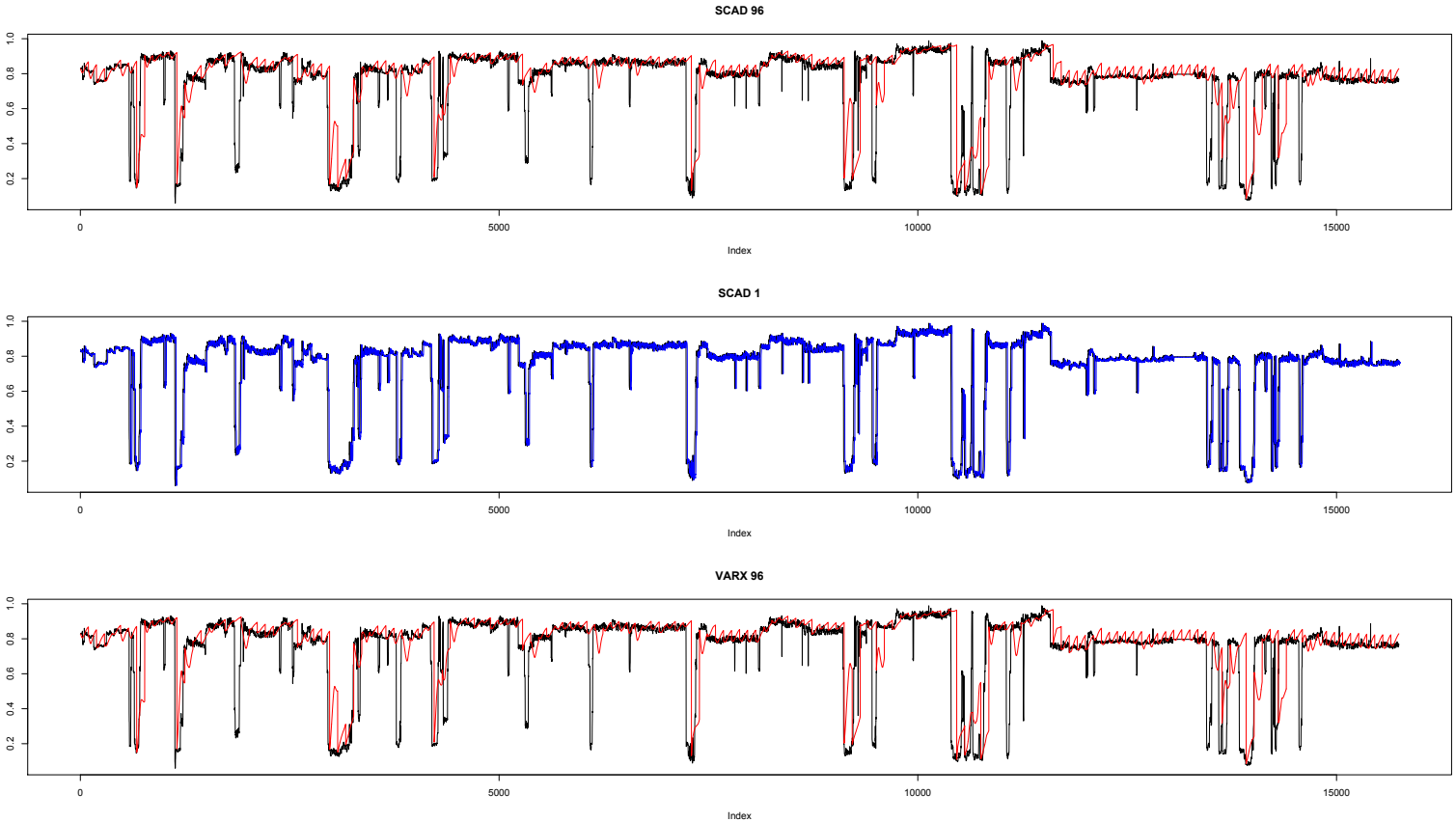
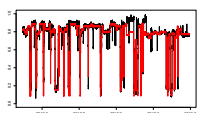


Figure 10: True curve and forecasts for SCAD 96, SCAD 1 and VARX 96: 1 day, 1 quarter hour and 1 day ahead.



Return

Results: ARX

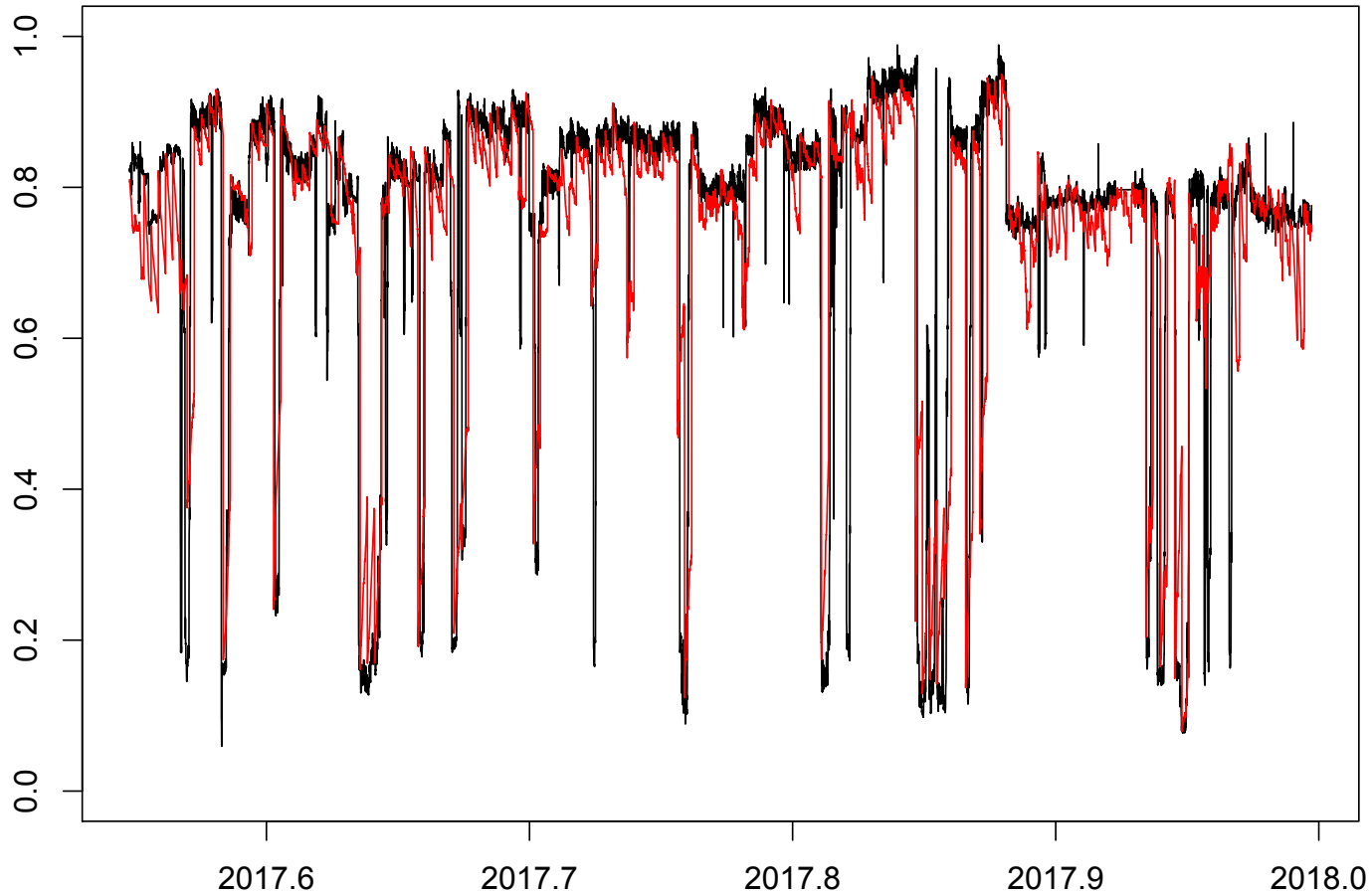
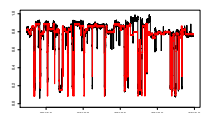


Figure 11: True curve and 1 day ahead forecast.



Return

Results: LSTM

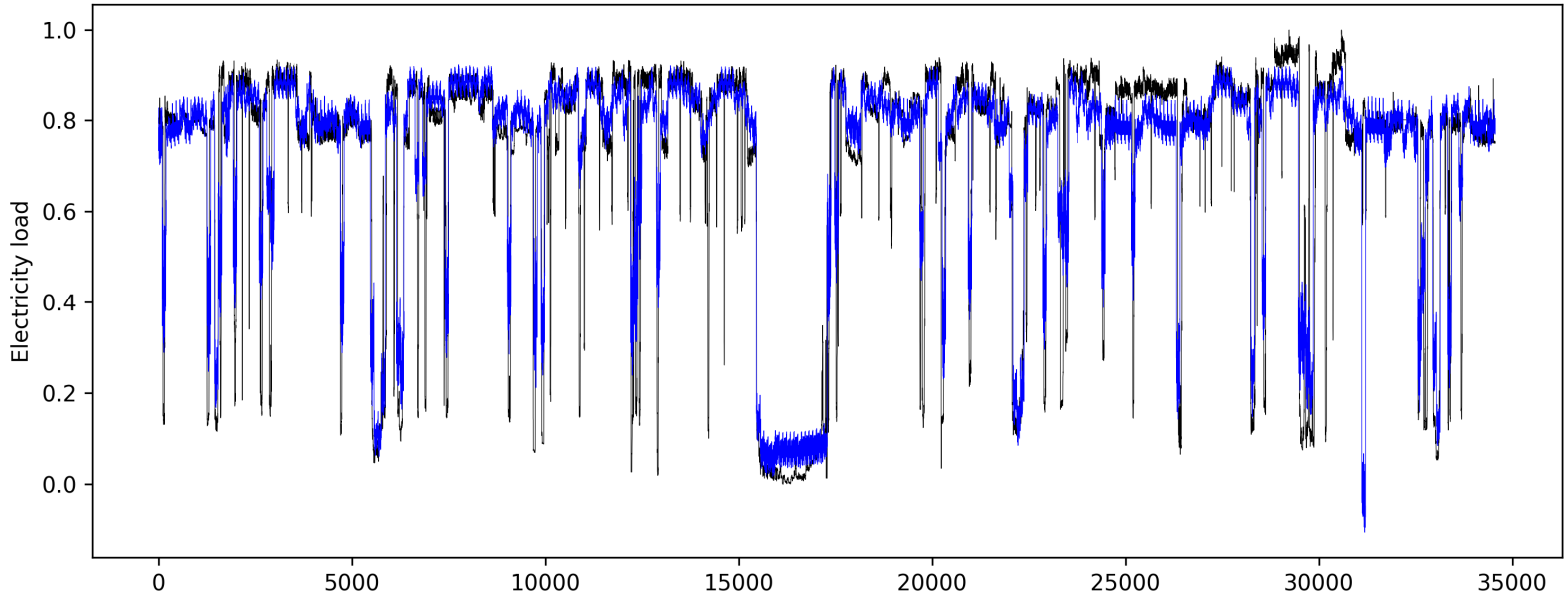
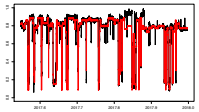


Figure 12: True curve and quarterly-hour 1-day ahead forecast.

Return



Results: LSTM

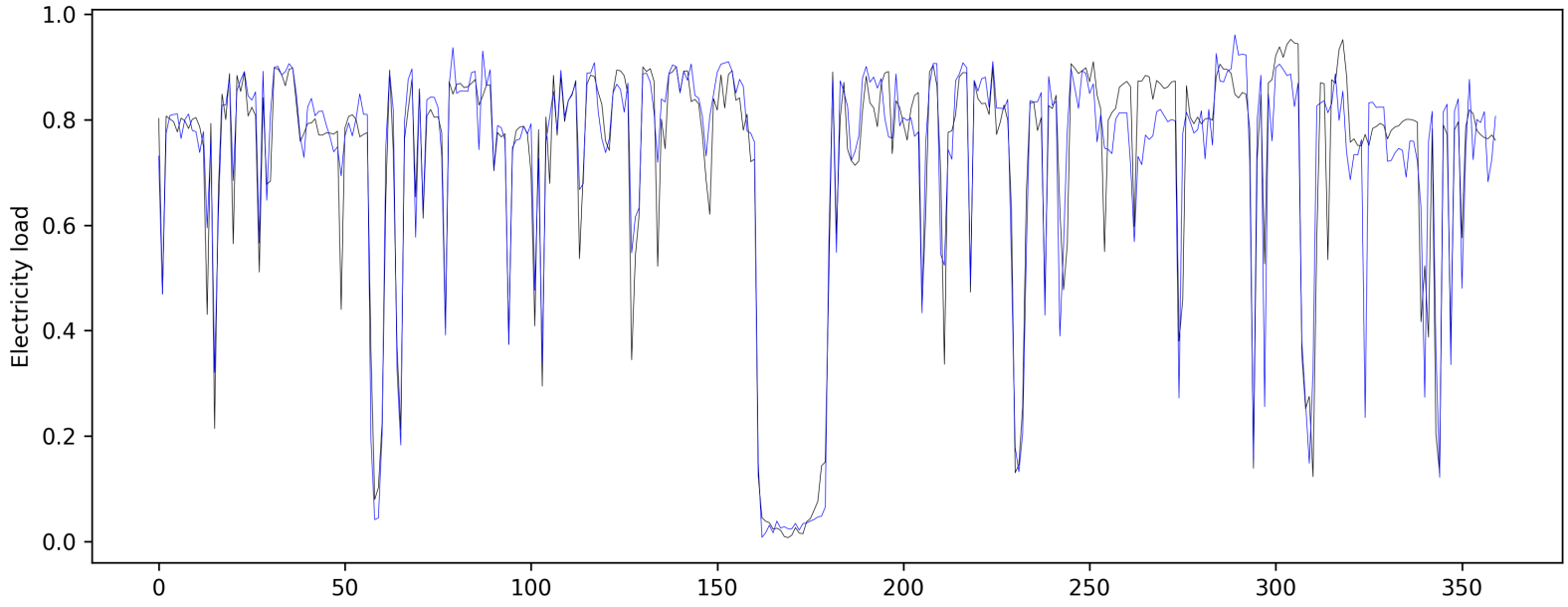
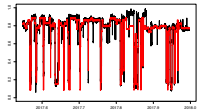


Figure 13: True curve and 1-day aggregated ahead forecast.

Return



Results: NN

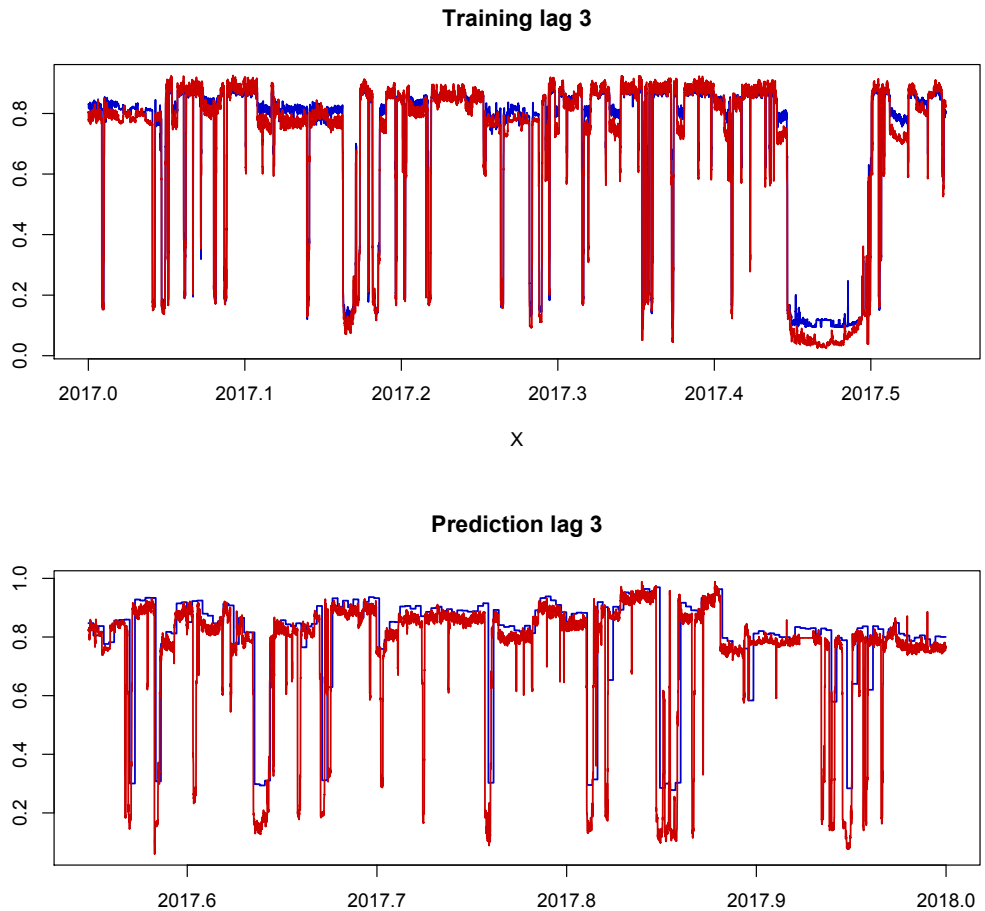
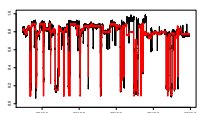
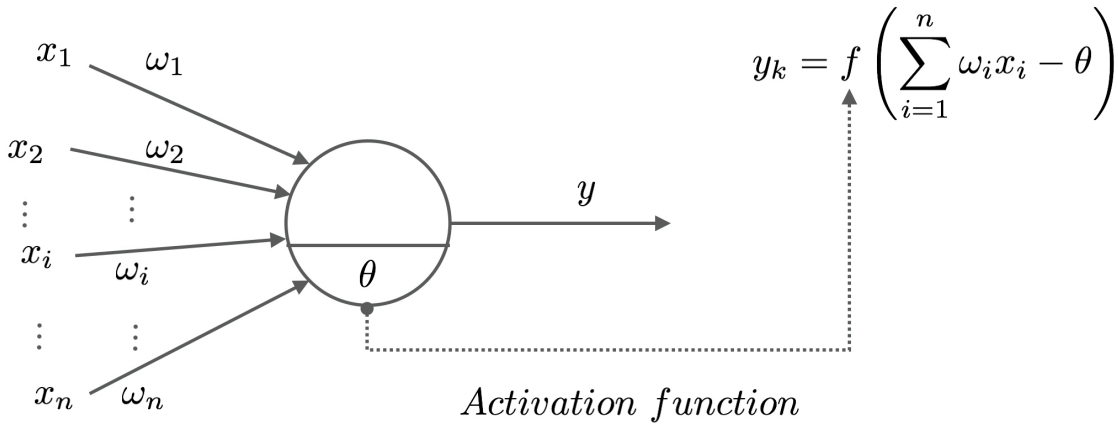


Figure 14: True curve and 1 day ahead forecast.

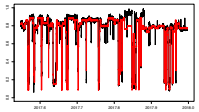
Return



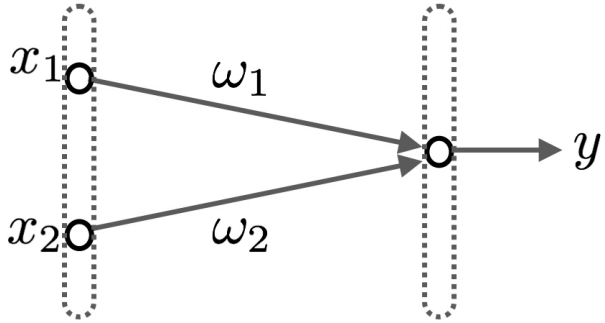
Neuron



- M-P neuron model, McCulloch and Pitts(1943)
- "Artificial Neural Networks" (Kohonen, T. ,1988): Interconnected networks of simple-adaptive elements with hierarchical interactions



Perceptron

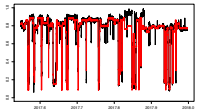


$$\omega_i \leftarrow \omega_i + \Delta\omega_i$$

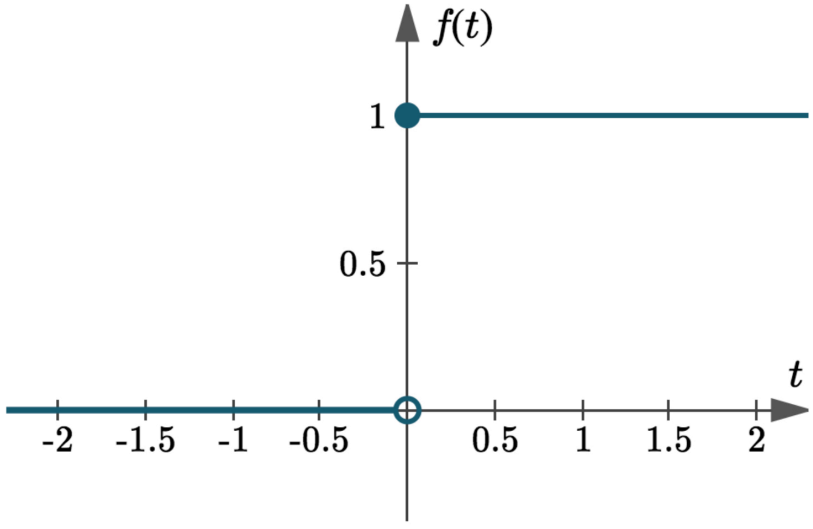
$$\Delta\omega_i = \eta(y - \hat{y})x_i$$

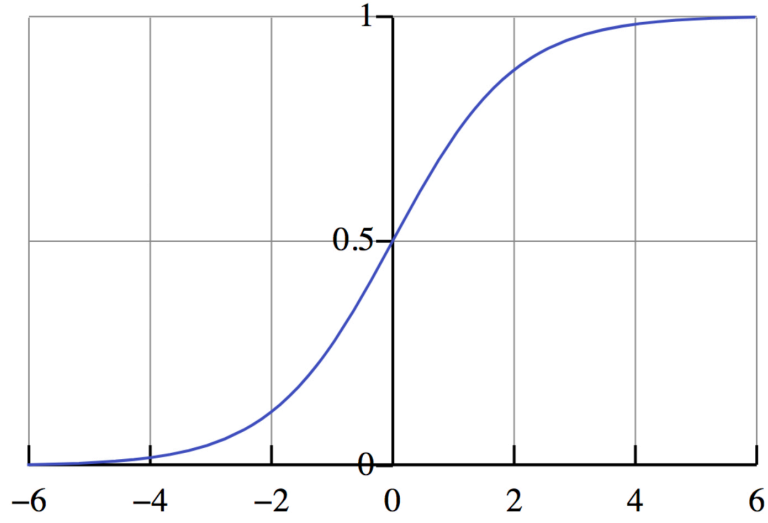
Weight adjustment

Figure 15: A perceptron is formed with 2 layers of neural networks, only output layer uses activation function, so one perceptron has only one functional layer



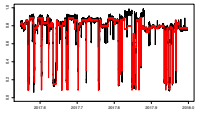
Activation Function



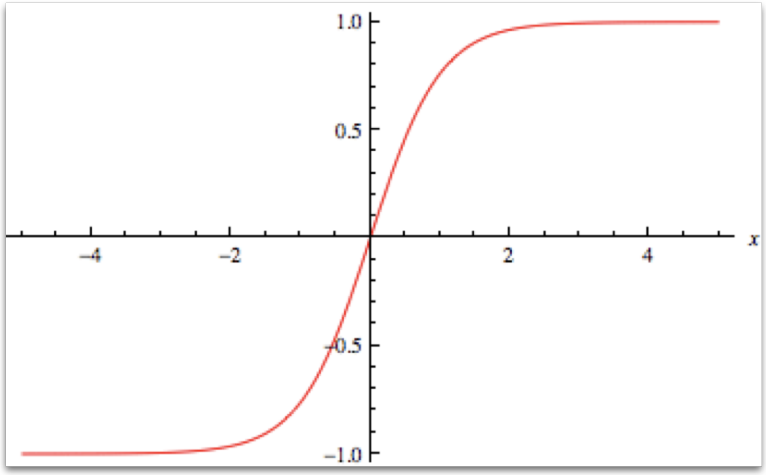


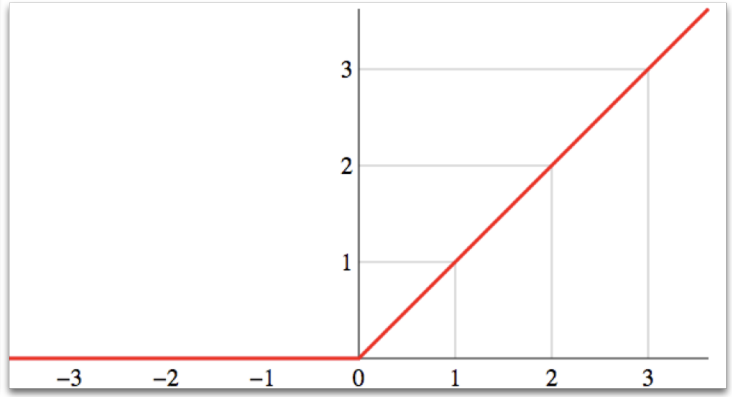
$$sgn(x) = \begin{cases} 1, x \geqslant 0 \\ 0, x < 0 \end{cases}$$

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



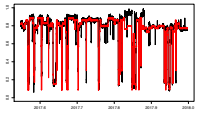
Activation Function





$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$ReLU(x) = max\{0, x\}$$



Backpropagation Algorithm

For training sample (x_k, y_k), assume the output is $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, ..., \hat{y}_l^k)$

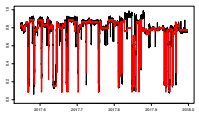
for j_{th} neuron of output layer: $\hat{y}_j^k = f(\beta_j - \theta_j)$

We can define mean-square error of sample (x_k, y_k) as:

$$E_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2$$

BP algorithm is based on gradient descent strategy, adjusting parameters to drive error E_k to the negative gradient:

$$\Delta\omega_{hj} = -\eta\frac{\partial E_k}{\partial \omega_{hj}}$$



Backpropagation Algorithm

Take the parameter, output weight of h_{th} neuron in hidden layer ω_{hj}, as an example:

Given learning rate η:

$$\Delta\omega_{hj} = -\eta \frac{\partial E_k}{\partial \omega_{hj}}$$

Use the chain rule:

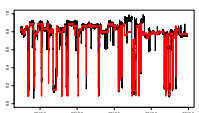
$$\frac{\partial E_k}{\partial \omega_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial \omega_{hj}}$$

According to the definition of β_j, we have: $\frac{\partial \beta_j}{\partial \omega_{hj}} = b_h$

Use the property of sigmoid function:

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

$$f'(x) = f(x)(1-f(x))$$



Backpropagation Algorithm

Derive the gradient term of output layer g_j:

$$\begin{aligned} g_j =& \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ =& -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \\ =& \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \end{aligned}$$

Finally, update term $\Delta\omega_{hj}$ reads as:

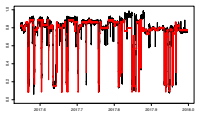
$$\Delta\omega_{hj} = \eta g_j b_h$$

Likewise, derive the other params:

$$\Delta\theta_j = -\eta g_j$$

$$\Delta v_{ih} = \eta e_h x_i$$

$$\Delta\gamma_h = -\eta e_h$$



Backpropagation Algorithm

Input

Training set $D = \{(x_k, y_k)\}_{k=1}^{m}$, and learning rate

Process

Initialize the parameters in the whole network, randomly

Repeat:

For all $(x_k, y_k) \in D$ do:

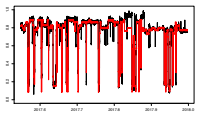
Calculate the output $\hat{y}_k$

Calculate the gradient term g_j and e_h

Update the parameters $\omega_{hj}, v_{ih}, \theta_j, \gamma_h$

Until reach some criterion

Note: standard BP algorithm only optimize the sample error



Long Short-Term Memory networks (LSTM)

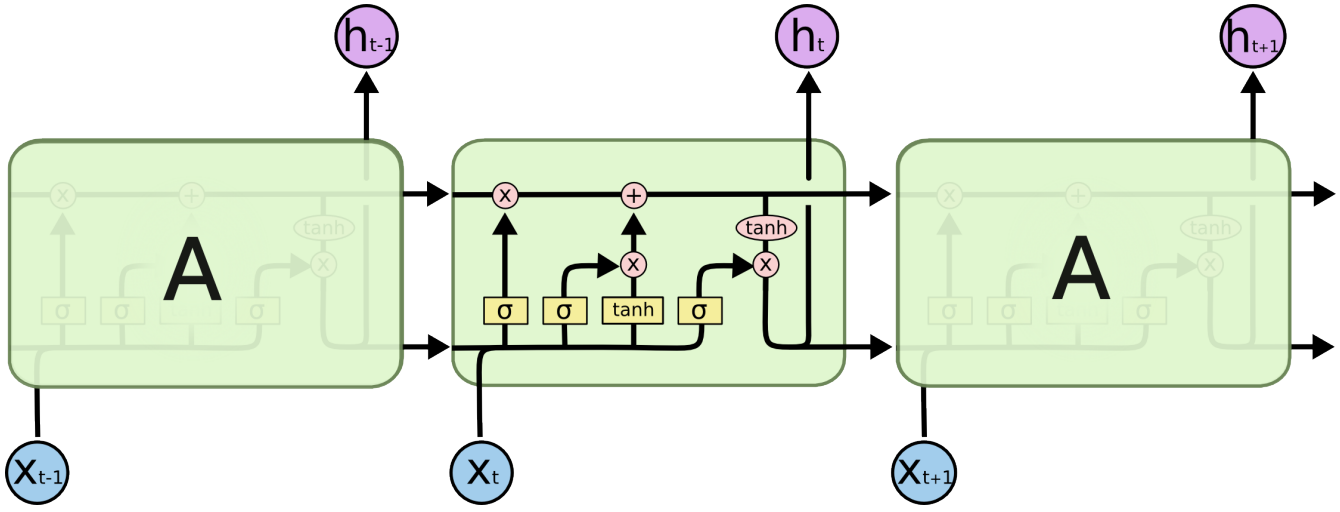
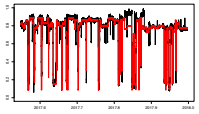


Figure 16: LSTM networks, Hochreiter & Schmidhuber (1997)



LSTM Inside Structure

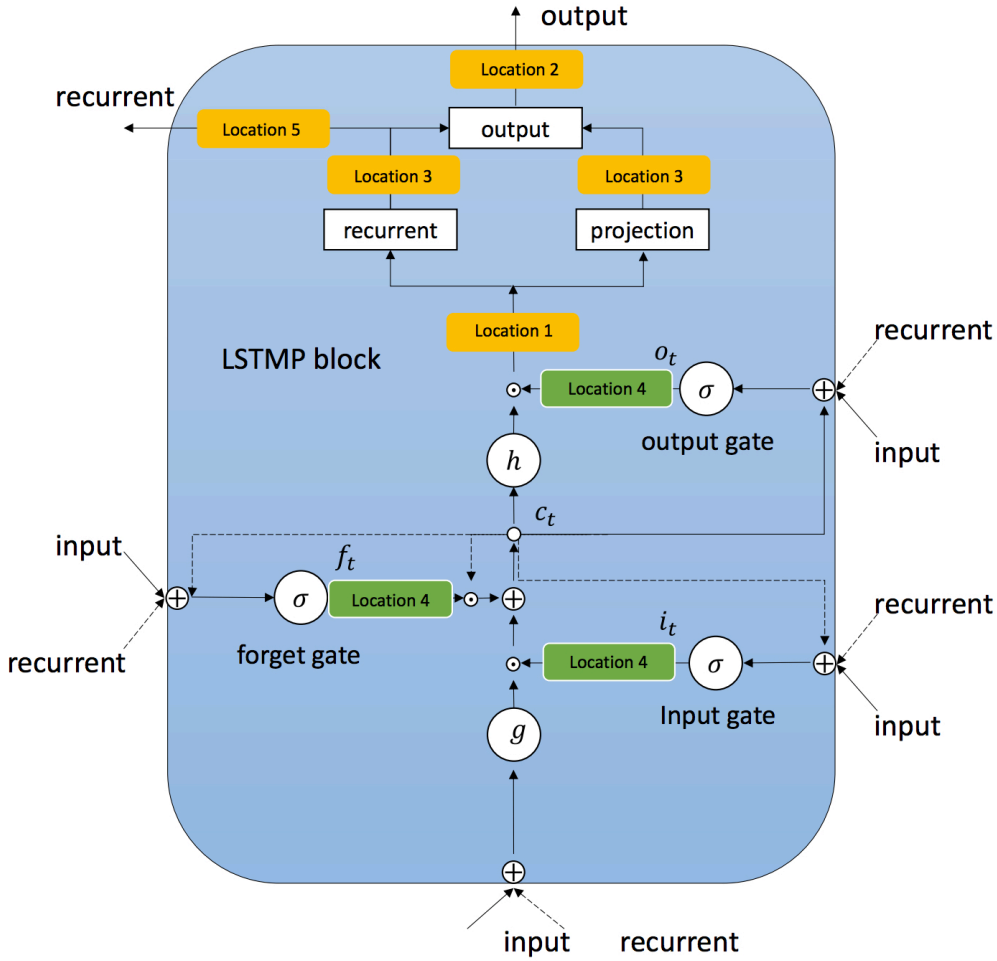
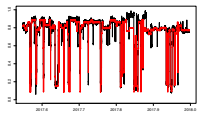


Figure 17: Ref: Cheng G, et al. (2017)



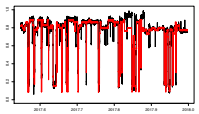
Sparse VAR model

1. Stage: partial spectral coherence (PSC):
neg. scaled inverse of spectral density

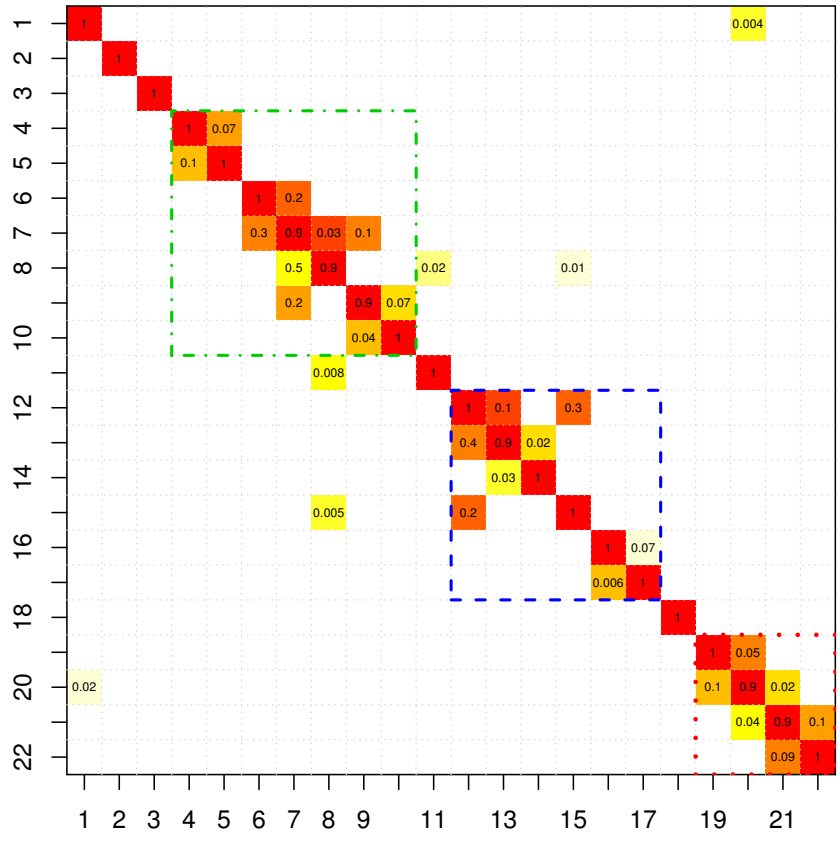
$$PSC_{ij}(\omega) = -\frac{g_{ij}^{Y}(\omega)}{\sqrt{g_{ii}^{Y}(\omega) g_{jj}^{Y}(\omega)}}, \quad \omega \in (-\pi, \pi]$$

$$S_{ij} = \sup_{\omega} |PSC_{ij}(\omega)|$$

$g^{Y}(\omega) = f^{Y}(\omega)^{-1}$: inverse density
$S_{i,j}$: conditional correlation of turbine i with j

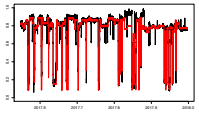


Sparse VAR model



2. Stage: variable selection by ranking according to t-stat

Dowell & Pinsen (2016)

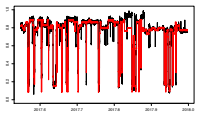


FASTEC-VAR(p) model

$$\Psi_k(\tau) = \sum_{i=1}^{p} \Theta_i \Psi_{k-i}(\tau) + \eta_k$$

$\Psi_k(\tau)$: vector of loadings at τ-level, $\Psi_{(365 \times r)}$
Θ_i: matrix of VAR coefficients
η_k: white noise error term
$\tau = \{1\%, \ldots, 50\%, \ldots, 99\%\}$

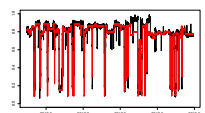
López Cabrera & Schulz (2016)



ARX(p)

$$\mathbf{Y}_k = \sum_{i=1}^{p} \Theta_i \mathbf{Z}_{k-i}(\tau) + \sum_{i=1}^{p} \Xi_i \mathbf{X}_{k-i}(\tau) + \eta_k$$

Y_k: $1 \times T$ vector of power load
Z_k: $p \times T$ matrix of p power load lags
Θ_i: $1 \times p$ vector of AR coefficients
X_k: $Mp \times T$ matrix of exogenous variables
Ξ_i: $1 \times Mp$ vector of coefficients
η_k: white noise error term

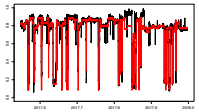


VAR(p)

Assuming all intra-day observations being "variables" of each day

$$\mathbf{Y}_k = \sum_{i=1}^{p} \Theta_i \mathbf{Z}_{k-i}(\tau) + \sum_{i=1}^{p} \Xi_i \mathbf{X}_{k-i}(\tau) + \eta_k$$

Y_k: $K \times T$ vector of power load
Z_k: $Kp \times T$ matrix of p power load lags
Θ_i: $K \times Kp$ vector of AR coefficients
X_k: $Mp \times T$ matrix of exogenous variables
Ξ_i: $M \times Mp$ vector of coefficients
η_k: white noise error term



SCAD VAR(p)

SCAD objective

$$\arg\min \frac{1}{2}(\mathbf{Y} - \Theta\mathbf{Z})^\top(\mathbf{Y} - \Theta\mathbf{Z}) + T^* \sum_{j=1}^{K \cdot P} \mathfrak{p}_\lambda(|\theta|_j).$$

$$\mathfrak{p}_\lambda(|\theta|) = \begin{cases} \lambda|\theta| & \text{if} |\theta| \leq \lambda \\ -\frac{|\theta|^2 - 2\alpha\lambda|\theta| + \lambda^2}{2(\alpha-1)} & \text{if} \lambda \leq |\theta| \leq \alpha\lambda \\ \frac{(\alpha+1)\lambda^2}{2} & \text{if } |\theta| \geq \alpha\lambda, \end{cases}$$