Ensemble Time Series Forecasting

with Applications in Renewable Energy

Candidate: Ren Ye
Supervisor: A/P P. N. Suganthan

ERI@N Joint Industry Programme (JIP)
EDB Energy Innovation Programme Office (EIPO) PhD Scholarship

March 21, 2016

Outline

Outline

Renewable Energy

Why Renewable Energy? $\Rightarrow$ Advantages

- Abundant
- Sustainable
- Environmental Friendly

Why NOT Renewable Energy? $\Rightarrow$ Challenges

- Unstable
- Time/Weather Dependent
- High Initial Cost

Forecasting

Benefit of Forecasting

- Reduce Reserve
- Reduce Risk of Overloading
- Precaution on Extreme Weather Conditions

Application of Forecasting

- *Very Short Term*: active generator control
- *Short Term*: power grid scheduling
- *Mid & Long Term*: maintenance scheduling

Development of Forecasting

Intuition

observation; empirical experiences

Statistics

exponential smoothing; autoregressive moving average

Physics, Meteorology & Geography

numeric weather prediction

Computational Intelligence

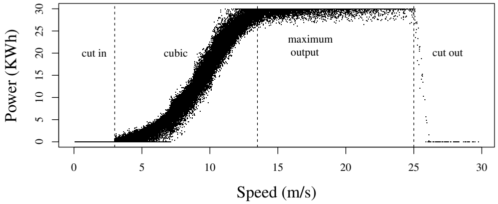
neural networks; fuzzy systems, etc.

Wind and Solar Energy Basics

Wind Energy[1]

$P = \frac{1}{2}\rho_a A_t C_p(\lambda, \beta) v^3$

where ρ_a: air density, A_t: area of the turbine, $C_p(\lambda, \beta)$: efficiency, λ: tip speed ratio, β: blade pitch, v: up-wind speed.



Wind Power vs Wind Speed Curve

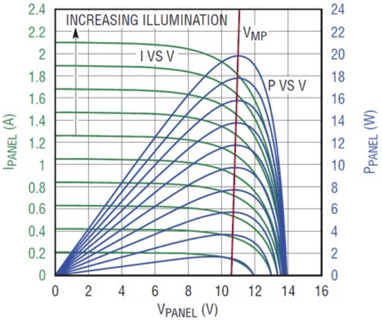
[1] Soman et al. 2010.

Wind and Solar Energy Basics

Solar Energy[1]

$$G = \alpha G_0 (1 + 0.033 \cos \frac{360d}{365}) \sin \theta_s$$
$$\sin \theta_s = \cos h \cos \delta \cos \phi + \sin \delta \sin \phi$$

where θ_s: solar elevation angle, h: solar hour angle, δ: sun declination, ϕ: local latitude, d: date sequence of a year, α: cloud/haze cover index, $G_0 = 1367 W/m^2$: solar irradiance constant.



Solar PV Current-Voltage and Power Curve

[1] Wang et al. 2012.

Time Series Basics

Probability Density Function (PDF)

distribution, likelihood of value

Stationary

statistical characteristics are independent of the time when it is observed.

Weakly stationary

1. the mean of the TS is constant and does not depend on time
2. the autocorrelation function (ACF) depends on s and t only through their difference $|s - t|$.

Time Series Basics

Autocorrelation Function

$\gamma(s,t) = \mathrm{E}[(X_s - \mu_s)(X_t - \mu_t)]$
$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} = \frac{\gamma(s,t)}{|\sigma_s \sigma_t|}$

where μ: sample mean, σ: sample standard deviation, γ: auto covariance, ρ: auto correlation.

Trend

additive term, slowly varying

Seasonality

repeated pattern, additive and/or multiplicative

Outline

Ensemble Methods[4]

- Uses multiple predictors to obtain an aggregated decision which is better than any of the base predictors[2]
- Three fundamental reasons for the success of ensemble methods: statistical, computational and representational[3]
- Bias-variance-covariance decomposition
- Divide and conquer

[2] Opitz and Maclin 1999.

[3] **Dietterich2000ensemble** .

[4] Ren, Suganthan, and Srikanth 2015b; Ren, Zhang, and Suganthan 2016.

Bias-variance-covariance decomposition

$$\mathrm{E}[\overline{f} - t]^2 = bias^2 + \frac{1}{M} var + (1 - \frac{1}{M}) covar$$

$$bias = \frac{1}{M} \sum_{i=1}^{M} (\mathrm{E}[f_i] - t)$$

$$var = \frac{1}{M} \sum_{i=1}^{M} \mathrm{E}[f_i - \mathrm{E}[f_i]]^2$$

$$covar = \frac{1}{M(M-1)} \sum_{i} \sum_{j \neq i} \mathrm{E}[f_i - \mathrm{E}[f_i]](f_j - \mathrm{E}[f_j])$$

where t: target, M: ensemble size, $\mathrm{E}(\cdot)$: expectation function, f_i: output from each single model.

Diversity

Data Diversity

Bagging, boosting, random subspace, spatial correlation

Parameter Diversity

Negative correlation learning, multiple kernel learning, multi-objective optimization

Structural Diversity

Heterogeneous ensemble

Preprocessing

Wavelet decomposition

Wavelet-ANN

Empirical mode decomposition

EMD-ANN/SVR

Complex Value

Complex value ANN

Postprocessing

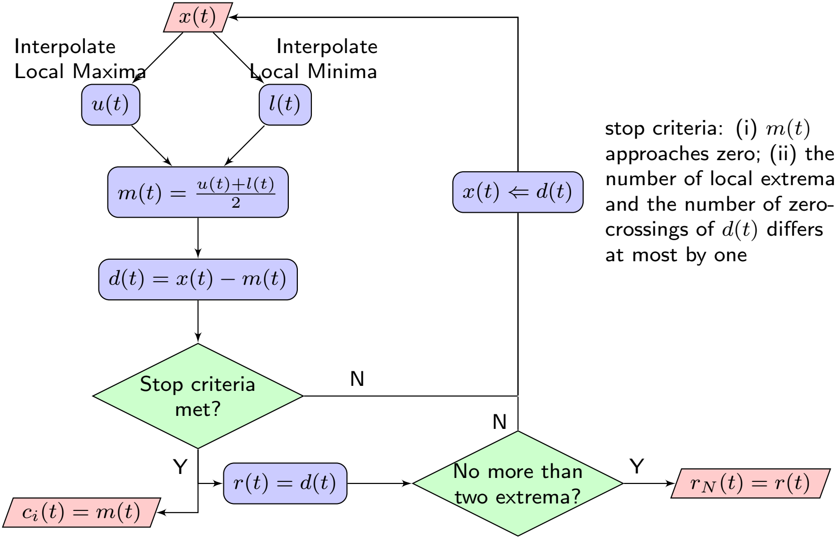
Linear-residual analysis
ARIMA-GARCH

Linear-nonlinear
ARIMA-ANN/SVR, RVFL

Converted from ensemble classification
SVR-SVC, extreme randomized discretization

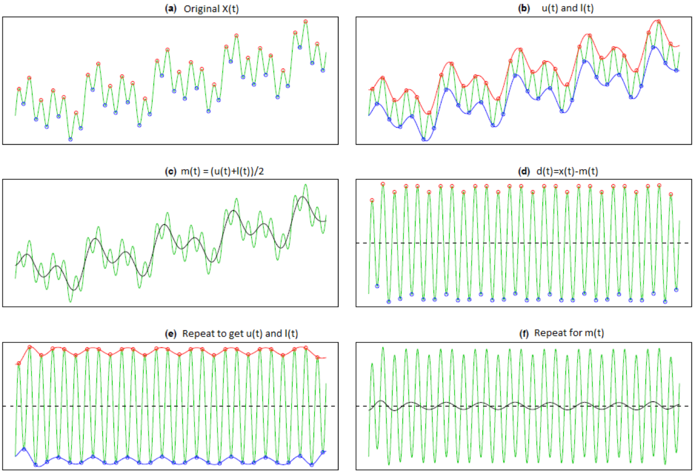
Outline

Flowchart



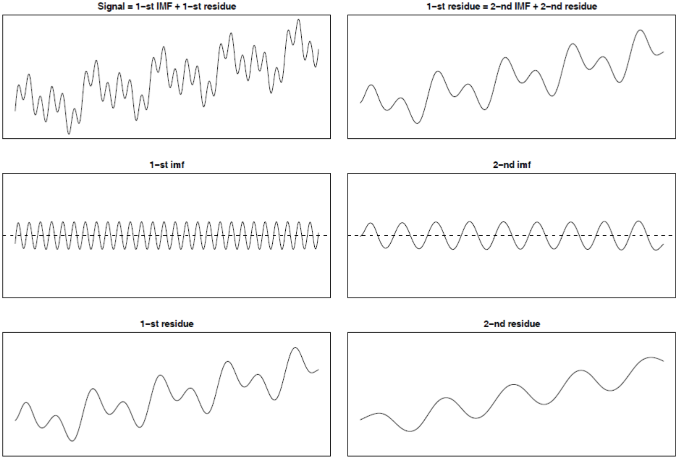
stop criteria: (i) $m(t)$ approaches zero; (ii) the number of local extrema and the number of zero-crossings of $d(t)$ differs at most by one

Example[5]



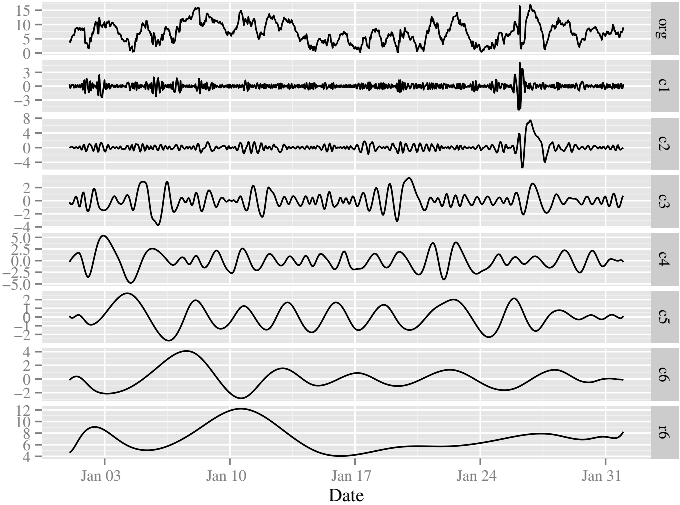
5 Kim and Oh 2009.

Example[5]



5 Kim and Oh 2009.

Example[5]



5 Kim and Oh 2009.

Characteristics[6]

$$x(t) = \sum_{i=1}^{N} c_i(t) + r_N(t)$$

- Decompose complex time series into simpler time series
- Narrow band, symmetric
- Reveal hidden features/correlations of the original time series

Advantages

- Adaptive
- Local
- Orthogonal
- Completeness

Disadvantages

- Mode mixing
- Uncertain iteration

[6] N. Huang et al. 1998.

EMD-ANN/SVR

Multiple predictors approach

$$\hat{x}(t+h) = \sum_{i=1}^{N} f_i(c_i(t), \ldots, c_i(t-p_i+1), \theta_{c_i}) + f(R_N(t), \ldots, R_N(t-l_i+1), \theta_R)$$

$\hat{x}(t+h)$: h-step ahead predicted value, $c_i(t)$: ith decomposed IMF, $R_N(t)$: residue, p_i: ith lag

- $N+1$ predictor needed
- Additive error

EMD-ANN/SVR

Single predictor approach[7]

$$\hat{x}(t+h) = f(c_i(t), R_N(t), \theta),\ i \in \{1, \dots, N\}$$

$\hat{x}(t+h)$: h-step ahead predicted value, $c_i(t)$: ith decomposed IMF, $R_N(t)$: residue, p_i: ith lag

- Reduce computation time
- Sacrifice accuracy
- Issue on feature selection

[7] Han and Zhu 2011.

Input Vector Reconstruction

Main Algorithm[8]

1. Decomposes $\mathbf{x}(t)$ by EMD into a collection of IMFs: $c_i(t)\ i \in \{1, \ldots, N\}$, and a Residue r_N,
2. Combines the IMFs up-to an optimal lag l based on a feature selection module $\Rightarrow$ input vector
3. Forecast the output from the input vector by ANN/SVR

Advantages

1. Does not require more than one predictor to model the decomposed time series $\Rightarrow$ low computational power
2. Retains the importance of autocorrelation of time series

8 Ren, Suganthan, and Srikanth 2014.

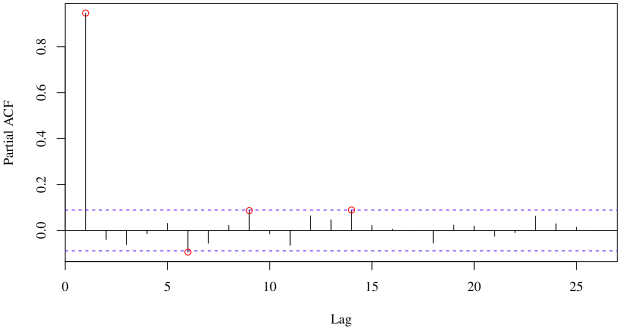
Input Vector Reconstruction

Feature selection

- PACF selection
- mRMR selection
- 2-days' historical data (FULL)

Input Vector Reconstruction

PACF selection



Input Vector Reconstruction

mRMR selection

- minimum Redundancy Maximum Relevance
- select the top 50% sailent features
- $$I(x, y) = -0.5\ln(1 - \rho(x, y)^2)$$

  where x: input feature, y: output data, I: mutual information, ρ: correlation coefficient (usually Pearson's or Spearman's correlation coefficient)
  First select:

  $$x_i = \underset{x_i \in X}{\operatorname{argmax}} I(x_i, y)$$

  Repeatedly select:

  $$x_j = \underset{x_j \in X}{\operatorname{argmax}} I(x_j, y) - \frac{1}{|S|}\sum_{x_i \in S} I(x_j, x_i)$$

Experiment Setup

- NDBC sites 41004, 44009, 46077, year 2011.
- Monthly, 70% for training, 30% for testing.
- 1 – 12 hour ahead forecasting
- Error metric: nRMSE, nMAE, MASE

$$\mathrm{nRMSE} = \frac{1}{x_{max} - x_{min}} \sqrt{\mathrm{E}[(\hat{x} - x)^2]}$$

$$\mathrm{nMAE} = \frac{1}{x_{max} - x_{min}} \sqrt{\mathrm{E}\left|\hat{x} - x\right|}$$

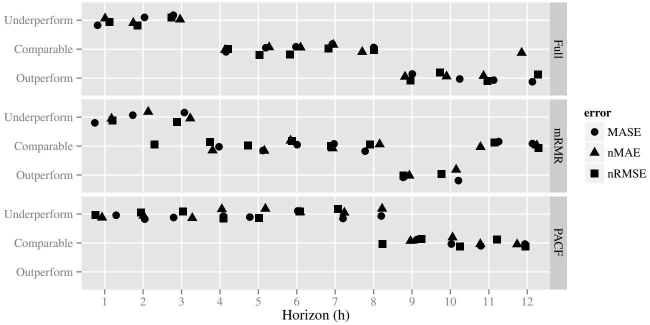
$$\mathrm{MASE} = \frac{\sum_{t=1}^{n} |\hat{x} - x|}{\frac{n}{n-1} \sum_{i=2}^{n} |x_i - x_{i-1}|}$$

$$\mathrm{sMAPE} = \frac{1}{n} \sum_{t=1}^{n} \frac{|x - \hat{x}|}{(|x| + |\hat{x}|)/2} \times 100\%$$

where $\hat{x}$: predicted data, x: desired data, n: number of data points in the time series.

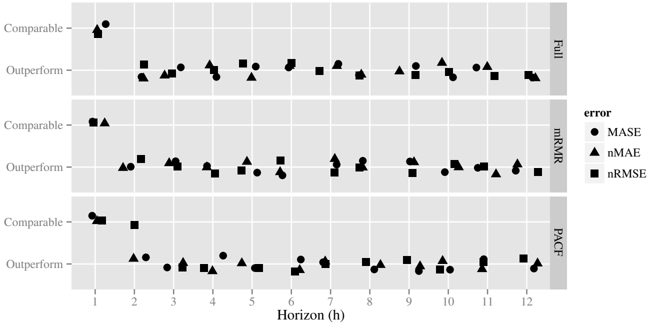
Results and Discussion

Improvement Contributed by EMD to ANN, Wilcoxon Signed Rank Test



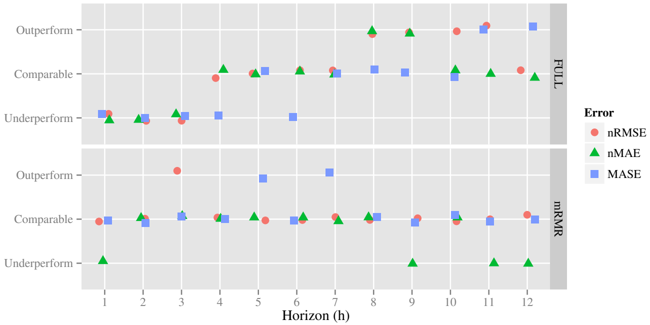
Results and Discussion

Improvement Contributed by EMD to SVR, Wilcoxon Signed Rank Test



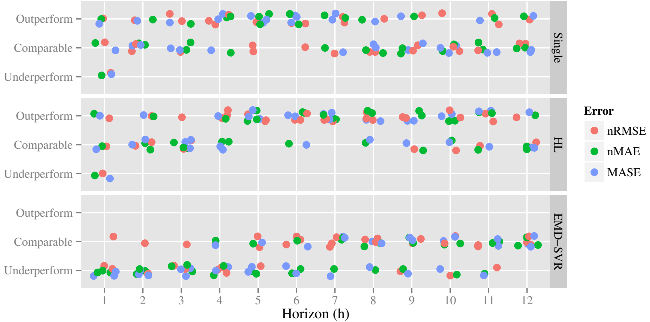
Results and Discussion

Feature Selection Comparison (PACF vs FULL and mRMR), Nemenyi Test



Results and Discussion

Comparison with Benchmark Methods (vs Single[8] and HL[9]), Nemenyi Test



8 Han and Zhu 2011.

9 Ye and Liu 2011.

Concluding Remarks

- Improvement of EMD on SVR is significant
- PACF reduces the input space complexity without degrading the performance
- Outperformed two benchmark methods and comparable with multiple predictor based EMD-SVR

Noise assisted ensemble EMD (EEMD)[10]

Noise assisting

- Uncorrelated Gaussian noise
- Multiple Trials
- Noise may cancel each other after aggregation

Side effect

- Completeness violated
- Large number of trials

[10] Wu and Norden E. Huang 2009.

Noise assisted ensemble EMD (EEMD)[10]

1. Create a collection of noise added original time series: $x^i(t) = x(t) + \varepsilon^i(t)$, $i \in \{1, \ldots, I\}$, where $\varepsilon(t)$ are independent Gaussian white noise.
2. For each $x^i(t)$, apply EMD to obtain the decomposed IMFs and residue: $x^i(t) = \sum_{j=1}^{N} c_j^i + r_N^i$.
3. In order to reconstruct back the original time series, one just needs to average on all trials:

$$x(t) = \frac{1}{I}(\sum_{i=1}^{I}\sum_{j=1}^{N} c_j^i + r_N^i) + \varepsilon_I$$

where $\varepsilon_I = \frac{\varepsilon}{\sqrt{I}}$ is the aggregated error.

[10] Wu and Norden E. Huang 2009.

Complementary ensemble EMD (CEEMD)[11]

Improvement

- Correlated Gaussian noise
- Noise can 100% cancel each other after aggregation

$$\varepsilon^i(t) \in \{\varepsilon_+^{i/2}(t), \varepsilon_-^{i/2}(t)\}$$

where $\varepsilon_+^{i/2}(t) + \varepsilon_-^{i/2}(t) = 0$, $i \in \{1, \ldots, I\}$.

Side effect

- Double the trials

[11] Yeh, Shieh, and N. E. Huang 2010.

Complete ensemble EMD with adaptive noise (CEEMDAN)[12]

Improvement

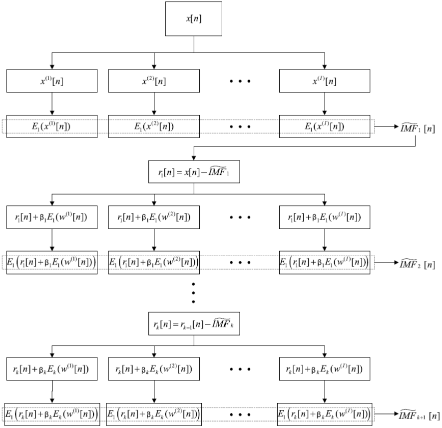
- Controlled noise generation
- Reduce number of iteration
- Completeness

Side effect

- Sequential, cannot use parallel computing
- Question on noise s.d. selection

12 Torres et al. 2011.

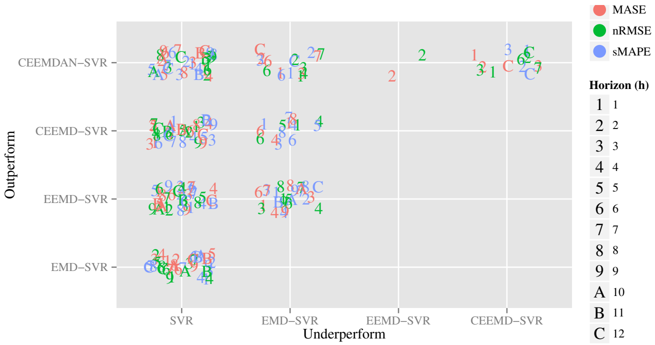
Complete ensemble EMD with adaptive noise (CEEMDAN)[12]



12 Torres et al. 2011.

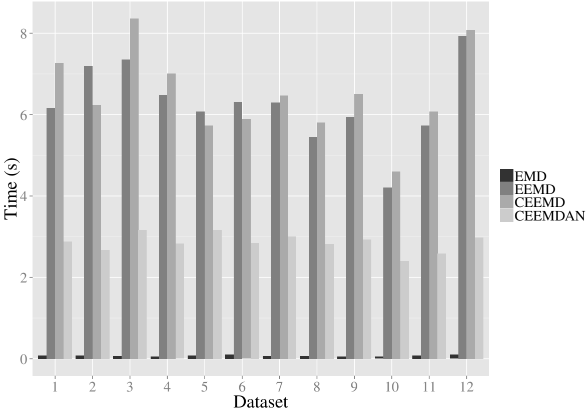
EEMD vs EMD on Wind Speed Forecasting

Wilcoxon Signed Rank Test among the SVR based Methods



EEMD vs EMD on Wind Speed Forecasting

CPU Time



EEMD vs EMD on Solar Irradiance Forecasting[13]

Wilcoxon Signed Rank Test between the Persistent model and the SVR based Methods

Horizon(h)	SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
24	0.006	0.999	0.000	0.001	0.001
48	0.017	0.992	0.000	0.006	0.001

13 Ren, Suganthan, and Srikanth 2015a.

EEMD vs EMD on Solar Irradiance Forecasting[13]

Wilcoxon Signed Rank Test among the SVR based Methods

Outperformance		Underperformance: SVR	EMD-SVR	EEMD-SVR	CEEMD-SVR	CEEMDAN-SVR
24 h ahead	SVR	–	0.000	0.993	0.935	0.978
	EMD-SVR	1	–	1	1	0.999
	EEMD-SVR	0.008	0.000	–	0.0319	0.017
	CEEMD-SVR	0.075	0.000	0.973	–	0.688
	CEEMDAN-SVR	0.026	0.001	0.986	0.338	–
48 h ahead	SVR	–	0.001	0.997	0.830	0.986
	EMD-SVR	0.999	–	0.999	0.999	0.999
	EEMD-SVR	0.003	0.001	–	0.008	0.000
	CEEMD-SVR	0.190	0.001	0.993	–	0.604
	CEEMDAN-SVR	0.017	0.001	0.999	0.425	–

13 Ren, Suganthan, and Srikanth 2015a.

Concluding Remarks

Wind Speed Forecasting

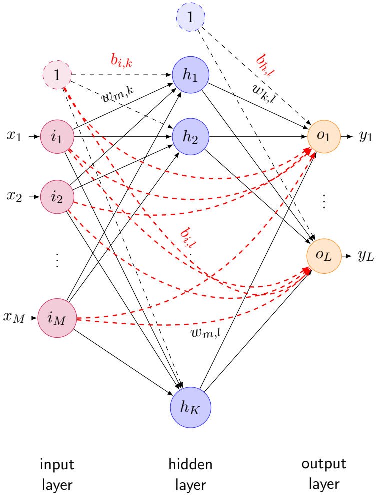
- Noise assisted Ensemble EMD-SVR outperform EMD-SVR
- CEEMDAN-SVR has the best performance (accuracy and time)

Solar Irradiance Forecasting

- EMD-SVR underperform SVR and persistent method (mode mixing)
- Noise-assisted ensemble EMD versions have attenuated the mode mixing problem and the performance has been elevated significantly
- EEMD-SVR has the best performance

Outline

Schematic Diagram



Dashed red arrows: direct connections between the input neurons and the output neurons, Dashed black arrows: connections from a bias neuron, $f(\cdot)$: $logsig$ activation function. N: number of samples, M: number of input neurons, K: number of hidden neurons. $w_{m,k}$: weights between the input and hidden layer neurons, $w_{k,l}$: weights between the hidden and output layer neurons and $b_{i,k}$ and $b_{h,j}$: input layer and hidden layer biases.

Feature

Compare with ANN[14]

- No Need backpropagation to train the network
- Random weights for $w_{m,k}$
- Least square estimation to obtain $w_{k,l}$ and $w_{m,l}$

14 Pao, Park, and Sobajic 1994; Chen 1996.

Variations[17]

Method	Input Layer Bias	Hidden Layer Bias	Input-Output Connection	Formula
M1	√	√	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l} + \sum_{m=1}^{M} w_{m,l} i_m + b_{i,l}$
M2 [15]	√	√	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l}$
M3 [16]	√	×	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + \sum_{m=1}^{M} w_{m,l} i_m + b_{i,l}$
M4	√	×	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m + b_{i,k})$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k$
M5	×	√	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l} + \sum_{m=1}^{M} w_{m,l} i_m$
M6	×	√	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + b_{h,l}$
M7	×	×	√	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k + \sum_{m=1}^{M} w_{m,l} i_m$
M8	×	×	×	$h_k = f(\sum_{m=1}^{M} w_{m,k} i_m)$ $o_l = \sum_{k=1}^{K} w_{k,l} h_k, \forall k \in \{1, \ldots, K\}, \forall l \in \{1, \ldots, L\}$

15 Schmidt, Kraaijveld, and Duin 1992.

16 Pao, Park, and Sobajic 1994.

17 Ren, Suganthan, Srikanth, and Amaratunga 2016.

Influence of Input Layer Bias

Wind Speed Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M5	0.081	**0.013**	**0.011**	0.115
M2 v.s. M6	0.101	0.229	**0.008**	**0.001**
M3 v.s. M7	0.408	**0.007**	**0.002**	0.057
M4 v.s. M8	0.377	0.121	**0.012**	**0.001**

Solar Irradiance Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)	
	24	48
M1 v.s. M5	0.694	0.952
M2 v.s. M6	0.910	0.608
M3 v.s. M7	0.435	0.333
M4 v.s. M8	0.805	0.201

Influence of Hidden Layer Bias

Wind Speed Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M3	0.121	0.281	0.313	0.637
M2 v.s. M4	0.144	0.964	0.242	0.187
M5 v.s. M7	0.387	0.387	**0.006**	0.096
M6 v.s. M8	0.906	0.281	0.523	0.160

Solar Irradiance Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)	
	24	48
M1 v.s. M3	**0.026**	0.777
M2 v.s. M4	0.847	0.744
M5 v.s. M7	0.675	0.157
M6 v.s. M8	0.142	0.221

Influence of Direct Input–Output Connections

Wind Speed Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)			
	1	4	8	12
M1 v.s. M2	*0.290*	0.002	7.0e-07	3.5e-05
M3 v.s. M4	*0.550*	1.8e-04	5.2e-07	1.6e-05
M5 v.s. M6	*0.560*	0.002	8.4e-07	1.3e-05
M7 v.s. M8	*0.632*	0.002	3.7e-06	2.5e-05

Solar Irradiance Forecasting, Wilcoxon Signed Rank Test

Comparison	Horizon (h)	
	24	48
M1 v.s. M2	0.029	0.004
M3 v.s. M4	0.010	0.005
M5 v.s. M6	0.029	0.017
M7 v.s. M8	0.004	0.006

Compare with Benchmark Methods

Wilcoxon Signed Rank Test

RVFL vs hierarchical ensemble methods

	Horizon (h)	ARIMA-ANN	ARIMA-SVR
RVFL	1	9.09e-13	1.00
	4	9.09e-13	0.99
	8	9.09e-13	0.89
	12	1.18e-13	0.64

RVFL vs other machine learning methods

	Horizon (h)	ANN	SVR	RF
RVFL	1	0.006	0.125	0.500
	4	0.006	0.187	0.437
	8	0.006	0.437	0.312
	12	0.006	0.312	0.312

Concluding Remarks

- Hidden layer bias: no significant improvement
- Input layer bias: improve for wind forecasting only
- Direct input–output connections: significant improvement
- RVFL better than two stage hierarchical ensemble methods (ARIMA-ANN)

Wind Power Ramp[18]

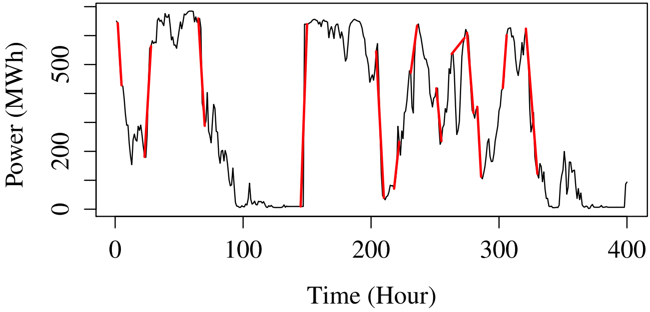
$$\Gamma_{ext} = max(P(t, \ldots, t + \Delta t)) - min(P(t, \ldots, t + \Delta t)) > \Gamma_{val}$$

$$\Gamma_{end} = |P(t + \Delta t) - P(t)| > \Gamma_{val}$$

where $P(t)$: wind power generated at time t, Δt: time interval, Γ_{val}: threshold

18 Bossavy, Girard, and Kariniotakis 2013.

Wind Power Ramp[18]



18 Bossavy, Girard, and Kariniotakis 2013.

Dataset Processing[19]

Imbalance

- Upsample minority class training data
- Keep test data untouched
- Use F Score instead of accuracy

	$\hat{x}=+1$	$\hat{x}=-1$
$x=+1$	TP	FN
$x=-1$	FP	TN

$$\text{Recall} = \frac{TP}{TP+FN}$$
$$\text{Precision} = \frac{TP}{TP+FP}$$
$$F\,\text{Score} = 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

where TP stands for true positive, FN stands for false negative, FP stands for false positive and TN stands for true negative.

19 Ren, Qiu, et al. 2015.

Classification Performance

Friedman Test

Measure	Γ_{ext} 6 Hour χ^2	Γ_{ext} 6 Hour p	Γ_{ext} 12 Hour χ^2	Γ_{ext} 12 Hour p	Γ_{end} 6 Hour χ^2	Γ_{end} 6 Hour p	Γ_{end} 12 Hour χ^2	Γ_{end} 12 Hour p
F Score	4.92	0.177	10.92	**0.0121**	3.7826	0.286	7.8	0.050
Precision	1.08	0.781	12.12	**0.0069**	1.1739	0.759	7.8	0.050
Recall	8.1875	**0.0423**	9.72	**0.021**	7.6957	0.053	4.4667	0.215

Nemenyi Test

		Γ_{ext} ANN	Γ_{ext} RF	Γ_{ext} SVM	Γ_{end} ANN	Γ_{end} RF	Γ_{end} SVM
RVFL	F Score	0.203	0.961	**0.017**	0.20	0.99	0.12
	Precision	0.068	0.995	0.122	0.32	0.99	0.20
	Recall	0.122	0.611	**0.017**	0.53	0.69	0.20

CPU Time (s)

Dataset	6 Hour								12 Hour							
	RF		SVM		ANN		RVFL		RF		SVM		ANN		RVFL	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Γ_{ext}																
D1	1.2	1.13	85.85	0.24	205.42	7.55	1.72	0.09	0.81	0.78	64.36	0.22	151.3	3.95	1.37	0.08
D2	0.88	0.9	75.56	0.2	176.92	4.52	1.7	0.09	0.7	0.67	50.3	0.17	118.95	3.62	1.27	0.06
D3	1.17	1.22	88.48	0.22	181.02	6.27	1.67	0.88	0.67	0.7	57.47	0.16	113.88	4.57	1.22	0.06
D4	0.78	0.81	72.88	0.21	157.63	4.48	1.89	0.08	0.62	0.63	51.01	0.15	122.21	3.71	1.26	0.06
D5	0.98	1.01	92.18	0.33	187.18	5.04	1.73	0.1	0.89	0.85	64.46	0.2	141.37	3.32	1.39	0.08
Ave	1	1	82.99	0.24	181.63	5.57	1.74	0.25	0.74	0.73	57.52	0.18	129.54	3.83	1.30	0.07
Sd	0.182	0.166	8.37	0.05	17.29	1.32	0.09	0.35	0.11	0.09	6.88	0.03	16.00	0.47	0.07	0.01
Γ_{end}																
D1	1.06	1.07	86.03	0.25	214.61	5.55	1.72	0.09	0.98	0.8	70.18	0.21	154.19	4.42	1.55	0.08
D2	0.92	1.12	78.76	0.12	173.92	5.48	1.75	0.08	0.66	0.63	46.34	0.13	115.34	4.64	1.22	0.09
D3	1.14	1.17	89.94	0.25	183.11	5.3	1.76	0.09	0.78	0.98	64.57	0.2	139.76	4.15	1.61	0.05
D4	1	0.78	71.74	0.21	157.38	4.07	1.81	0.08	0.64	0.61	51.28	0.16	124.77	3.2	1.26	0.05
D5	1	1.14	93.96	0.3	176.78	4.92	1.73	0.09	0.85	1.09	71	0.21	146.03	3.52	1.48	0.08
Ave	1.03	1.06	84.09	0.23	181.16	5.06	1.75	0.09	0.78	0.82	60.67	0.18	136.02	3.99	1.42	0.07
Sd	0.08	0.16	8.89	0.07	20.97	0.61	0.04	0.01	0.14	0.31	11.25	0.04	15.80	0.61	0.17	0.02

Concluding Remarks

- RVFL has better performance than SVM for 12 h ahead forecasting; comparable as ANN and RF
- RVFL has fast training time as RF but has the fastest testing time

Outline

Conclusion I

- Introduction of time series characteristics
- Review of ensemble forecasting methods

EMD based method

- EMD-ANN/SVR with input vector reconstruction
- Significant improvement over SVR
- Outperformed two benchmark methods
- Noise assisted ensemble EMD improved the performance of SVR for wind speed forecasting, best method is CEEMDAN-SVR
- EEMD-SVR is the best for solar irradiance forecasting

Conclusion II

RVFL network

- Direct input output connections have significantly improved the performance
- No evidence of hidden layer bias
- Advantageous over hierarchical forecasting method
- Outperformed ANN, comparable performance as SVR and RF
- Short computational time

- For higher accuracy: ensemble EMD, e.g. CEEMDAN-SVR
- Rapid updating, fast training: RVFL network
- IoT applications: RVFL network

Future Work

- Multivariate datasets, multiple renewable energy sources
- 2-dimensional EMD
- Ensemble of RVFL network
- Resource constrained RVFL network

Bibliography I

Arthur Bossavy, Robin Girard, and Georges Kariniotakis. “A Novel Methodology for comparison of different wind power ramp characterization approaches”. In: *EWEA 2013-European Wind Energy Association annual event*. 2013, pp. 1–6.

C. L. Philip Chen. “A rapid supervised learning neural network for function interpolation and approximation”. In: *IEEE Trans. Neural Netw.* 7.5 (Sept. 1996), pp. 1220–1230.

Thomas G. Dietterich. “Proc. Multiple Classifier Systems (MCS’00)”. In: Berlin, Heidelberg, 2000. Chap. Ensemble Methods in Machine Learning, pp. 1–15.

Harris Drucker. “Improving regressors using boosting techniques”. In: *Proc. International Conference on Machine Learning (ICML1997)*. Vol. 97. 1997, pp. 107–115.

N.E. Huang et al. “The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis”. In: *Proc. Royal Society London A* 454 (1998), pp. 903–995.

Zhonghe Han and Xiaoxun Zhu. “Training Set of Support Vector Regression Extracted by Empirical Mode Decomposition”. In: *Proc. Asia-Pacific Power and Energy Engineering Conference (APPEEC’11)*. Wuhan, China, 25-28 03 2011, pp. 1–4.

Donghoh Kim and Hee-Seok Oh. “EMD: A package for empirical mode decomposition and Hilbert spectrum”. In: *The R Journal* 1.1 (2009), pp. 40–46.

D. Opitz and R. Maclin. “Popular ensemble methods: An empirical study”. In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 169–196.

Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. “Learning and generalization characteristics of the random vector functional-link net”. In: *Neurocomputing* 6.2 (1994), pp. 163–180.

Bibliography II

Ye Ren, Xueheng Qiu, et al. "Detecting Wind Power Ramp with Random Vector Functional Link (RVFL) Network". In: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL2015)*. Cape Town, South Africa, Dec. 2015.

Ye Ren, P. N. Suganthan, N. Srikanth, and G. Amaratunga. "Single Hidden Layer Neural Networks with Random Weights for Short-term Electricity Load Demand Forecasting". In: *Information Sciences* (2016). DOI: 10.1016/j.ins.2015.11.039.

Ye Ren, P. N. Suganthan, and N. Srikanth. "A Novel Empirical Mode Decomposition With Support Vector Regression for Wind Speed Forecasting". In: *IEEE Trans. Neural Netw. Learn. Syst.* PP (Sept. 2014). DOI: 10.1109/TNNLS.2014.2351391.

Ye Ren, P. N. Suganthan, and N. Srikanth. "A comparative study of empirical mode decomposition-based short-term wind speed forecasting methods". In: *IEEE Trans. Sustain. Energy* 6.1 (Jan. 2015), pp. 236–244.

Ye Ren, P. N. Suganthan, and N. Srikanth. "Ensemble methods for wind and solar power forecasting: a state-of-the-art review". In: *Renewable and Sustainable Energy Reviews* 50 (Oct. 2015), pp. 82–91.

Ye Ren, Le Zhang, and P. N. Suganthan. "Ensemble Classification and Regression –Recent Developments, Applications and Future Directions". In: *IEEE Comput. Intell. Mag.* (2016). DOI: 10.1109/MCI.2015.2471235.

Wouter F Schmidt, Martin A Kraaijveld, and Robert PW Duin. "Feedforward neural networks with random weights". In: *Proc. IEEE International Conference on Pattern Recognition Conference B: Pattern Recognition Methodology and Systems*. 1992, pp. 1–4.

S.S. Soman et al. "A review of wind power and wind speed forecasting methods with different time horizons". In: *Proc. North American Power Symposium (NAPS'10)*. Sept. 2010, pp. 1–8.

Bibliography III

M.E. Torres et al. "A complete ensemble empirical mode decomposition with adaptive noise". In: *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'11).* 1520-6149. Prague, Czech Republic, 22-27 05 2011.

Fei Wang et al. "Short-term solar irradiance forecasting model based on artificial neural network using statistical feature parameters". In: *Energies* 5 (2012), pp. 1355–1370.

Zhaohua Wu and Norden E. Huang. "Ensemble empirical mode decomposition: a noise-assisted data ananlysis method". In: *Advances in Adaptive Data Analysis* 1.1 (2009), pp. 1–41.

Lin Ye and Peng Liu. "Combined Model Based on EMD-SVM for Short-term Wind Power Prediction". In: *Proc. Chin. Soc. Elec. Eng.* Vol. 31. 31. May 2011, pp. 102–108.

J.-R. Yeh, J.-S. Shieh, and N. E. Huang. "Complimentary ensemble empirical mode decomposition: a novel noise enhanced data analysis method". In: *Advances in Adaptive Data Analysis* 2.2 (2010), pp. 135–156.

List of Publications I

- **Y. Ren**, P. N. Suganthan, "Empirical comparison of bagging-based ensemble classifiers," in: *Proc. International Conference on Information Fusion (Fusion'12)*, Singapore, Jul. 2012, pp. 917–924.
- **Y. Ren**, P. N. Suganthan, "A kernel-ensemble bagging support vector machine," in: *Proc. International Conference on Intelligent Systems Design and Applications (ISDA'12)*, Brunei, Oct. 2012.
- **Y. Ren**, P. N. Suganthan, N. Srikanth, "Wind speed forecasting: a comparison between statistical approach and learning-based approach," in: *Smart Microgrids, New Advances, Challenges and Opportunities in the Actual Power Systems*, US:NOVA, 2013.
- **Y. Ren**, L. Zhang, P. N. Suganthan, "K-nearest neighbor based bagging SVM pruning," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'13)*, Singapore, Apr. 2013.
- L. Zhang, **Y. Ren**, P. N. Suganthan, "Instance based random forest with rotated feature space," in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'13)*, Singapore, Apr. 2013.
- **Y. Ren**, P. N. Suganthan, N. Srikanth, "A hybrid ARIMA-DENFIS method for wind speed forecasting," in: *Proc. IEEE International Conference on Fuzzy Systems (Fuzz'13)*, India, Jul. 2013.
- **Y. Ren**, P. N. Suganthan, and N. Srikanth, "A comparative study of empirical modedecomposition-based short-term wind speed forecasting methods," *IEEE Trans. Sustain.Energy*, vol. 6, no. 1, pp. 236–244, Jan. 2015.
- **Y. Ren** and P. N. Suganthan, "EMD-kNN models for wind speed forecasting," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 176185, 2014.
- **Y. Ren**, P. N. Suganthan, and N. Srikanth, "A novel empirical mode decomposition with support vector regression for windspeed forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, 2015, doi: 10.1109/TNNLS.2014.2351391.

List of Publications II

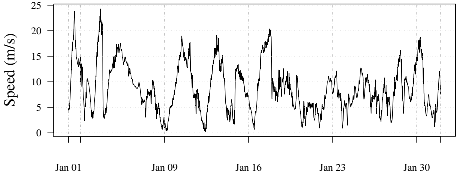
- **Y. Ren**, X. Qiu, and P. N. Suganthan, “EMD based adaboost-BPNN method for windspeed forecasting,” in *Proc. IEEE Symposium on Computational Intelligence and En-semble Learning (CIEL’14)*, Orlando, US, Dec. 2014.
- D. Laha, **Y. Ren** and P. N. Suganthan, “Modeling of Steelmaking Process with Effective Machine Learning Techniques, ” *Expert Systems With Applications*, vol. 42, pp. 4687-4696, 2015.
- **Y. Ren**, P. N. Suganthan, N. Srikanth, “Ensemble methods for wind and solar power forecasting – a state-of-the-art review,” *Renewable and Sustainable Energy Reviews, Elsevier*, vol.50, pp. 82-91, 2015.
- **Y. Ren**, X. H. Qiu, P. N. Suganthan, N. Srikanth and G. Amaratunga, “Detecting Wind Power Ramp with Random Vector Functional Link (RVFL) Network,” in: *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL’15)*, Cape Town, South Africa, Dec. 2015.
- **Y. Ren**, L. Zhang, and P. N. Suganthan, “Ensemble Classification and Regression – Recent Developments, Applications and Future Directions,” *IEEE Comput. Intell. Mag.*, 2016, doi: 10.1109/MCI.2015.2471235.
- **Y. Ren**, P. N. Suganthan, N. Srikanth and G. Amaratunga, “Single Hidden Layer Neural Networks with Random Weights for Short-term Electricity Load Demand Forecasting,” *Information Sciences, Elsevier*, 2015, doi: 10.1016/j.ins.2015.11.039.

The End

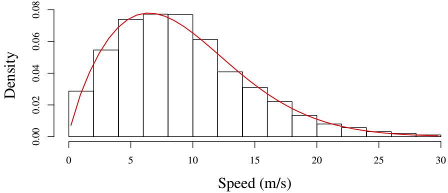
Thank you!

Wind speed time series plot

Plot of wind speed time series

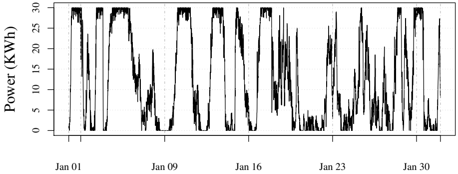


Histogram of wind speed time series

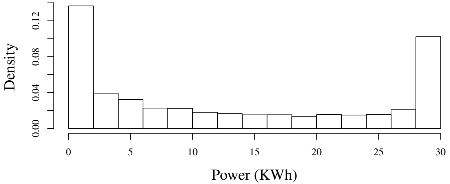


Wind speed time series plot

Plot of wind power time series

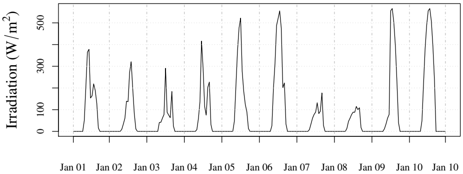


Histogram of wind power time series

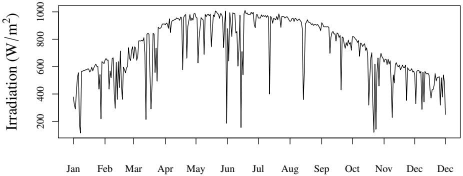


Wind speed time series plot

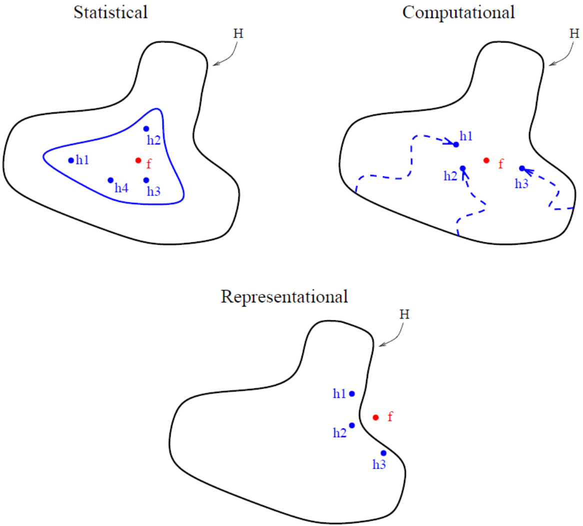
Plot of solar irradiance time series



Daily maximum of solar irradiance time series



Reason for good ensembles



Statistical Forecasting Methods

Exponential Smoothing

$$\hat{x}(t+1) = \alpha x(t) + \alpha(1-\alpha)x(t-1)$$

α: exponentially decreasing weight over time

Statistical Forecasting Methods

ARIMA

$$\nabla^d x(t) = \underbrace{\sum_{i=1}^{p} \phi_i \nabla^d x(t-i)}_{\text{AR Term}} + w_t + \underbrace{\sum_{j=1}^{q} \theta_j w(t-j)}_{\text{MA Term}}$$

$$\nabla^d x = [x(t)-x(t-1)]-[x(t-1)-x(t-2)]-\cdots-[x(t-d+1)-x(t-d)]$$

Statistical Forecasting Methods

ARIMA Order

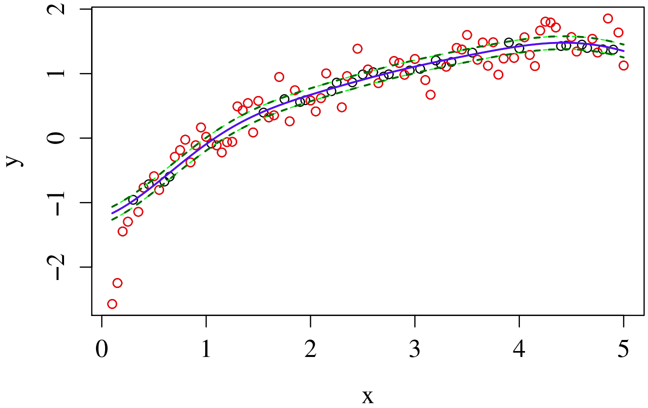
		PACF	
		Decay	Cut-off at Lag p
ACF	Decay	—	ARMA(p,d,0)
	Cut-off at Lag q	ARIMA(0,d,q)	ARIMA (p,d,q)

Statistical Forecasting Methods

AIC and BIC

$$\mathrm{AIC} = \log \frac{\sum_{t=k}^{n}(x(t) - \bar{x})^2}{n} + \frac{n + 2k}{n}$$
$$\mathrm{BIC} = \log \frac{\sum_{t=k}^{n}(x(t) - \bar{x})^2}{n} + \frac{k \log n}{n}$$

SVR



SVR

$$\min \frac{1}{2}||w||^2 + C\sum_i(\xi_i + \xi_i^*)$$

subject to:

$$y_i - w^T x_i - b \leq \epsilon + \xi_i$$
$$w^T x_i + b - y_i \leq \epsilon + \xi_i^*$$
$$\xi_i^{(*)} \geq 0$$

where C is the cost factor for trade-off between flatness and tolerance. $\xi_i^{(*)}$ is the slack variable.

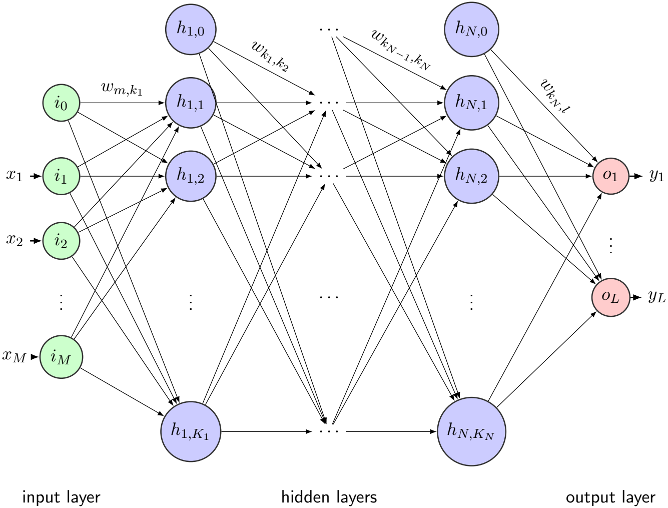
SVR

$$\text{Polynomial Kernel: } K(x_i, x_j) = (\gamma x_i^T x_j)^d$$

$$\text{RBF Kernel: } K(x_i, x_j) = e^{\gamma |x_i - x_j|^2}$$

$$\text{Sigmoid Kernel: } K(x_i, x_j) = \tanh(\gamma x_i^T x_j)$$

Neural Network



Neural Network

$$h_{1,k_1} = f(\sum_{m=0}^{M} w_{m,k_1} i_m),\ \forall k_1 \in \{1, \dots, K_1\}$$

$$h_{n,k_n} = f(\sum_{k_{n-1}=0}^{K_{n-1}} w_{k_{n-1},k_n} h_{n-1,k_{n-1}}),\ \forall k_n \in \{2, \dots, K_n\}$$

$$o_l = \sum_{k_N=0}^{K_N} w_{k_N,l} h_{N,k_N},\ \forall l \in \{1, \dots, L\} \tag{1}$$

Neural Network

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

AdaBoost[20] EMD

$\{\mathbf{x}\} := m \times n$: training data
y: corresponding future value
I_m: maximum number of iterations
$f(\cdot)$: base learning algorithm
$\mathbf{w} := m \times 1$: weight vector

$\mathbf{w}^1 = \{w_i^1 | 1/m, 1 \le i \le m\}$
$\overline{L} = 0$

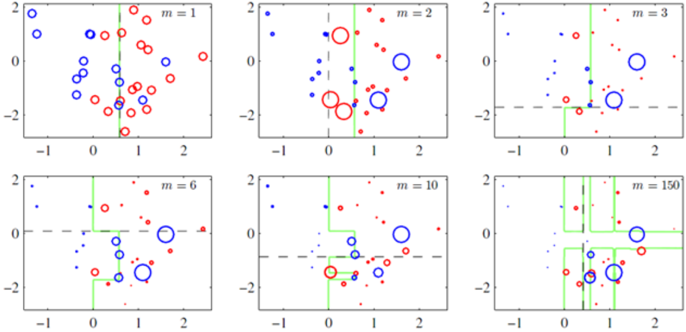
FOR $t = 1$ to I_m & $\overline{L} < 0.5$
    Sample with replacement from $\mathbf{x}$ with distribution $\mathbf{w} \Rightarrow \mathbf{x}^t$
    Obtain a trained model $h_t = f(\mathbf{x}^t)$
    Calculate maximum loss $L_m = \max |y_i - h_t(x_i)|$
    Calculate individual loss $L_i = \frac{|y_i - h_t(x_i)|}{L_m}$
    Calculate weighted loss $\overline{L} = \sum_{i=1}^{m} L_i w_i^t$
    Update weight vector $w_i^{t+1} = \frac{w_i^t \beta_t^{(1-L_i)}}{Z_t}$,
    where $\beta_t = \frac{\overline{L}}{1-\overline{L}}$ and Z_t is a normalization factor.
END

RETURN $\hat{y}^* = \text{median}(h_t(x^*) * \ln \frac{1}{\beta_t})$, $t = 1 \cdots I_m$.
where x^* is a testing data and $\hat{y}^*$ is the predicted output.

20 Drucker 1997.

AdaBoost[20] EMD

AdaBoost Example



20 Drucker 1997.

AdaBoost[20] EMD

AdaBoost Characteristics

- Weak learners
- Diversity
- Sequential
- Apply AdaBoost to each decomposed series

[20] Drucker 1997.

AdaBoost-EMD-ANN Performance[21]

Nemenyi Test

Nemenyi Test of AdaBoost-EMD-ANN vs Benchmarks

Horizon (h)	Persistent	AdaBoost-RT	ANN	AdaBoost-ANN	EMD-ANN	ABEMD-RT
1	5.5e-08	0.36529	0.17533	7.5e-05	0.11252	0.79266
3	3.0e-07	0.48706	0.03479	0.00856	0.00052	1.00000
5	4.7e-05	0.61549	0.00028	0.14133	7.0e-06	1.00000

Concluding Remarks

AdaBoost framework improved EMD-ANN's performance

21 Ren, Qiu, et al. 2015.

ELIA power ramp data

	6 Hour				12 Hour			
	Training		Testing		Training		Testing	
	+1	-1	+1	-1	+1	-1	+1	-1
Γ_{ext}								
D1	64	387	32	184	153	292	71	145
D2	332	136	55	168	252	210	111	112
D3	103	365	54	169	220	242	120	103
D4	77	340	58	144	152	259	75	127
D5	78	389	58	165	164	297	107	116
Γ_{end}								
D1	62	389	26	190	124	321	54	162
D2	130	228	45	178	237	225	85	138
D3	93	375	60	163	167	295	115	108
D4	82	335	78	124	150	261	62	140
D5	84	383	59	164	144	317	117	106