An Introduction to Solar Energy Forecasting

Jitkomut Songsiri

Smart Grid Research Unit Center
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

CUEE

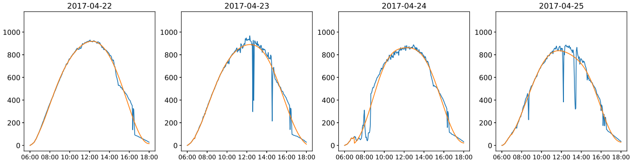
Jul 25, 2020

Outline

Importance of solar energy forecasting

- the growth of power consumption leads to more renewable energy production
- the need of energy management in all power units

challenge: solar power highly depends on solar irradiance which varies upon weather condition



blue line is the solar irradiance; orange line is from a clear-sky model

What to forecast ?

Forecasted variables

- **solar power**
- **solar irradiance**
- wind power
- electrical load (in buildings, regions)
- meteoreological variables (temperature, relative humidity)

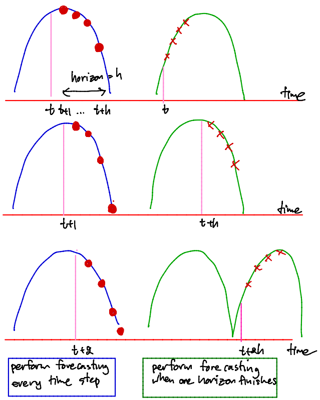
Problem statements

forecasting configurations:

- horizon of forecasts
- resolution of forecasts
- time step of forecast computing
- sliding/non-sliding

these specifications lead to

- different problem statements (easy or hard problems)
- different implementation schemes



Typical horizons in solar forecasting

forecasting specifications are determined from application point of view

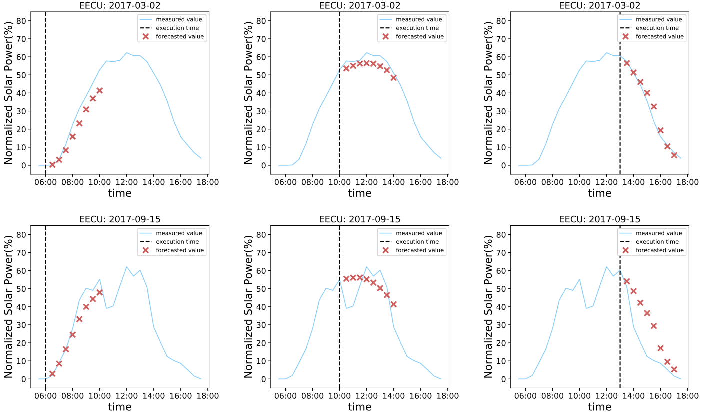
forecasting	horizon	applications	input
nowcasting	5-60 mins	spinning reserve, demand response	cloud
intra-day	1-6 hours	load-following	weather forecasts
short-term	1-3 days	planing, unit commitment	weather forecasts

typical input variables:

- weather variables and solar irradiance (from sensors)
- solar power generation (from sensors)
- weather forecasts (numerical weather prediction – NWP)
- synthesized inputs (e.g. clear-sky irradiance, exponential moving avarage)

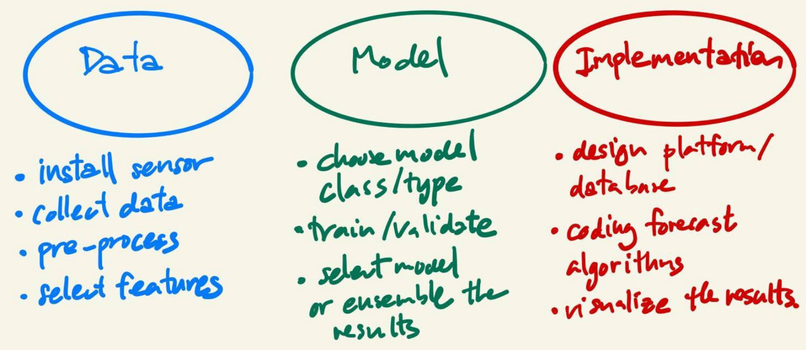
Example of intra-day forecasting at CUEE

problem: give forecasts at 6:00 to 17:30 with 4-hour horizon, resolution of 30 mins; forecast at every 30 min



Elements in solar forecasting

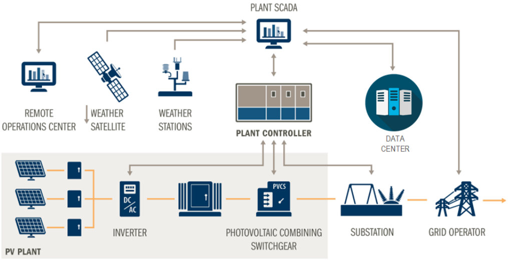
Key elements

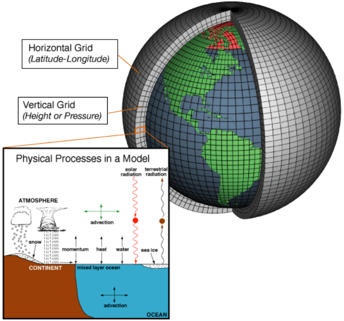


involves disciplinary fields: remote sensing, electrical engineering, atmospheric science, data analytics, system engineering, programming

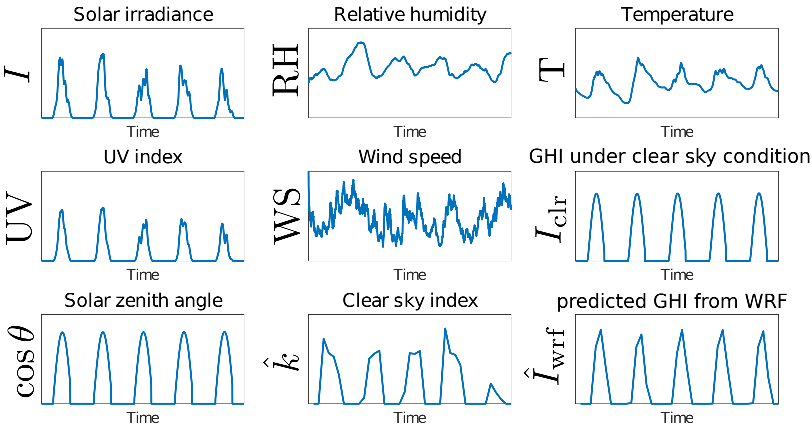
Data

sources of data: sensor, deterministic models, numerical weather prediction



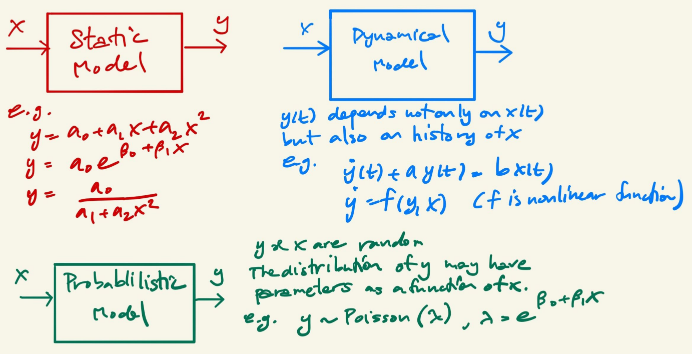


Time series example



Model

model is a relationship between inputs and outputs (typically is a mathematical mapping)



other names

- inputs: predictors, features, explanatory variables, independent variables
- outputs: response variables, explained variables, dependent variables

Implementation

a well-designed ICT infrastructure is needed

- database
- processing units (tasks to be done)
- data flow
- visualization

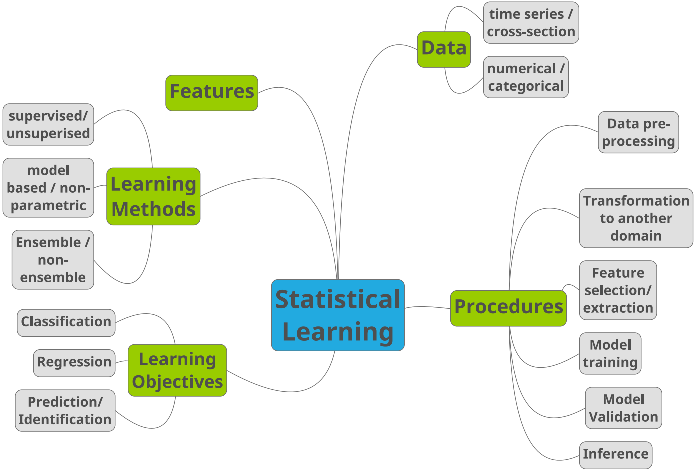
then select hardwares to serve the proposed structure

note: issues in practical implementation

- data are not perfect (missing, errors, etc.)
- data come with delay (cannot predict at the exact specified time)

Statistical learning methods

Elements in statistical learning



Methods

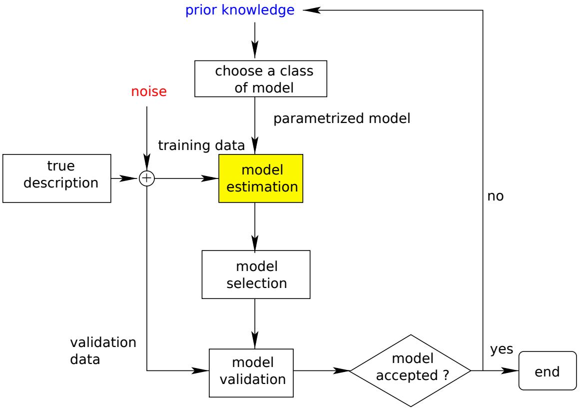
categorized based on how to guide the learning process

- **supervised learning**
  - the presence of outcome variable is used to guide the learning process
  - examples are regression, support vector machine, neural network
- **unsupervised learning**
  - we observe only the features (no measurements of outcome) and describe how the data are clustered
  - examples are k-means clustering, k-nearest-neighbor, principal components analysis

Procedures in Statistical Learning

- data pre-processing: missing-value imputation, removing artifacts, normalization, preparation of data sets for experiments
- feature selection/extraction: to choose relavant input variables for the output
- model training: this is to estimate f from (X, Y) data where $Y = f(X)$
  - this steps involve varying complexity of models
  - one obtain many candidate models in this step
- model validation: compare candidate model performance evaluated on unseen data (validation set)
  - example of methods: leave-one-out cross-validation, k-fold cross-validation, residual analysis, white-ness test
- inference: use the selected model to further infer about the learning goal

Flow chart of training and validation process



Solar forecasting techniques

Typical features

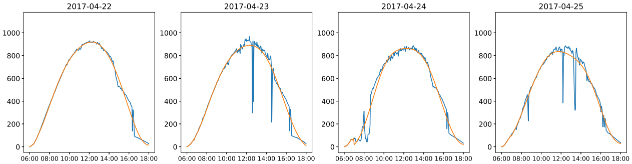
response variable: $I(t+1)$ or $P(t+1)$ (irradiance or power at the next time step)

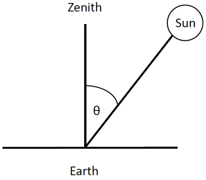
- features from measurement:
  - $I(t), I(t-1), \ldots, I(t-7)$
  - $I^{(d-1)}(t+1)$ (from the previous day)
  - $T(t), \mathrm{RH}(t), \mathrm{UV}(t), \mathrm{WS}(t)$
- determinisic features:
  - solar zenith angle $\cos(\theta(t+1))$
  - clear-sky irradiance $I_{\mathrm{clr}}(t+1)$
  - exponential moving average $I_{\mathrm{ema}}(t+1)$ (but use only past data to calculate)

methods of feature selection: correlation, partial correlation, step-wise regression

Clear-sky model

clear-sky models give irradiance under clear-sky assumption





- static models; many have been proposed
- Ineichen model [IPC13]

$$I(t) = a_1 I_0 \cos(\theta(t)) e^{-a_2 \mathrm{AM}(t)(f_{\mathrm{h1}} + f_{\mathrm{h2}}(T_L - 1))}$$

- clear-sky index: $k(t) = I(t)/I_{\mathrm{clr}}(t)$ indicates the degree of clear-sky condition

Linear regression

simplest model; typically used as a baseline

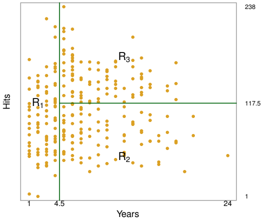
$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

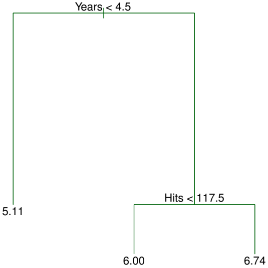
response	irradiance	power
predictors	lagged irradiance	lagged power irradiance and lags
common predictors	temperature, relative humidity solar zenith angle, clear-sky irradiance	

extensions:

- adaptive coefficients (known as model output statistic – MOS)
- predictors can be chosen by stepwise-regression

Regression trees [Jam+13]





- a regression tree consists of a series of splitting rules
- predictor and response data are partitioned according to the rules
- the predicted response of each group is the **mean** of the response values
- example of 'Hitters' data where the response is player's salary
- tree terminology:
  - leaves or terminal nodes: regions $R_1, R_2, \ldots$
  - internal nodes: points where predictor space is split
  - branch: segments of trees that connect the nodes

Recursive binary splitting (top-down approach)

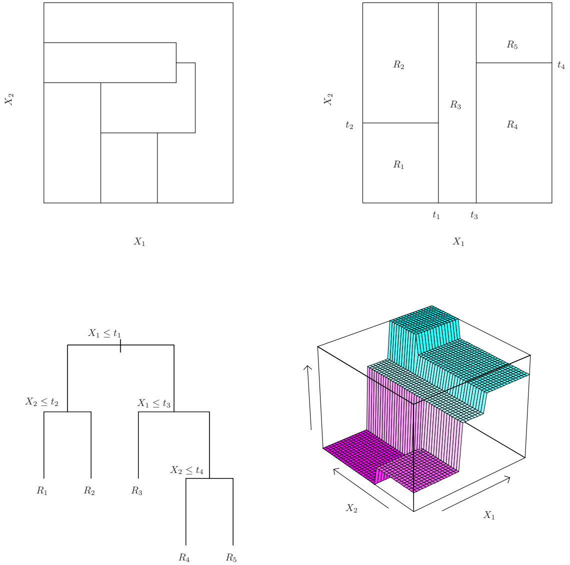
the goal is to find boxes $R_1, R_2, \ldots, R_J$ that minimize the RSS

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2, \quad \hat{y}_{R_j} \text{ is the mean response}$$

- begins at the top of the tree and successively splits the predictor space
- each split is indicated via two new brances further down

the pair of half-planes: $R_1(j,s) = \{X | X_j < s\}$ and $R_2(j,s) = \{X | X_j \geq s\}$
we seek the value of j and s that minimizes

$$\sum_{i: x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$



- select X_j and cutpoint s such that splitting leads to greatest reduction in RSS
- repeat the process but split one of the two previously identified regions
- continues until a stopping criterion is done (e.g. until no more regions contains more than five observations)

Bagging

procedure: bootstrap and average

- resample to get B boostrap samples from the training set and calculate

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

- average all predictions: $\hat{f}_{\text{bag}}(x) = (1/B)\sum_{b=1}^{B} \hat{f}^{(b)}(x)$

why bagging is useful for decision trees?

- since decision trees typically suffer from *high variance*, we can apply the bagging which is averaging step to reduce the variance
- recall: given a set of n independent observations $Z_1, \ldots, Z_n$ with σ^2
- variance of $\bar{Z}$ is given by σ^2/n (variance reduction)

Random forest

average the decision trees built from using only **subset** of predictors

- building trees: random sample of m predictors as split candidates from full set of p predictors (choose $m \approx \sqrt{p}$)

$$\hat{f}^{(1)}(x), \hat{f}^{(2)}(x), \ldots, \hat{f}^{(B)}(x)$$

  (unlike bagging, all theese trees may not look similar to each other)

- averaging all predictions from **decorrelated** trees

main difference of RF from bagging

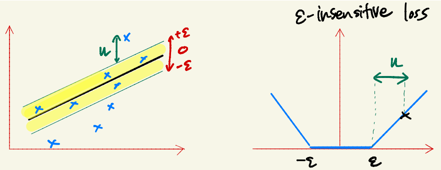
- averaging many highly correlated quantities may not lead to a large variance reduction (as in bagging, all trees may look very similar)
- some trees in RF do not contain the same set of predictors, so all trees are decorrelated, making average of the trees less variable and more reliable

Support vector machine [SS02]

given data sets $\{(x_i, y_i)\}_{i=1}^N$ (feature, response)
a support vector regression problem is to solve

$$\begin{array}{ll} \text{minimize} & (1/2)\|w\|^2 + C\sum_{i=1}^n (u_i + v_i) \\ \text{subject to} & y_i - w^T x_i - b \leq \epsilon + u_i \\ & w^T x_i + b - y_i \leq \epsilon + v_i \\ & u_i, v_i \geq 0 \end{array}$$

with variables w, b (hyperplane parameters) and u, v (slack variables)



- (x, y) have an approximate linear relationship (explained by a hyperplane); determined by the yellow tube and parametrized by ϵ
- the number of $u_i, v_i > 0$ tell us how much we allow the pair (x_i, y_i) stay outside the tube

Applying SVR to solar forecasting

forecast output (vector): $\hat{P}(t+1), \hat{P}(t+2), \ldots, \hat{P}(t+k)$ where $k = 8$

since output of SVR is scalar, we break down to have sub-model of SVR

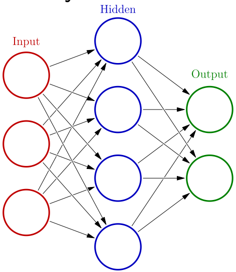
- example of forecasts at 6:00-9:00
- response y: $\hat{P}(t+k)$
- features x:

$$I(t), P(t), P^{(d-1)}(t+k), \cos(\theta(t+k)), I_{\mathrm{clr}}(t+k)$$

(choice of features depend on the time of forecasts – more on this later)

ANN

a fully-connected networks used for explaining nonlinear relationships



- input layer: affine transformation (output $= w^T x + b$)
- hidden layers: a series of nonlinear activation function ($\tanh$, sigmoid, ReLU, and more) composite with affine transformation
- output layer: affine transformation
- model parameters: weight and bias terms in all layers

ANN can be used for both classification and regression problems

Fitting neural networks [WBK20]; [HTF09]

some issues you need to know

- selecting the right network architecture
  - use prior knowledge about the application domain
  - define an appropriate input/target mapping – e.g. use ANN as a static or dynamic model in your application ?
  - capacity gained by adding new unit is typically smaller relative to capacity gained by adding new hidden layers
- selecting the cost objective function: ℓ_1 or ℓ_2
- training neural network: starting values, multiple minima, overfitting, scaling of the inputs
- algorithm setting: algorithm choice, effect of algorithm parameters (batch, epoch), stopping criterions

Applying ANN to solar forecasting

forecast output (vector): $\hat{P}(t+1), \hat{P}(t+2), \ldots, \hat{P}(t+8)$

- inputs:
  - current and past irradiance

$$I(t), I_{\text{ema}}^{(d-1)}(t+1), \ldots, I_{\text{ema}}^{(d-1)}(t+8)$$

    where $I_{\text{ema}}^{(d-1)}(t) = \alpha I^{(d-1)}(t) + (1-\alpha) I_{\text{ema}}^{(d-2)}(t)$ and $\alpha \in [0.8, 1)$
  - past solar power: $P(t)$
  - ambient temperature: $T(t)$
- model structure: 3 hidden layers, 128 neurons
- test on 10-fold cross validation
- algorithm: mini-batch Stochastic gradient (Adam optimizer), stopping criterion = 'validation loss decreases within MAX no. of epoch'

Solar forecasting techniques in literature

the literature is huge (and keeps growing)

models can be parametrized in three categories

- physical methods: NWP (such as GFS, WRF, ECMWF)
- statistical methods: time series, regression (linear and nonlinear)
- machine learning: NN, SVR, RF, wavelet, k-mean, SOM, fuzzy

check out references:

[IPC13]; [YC14]; [Ant+16]; [Voy+17]; [Das+18]; [Yan+18]

Model configurations

Approaches for differentiating frameworks [Suk19]

different combinations of applying existing tools can be based on

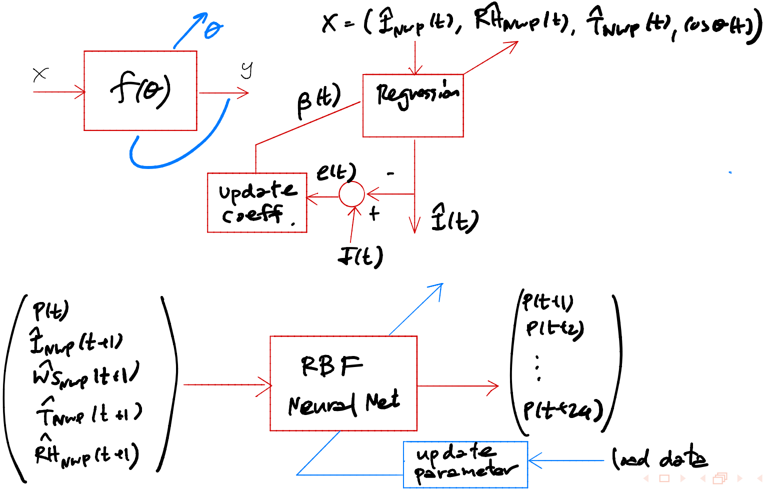
- adaptive models
- parallel models
- cascade models
- bias correction model

denote a relationship between predictors and response as $y = f(x;\theta)$

- f can be nonlinear from any methods
- θ is model parameter

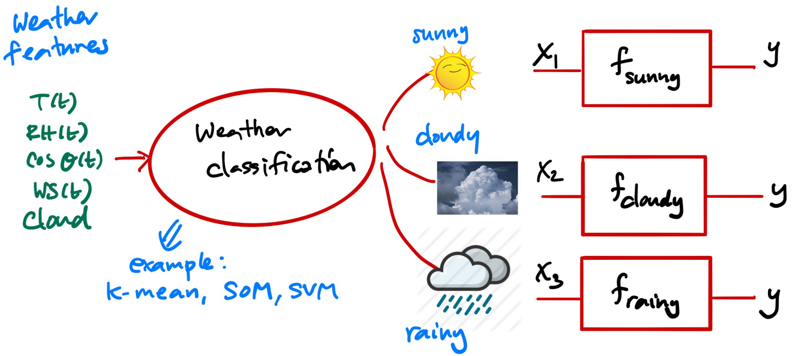
Adaptive models

model f can be trained off-line to obtain θ but use real-time data to update θ adaptively



Parallel models: weather classification

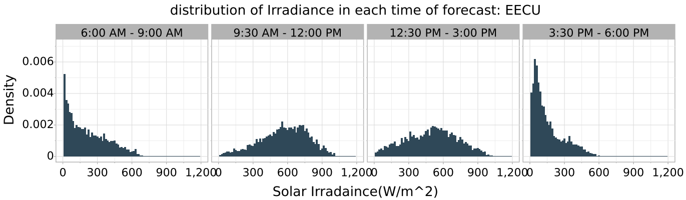
assumption: solar power varies upon different weather conditions



switch forecasting models based on a classified weather condition (each model has different features and different parameters)

Parallel models: time split

assumption: weather uncertainty varies according to different hours of the day



$$X_{set\,1} \longrightarrow \boxed{\text{Model for 6:00}} \longrightarrow \hat{P}(6{:}00)$$

$$X_{set\,2} \longrightarrow \boxed{\text{Model for 7:00}} \longrightarrow \hat{P}(7{:}00)$$

$$\vdots \qquad\qquad \vdots$$

$$X_{set\,13} \longrightarrow \boxed{\text{Model for 18:00}} \longrightarrow \hat{P}(18{:}00)$$

sub-model responsible for each hour should be different and use different features

example of random forest model for CUEE solar forecasts

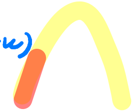
Intra-day forecasts: $\hat{P}(t+1), \hat{P}(t+2), \ldots, \hat{P}(t+8)$

X

$I(t), P(t), P^{(d-1)}(t+k)$
$\cos\theta(t+k), I_{clr}(t+k)$

Model for morning session (6:00-9:00)

$\hat{P}(t+k)$

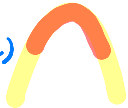


X

$I(t), I(t-1), \ldots, I(t-7)$
$P(t), P(t-1), \ldots, P(t-7)$
$P_{ema}(t), P^{(d-1)}(t+k)$
$\cos(\theta(t+k)), I_{clr}(t+k)$

Model for mid-day session (9:30-15:00)

$\hat{P}(t+k)$

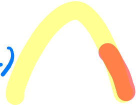


X

$I(t), P(t), P^{(d-1)}(t+k)$
$\cos(\theta(t+k)), I_{clr}(t+k)$

Model for evening session (15:30-17:00)

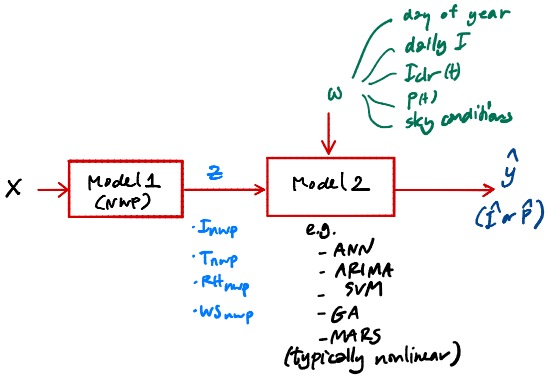
$\hat{P}(t+k)$



early morning and late evening data have less variation, so we use static features and only recent values of $I(t), P(t)$

Cascade model

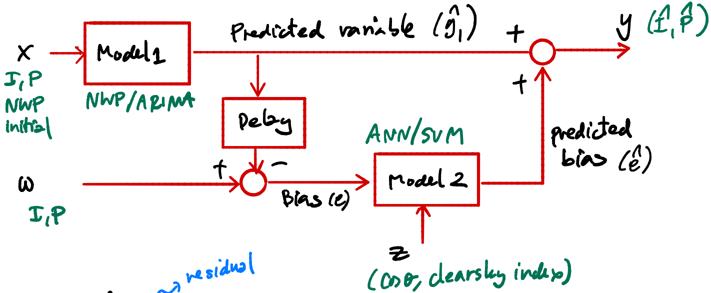
assumption: one model may not capture the whole nonlinear dynamics of I



use another nonlinear model to explain the rest

Bias correction model

assumption: use the second model to learn dynamics from the residual error of the first model



residual

$$y = \hat{y}_1 + e, \quad \hat{y}_1 = f_1(x)$$

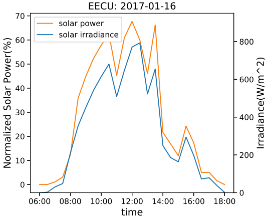
$$\hat{e} = f_2(z, e)$$

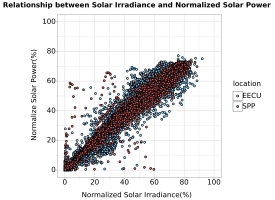
$$\hat{y} = \hat{y}_1 + \hat{e}$$ ← adjust to predicted bias

Illustrated results

Setting

source: CUEE solar rooftop (capacity of 8kW)

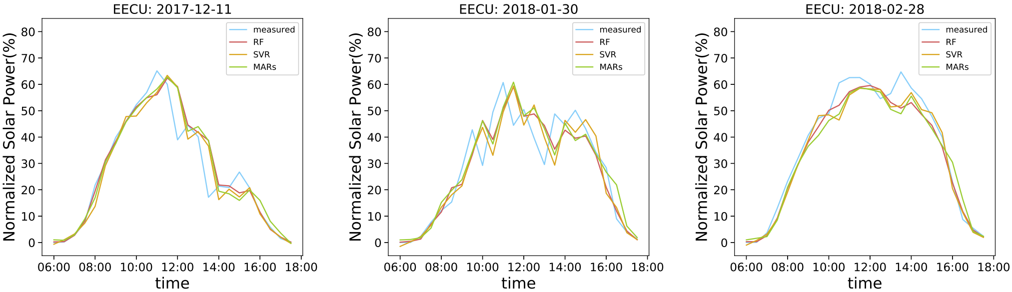




- $I(t), P(t), T(t)$ are downsampled to 30-min sampling
- give forecasts of 6:00,6:30,7:00,...,17:30
- models (RF,SVR,MARS,ANN) trained and tested using k-fold cross validation
- test the performance with normalized RMSE, MBE (by installed capacity)

Forecast results: time plot example

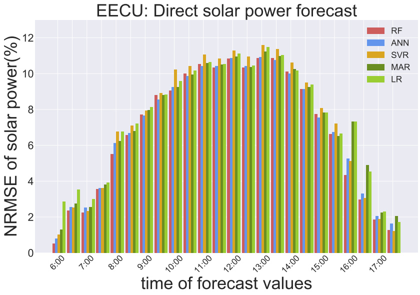
results contain good and bad performances

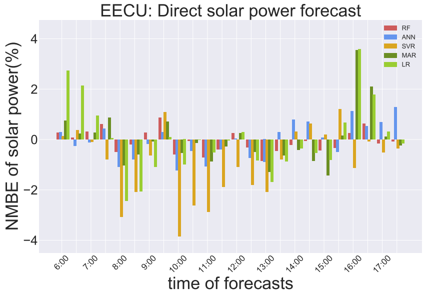


results shown are from one-step forecasts

Forecast results: each time of the day

performance indices split by time of the day



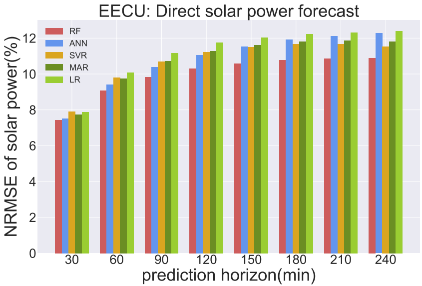


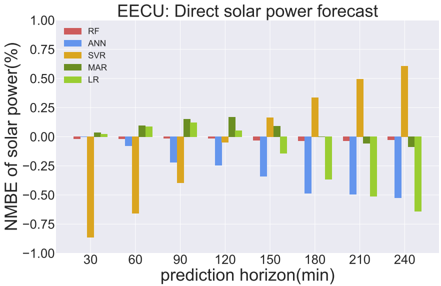
typical solar forecasting results

- less error in the early morning and late evening
- more variation of data during mid-day; lead to higher error
- in this experiment, RF gives the best performance

Forecast results: k-step

performance indices split by k-step predicted





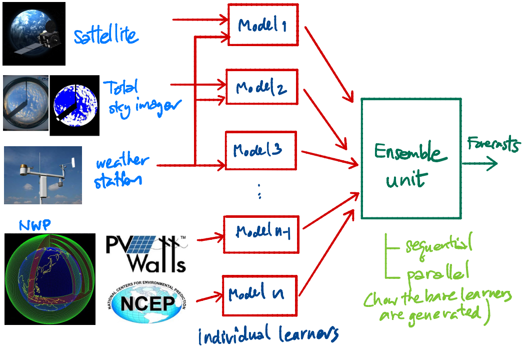
typical solar forecasting results

- higher error if we look further step ahead
- in this experiment, RF gives the best performance in both RMSE and MBE

Trends and open problems

Trends: ensemble forecasting

use different sources of input; trained many models and **ensemble**

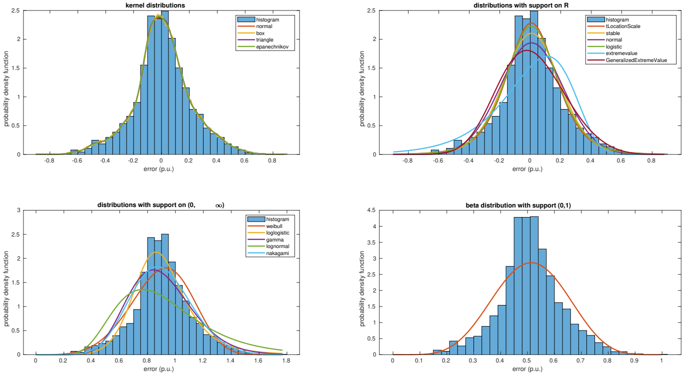


- in NWP, perturb initial conditions to get different forecasts and ensemble the results
- ensemble techniques from statistical learning: boosting, bagging, etc.

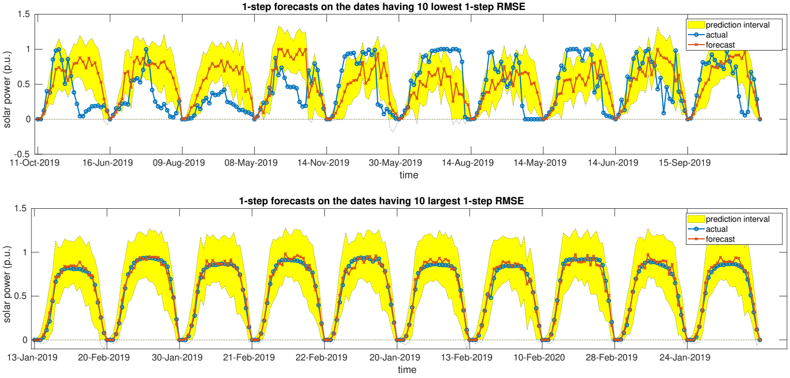
Trends: probablilistic forecasting [GPG16]; [Kho+11]; [DPP16]

give more **statistical properties** of the forecasts

- use probablilistic models in the training process
- use deterministic models but analyze statistical properties of forecast errors



Prediction interval



prediction interval of forecasts can be useful for operators

- the lower bound tells us to reserve some other source of generation to supply the load

Conclusion

- solar forecasting is an applied problem and requires knowledge from many disciplinary fields
- existing tools in statistical learning (regression, ML) have been applied extensively in literature; only twists in model configuration and how the data are pre-processed
- good forecasting results come from relevant inputs that explain fluctuated weather conditions, so an investment of good source of data or equipment is unavoidable (not presented here, e.g, satellite data, sky imagers, forecasts of NWP)

Authors

experimental results in this talk are based on final-year project of

Jitkomut Songsiri
jitkomut.s@chula.ac.th

Chayanont Potawananont
chayanont1997@hotmail.com

Sararut Pranonsatid
sararut.pra@gmail.com

two CUEE graduates from class of 2019

Websites

- Smart grid research unit
  `http://www.sgru.eng.chula.ac.th/`
- Jitkomut page
  `http://jitkomut.eng.chula.ac.th/`
- CUEE page
  `http://www.ee.eng.chula.ac.th/th/`

References I

- Antonanzas, J. *et al.* Review of photovoltaic power forecasting. *Solar Energy* **136,** 78–111 (2016).
- Das, U. *et al.* Forecasting of photovoltaic power generation and model optimization: A review. *Renewable and Sustainable Energy Reviews* **81,** 912–928 (2018).
- Dordonnat, V., Pichavant, A. & Pierrot, A. GEFCom2014 probabilistic electric load forecasting using time series and semi-parametric regression models. *International Journal of Forecasting* **32,** 1005–1011 (2016).
- Golestaneh, F., Pinson, P. & Gooi, H. Very short-term nonparametric probabilistic forecasting of energy generation: With application to solar energy. *IEEE Transactions on Power Systems* **31,** 3850–3863 (2016).
- Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* 2nd (Springer, 2009).

References II

- Inman, R., Pedro, H. & Coimbra, C. Solar forecasting methods for renewable energy integration. *Progress in energy and combustion science* **39**, 535–576 (2013).
- James, G., Witten, D., Hastie, T. & Tibshirani, R. *An introduction to statistical learning: with Applications in R.* (Springer, 2013).
- Khosravi, A., Nahavandi, S., Creighton, D. & Atiya, A. F. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on neural networks* **22**, 1341–1356 (2011).
- Schölkopf, B. & Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* (MIT press, 2002).
- Suksamosorn, S. *An Improvement of Photovoltaic Power Forecasting from Numerical Weather Prediction.* http://jitkomut.eng.chula.ac.th/group/supachai_thesis.pdf. MA thesis (Chulalongkorn University, 2019).

References III

- Voyant, C. *et al.* Machine learning methods for solar radiation forecasting: A review. *Renewable Energy* **105,** 569–582 (2017).
- Watt, J., Borhani, R. & Katsaggelos, A. *Machine learning refined: foundations, algorithms, and applications.* (Cambridge University Press, 2020).
- Yadav, A. & Chandel, S. Solar radiation prediction using Artificial Neural Network techniques: A review. *Renewable and Sustainable Energy Reviews* **33,** 772–781 (2014).
- Yang, D., Kleissl, J., Gueymard, C., Pedro, H. & Coimbra, C. History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining. *Solar Energy* **168,** 60–101 (2018).