Final Year Project

Load Prediction of Power Generators Using Semi-Supervised Learning

The Chinese University of Hong Kong
Department of Computer Science and Engineering
Jeon Cheol Su, 1155043327

April 2018

1 Introduction

With the advancement of technology in areas like IoT, high quality of data can be collected and managed in highly economical manner. The accumulated data can be used to solve different issues of interest through data analytics. Data analytics involve various disciplines such as statistics, machine learning, data mining, and visualization. Machine learning is a process of training a computer to give the ability to learn without being explicitly programmed. Machine learning is then used to train a function(model) for various tasks such as prediction and classification.

Machine learning methodologies can be generally divided into 3 parts, namely, supervised learning, unsupervised learning and semi-supervised learning. In supervised learning, a function(model) is trained from labeled training data. This function is then used to make a prediction when unseen data points are given. Elsewhere in unsupervised learning, a function is trained from unlabeled training data. Unsupervised learning is generally used for clustering(i.e. group similar data points into the same cluster) and dimensionality reduction(i.e. reduce the number of variables under consideration). The semi-supervised learning is the combination of both, where both labeled and unlabeled data points are exploited.

In traditional supervised learning scenario, in order to construct a well rounded generalized model for classification/ prediction, a large amount of labeled dataset is required. Unfortunately, in most practical machine learning and data mining applications, only small number of labeled data points are available since labeling data often requires human effort which can be costly. On the other hand, unlabeled data points are easily available and cheap, due to the aforementioned reason of advancement of IoT technology and a data crawling script can easily extract a huge amount of unlabeled data with relevant attributes. In fact, the combination of both labeled and unlabeled data points is the most common dataset setting in most real-life problems, such as classifying the web pages into categories, voice recognition, and protein sequence classification. Among few machine learning paradigms such as semi-supervised learning, transductive learning and active learning that exploits the unlabeled data points in the dataset, semi-supervised learning is used in this project in order to utilize the cheap and abundant unlabeled data points to effectively improve the model's performance that is developed only with supervised learning technique.

2 Problem set

Nowadays, with electricity demand escalating, there are two kinds of power generators to supply electric energy while causing much less pollution compared to other power generators: wind power generators and hydraulic power generators. However, these power generators are highly dependent on natural conditions and their location. For example, wind power generators depend on wind speed, surface pressure, and many others. Therefore, they cooperate with traditional power generators, such as thermal power generators in order to generate continuous energy and to prevent a blackout. In addition, different amount of energy is consumed in different time. To cope with the dependence on natural conditions and varying consumption, it is critical to predicting how many wind power generators are needed to be turned on.

2.1 Problem Setting

Assume there are $l + u$ instances of the status of wind power generators in an area, the on-off status at i^{th} timestamp of wind power generators is expressed as y_i. In the area where wind power generators are located, there are n attributes $\mathbf{x} \in \mathbb{R}^n$ which are used to predict the status of wind power generators. A set of history data with l labeled and u unlabeled instances of the status of wind power generators are collected. The formulation of the load prediction problem is as follows:

Problem. Given a set of history data $D = \{\mathbf{x}_i, y_i\}_{i=1}^{l} \cup \{\mathbf{x}_i\}_{i=u}^{l+u}$ with l known status of wind power generators and u unknown status of wind power generators, train a prediction function f with D to predict any unseen status of wind power generators when $\mathbf{x}$ is given, i.e. $y_j = f(\mathbf{x}_j)$ with maximum of $P(y_j|\mathbf{X},\mathbf{y})$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_{l+u}]$ and $\mathbf{y} = [y_1, y_2, y_3, ..., y_l]$.

2.2 Problem Solution

In this project, prediction models will be constructed with the dataset explained above. The value of the status of the power generators in the dataset refers to the percentage of power generators turned on out of all power generators. The value is a continuous numeric value and hence the model will be trained using regression.

Different prediction models will be constructed using different machine learning techniques. As mentioned above, the semi-supervised learning which exploits both the labeled and unlabeled data points will be used. Out of various semi-supervised learning algorithms, a co-training algorithm will be used. First, supervised learning techniques such as the support vector regression and random forest regression will each be co-trained with its alike respectively to produce the effect of semi-supervised learning. The co-training algorithm will be further discussed in the following section. In addition, other models will be constructed and trained with ordinary supervised learning techniques that are mentioned above (i.e. Support vector regression and random forest regression). These models will be used for comparing the performances of the models that are trained with just the labeled data points (supervised learning) and the models that are trained with both the labeled and unlabeled data points(semi-supervised learning) to identify any improvement in performance when the unlabeled data points are augmented during the training phase. Lastly, attributes that reflect the most significant contribution in predicting the load will be deduced (a feature of random forest regression).

3 Semi-supervised Learning

Semi-supervised learning is one of the machine learning paradigms which exploits both the labeled and unlabeled data points in the dataset. With the rapid progress of machine learning, especially the explosive bloom of statistical learning research, and the increasing requirement of exploiting unlabeled data, semi-supervised learning has become a hot topic in both machine learning and data mining. There are several effective approaches to semi-supervised learning and can be generally categorized into 3 paradigms. In the first paradigm, a generative model such as Naive Bayes classifier or a mixture of Gaussians is used for the classifier, and EM is employed to model the label estimation or parameter estimation process. In the second paradigm, unlabeled data is used to regularize the learning process in various ways. For example, a graph can be defined on the data set, where the nodes correspond to the labeled or unlabeled examples while the edges reflect the similarity between the examples; then, the label smoothness can be enforced over the graph as a regularization term. Representative approaches of this paradigm include. The third paradigm, i.e. co-training, is closely related to the work described in this paper, therefore here we introduce it with more details. [7]

4 Co-Training Algorithm

In this project, supervised learning techniques such as support vector regression and the random forest will utilize the co-training algorithm in order to exploit the unlabeled data points in the dataset and produce the effect of semi-supervised learning. The co-training algorithm is explained as follows:

"Two regressors, i.e. $h1$ and $h2$, are generated from L, each of which is then refined with the help of unlabeled examples that are labeled by the latest version of the other regressor. Here the support vector regression, random forest regression regressor is used as the base learner to instantiate $h1$ and $h2$, which labels a new instance through measuring the most confidently labeled sample and augmenting the newly labeled sample into the other regressor.

In order to choose appropriate unlabeled examples to label, the labeling confidence should be estimated such that the most confidently labeled example can be identified. Note that both active learning and semi-supervised learning try to select “valued” unlabeled examples to use. In active learning, the selected unlabeled example will be passed to an oracle to ask for its ground- truth label. Therefore, the unlabeled example on which the learner is with the least confidence is usually selected since it would be most valuable for improving the learner. While in semi-supervised learning, since there is no oracle that can be relied on, the unlabeled example on which the learner is with the most confidence is usually selected to be labeled.

Intuitively, the most confidently labeled example of a regressor should be with such a property such that the error of the regressor on the labeled example set should decrease the most if the most confidently labeled example is utilized. In other words, the most confidently labeled example should be the one which makes the regressor most consistent with the labeled example set. Thus, the mean squared error (MSE) of the regressor on the labeled example set can be evaluated first. Then, the MSE of the regressor utilizing the information provided by (x_u,$\hat{y}_u$) can be evaluated on the labeled example set, where x_u is an unlabeled instance while $\hat{y}_u$ is the real-valued label generated by the original regressor. Let Δu denote the result of subtracting the latter MSE from the former MSE. Note that the number of Δu to be estimated equals to the number of unlabeled examples. Finally, (x_u,$\hat{y}_u$) associated with the biggest positive Δu can be regarded as the most confidently labeled example. Then the most confidently labeled example $\tilde{x}$ is identified through maximizing the value of δx_u in equation 1,

$$\delta x_u = \sum_{x_i \in L} ((y_i - h(x_i))^2 - (y_i - h'(x_i))^2) \tag{1}$$

where L denotes the labeled dataset of the regressor, h denotes the original regressor while h' denotes the refined regressor which has utilized the information provided by (x_u, $\hat{y}_u$), $\hat{y}_u \in h(x_u)$. Then, choose the (x_u, $\hat{y}_u$) having the maximum of δx_u to be the most confidently labeled sample, and add it to the training set of other regressor being co-trained, $h2$. Repeat the process for the regressor $h2$. Note that the predictions can be made by any kinds of regressors. For example, in the project support vector regression is used. After using the two SVRs to select and label the unlabeled examples, we get two augmented labeled training sets. Then, after the models are fully trained, the predictions of these two SVRs regressors are averaged as the final prediction. The diagram below illustrates the overall process of the algorithm graphically. 1 The semi-supervised random forest can be implemented in the same way." [7][6]

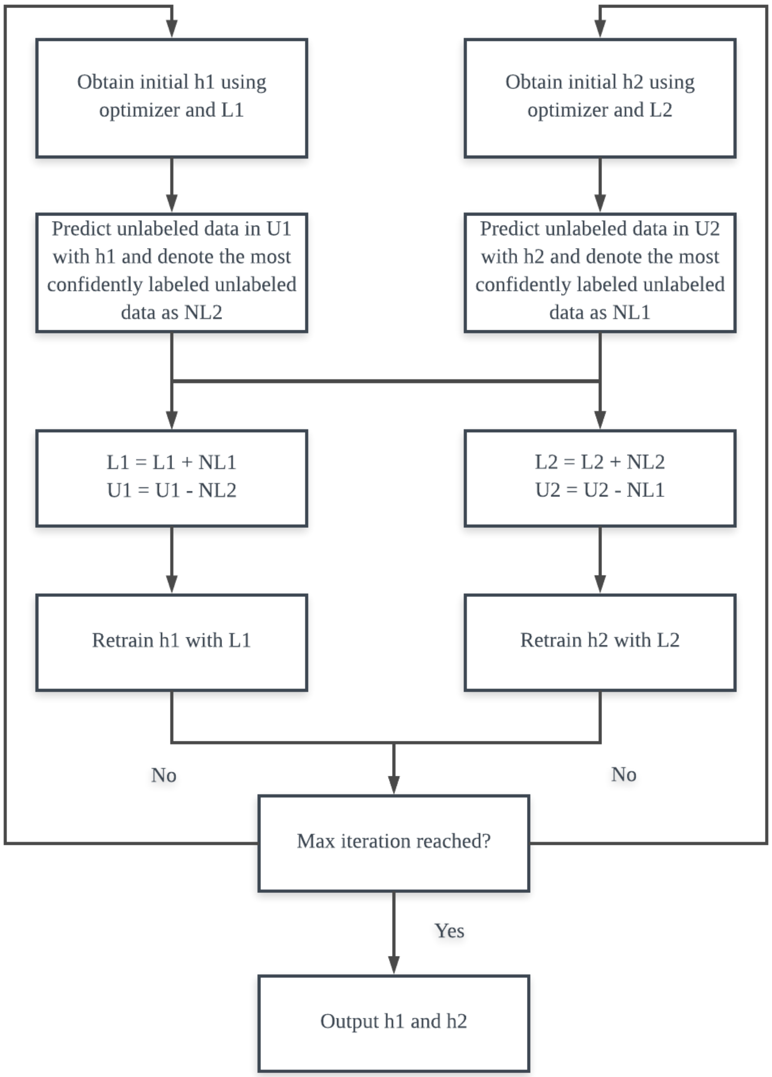

Obtain initial h1 using optimizer and L1
Obtain initial h2 using optimizer and L2
Predict unlabeled data in U1 with h1 and denote the most confidently labeled unlabeled data as NL2
Predict unlabeled data in U2 with h2 and denote the most confidently labeled unlabeled data as NL1
L1 = L1 + NL1
U1 = U1 - NL2
L2 = L2 + NL2
U2 = U2 - NL1
Retrain h1 with L1
Retrain h2 with L2
No
No
Max iteration reached?
Yes
Output h1 and h2

Figure 1: Illustration of co-training process

5 Learning Models

In this project, two different semi-supervised learning algorithms, namely, semi-supervised support vector regression and semi-supervised random forest regression are introduced to build predictive models for the regression problem. Since the semi-supervised support vector regression and semi-supervised random forest are just co-trained versions of the models trained with supervised learning technique, the explanation of their supervised learning algorithms is stated. Below are the definitions of these algorithms.

5.1 Support Vector Regression

Proposed by Vladimir N. Vapnik, Support Vector Machine (SVM)[1] is one of machine learning classification algorithms that provides the optimal hyperplane to classify two groups. SVM model is a representation of the instances as points in space, mapped so that the instances of the separate categories are divided by a clear gap that is as wide as possible.[5]. For the load prediction, Support Vector Regression (SVR), which is the generalized form of SVM is used for regression problem[2].

As well proposed by Vapnik, ε-SVR (Vapnik,1995)[1]'s basic idea is to find the most flat regression function $f(x)$ such that all training data's target variable y_i exists within the boundary of ε deviation, given a training set $(x_1 - y_1), ..., (x_l - y_l) \subset \mathcal{X} \times \mathbb{R}^d$. However, it is almost impossible to find a flat function $f(x)$ where all y_i are within the deviation ε. To cope with this infeasibility, Vapnik applied the loss function and the slack variable ξ_i for y_i which deviates above $f(x)$ and the slack variable ξ_i^* for y_i which deviates below $f(x)$ are added with ε deviation.

The optimal equation of SVR is shown below (equation 2). In the equation, x is input vector, y is output vector, w is coefficient vector, C is unit cost, b is bias and ξ_i,ξ_i^* are slack variables to cope with unfeasible constraints of the optimization problem.[4]

$$\begin{aligned} min. \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ s.t. \quad & y_i - < w, x_i > -b \leq \varepsilon + \xi_i \\ & < w, x_i > +b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{2}$$

The loss function discussed earlier is defined as:

$$|\xi|_\varepsilon := \begin{cases} 0, & \text{if } |\xi| \leq \varepsilon. \\ |\xi| - \varepsilon, & \text{otherwise.} \end{cases} \tag{3}$$

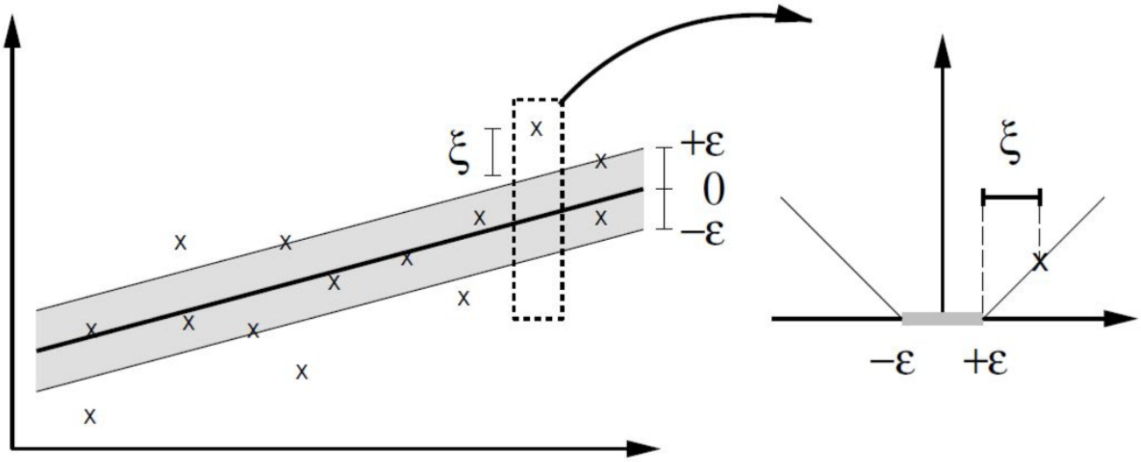


Figure 2: Loss function applied to ε-SVR

The regularization parameter C > 0 determines the trade-off between the flatness of the function and the amount up to which deviations larger than ε are tolerated. Large C value means to give a big penalty for those y_i which deviates out of ε. The result is a overfit model to the training set. Conversely, small C value means to give a little penalty for those y_i which deviates out of ε. The result is a underfit model. Hence, selection of appropriate C value is crucial in generating a high performance generalized SVR model.

To resolve the optimization problem of the equation 2, optimization function and constraint can be expressed as Lagrange function as follows:

$$
\begin{aligned}
L := \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\
- \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i - y_i + < w, x_i > +b) \\
- \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* + y_i - < w, x_i > -b) \\
where \quad \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0
\end{aligned}
\tag{4}
$$

L is the Lagrange function, and $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$ represents the Lagrange multipliers. For optimality, Taking the partial derivatives of L with respect to (w, b, ξ_i, ξ_i^*), the following equations are derived.

$$
\begin{aligned}
\partial_w L = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i = 0 \\
\partial_b L = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\
\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \\
\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0
\end{aligned}
\tag{5}
$$

Substituting equation 5 to the equation 4 populates the Dual Optimization Problem like equation 6. Two variables, η_i, η_i^* are eliminated and regression function can be expressed as equation 7.

$$
\begin{aligned}
max. \quad & \begin{cases} -\frac{1}{2}\sum\limits_{i,j}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < x_i, x_j > \\ -\varepsilon\sum\limits_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum\limits_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \end{cases} \\
s.t \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]
\end{aligned}
\tag{6}
$$

$$
w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i, \text{ thus } f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) < x_i, x > +b \tag{7}
$$

The function in equation 7 is formed with linear combination by w and w is expressed with the attributes x_i of training set. Non-linear classification requires mapping of input space to feature space. This is done using the trick of kernel functions, such as Polynomial Kernel and Gaussian Radial Basis Function Kernel. The SVR algorithm model that is used in this project is ε-SVR using Radial Basis Function(RBF) kernel.

5.2 Random Forest

Random Forest is a collection of many decision trees. Traditionally, the problem for decision tree is that it may over-fit to its training data. To overcome this issue, the random forest is introduced. Rather than taking entire instances and attributes like in a decision tree, a random forest randomly selects instances and attributes to build multiple decision trees and then averages the results. This results in a single output of many less biased decision trees rather than a single tree biased to its training sample.

To construct a random forest, the number of decision trees to construct needs to be decided. These trees must be constructed in such a way that each tree is unique and thus the algorithm to construct the random forest must perform a random selection of instances and attributes. To construct a tree, a bootstrap sample of the data is generated. If there are n data points, randomly extract a data point from the set n times (same data point can be extracted multiple times). Then the average of the prediction made by each decision tree is deduced.

More formally, the random forests algorithm (for both classification and regression) is as follows [3]:

1. Draw n_{tree} bootstrap samples from the original data.

2. For each of the bootstrap samples, grow an unpruned classification or regression tree, with the following modification: at each node, rather than choosing the best split among all attributes, randomly sample m_{try} of the attributes and choose the best split from among those attributes. (Bagging can be thought of as the special case of random forests obtained when $m_{try} = p$, the number of attributes.)

3. Predict new data by aggregating the predictions of the n_{tree} trees (i.e., taking their

average).

An estimate of the error rate can be obtained, based on the training data, by the following:

1. At each bootstrap iteration, predict the data not in the bootstrap sample ("out-of-bag(OOB)" data) using the tree grown with the bootstrap sample.
2. Aggregate the OOB predictions. (On the average, each data point would be out-of-bag around 36% of the times, so aggregate these predictions.) Calculate the error rate, and call it the OOB estimate of error rate.

The random forest algorithm estimates the importance of a variable by looking at how much prediction error increases when (OOB) data for that variable is permuted while all others are left unchanged.

The Figure 3 of the random forest is illustrated as follows:

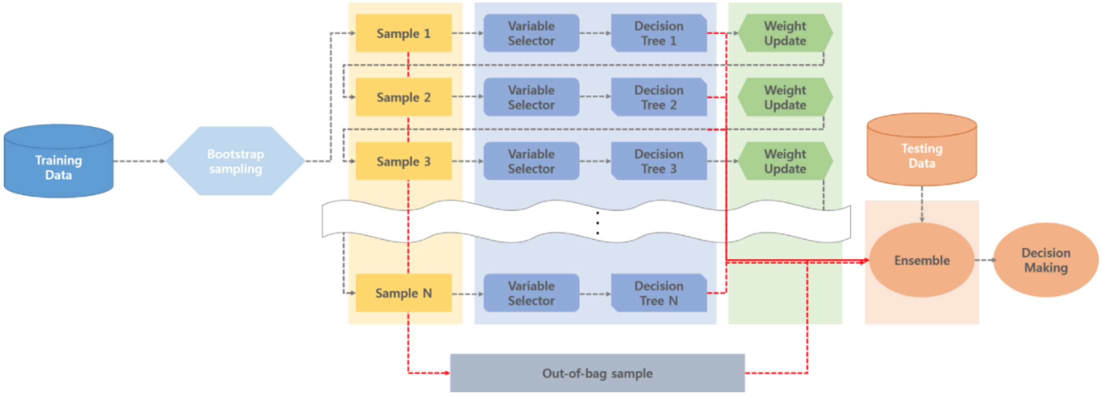


Figure 3: Procedure on how the random forest works

In the random forest, the splitting in a node is done in this way. In each tree in the forest, the label is fit using each of attributes in the dataset. Then for each attribute, the data is partitioned to different data points. At each split point, a "Mean Squared Error"(MSE) is calculated based on the predicted value and actual value of the label. The MSE error for each attribute is compared with every other attribute and the attribute which yields the lowest MSE is chosen to be a split point for that node.

6 Dataset and Implementation

In this project, Python will be used as the main programming language to preprocess the dataset and training the model. For the dataset, a real-world dataset was received by a Chinese wind power generator company. The model will trained with this set of data for the load prediction.

6.1 The Dataset

The dataset is comprised with 2 different files, namely "NWP.xls" and "Wind_Power_Time_Serial.xlsx". For the data in both files, each data point is distinguished by a distinct "time" value.

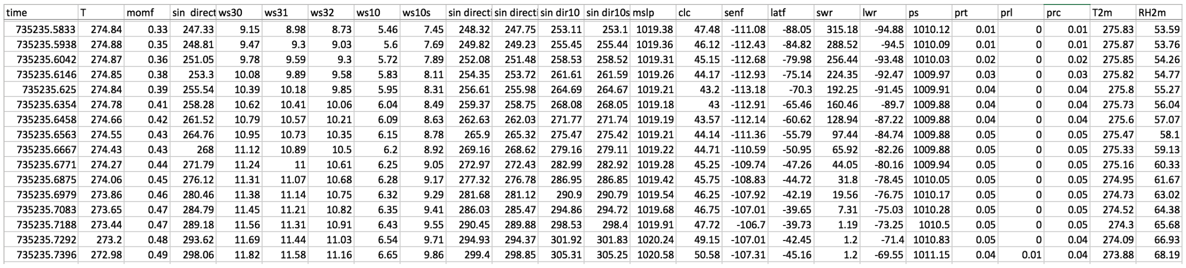
There are total of 24 attributes excluding "time", as "time" is an identifier for each data point in the dataset. A description and data type of each attribute are shown in the Table 1 below:

Attribute	Description	Type
T	temperature	nominal
momf	momentum flux	nominal
sin direction32	direction 30m	nominal
ws30	wind speed at 170m	nominal
ws31	wind speed at 100m	nominal
ws32	wind speed at 30m	nominal
ws10	wind speed at 10m	nominal
ws10s	wind speed at 0m	nominal
sin direction30	wind direction at 170m	nominal
sin direction31	wind direction at 100m	nominal
sin dir10	wind direction at 10m	nominal
sin dir10s	wind direction at 0m	nominal
mslp	mean sea level pressure	nominal
clc	fraction of clouds [0-1]	nominal
senf	sensible heat flux	nominal
latf	latent heat flux	nominal
swr	shortwave radiation	nominal
lwr	longwave radiation	nominal
ps	surface pressure	nominal
prt	(unspecified)	nominal
prl	large scale precipitation	nominal
prc	convective scale precipitation	nominal
T2m	(unspecified)	nominal
RH2m	humidity	nominal

Table 1: List of attributes and their description and data type

The attributes prt and T2m were not specified at the time of this project and require further clarification.

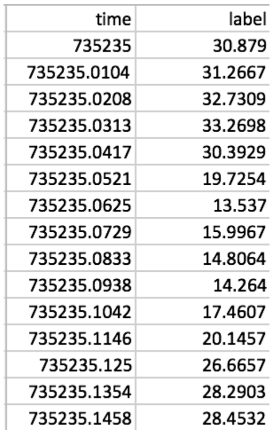
The "NWP.xls" file contains the value of each attribute in a given instance of "time". The Figure 4 illustrates a part of data points in the file.



time	T	momf	sin direct	ws30	ws31	ws32	ws10	ws10s	sin directi	sin directi	sin dir10	sin dir10s	mslp	clc	senf	latf	swr	lwr	ps	prt	prl	prc	T2m	RH2m
735235.5833	274.84	0.33	247.33	9.15	8.98	8.73	5.46	7.45	248.32	247.75	253.11	253.1	1019.38	47.48	-111.08	-88.05	315.18	-94.88	1010.12	0.01	0	0.01	275.83	53.59
735235.5938	274.88	0.35	248.81	9.47	9.3	9.03	5.6	7.69	249.82	249.23	255.45	255.44	1019.36	46.12	-112.43	-84.82	288.52	-94.5	1010.09	0.01	0	0.01	275.87	53.76
735235.6042	274.87	0.36	251.05	9.78	9.59	9.3	5.72	7.89	252.08	251.48	258.53	258.52	1019.31	45.15	-112.68	-79.98	256.44	-93.48	1010.03	0.02	0	0.02	275.85	54.26
735235.6146	274.85	0.38	253.3	10.08	9.89	9.58	5.83	8.11	254.35	253.72	261.61	261.59	1019.26	44.17	-112.93	-75.14	224.35	-92.47	1009.97	0.03	0	0.03	275.82	54.77
735235.625	274.84	0.39	255.54	10.39	10.18	9.85	5.95	8.31	256.61	255.98	264.69	264.67	1019.21	43.2	-113.18	-70.3	192.25	-91.45	1009.91	0.04	0	0.04	275.8	55.27
735235.6354	274.78	0.41	258.28	10.62	10.41	10.06	6.04	8.49	259.37	258.75	268.08	268.05	1019.18	43	-112.91	-65.46	160.46	-89.7	1009.88	0.04	0	0.04	275.73	56.04
735235.6458	274.66	0.42	261.52	10.79	10.57	10.21	6.09	8.63	262.63	262.03	271.77	271.74	1019.19	43.57	-112.14	-60.62	128.94	-87.22	1009.88	0.04	0	0.04	275.6	57.07
735235.6563	274.55	0.43	264.76	10.95	10.73	10.35	6.15	8.78	265.9	265.32	275.47	275.42	1019.21	44.14	-111.36	-55.79	97.44	-84.74	1009.88	0.05	0	0.05	275.47	58.1
735235.6667	274.43	0.43	268	11.12	10.89	10.5	6.2	8.92	269.16	268.62	279.16	279.11	1019.22	44.71	-110.59	-50.95	65.92	-82.26	1009.88	0.05	0	0.05	275.33	59.13
735235.6771	274.27	0.44	271.79	11.24	11	10.61	6.25	9.05	272.97	272.43	282.99	282.92	1019.28	45.25	-109.74	-47.26	44.05	-80.16	1009.94	0.05	0	0.05	275.16	60.33
735235.6875	274.06	0.45	276.12	11.31	11.07	10.68	6.28	9.17	277.32	276.78	286.95	286.85	1019.42	45.75	-108.83	-44.72	31.8	-78.45	1010.05	0.05	0	0.05	274.95	61.67
735235.6979	273.86	0.46	280.46	11.38	11.14	10.75	6.32	9.29	281.68	281.12	290.9	290.79	1019.54	46.25	-107.92	-42.19	19.56	-76.75	1010.17	0.05	0	0.05	274.73	63.02
735235.7083	273.65	0.47	284.79	11.45	11.21	10.82	6.35	9.41	286.03	285.47	294.86	294.72	1019.68	46.75	-107.01	-39.65	7.31	-75.03	1010.28	0.05	0	0.05	274.52	64.38
735235.7188	273.44	0.47	289.18	11.56	11.31	10.91	6.43	9.55	290.45	289.88	298.53	298.4	1019.91	47.72	-106.7	-39.73	1.19	-73.25	1010.5	0.05	0	0.05	274.3	65.68
735235.7292	273.2	0.48	293.62	11.69	11.44	11.03	6.54	9.71	294.93	294.37	301.92	301.83	1020.24	49.15	-107.01	-42.45	1.2	-71.4	1010.83	0.05	0	0.04	274.09	66.93
735235.7396	272.98	0.49	298.06	11.82	11.58	11.16	6.65	9.86	299.4	298.85	305.31	305.25	1020.58	50.58	-107.31	-45.16	1.2	-69.55	1011.15	0.04	0.01	0.04	273.88	68.19

Figure 4: NWP.xls snippet

The "Wind_Power_Time_Serial.xlsx" file contains the label of each given instance of "time". The Figure 5 illustrates a part of data points in the file.



time	label
735235	30.879
735235.0104	31.2667
735235.0208	32.7309
735235.0313	33.2698
735235.0417	30.3929
735235.0521	19.7254
735235.0625	13.537
735235.0729	15.9967
735235.0833	14.8064
735235.0938	14.264
735235.1042	17.4607
735235.1146	20.1457
735235.125	26.6657
735235.1354	28.2903
735235.1458	28.4532

Figure 5: Wind_Power_Time_Serial.xlsx snippet

After a sequence of preprocessing, the final dataset that was used to train the models was created. In the dataset, there are 11399 data points, 9022 labeled data points, and 2377 unlabeled data points. A label is a continuous value ranging from -0.395085 to 95.5206, which represents the percentage of total generators working. Since it's not possible to have a minus percentage of generators working, it is regarded as an error during data collection and is rounded up to 0.

6.2 Parameter and Hyperparameter

Parameters: Parameters are the training dataset's properties and are learned during training by the machine learning models. Model parameters are different from an experiment to experiment and depend on the type of data to be handled. In this project, the weight of each attribute and split points are the examples of parameters.

Hyperparameters: Hyperparameters are properties of a dataset that cannot be learned during training and are pre-set before training. In this project, the hyperparameters of semi-supervised SVR are C, the unit cost, and ε in its loss function and gamma of RBF kernel. These hyperparameters are optimized using the genetic algorithm. The hyperparameters for the semi-supervised random forest are n_estimators, max_depth, max_features, min_sample_splits and min_sample_leaf. The hyperparameters are optimized using the random grid search.

7 Result and Analysis

7.1 Performance Metrics

The models produced in this project are compared based on 3 criteria, namely correlation coefficient, root mean squared error and mean absolute error. The performance of a model is said to be better if it has higher correlation coefficient and lower root mean squared error and mean absolute error.

Let the actual value of label be θ and the predicted value of from a model be $\hat{\theta}$.

Correlation Coefficient: Correlation coefficient shows how related θ and $\hat{\theta}$ are. The value is between -1 and 1 where 0 is no relation, closer to 1 means more positive correlation and closer to 0 means less positive correlation. The negative value means the inverse correlation, a negative correlation. The Figure 6 illustrates how the correlation would look for different correlation coefficient values.

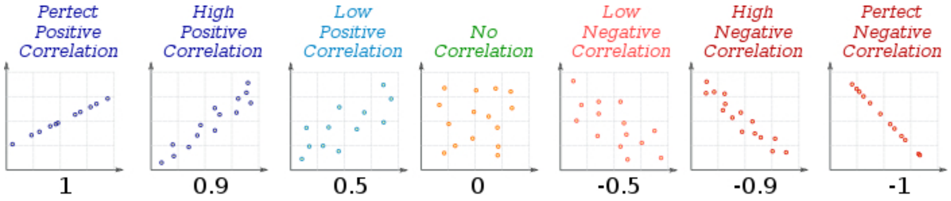


Figure 6: Correlation Examples

The equation for correlation coefficient of θ and $\hat{\theta}$ is,

$$CC_{\hat{\theta}\theta} = \frac{C_{\hat{\theta}\theta}}{\sqrt{C_{\theta\theta} * C_{\hat{\theta}\hat{\theta}}}} \tag{8}$$

where C is the covariance matrix.

Root Mean Squared Error: Root mean square error (RMSE) is the standard deviation of errors between θ and $\hat{\theta}$. Residual is the difference between the value of estimated label and the true value of that label, $\hat{\theta} - \theta$ and root mean square error is then a measure of how spread these residuals are. The equation for RMSE is shown in the equation 9

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2} \tag{9}$$

Mean Absolute Error: Mean absolute error (MAE) measures the average of difference between θ and $\hat{\theta}$. The equation for MAE is shown in equation 10

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{\theta}_i - \theta_i| \tag{10}$$

7.2 Result

The following tables illustrate the results of the performances of the models after testing them with a test set. Each supervised learning model (i.e. Support vector regression and random forest) is trained with 200 labeled data points and each semi-supervised learning model (i.e. Semi-supervised support vector regression and semi-supervised random forest) is trained with 200 labeled data points and 30 unlabeled data points for SSSVR and 500 unlabeled data points for SSRF. The SSSVR is trained with 30 unlabeled data points due to high computational cost for each retraining with new unlabeled data points. The support vector regression and semi-supervised support vector regression are compared head to head while random forest and semi-supervised random forest are compared each other and their performance differences are denoted at the bottom of the table as shown below:

Table 2: Support Vector Regression vs Semi-supervised Support Vector Regression (30 unlabeled data points)

Model	CC	RMSE	MAE
Support Vector Regression	0.646839	21.338619	10.896001
Semi Supervised Support Vector Regression	0.653136	14.747316	10.173029
Performance Difference	0.006297	-6.591303	-0.722972

Table 3: Random Forest vs Semi-supervised Random Forest (500 unlabeled data points)

Model	CC	RMSE	MAE
Random Forest	0.777218	13.903623	10.529350
Semi Supervised Random Forest	0.792573	13.754622	10.403035
Performance Difference	0.015356	-0.149000	-0.126316

The top 5 attributes importance which are deduced by the random forest are : 1.ws21 2.ws10s 3.ws20 4.ws10 5.dir

7.3 Conclusion

After running the training and testing phases of each model, the performances of each model is deduced. From the result, the augmentation of unlabeled data points improved the correlation coefficient, RMSE and MAE for both support vector regression and random forest.

The correlation coefficient of support vector regression improved by 0.006297, RMSE reduced by 6.591303 and MAE reduced by 0.722972. Likewise for random forest, the correlation coefficient improved by 0.015356, RMSE reduced by 0.149000 and MAE reduced by 0.126316. Clearly, the co-training algorithm has improved the models but it is worth noticing the trade-off between the additional computational power to generate the semi-supervised model and the performance of each model being produced. Semi-supervised learning with co-training algorithm requires greatly more computational power than supervised learning approach (For example, training with 200 labeled data points and 1000 unlabeled data points for semi-supervised learning random forest require 82816.0064 seconds while supervised learning random forest to train with 200 labeled data points only require 24.6734 seconds), since each time unlabeled data points is augmented, the regressor needs to be retrained with the augmented dataset.

8 Summary

This project intends to solve a real world problem through data analytics. The problem in the context of this project is to predict the load of wind power generators in order to cope with the dependence on natural conditions and varying consumption so that the wind power generators can be effectively cooperated with the traditional power generators. Machine learning is used since it is capable of producing a predictive model out of given dataset with labels. Since the dataset contains both the labeled and unlabeled data points, the project incorporates the semi-supervised learning which exploits both the labeled and unlabeled data points in the dataset. The prediction models are built with different machine learning techniques, namely support vector regression, random forest and co-trained versions of these regression techniques which utilizes the unlabeled data points in the dataset to produce the effect of the semi-supervised learning. The models constructed by these techniques are then tested with a test set in order to figure out the performance of each model. The performances of the models are then compared and the effect of augmenting the unlabeled data points to the training phase was deduced. The semi-supervised learning which exploits the unlabeled data points improved the models but the trade-off between the computational power and performance exists, hence the balancing between the additional computational power and better performance needs to be done before actually implementing the models into solving the problem in hand.

References

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] Yeong-ju Kim, Min-a Jeong, and Nam-rye Son. Forecasting of short-term wind power generation based on svr using characteristics of wind direction and wind speed. 42:1085–1092, 05 2017.

[3] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. 23, 11 2001.

[4] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[5] Support Vector Machine. Support vector machine — Wikipedia, the free encyclopedia, 2017. [Online; accessed 20-November-2017].

[6] Xili Wang, Li Fu, and Lei Ma. Semi-supervised support vector regression model for remote sensing water quality retrieving. *Chinese Geographical Science*, 21(1):57–64, 2011.

[7] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005.