Assignment 2:

Introduction

As mentioned in the description of the assignment, accurate wind power forecasts can assist different agents in the power grid to make intelligent decisions on grid sustainability, load balancing and setting pricing policies. In this assignment, we have utilized different machine learning techniques and data sources to provide hourly wind forecasts of a wind farm in Australia.

This assignment has three parts.

- Part1: we build and evaluate the ML models for wind power prediction, based on one meteorological variable (historical wind speed).
- Part2: we add one more variable (wind direction) to the input of the models to evaluate how wind direction can affect the model accuracy.
- Part 3: we only consider historical wind power observations as the influential factor on future power generation.

Regarding the implementation, we used Python as the programming language. Python libraries as Pandas, Numpy and Matplotlib were used for data analysis and plotting; Sklearn and Keras libraries were used to build the classical ML techniques and ANNs respectively.

For the rest of the report, in Section 1, we briefly explain the applied models to this problem and the important parameters that we set to build the models. Then, in Section 2, we introduce the error metric used for the evaluations. In Section 3, for each part, we explain the training and test data to fit and test the models. We also evaluate the prediction results based on the error metrics. The graphs, in addition, show the comparisons between the real and predicted values.

1. Machine Learning Algorithms

Models used for the present study are listed below:

1.1. Linear regression:

In statistics, linear regression is a linear approach to model the relationship between a dependent variable and one or more explanatory variables. The case of one explanatory variable is called a simple linear regression (LR). For more than one explanatory variable, the process is called multiple linear regression (MLR). Linear regression models which are used in this assignment use the ordinary least squares (OLS) approach. OLS is a type of linear least squares method for estimating the unknown parameters in a linear regression model. It chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares: minimizing the sum of the squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function [1].

1.2. K-Nearest-Neighbors

The k-nearest-neighbours algorithm is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. In this assignment, since the target variable to be forecasted (wind power generation) has a continuous value, the forecasting problem is called a regression problem. In the k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors. k-NN is a type of lazy algorithms, where the function is only approximated locally, and all computation is deferred until regression or classification. This algorithm is among the simplest of all machine learning algorithms in terms of both computation time and complexity [2]. In this assignment, to build the KNN model, we set *k*=7. (We tested a few numbers of *k;* the number 7 as the neighbor's number produced more accurate predictions). *All* points in each neighborhood were weighted equally and `Minkowski` (as a generalization of both the Euclidean distance and the Manhattan distance) was used as the distance metric.

1.3. Support Vector Regression

The Support Vector Regression, as a version of Support Vector Machine (SVM) for regression, is widely used for time series prediction. In SVR, we set a margin between a hyperlane and closest nodes to the hyperlane which shows relationships between inputs and output. We ignore the errors as long as they are less than margins. The linear function in SVR can describe the nonlinear relationship between variables in high dimensional feature space, using the kernel trick [3].

To build the SVR model, we used a radial basis function known as `rbf` as the kernel. We preferred a non-linear SVR to a linear SVR because, it can estimate the non-relationship between dependent and non-dependent variables, more accurately. We also set the kernel coefficient (gamma) to 0.1, and Penalty parameter C of the error term to 100. Other parameters were set to the default values of the model in the `Scikit Learn` library.

1.4. Artificial Neural Network (ANN)

An artificial neural network consists of several highly interconnected artificial neurons. The most common architecture of the neural networks is the Multilayer Perceptron (MLP) type where neurons organized in layers. They are powerful prediction models since, they are self-adaptive and can capture non-linear relationships between inputs and outputs. In this assignment, we built an MLP with one hidden layer. The number of neurons in the first layer is set to the number of input variables, the number of neurons in the output layer is set to the number of outputs (in our case, it is 1) and the number of neurons in the hidden layer was set to 10. Weights were initialized randomly. 'Adam' was used as the optimization algorithm and 'Relu' as the activation function.

1.5. Recurrent Neural Network (RNN)

RNNs are neural networks which use feedback connections among the nodes to remember the values from previous time steps, therefore they will be able to capture the temporal behavior of time series data. In this assignment, we used an RNN with Long-short term memory cells (LSTM). One hidden layer, 30 memory cells, and 'Adam' optimizer were used for building the LSTM.

2. Error metric:

Root Mean Squared Error (RMSE): $\sqrt{\sum_{i=1}^{N} \left(\frac{Prediction_i - Actual_i}{N}\right)^2}$, where N is the total number of predicted values.

3. Evaluation and Results

3.1. Part 1

In this part, we have built and trained four ML for power prediction. The training set includes information on wind speed at 10 meters above ground level from 1.1.2012 to 31.10.2013, and the test set is the wind generation over November 2013. The ML techniques include simple LR, KNN, SVR and an ANN. All the models only approximate the relation between wind speed and wind power generation. Some parameters which were mentioned in Section 1, were set manually, but all other parameters of the models were set to the default values in the Sklearn library and the depending packages.

Regarding the ANN model (MLP with one hidden layer), the training data first normalized between 0 and 1 (having zero mean and unit variance). Then it was fed to the model. Normalization is important since, it transposes the input variable(s) into the same range where the ReLu activation functions of the MLP lie in. The number of iterations for updating the weights (epochs) is set to 70 and the number of training records in each iteration (batch size) was set to 50. Table 1 shows the prediction results of the models.

Table 1: RMSE results over four ML techniques, Part 1

Machine Learning Techniques	**Root Mean Squared Error (RMSE)**
Linear Regression	0.2164
KNN	0.2318
SVR (with RBF kernel)	**0.2142**
ANN (with one hidden layer and 10 neurons)	0.2158

According to this table, SVR outperforms the other three techniques and KNN performance is the worst. The KNN result is considerably far from the results of the others since, it is the simplest approach for regression and the other three, as described in Section 1, apply more complicated approximations. ANN and SVR with an RBF kernel, which have similar performance in this case, compared to linear regression, in addition to linear relationships between the wind speed and power generation, can capture the non-linearities and subtle patterns in the data. Figures 1,2 3, and 4 show the predicted and true wind power measurements over the one-month test period for LR, KNN, SVR and ANN respectively.

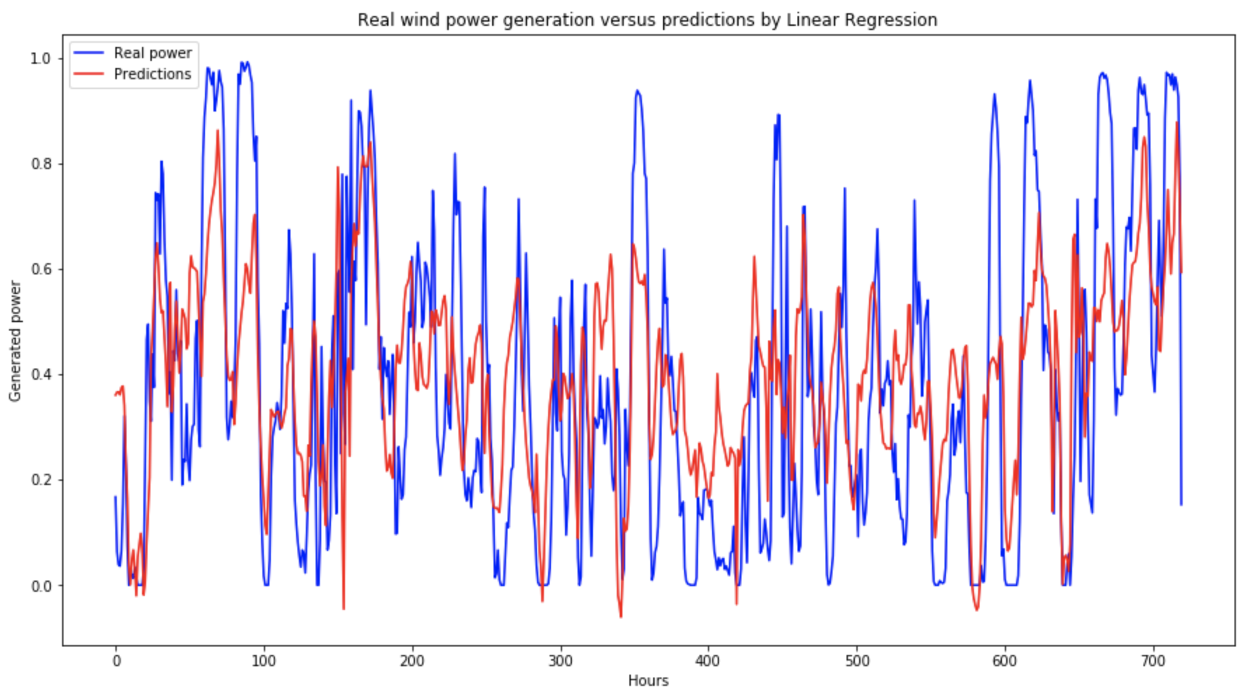

Fig.1

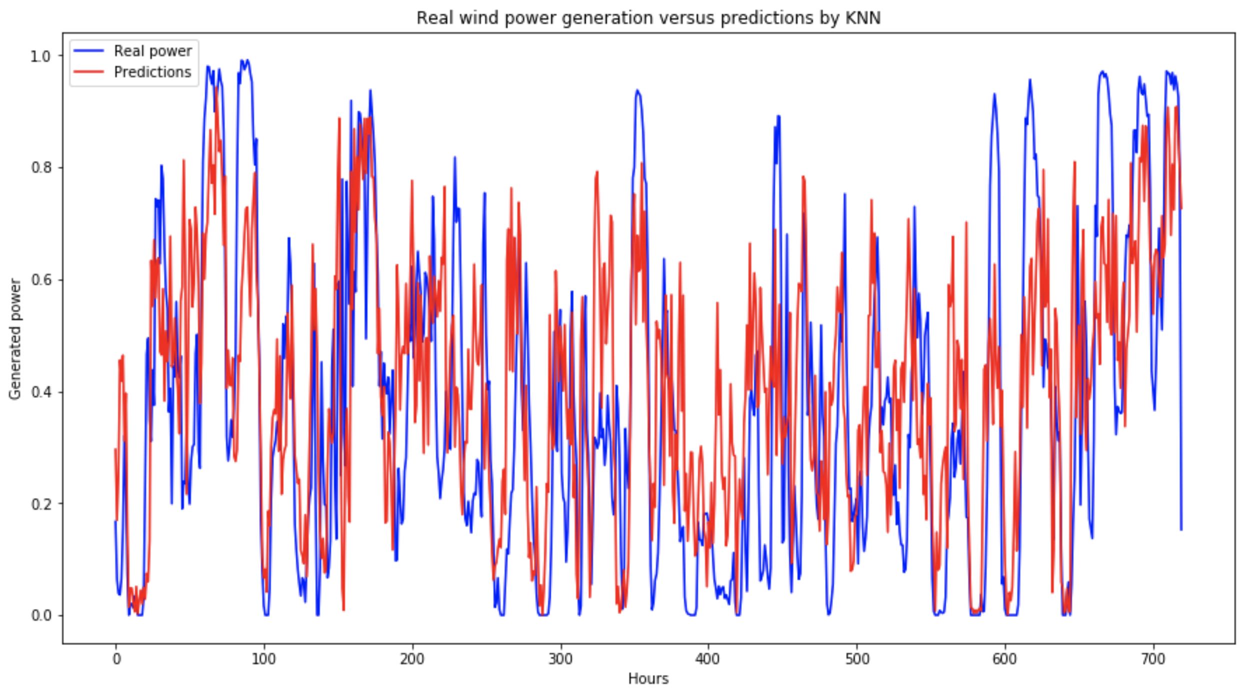
Real wind power generation versus predictions by KNN
Real power
Predictions
Generated power
Hours
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
600
700

Fig. 2

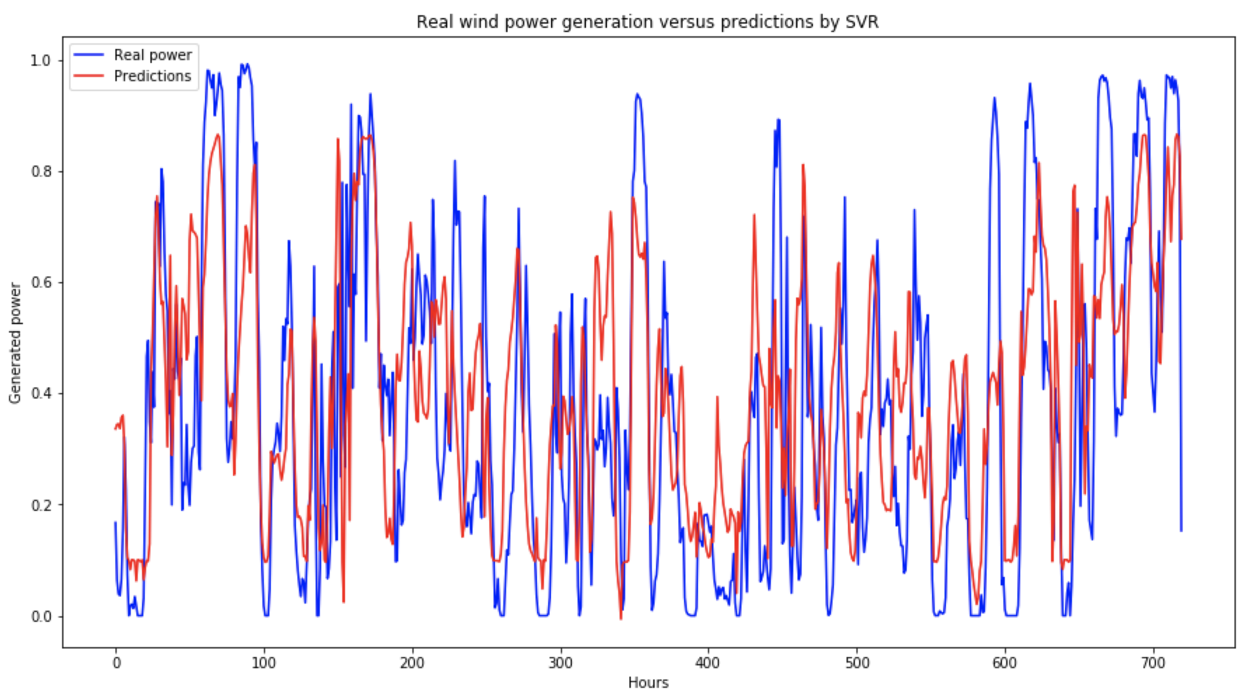
Real wind power generation versus predictions by SVR
Real power
Predictions
Generated power
Hours
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
600
700

Fig. 3

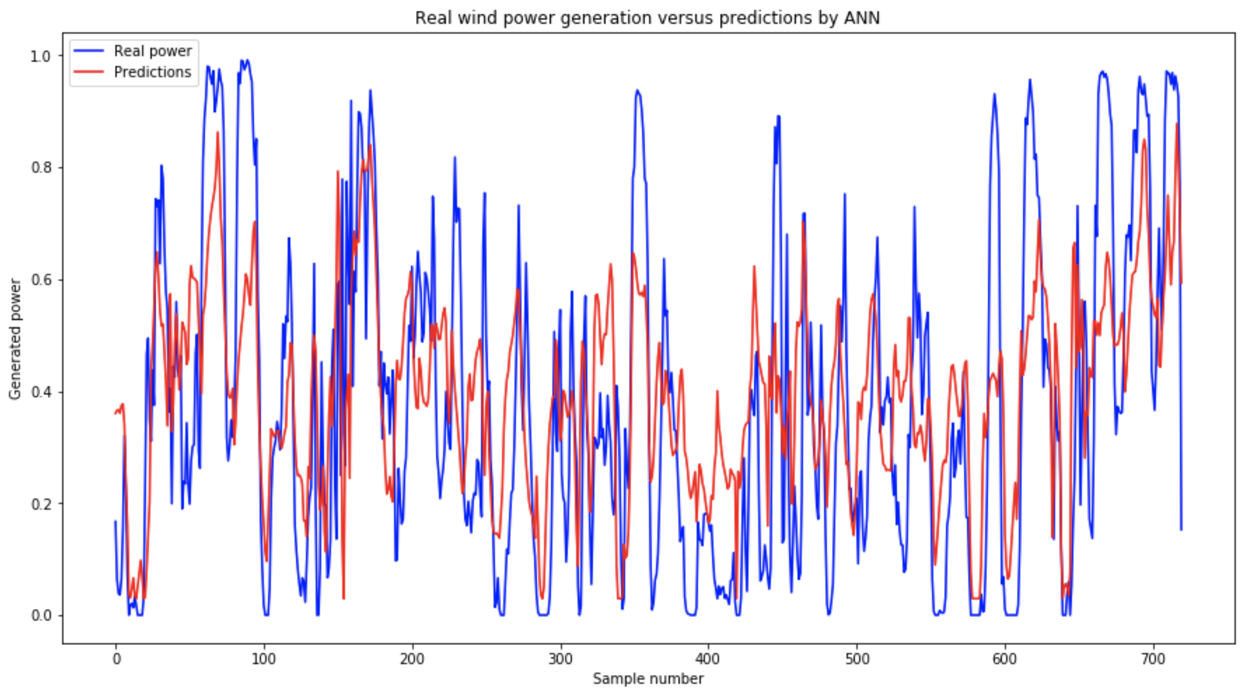

Fig. 4

3.2. Part 2

In this part, we have built and trained a multiple linear regression model in order to forecast wind power generation over the test period. This model uses wind speed at 10m above ground level in addition to wind direction, as its input (training data). The wind direction was calculated based on the zonal component U10 and meridional component V10 derived from the weather forecast dataset. To calculate the wind direction (between 0 and 360 degrees) first, we should apply an `atan2` function on U10 and V10 and then convert it to degrees. This function is defined as the angle in the Euclidean plane given in radians, between the positive *x*-axis and the ray to the point (*U10,,V10*) ≠ (0,0) [4].The complete computation formula is as follows:

$$r2d = \frac{45}{atan(1.0)}$$ as the conversion factor (from radians to degrees)

$$Wind\,direction = atan2(U10, V10) * r2d + 180$$

Moreover, we did a comparison between the performance of linear regression with one input (Part1) and the performance of LR with two inputs (Part2). Table 2 shows the RMSE and MAPE results of the two models. `MAPE` represents the Mean Average Percentage Error.

Table 2. RMSE and MAPE results of LR and MLR

Machine Learning Techniques	**Root Mean Squared Error (RMSE)**	**MAPE**
Linear Regression (LR)	0.2164	18.0478
Multiple-Linear Regression (MLR)	**0.2149**	**17.9413**

The results show that the prediction accuracy of the linear model was slightly improved due to the addition of one more predictive variable (wind direction) to the input. (0.6 % and 0.5 % reductions in terms of RMSE and MAPE respectively.) This implies that linear regression model like other data-driven models, learn from the data, and if it is fed by more variables (preferably more informative ones) can produce more accurate approximations.

Figure 5 and Figure 6 show the comparisons among predictions of LR, MLR versus real values over the one month and one day test period, respectively.

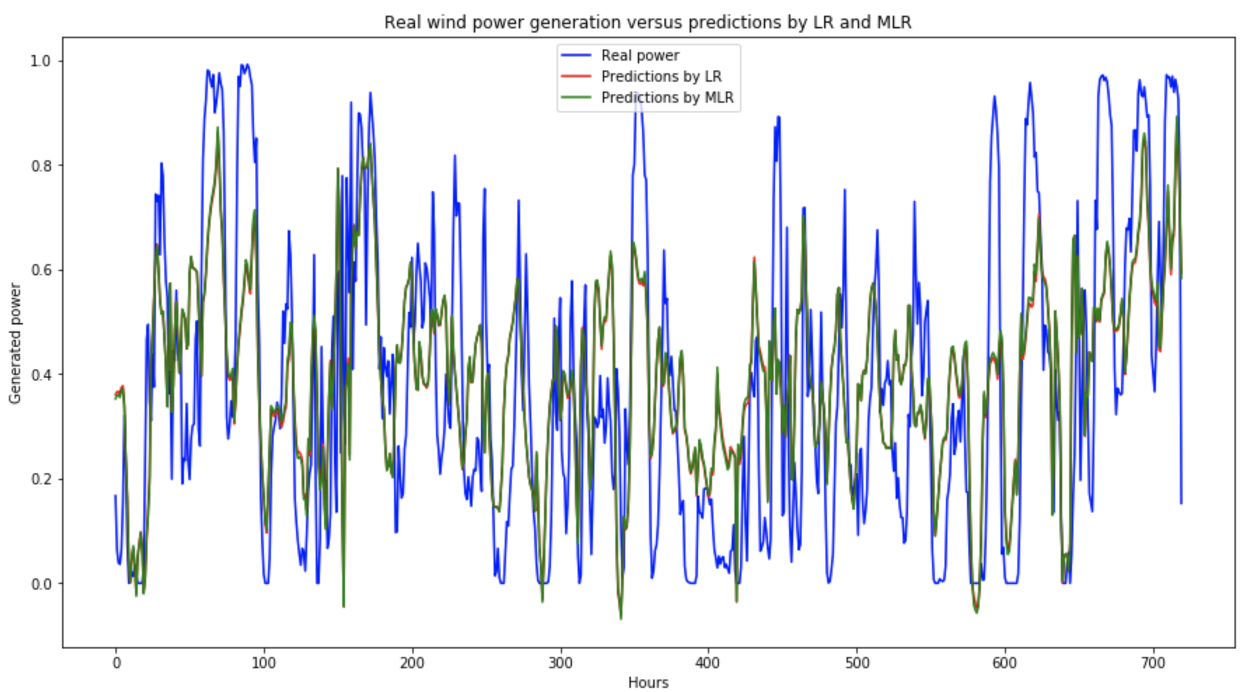

Fig. 5

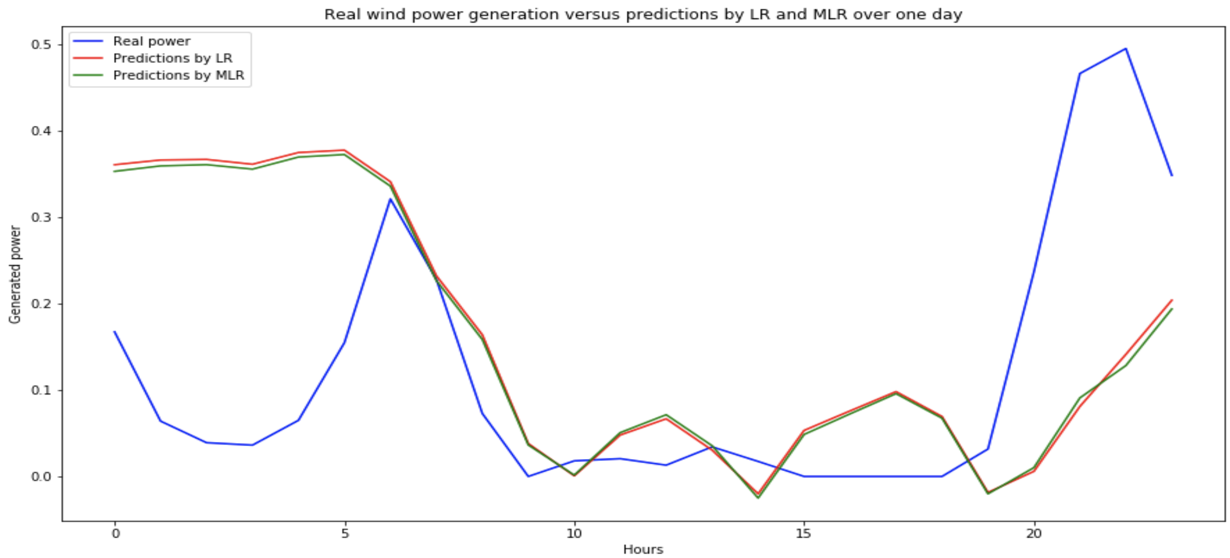

Fig. 6

3.3. Part 3

In this part, we convert the problem to a time series forecasting problem, where we only use historical observations of wind power to forecast future generations. This time we applied linear regression, SVR, MLP and an advanced version of RNN known as LSTM, to provide the forecasts.

The first step, before training the models, was to convert the collections of observations to a time series dataset. For this purpose, we removed all the meteorological observations and defined a dataset with just two columns: TIMESTAMP and POWER. Then we converted it to a supervised learning dataset using a window slicing method. We used previous time steps as input variables to predict the next time step (the next hour observation) as the output variable. We also preserved the order between the observations. (We defined a function for this conversion, having a parameter called: *look back*; we can change the value of this parameter to have a time series dataset with a different number of previous time steps or lags). For this part, we created two different time series and applied them to each model: one with 1 input variable (1 lag observation) and one with 24 input variables (24 lags or window size of 24). The reason was to test the effect of using more historical information on the capability of the models in time series prediction.

The training set covers the hourly power observations from almost the same 22 months (1.1.2012 to 31.10.2013) and the test set is almost one month in November 2011. We use these sets for all ML models.

- Note 1: In order to have fair comparisons between the results of the models in Part 1 and Part3, we did not change the parameter setting of the applied models. (Except for the ones that need changes due to change in input dimensions.)
- Note 2: Since we had no previous values that we could use to predict the first 24 values in the training sequence, we deleted these rows. Similarly, we deleted the last rows in our test set, since we did not have a known next value to predict for the last values.
- Note3: For both the MLP model and LSTM, we normalized the dataset like before.

Table 3 shows the performance results of all the applied techniques in Part 3 and Part1.

Table 3. Comparison of the results between Part 3 and Part1

Machine Learning Techniques	**RMSE using timeseries data with 1 lag**	**RMSE using timeseries data with 24 lags**	**RMSE using wind speed data**
Linear Regression	0.1241	**0.1213**	0.2164
SVR	**0.1254**	0.1280	0.2142
ANN	0.1240	**0.1214**	0.2158
LSTM	0.1238	**0.1181**	-----

It is shown that, using time series data, has helped the models to learn more about the power generation, compared to the scenarios where only meteorological data (wind speed) was used as the training set. The error results show more than 75 % improvement in the prediction accuracies of all the models.

A fair justification could be that time series adds an explicit order dependence between observations: a time dimension which provides a source of additional information about the target variable (power generation).Moreover, we can see that, except for SVR, the prediction accuracy of the models has been improved by adding more previous observations (features) to the dataset.

Among the models which were used in Part 3 for time series forecasting, the LSTM model outperformed others. The reason might be that LSTM model has been developed specifically to deal with time series data, using memory cells to remember long and short-term dependencies among the observations. It can solve the problem of the vanishing or exploding gradient problem of simple RNNs.

Figure 7 shows the predictions by SVR and Linear regression versus real observations, using 24 past observations in the dataset. (Multiple Linear regression performs better than SVR, 5% reduction in RMSE)

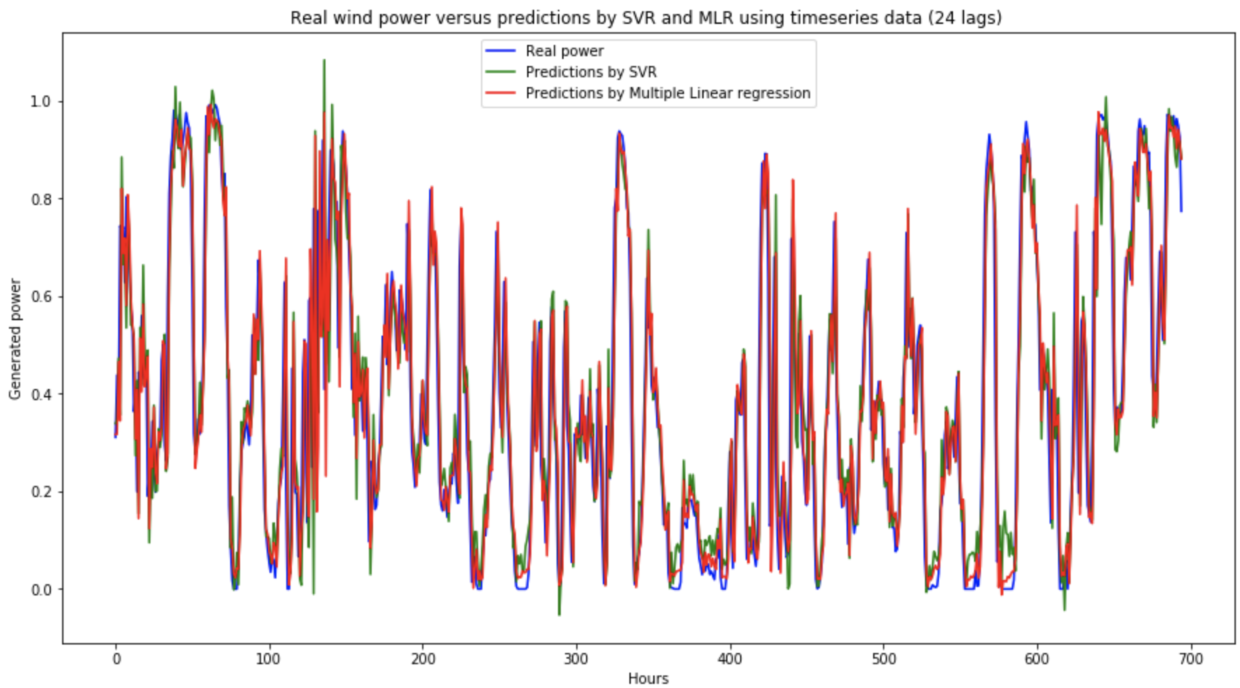

Fig. 7

Figure 8 shows the predictions by ANN and LSTM versus real observations, using 24 lag observations. (2.7% improvement in accuracy by LSTM versus ANN).

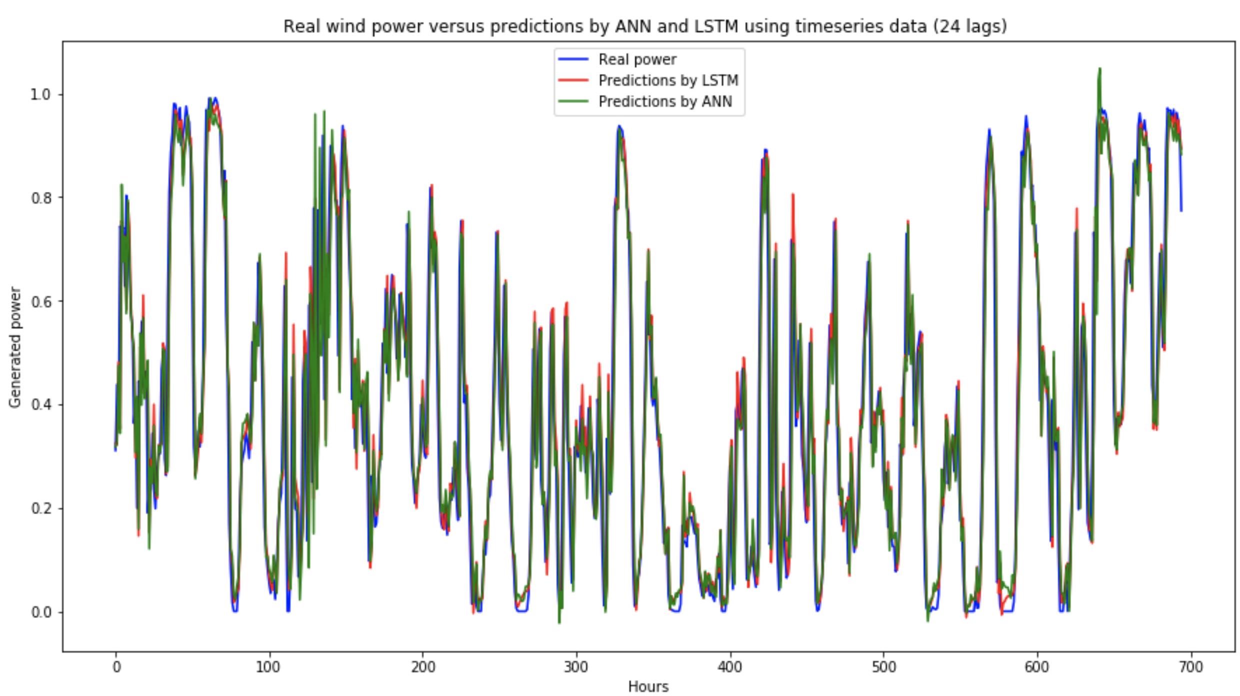

Fig. 8

References:

1. https://en.wikipedia.org/wiki/Ordinary_least_squares
2. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
3. Machine learning for renewable energy forecasting, Yan Zhang, University of Oslo, Norway
4. https://en.wikipedia.org/wiki/Atan2