University of Oslo

IN5410 - Energy informatics

Machine learning for wind energy forecasting

Authors:

Ingrid Bye Løken

Kristine Jun Grøttland

Eirik Steen Valen

UiO : Institutt for informatikk

Det matematisk-naturvitenskapelige fakultet

Index

1.0 Relationship between wind power generation and wind speed

1.1 Introduction

In task 1, different machine learning techniques were used to predict wind power generation for November 2013 based on wind speed data. The machine learning techniques used were:

- Linear regression (LR)
- Supported vector regression (SVR)
- k-nearest neighbor (kNN)
- Artificial neural networks (ANN)

The code was written in Python, with machine learning packages imported from sklearn and keras.

The following method was used for each of the different machine learning techniques/models:

1. The train data with actual wind speed and resulting wind power generation from the file *TrainData.csv* was used to train the different machine learning models.
2. Each training model was used to predict wind power generation output from wind speed from the file *WeaterForecastInput.csv.*
3. The predicted wind power generation was compared to actual wind power generation from the file *Solution.csv* (for November 2013)
    a. The predicted power and the actual power was plotted in the same figure with wind speed on the x-axis and power generation og the y-axis
    b. The predicted power and the actual power was plotted in the same figure with time on the x-axis and power generation on the y-axis.
    c. Root Mean Square Error (RMSE) was calculated for each machine learning model to compare the prediction accuracy

1.2 Implementation

The code for task 1 can be found in the task1.py file. To run the code, please make sure all files are in the same directory: *TrainData.csv, WeatherForecastInput.csv, Solution.csv, task1.py, read_from_file.py* and *plot_methods.py*. In addition to this, the following Python packages should be installed: pandas, numpy, sklearn and keras/tensorflow.

The main methods that run the implementations are `linear_regression()`, `k_nearest_neighbor()`, `supported_vector_regression()` and `artificial_neural_network()`. They contain the machine learning algorithms for linear regression, k-nearest neighbor and artificial neural networks respectively. To run all the implementations run the method `run_all_models_task1(show_plots, write_to_csv)`. The argument show_plots and write_to_csv can be set to either True or False depending on the wish to see the plots of the results and writing the results to csv files.

1.3 Results

1.3.1 Linear regression

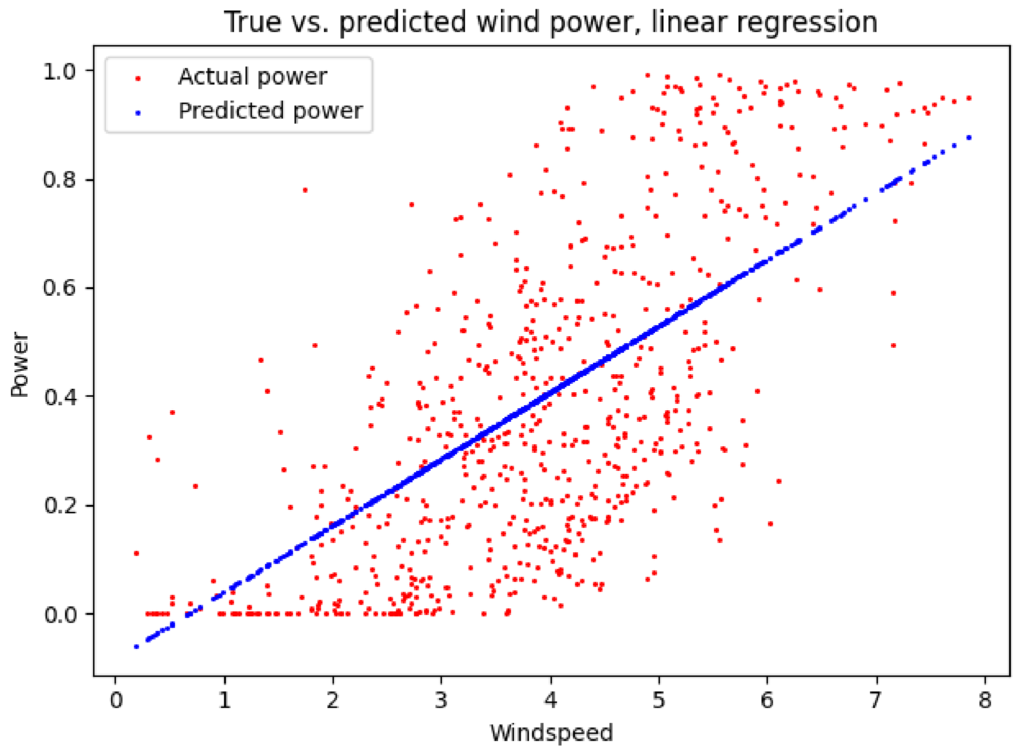

Figure 1: Predicted power generation vs actual power generation for different wind speeds using a linear regression model.

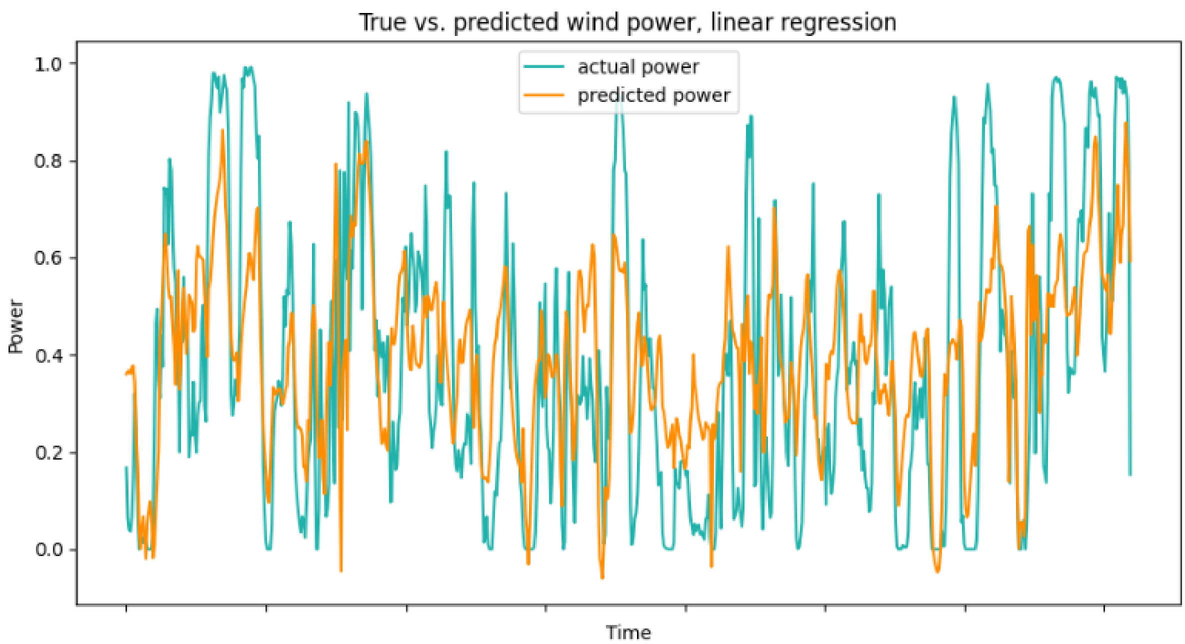

Figure 2: Time series for November 2013 of predicted power generation vs actual power generation using a linear regression model.

1.3.2 Supported vector regression (SVR)

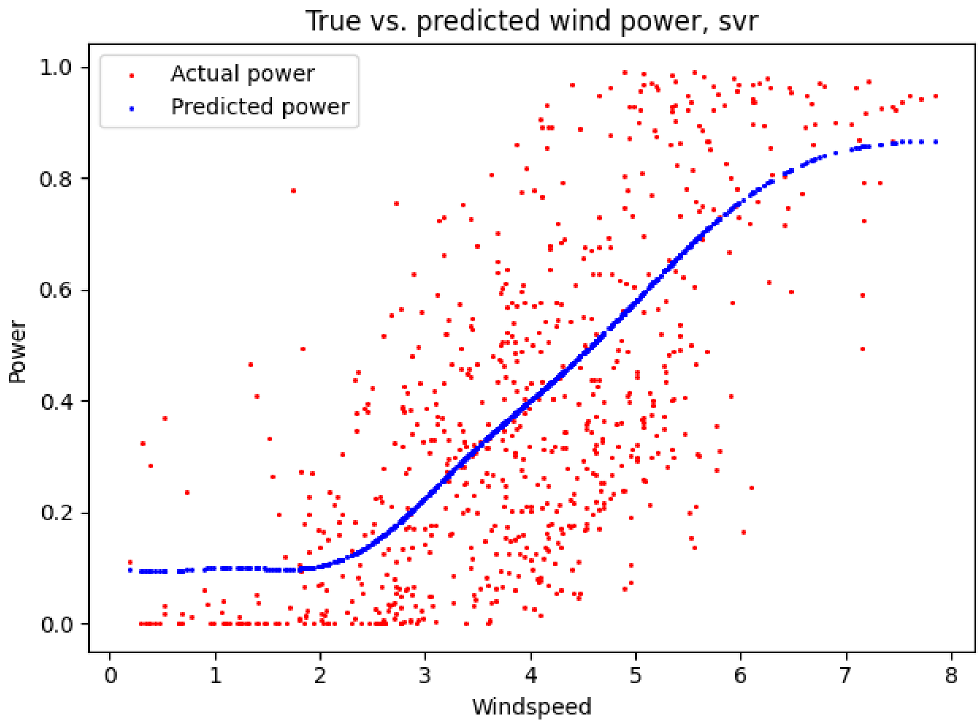

Figure 3: Predicted power generation vs actual power generation for different wind speeds using a SVR model

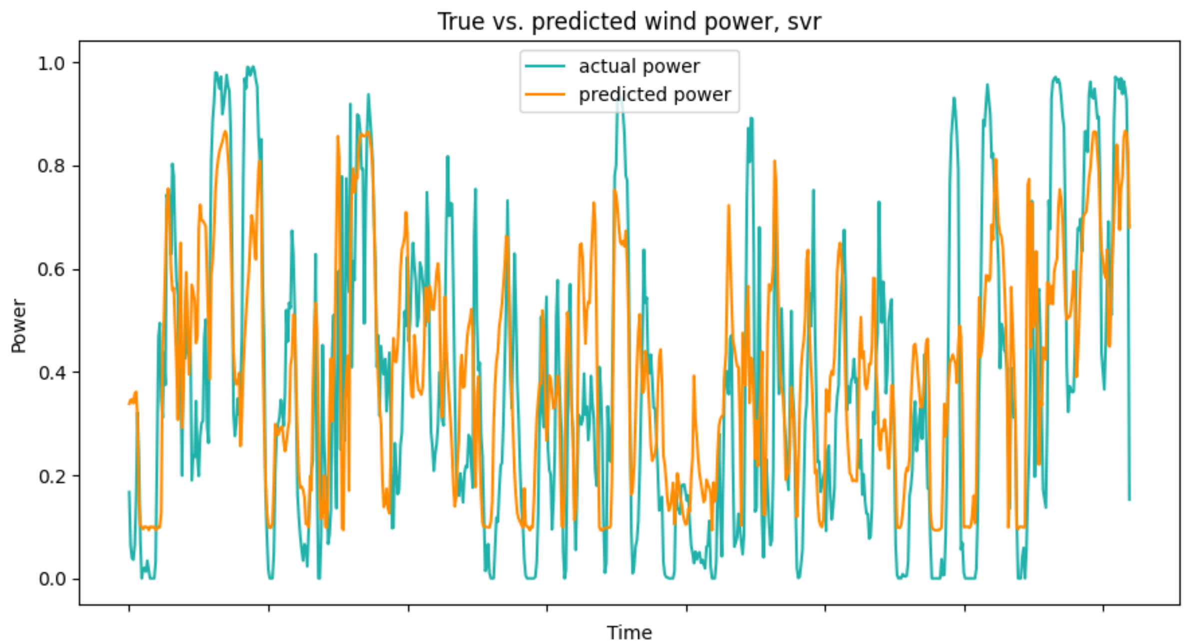

Figure 4: Time series for November 2013 of predicted power generation vs actual power generation using a SVR model

1.3.3 k-Nearest neighbor (kNN)

The kNN model was tested with different values for k, i.e. different amounts of neighbors to look at. The RMSE for different values of k were calculated, and is presented in figure 5. The RMSE decreases clearly with increasing k until approximately k = 20. From k = 20 and onwards, RMSE stays fairly constant.

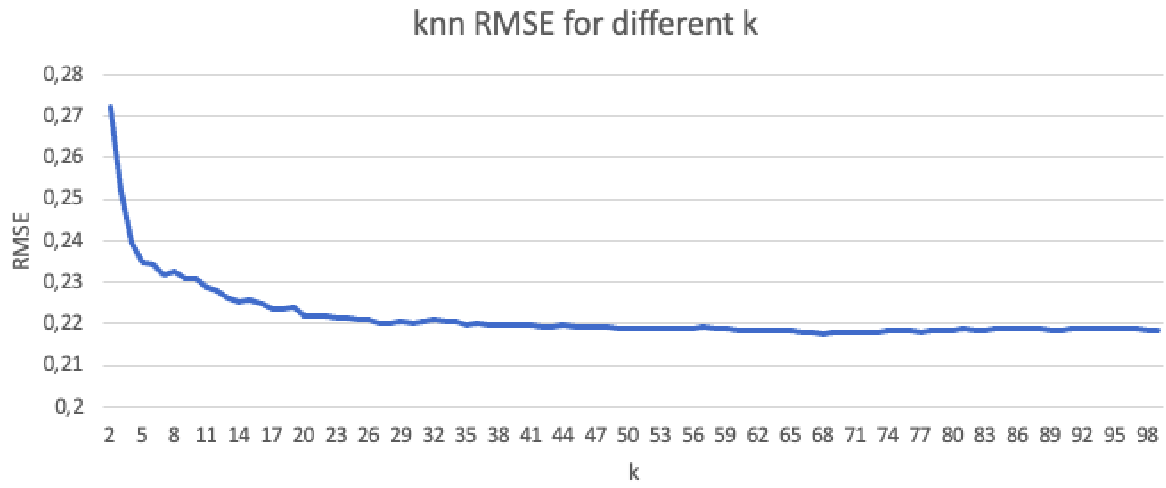

Figure 5: RMSE for kNN models with different k

The actual power vs predicted power is plotted for k=20:

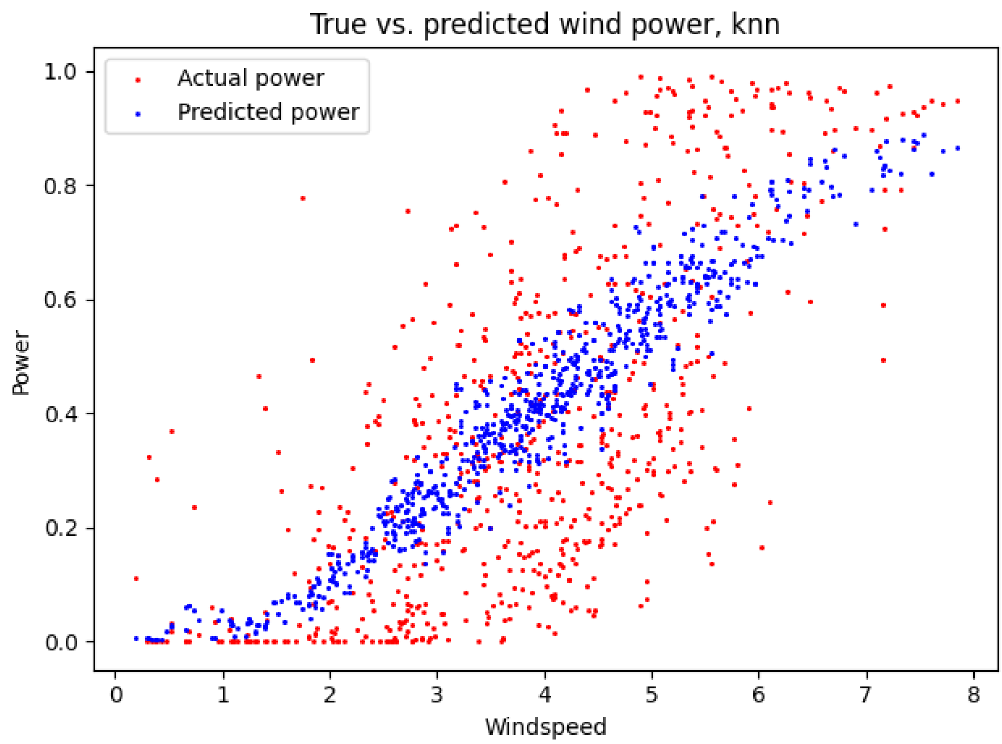

Figure 6: Predicted power generation vs actual power generation for different wind speeds using a kNN model with k = 20

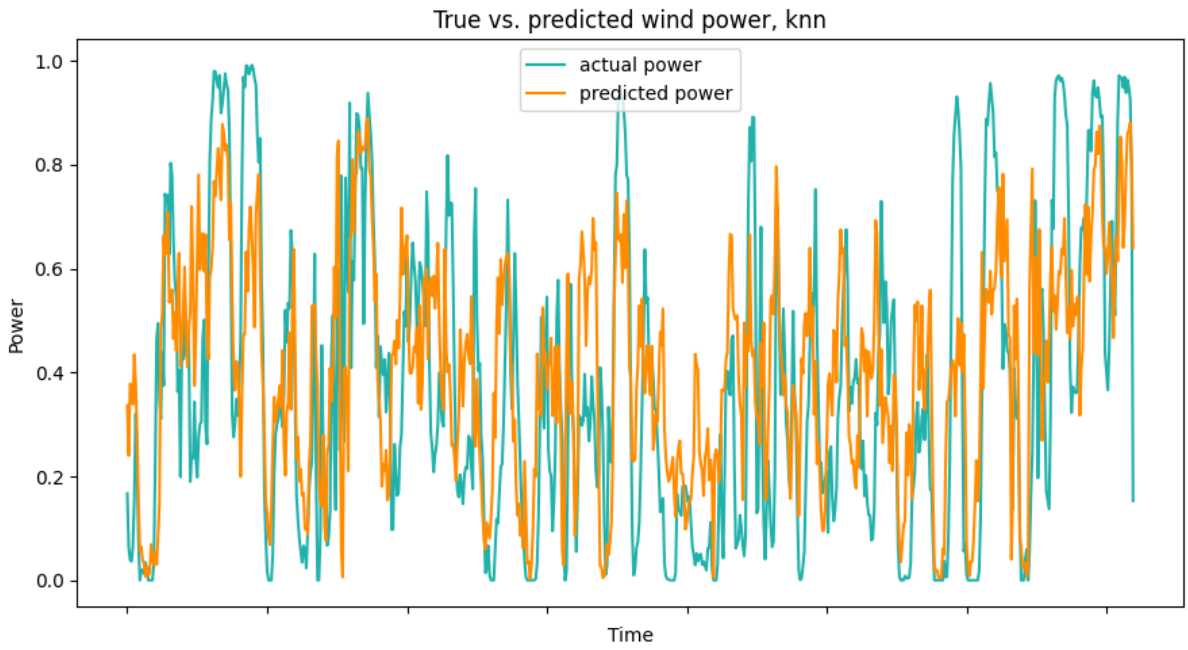

Figure 7: Time series for November 2013 of predicted power generation vs actual power generation using a kNN model with k = 20

1.3.4 Artificial neural networks (ANN)

For the ANN model, we tried 15 different combinations of layers, neurons, scaling, epoch and batch size, and chose the values that gave the lowest RMSE. 3 hidden layers with different numbers of neurons (32, 64 and 100) were selected.

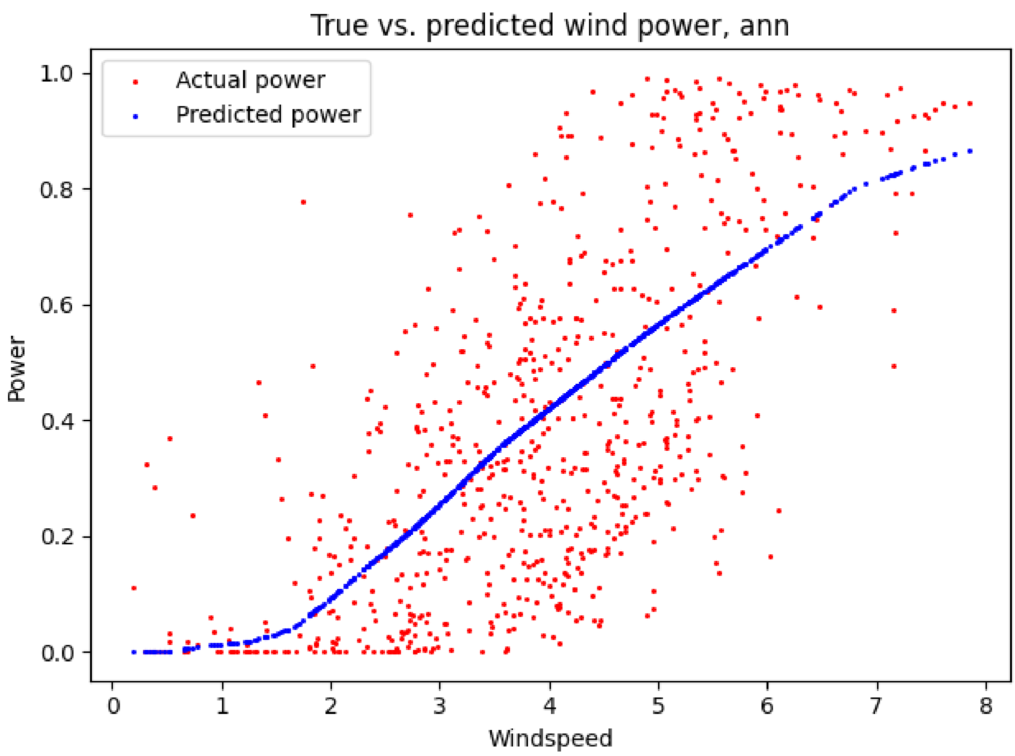

Figure 8: Predicted power generation vs actual power generation for different wind speeds using an ANN model

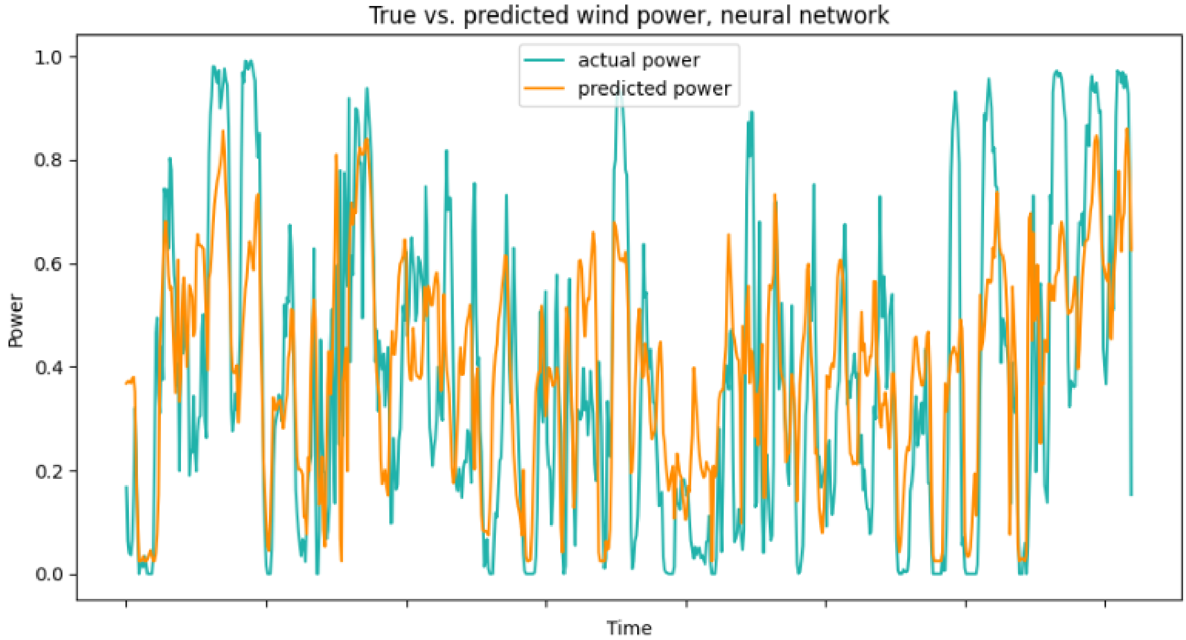

Figure 9: Time series for November 2013 of predicted power generation vs actual power generation using an ANN model

1.4 Discussion and comparison of results

The RMSE for the different prediction models compared to the actual power generation are listed in the tables below. The results were very similar, but the SVR model ended up being the most accurate one, with RMS of 0.21374, while the kNN model ended up being the least accurate one. It is clear from the plots above that the accuracy is not particularly good for any of the models. The models are not very good at predicting the highest and lowest power generation. This is due to the scattered nature of the data, showing that there is not a very clear correlation between wind speed and wind power, and that there are other factors influencing the power generation output. This makes it difficult to predict the resulting generation only based on the wind speed. The reason for the ANN and SVR models being more accurate than linear regression is that the wind generation plotted against wind speed forms a shape fitting more with a curved line than a straight line.

The relatively weak correlation between wind speed and wind power also makes the kNN model have a low accuracy, because the nearest wind speed from the training data set can have a power output significantly different from the output of the prediction.

Table 1: Comparison of RMSE for different prediction models

Model	**Root Mean Square Error (RMSE)**
Linear regression	0.21638
Supported vector regression (SVR)	0.21374

Artificial neural network (ANN)	0.21389
k - nearest neighbors (k=20)	0.222051

2.0 Relationship between wind power generation, wind speed and wind direction

2.1 Introduction

In task 2 we introduce a new parameter in addition to wind speed; wind direction. Therefore, a multiple linear regression (MLR) model was used to predict wind power generation. The wind direction is calculated based on data on zonal and meridional wind speed components in the file *TrainData.csv.*

2.2 Implementation

The code for task 2 can be found in the task2.py file. To run the code, please make sure all files are in the same directory: *TrainData.csv, WeatherForecastInput.csv, Solution.csv, task2.py, read_from_file.py* and *plot_methods.py*In addition to this, the following Python packages should be installed: pandas, numpy, sklearn.

The main method for task 2 is `multiple_linear_regression()`, which contains the machine learning method for multiple linear regression. To run task 2, please run `run_all_models_task2(show_plots, write_to_csv)`. The method will run both `linear_regression()` and `multiple_linear_regression()` so that the results can be compared. The argument show_plots and write_to_csv can be set to either True or False depending on the wish to see the plots of the results and writing the results to csv files.

2.3 Results and discussions

The MLR model was used to predict wind power production for november 2013 from wind data in the file *WeatherForecastInput.csv*. A comparison of the predicted power and the actual power generation from the file *Solution.csv* is plotted in Figure 10.

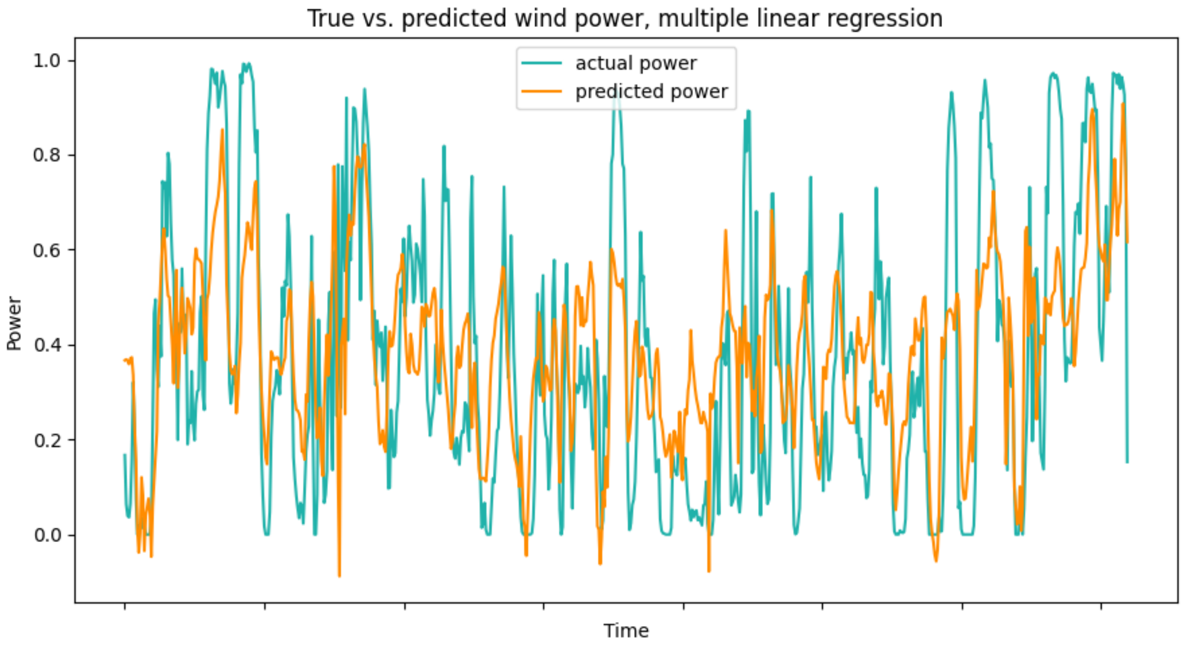

Figure 10: Time series for November 2013 of predicted power generation vs actual power generation using a multiple linear regression model

The resulting RMSE is 0.21556. Figure 11 shows the time series of predicted power generation of the multiple linear regression model vs the linear regression (LR) model from task 1, and Figure 12 compared the two models and the actual power generation.

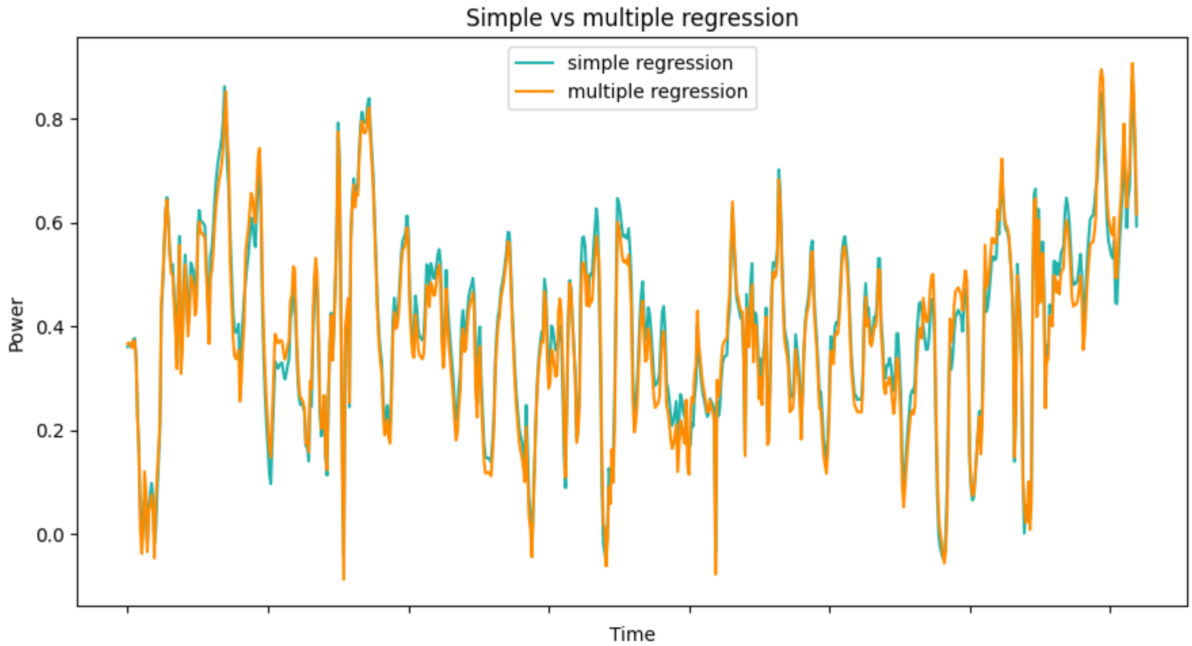

Figure 11: Time series for November 2013 of predicted power generation from a linear regression model vs a multiple linear regression model

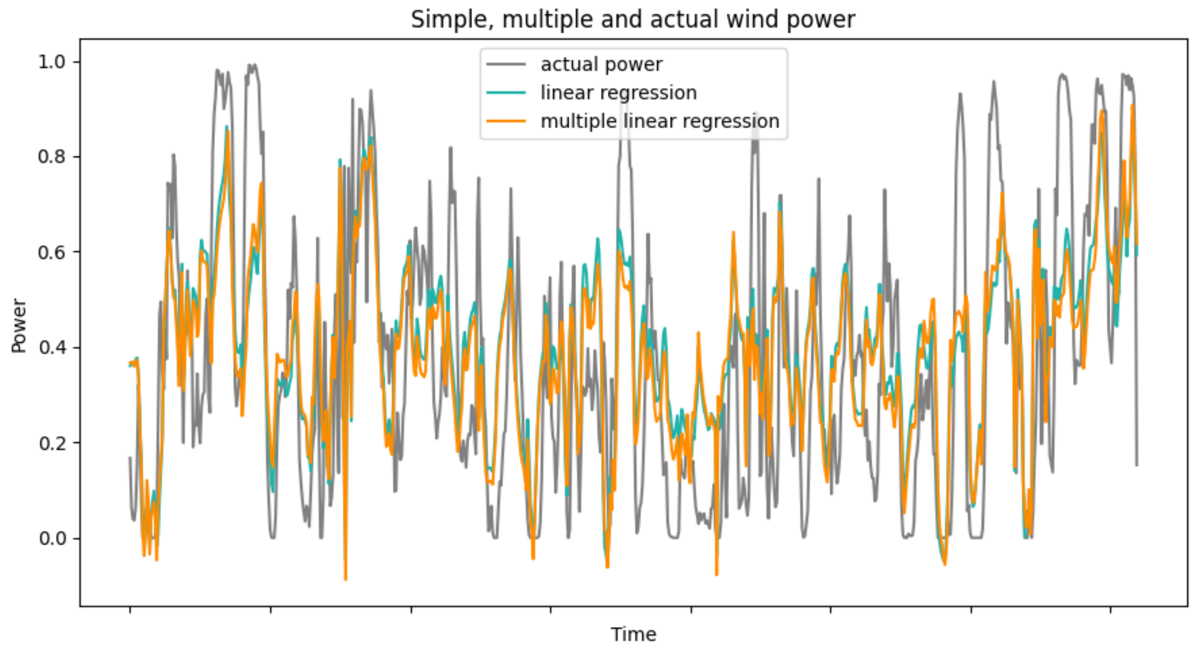

Figure 12: Time series for November 2013 of predicted power generation from a linear regression model and a multiple linear regression model, compared with actual power generation.

Table 2 compares the RMSE of the LR model from task 1 with the MLR model from task 2. The MLR model has a lower RMSE and thus a better accuracy than the LR model, and hence, wind direction plays a role in wind power prediction. Looking at the time series figures it can be seen that the MLR model is slightly more able to follow the actual wind power, but the difference between the two prediction models are small.

Table 2: Comparison of RMSE for LR vs MLR models

Model	**Root Mean Square Error (RMSE)**
Linear regression (LR)	0.216384
Multiple linear regression (MLR)	0.215560

The largest power output from a wind turbine is when the turbine faces the wind direction. Most modern large wind turbines are actively controlled to face the wind, minimizing the jaw angle (the misalignment between wind and turbine pointing direction). The higher prediction accuracy when including wind direction can indicate that the turbine in this assignment is at least not completely flexible when it comes to rotating towards the wind.

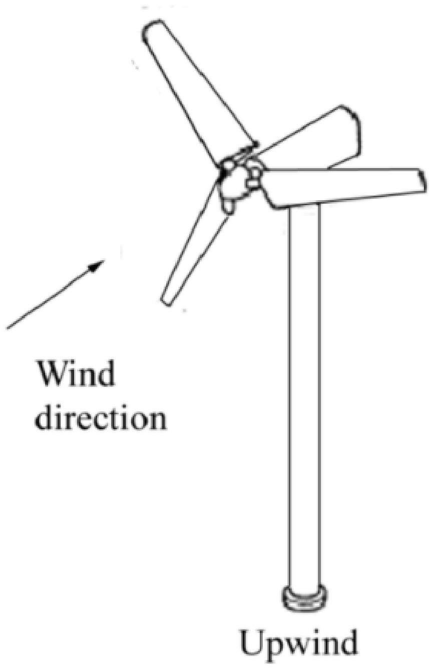

Figure 13: Illustration of wind turbine and wind direction

3.0 Forecasting of power generation using time series

3.1 Introduction

In task 3, wind power generation is predicted by using historical wind power generation instead of weather data. Four different models with four different machine learning techniques are used; linear regression, SVR, ANN and Recurrent Neural network (RNN).

The models are trained with time-series data from the file *TrainData.csv,* using the column POWER. The sliding window method is used, where previous time steps are used to predict the next time step. The models were trained with different window sizes (WS). Meaning that the power generation in one hour is predicted based on the power generation in the WS-previous hour(s). The training data was formulated as following

$POWER\ =\ x_1,\ x_2,\ x_3,\ \dots\ ,\ x_n$

ws = 1

$x_{input}\ =\ [[x_1], [x_2],\ [x_3], \dots\ ,\ [x_{n-1}]]$

$y_{output} = [x_{ws+1},\ x_{ws+2},\ \dots,\ x_n]$

ws = 2

$x_{input}\ =\ [[x_1,\ x_2], [x_2,\ x_3],\ [x_3,\ x_4], \dots\ ,\ [x_{n-2}, x_{n-1}]]$

$y_{output} = [x_{ws+1},\ x_{ws+2},\ \dots,\ x_n]$

ws = 3

$x_{input}\ =\ [[x_1,\ x_2,\ x_3], [x_2,\ x_3,\ x_4], \dots\ ,\ [x_{n-3},\ x_{n-2}, x_{n-1}]]$

$y_{output} = [x_{ws+1},\ x_{ws+2},\ \dots,\ x_n]$

ws = 4

$x_{input}\ =\ [[x_1,\ x_2,\ x_3,\ x_4], \dots\ ,\ [x_{n-4},\ x_{n-3},\ x_{n-2}, x_{n-1}]]$

$y_{output} = [x_{ws+1},\ x_{ws+2},\ \dots,\ x_n]$

3.2 Implementation

The code for task 2 can be found in the task3.py file. To run the code, please make sure all the following files are in the same directory: *TrainData.csv, WeatherForecastInput.csv, Solution.csv, task3.py, read_from_file.py* and *plot_methods.py*. In addition to this, the following Python packages should be installed: pandas, numpy, sklearn and keras/tensorflow.

The main methods for task 2 are `timeseries_linear_regression()`, `timeseries_supported_vector_regression()`, `timeserie_artificial_neural_networks()` and `timeserie_recurrant_neural_networks()`, which implements the machine learning methods for linear regression, supported vector regression, artificial neural networks and recurrent neural networks respectively. In addition to this, the method `create_timeseries(power, windowsize)` creates the training data (x_input and y_output) based on the POWER from TrainData.csv. The structure of the training data depends on the window size. See “3.1 Introduction” for the illustration of this.

To run task 3, please run `run_all_models_task3(actualPower, windowsize, show_plots)`. When running this method the window size can be specified accordingly. The argument show_plots and write_to_csv can be set to either True or False depending on the wish to see the plots of the results and writing the results to csv files.

3.3 Results and discussion

The predicted wind power generation for November 2013 is plotted and compared to the actual wind power generation in Figure 14-17. All the plots are with window size 1. Furthermore, the linear regression model and the SVR model are compared in Figure 18, while the ANN and RNN models are compared in figure 19. The prediction accuracy of the different models is evaluated using RMSE, and is listed in Table 3.

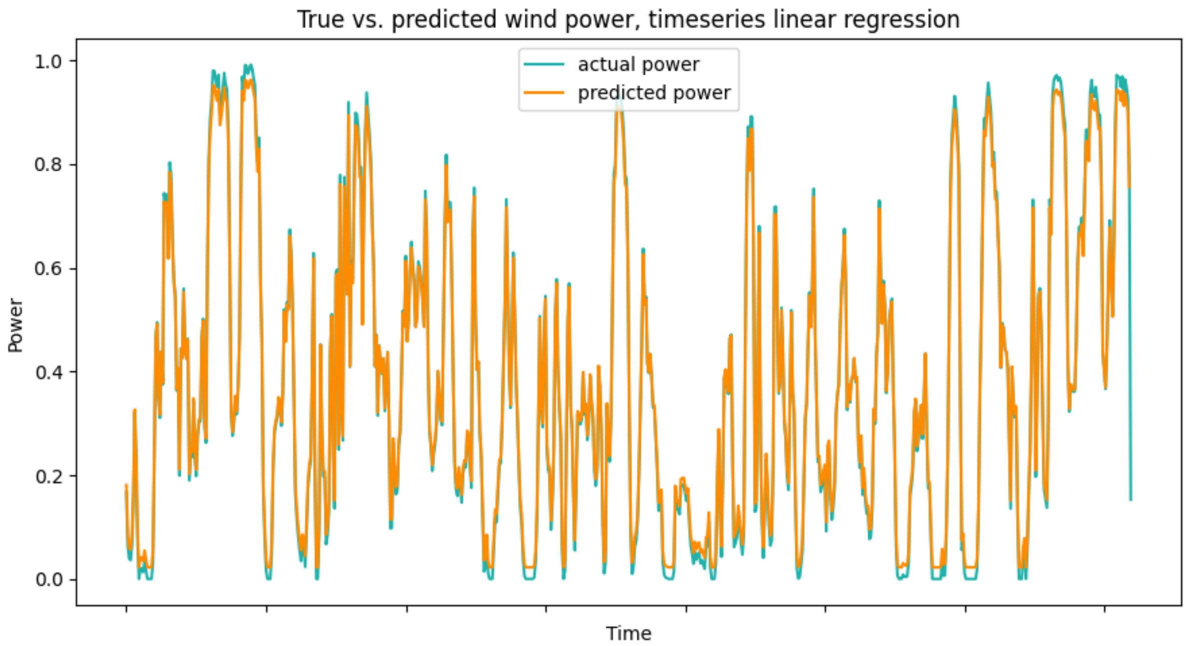

Figure 14: Time series for November 2013 of predicted power generation vs actual power generation using a multiple linear regression model and time-series analysis

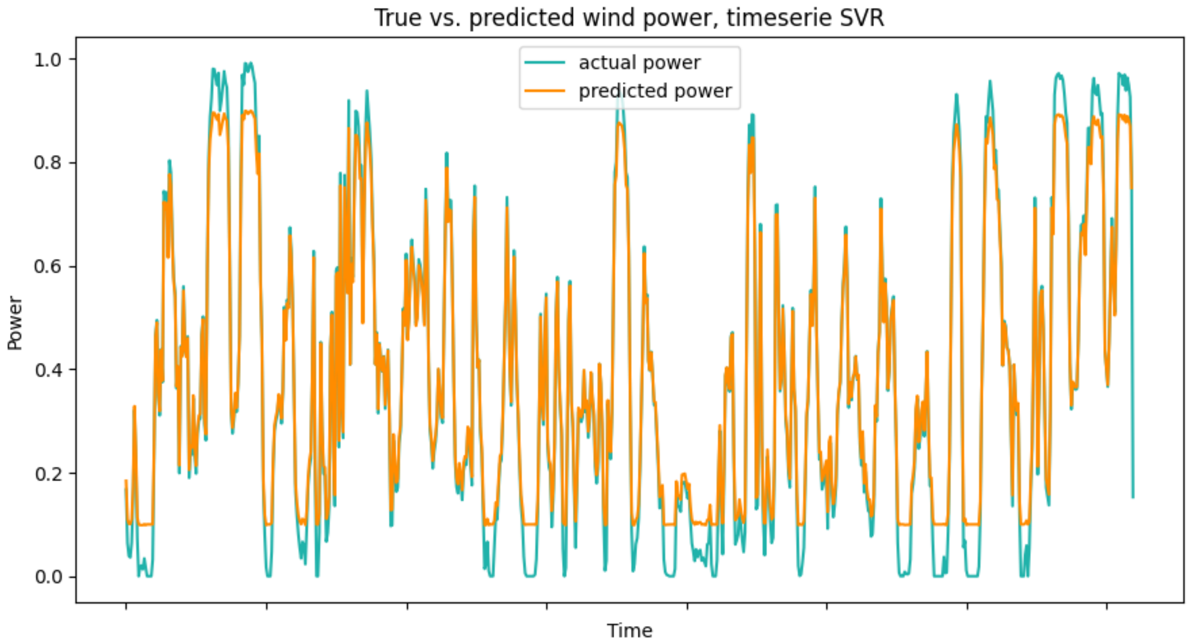

Figure 15: Time series for November 2013 of predicted power generation vs actual power generation using a SVR model and time-series analysis

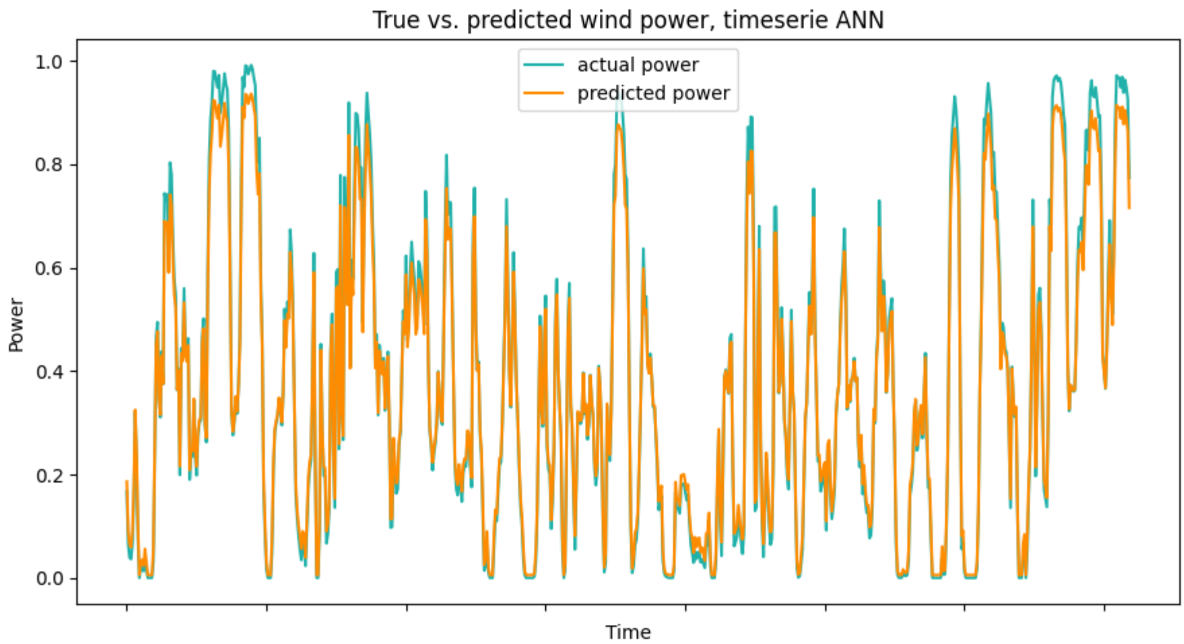

Figure 16: Time series for November 2013 of predicted power generation vs actual power generation using ANN model and time-series analysis

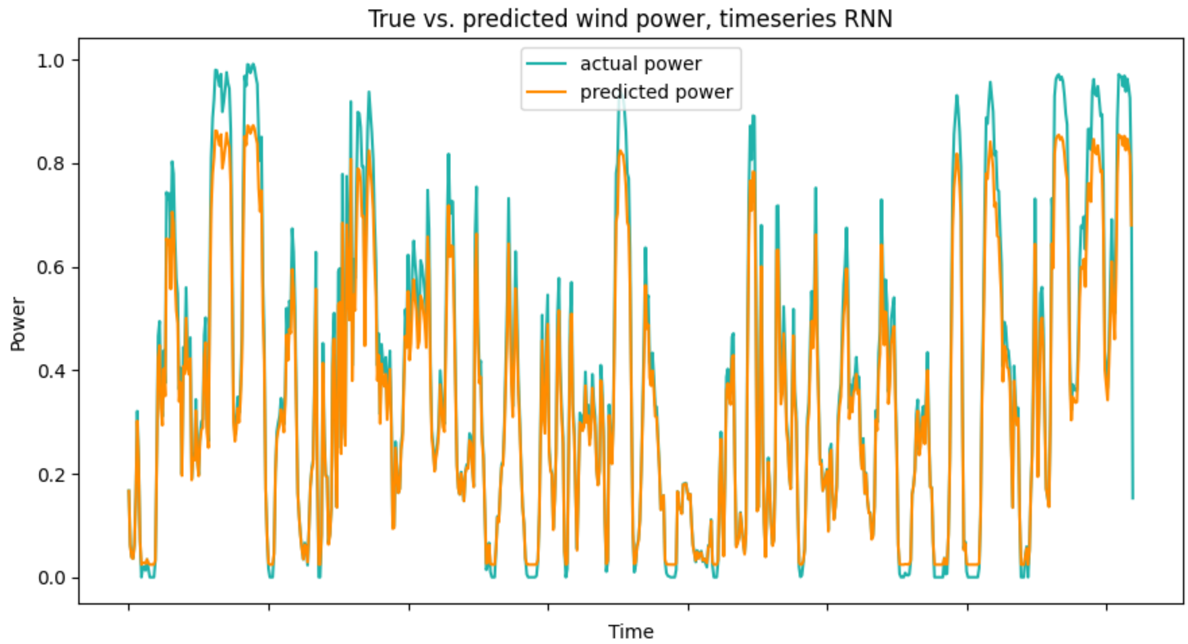

Figure 17: Time series for November 2013 of predicted power generation vs actual power generation using RNN model and time-series analysis

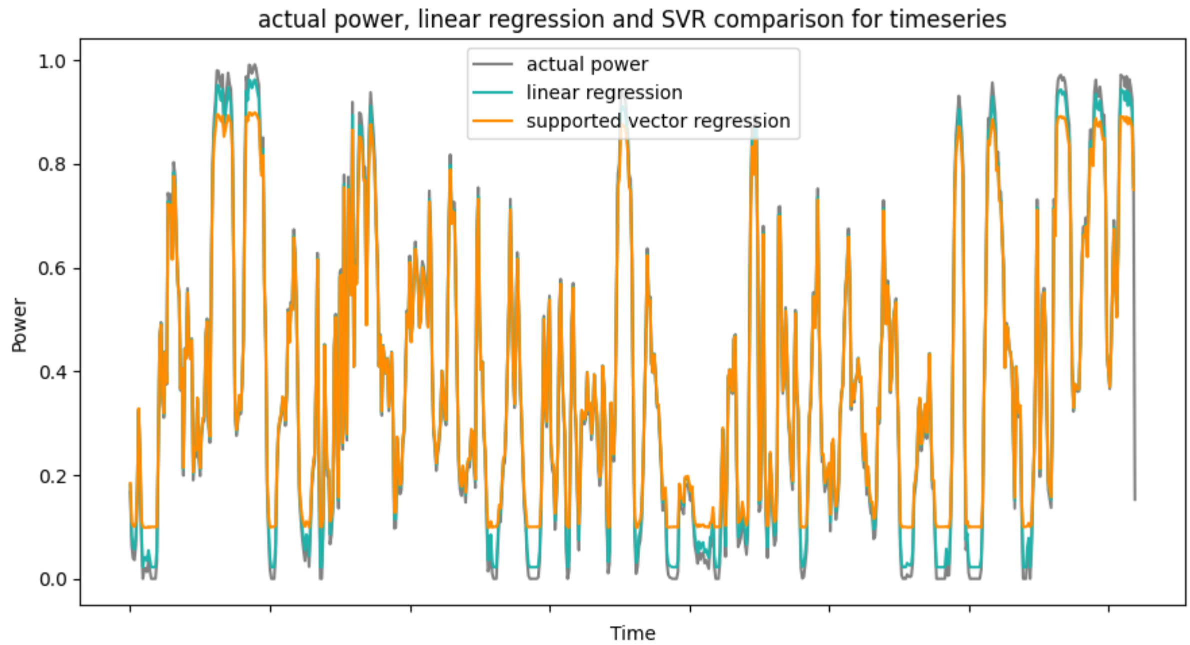

Figure 18: Comparison of time series for November 2013 of predicted power generation vs actual power generation using linear regression and SVR model with time-series analysis

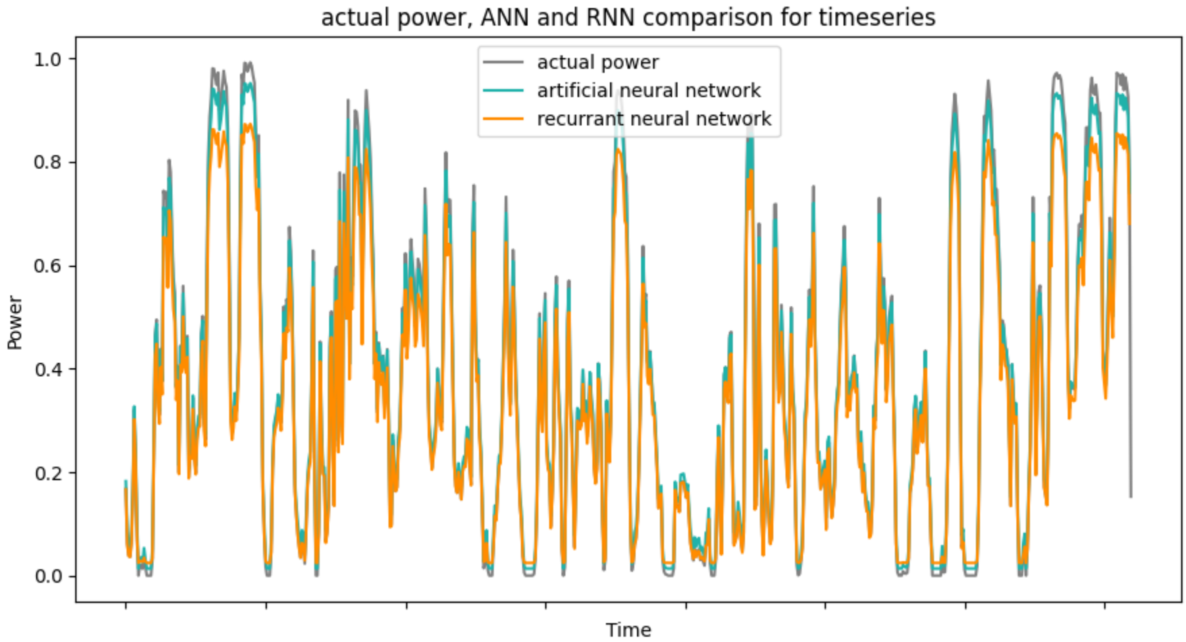

Figure 19: Comparison of time series for November 2013 of predicted power generation vs actual power generation using ANN and RNN model with time-series analysis

It is clear from the figures that the LR model and the ANN model are better than the SVR model and the RNN model at capturing the peaks and bottoms of the power generation. Table 3 and Figure 20 show the RMSE of the different models with different window sizes. The RMSEs underline what was seen from the time series plots - that the LR and the ANN models have lower RMSE and better prediction accuracy than the SVR and the RNN model with window size 1. The prediction accuracy increases for all models with increase in window size, which is expected. The increase in accuracy is the most prominent for RNN, which makes the RNN model with window size 4 the most accurate model, with RMSE = 0.119290.

Table 3: Comparison of RMSE for different prediction models with different window sizes using time-series analysis

Model	**RMSE ws = 1**	**RMSE ws = 2**	**RMSE ws = 3**	**RMSE ws=4**
LinReg	0.125992	0.123502	0.123411	0.123253
SVR	0.129280	0.123592	0.122657	0.123185
ANN	0.125295	0.12082	0.11994	0.120876
RNN	0.127599	0.122814	0.119390	0.119290

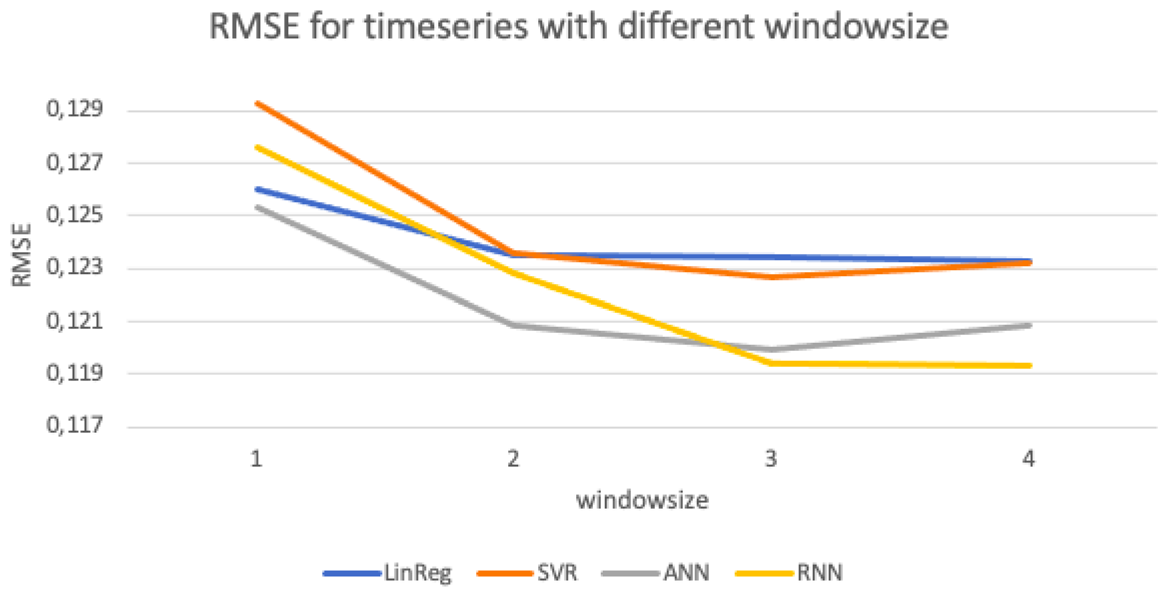

Figure 20: Comparison of RMSE for different prediction models with different window sizes using time-series analysis