



TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Department of Material and Environmental Technology

Machine learning application on wind power variation and weight allocation

MASINÕPPE RAKENDAMINE TUULEENERGIA VARIATSIOONIDELE JA RASKUSJAOTUSELE

MASTER THESIS

Student: Kota Taharaguchi

Student code: 177349KAYM

Supervisor: Dr. Sambeet Mishra

Tallinn, 2019

(On the reverse side of title page)

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently.

No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"....." 2019

Author:

/signature /

Thesis is in accordance with terms and requirements

"....." 2019.

Supervisor:

/signature/

Accepted for defense

".....".....2019 .

Chairman of theses defense commission:

/name and signature/

TUT Department of Materials and Processes for Sustainable Energetics

THESIS TASK

Student: Kota Taharaguchi, 177349KAYM

Study programme, KAYM09/09 - Materials and Processes for Sustainable Energetics

main specialty:

processes for sustainable energetics

Supervisor(s): Researcher, Sambheet Mishra,+372-620-3759

Thesis topic:

(in English) *Machine learning application on wind power variation and weight allocation*

(in Estonian) *Masinõppe rakendamine tuuleenergia variatsioonidele ja raskusjaotusele*

Thesis main objectives:

There are two main objectives in this paper. One is to predict the time series data of Wind power , temperature and events in Estonia. Second is to segregate the power system network into districts/zones from secure to critical.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Time series prediction for wind power temperature by 5 machine learning models.	February in 2019
2.	Time series prediction for events by 5 machine learning models	March in 2019
3.	Apply Unsupervised and supervised machine learning models to the network data.	April in 2019
4.	Writing thesis	May in 2019

Language: English **Deadline for submission of thesis:** 27/05/2019

Student: "....."2019.a

/signature/

Supervisor: "....."2019.a

/signature/

CONTENTS

PREFACE	6
List of abbreviations and symbols	7
INTRODUCTION	7
1. Prediction of time series data	10
1.1 Input data description on wind power capacity factor and temperature	10
1.2 Input data description on Wind RBA(Ramp Behavior Analysis) Data	12
1.3 Input and Output data structure on Prediction	16
1.4 Input data Filter	17
1.5 Wavelet Transformation	17
1.5.1 Basic principle of wavelet	17
2. Prediction structure	22
2.0 Model description	23
2.1 Deep Feedforward (DFF)	23
2.1.1 DFF Prediction structure	24
2.2 Long Short Term Memory Networks	24
2.2.1 LSTM Prediction structure	25
2.3 Attention Mechanism	25
2.3.1 Attention Mechanism Prediction structure	26
2.4 Deep Convolutional Network(DCN)	26
2.4.1 DCN Prediction structure	27
2.5 indRNN Mechanism	28
2.5.1 indRNN Mechanism Prediction structure	29
2.6 Prerequisites for prediction	29
2.6.1 Initial parameter for each model	29
2.2.2 Machine spec	31
2.7 Method for prediction evaluation	32
3. Prediction Result	34
3.1 Result on Temperature and Wind power capacity factor prediction	34
3.2 Result on RBA parameters prediction	36
4. Hyperparameter tuning	39
4.1 Prediction 1 hyperparameter comparison result	40
4.2 Prediction 2 hyperparameter comparison result	43
2. Classification	45
2.0 Motivation	45

2.1 Research Strategy and Methods	45
2.1.1 Data Preparation	46
Markov-Switching Autoregressive model (MS-AR)	46
Brownian motion	47
2.2 Model description	47
2.2.1 DBSCAN	47
2.2.2 K-means	47
2.2.3 Principal component analysis (PCA)	48
2.2.4 T-SNE	48
2.2.5 Convolutional Neural Network (CNN)	48
2.2.6 Decision Tree (DCT)	49
2.2.7 Support Vector Machine (SVC)	49
2.2.8 XGBoost (XGB)	50
2.3 Python for Power System Analysis (Pypsa)	50
2.4 Ranking	50
2.5 Classification and ranking Result	51
Future steps	53
SUMMARY	54
LIST OF REFERENCES	56

PREFACE

This thesis work is done as two parts. One is a predication part which uses 5 machines learning models to forecast wind power , temperature and wind power events in Estonia. Another part is a classification that analyzes the power flow network to embed the geographical data in Estonia.

I would like to thank Dr. Sambeet Mishra for supervising this master thesis. With his kind guidance, I could archive this work and learn a lot about data analysis.

keywords: Machine Learning, supervised learning, Deep learning, multivariate prediction, wind power, classification, network information visualization, hyper parameter tuning

List of abbreviations and symbols

WP - Wind Power

TEMP - Temperature

LSTM - Long Short Term memory

DFF - Deep Feed Forward

CNN - Convolutional Neural Network

HT - Hyper Parameter Tuning

indRNN - Independent Recurrent Neural Network

FFT - Fast Fourier Transform

WT - Wavelet Transform

INTRODUCTION

Quantification of wind power variation improves power system planning. Specifically, an accurate prediction ensures the system balance and reserve power capacity allocation. The electrical power network is deteriorating, and efficiency decreases with time. Determining the network conditions based on the network reliability framework is a pre-condition for network expansion to determine investments.

The objective of this thesis is to implement different machine learning methodologies and compare their performance within two main power systems related applications. In particular, two main tasks will be developed:

1) prediction of time series ramp events

A ramp event is a measure to quantify the wind power variations. Each event has certain properties such as peak, ramp-up rate, ramp-down rate, rise-time, and full-time. The wind farm operator needs to make predictions for future power production. This information is then used by the power system operator to maintain the supply-demand balance. In place of predicting time-series data, it is possible to predict the events that can take place. Thereby, the system planner can decide the actions according to the forecast events. The main actions are related to planning maintenance operations and preparing bids on quantity. Therefore, in 1, the ramp events extracted from the time-series wind power production will be used as an input for short-term prediction.

2) power network topology reduction using dimensionality reduction techniques.

The power network typically has elements - transmission lines, power system apparatus (transformer, switchgear, etc.), consumers and terrain information. However, these elements are in different condition such as efficiency and numbers of faults that took place. Based on the condition of the aforementioned elements the objective of this task will be to segregate the network into districts/zones from critical to normal. This is a way to allocate weight to a zone that in turn allocate weights to the lines and nodes in that zone. Thereby, this weight factor determines the importance of the zones for the planner. Therefore, in the proposed task the original network topology will be reduced using power transmission distribution factors. Additionally, weights will be allocated to create zones. The outcome will be the definition of clusters of nodes and arcs: those having similar properties together will be merged such that the total network size is reduced. Machine learning methodologies for clustering will be tested for this purpose.

In order to achieve the objectives discussed above, the existing machine learning models from the literature will be adapted for comparison and benchmarking. Particular attention will be given to supervised learning algorithms (support vector machine, XGBoost, Decision Tree), unsupervised learning algorithms (DBSCAN, K-means, PCA and T-sne) and deep learning models (Convolutional Network).

Optimal model parameters will be selected for improvements in prediction. The thesis will contribute to the topics time-series prediction and optimal weight allocation through model comparison. In addition, optimal parameter selection for improving model performance will represent another novel contribution to the machine-learning model.

Until now, energy analytic has been done as an important area of research because there is a huge impact on the economical and environmental development in a country. A bunch of studies focus on power network analysis and prediction the future data based on the past data. In machine learning methods, the basic technique is to use linear regression for modeling the relationship between predictors and predictant. In [1], it reported a statistical approach to forecast with internal and solar gains. [2] proposed the combination approach based on Extreme Learning Machine (SLM) and an error correction model to predict wind power in the short-term time scale. From these researches, machine learning method is very much effective to solve short and long term prediction problems. Based on this background information, we have compared 5 machine learning models to predict the wind power and temperature, wind power events.

1. Prediction of time series data

The Prediction part consists of Prediction 1 which predicts Wind power capacity factor and Temperature in Estonia, and Prediction 2 which predicts Ramp Behavior Analysis parameters extracted from Wind power capacity factor.

1.1 Input data description on wind power capacity factor and temperature

Firstly, we focused on predicting time series data about wind power capacity factor and temperature in 2016 based on 2011 to 2015 data.

The wind power and temperature input data are taken from Renewables.ninja [3] which contains global reanalysis models and satellite observations. The wind power capacity factor (WF) and

temperature (TEMP) data are retrieved from Renewables.ninja for 5 years (2011-2016). The data is segregated into model training (2011-2015) and testing (2016).

Table 1 presents the statistical information of the wind power capacity factor (Kwh/kW) from 2011 to 2016. The table contains the number of data count, mean, standard deviation (std), min, 0.25, 0.50 and 0.75 which give percentile values of all numeric values in a column and max in each year.

Table1. WF and TEMP input data structure in 2011-2016

Year	2011		2012		2013		2014		2015		2016	
Name	WF	TEMP	WF	TEMP	WF	TEMP	WF	TEMP	WF	TEMP	WF	TEMP
count	8760	8760	8760	8760	8760	8760	8760	8760	8760	8760	8760	8760
mean	0.26	6.27	0.248	4.57	0.228	5.83	0.219	6.31	0.266	6.87	0.232	6.09
std	0.21	10.19	0.183	10.47	0.196	10.32	0.185	9.48	0.216	7.64	0.192	9.30
min	0.00	-29.69	0.000	-30.11	0.001	-23.57	0.001	-19.10	0.001	-16.83	0.000	-18.08
0.250	0.09	-0.22	0.101	-1.80	0.072	-1.78	0.074	-0.38	0.098	0.93	0.079	-0.47
0.500	0.20	6.06	0.207	5.43	0.165	5.57	0.159	5.50	0.202	5.77	0.179	4.36
0.750	0.39	14.51	0.358	13.09	0.340	14.89	0.327	13.72	0.384	13.20	0.335	14.42
max	0.97	28.20	0.892	27.97	0.971	28.19	0.852	29.47	0.962	27.27	0.908	26.73

We created a heat map and 3-d plot for each data to check the data structure in more detail as shown in Fig. 1. Fig. 1 shows the average time-series data of TEMP in Estonia. It is evident that the temperature is high from June to August due to the influence of the summer season, and the temperature is low from December to February. According to the result of the Prediction of Renewables.ninja, the highest temperature in Estonia from 2011 to 2016 was 29.47 ° C, and the lowest temperature was -30.11 ° C.

In addition to that, we plotted 3-d graph in the time series of year and month for Temperature. According to the plot, the temperature rises from April to May in all the years. Also, a comparison between January in 2011 and 2016 shows that the temperature was lower in 2011.

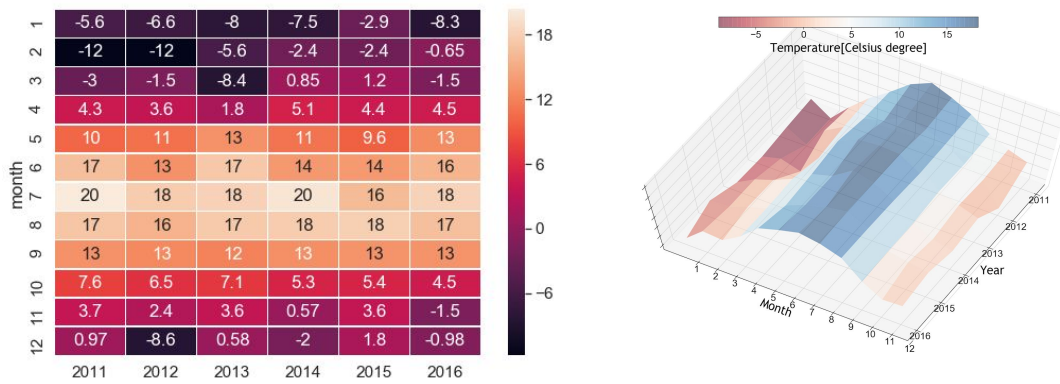


Fig 1. TEMP heat map (Left) and 3-d plot (Right)

Fig. 2 shows the heatmap and 3-d plot of WF in Estonia. It turns out that relatively wind turbine produces more power from November to December, which is the winter season. On the other hand, we found the smallest amount of WF in July most years. From 2011 to 2016, the most significant wind power capacity factor was 0.97, the lowest was 0. Also, we plotted 3-d graph in the time series of year and month for WF. The figure shows that the capacity factor for December is the highest in any year. Furthermore, in January and March on 2013 and 2014, the capacity factor is lower compared to other years, and the same trend can be seen in May and July.

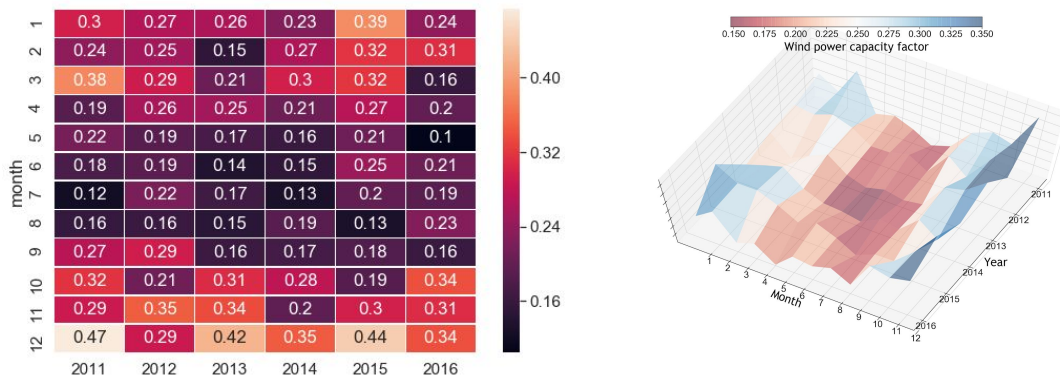


Fig 2. WF heat map (Left) and 3-d plot (Right)

1.2 Input data description on Wind RBA(Ramp Behavior Analysis) Data

Wind ramp events are extracted using the RBA methodology proposed in [4]. The events parameters presented in Fig. 3. There are five parameters to analyze in this part. First one is a graphical representation. $w_s(t)$ that indicates the value of ramp behavior and the value of capacity factor at the time of vertex. Next, t represents the time when the ramp event occurred.

Thirdly, Δw_s is the difference from the vertex based on threshold set when extracting RBA data from wind power capacity factor data. Fourthly, Δt indicates the occurrence interval of the ramp event. Lastly, $\theta(w_s)$ is the angle from the first departure point in contact with the threshold to the top.

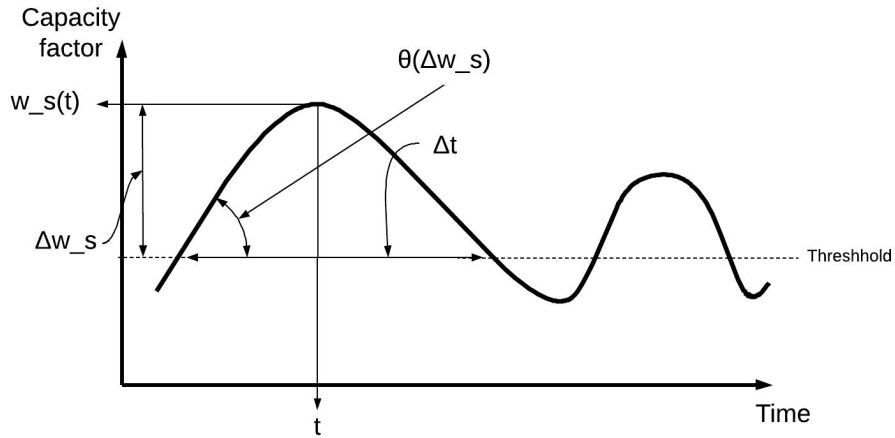


Fig. 3. Wind ramp behavior classification

Table. 2 shows statistical properties of the RBA parameters. This table contains the same statistic information with Table. 1.

Table. 2. RBA Data extracted from WF from 2011 to 2016

	$w_s(t)$	t	Δw_s	Δt	$\theta(\Delta w_s)$
count	6131	6131	6131	6131	6131
mean	0.25	26326.40	0.00	9.53	0.37
std	0.20	15242.50	0.22	6.60	10.72
min	0.00	0.00	-0.95	1.00	-36.37
0.25	0.09	13124.00	-0.10	5.00	-7.36
0.50	0.20	26469.00	-0.02	8.00	-1.58
0.75	0.37	39550.00	0.11	13.00	8.05
max	0.97	52604.00	0.83	56.00	45.62

On the same way to wind power capacity factor and temperature, the cross heat map and 3-d plot of year and month are shown in Fig 4-8 for each value.

Fig. 4 shows the heatmap and 3-d plot for the mean value of time gap (Δt). It can be seen that Δt of the Ramp behavior is higher in December-January all the years, as the same trend that

capacity factor was high in the period. Among them, it recorded the maximum value of 12.265 in 2014. Conversely, the value is generally lower in June and August. In the 3-d plot, It can be seen that Δt , which is the occurrence time of the ramp event, increases in proportion to the increase of the Wind power from November to December. We could also find that the value of Δt has generally decreased from 2011 to 2016 from the plot.

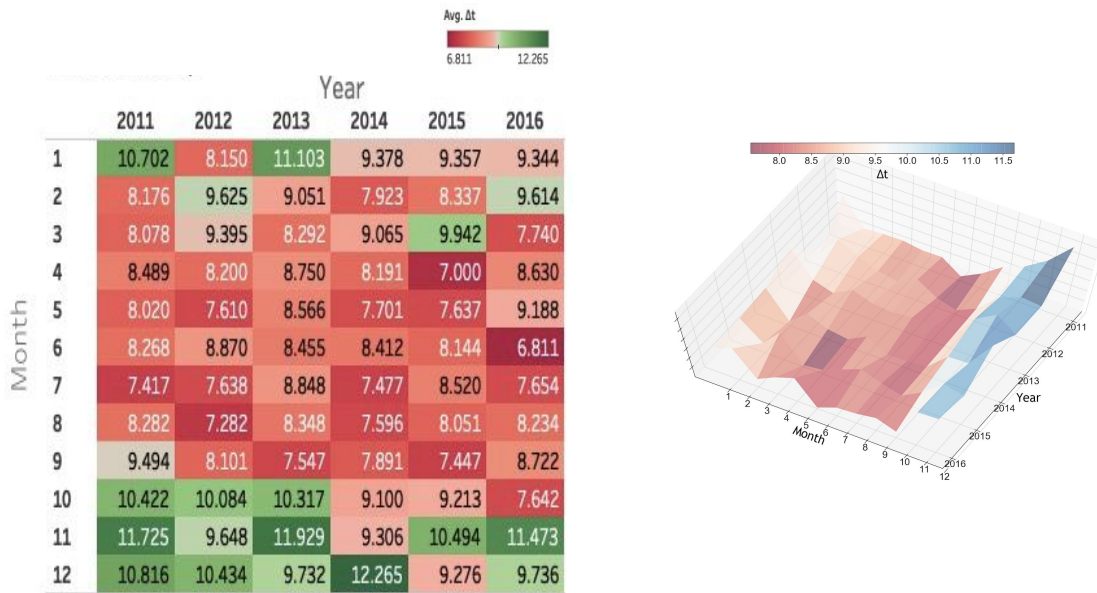


Fig. 4. Time gap (Δt) heatmap

Fig. 5 shows the heat map and 3-d plot of the ramp event value ($w_s(t)$). The trend shows that the maximum value in each year is recorded in December. We can also find that the value has risen significantly from September to October except 2012 and 2015. In the 3-d plot, It can be seen that $w_s(t)$ risen in January-February 2015-2016. Also, the value of $w_s(t)$ has been decreasing in October-November since 2013.

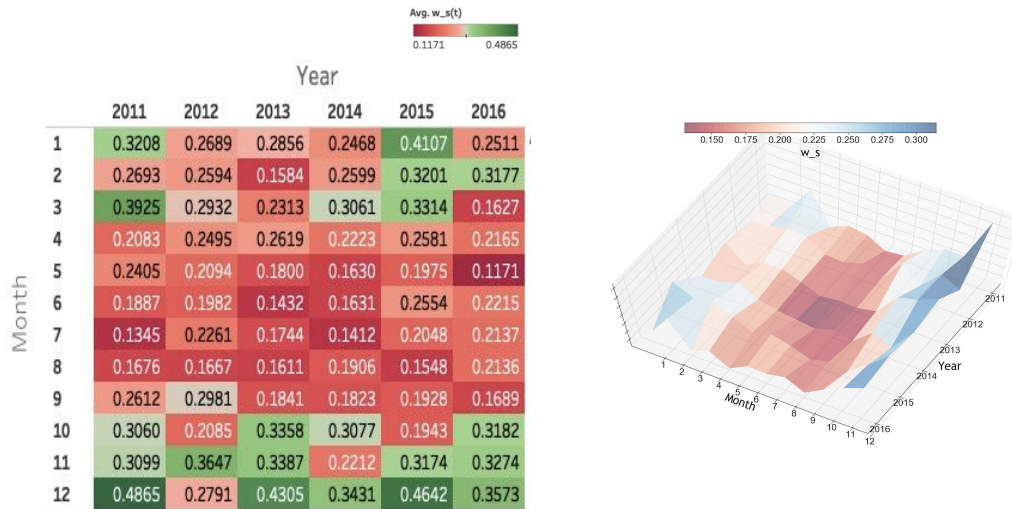


Fig. 5. Ramp event value($w_s(t)$) heatmap

Fig. 6 shows the heat map and 3-d plot of Amplitude of ramp behavior (Δw_s). Δw_s indicates the distance to the set threshold, but we couldn't find much difference in each year and month. The values for December are lower for all years. From this point, we could find that Δw_s becomes smaller in December when wind power is higher.

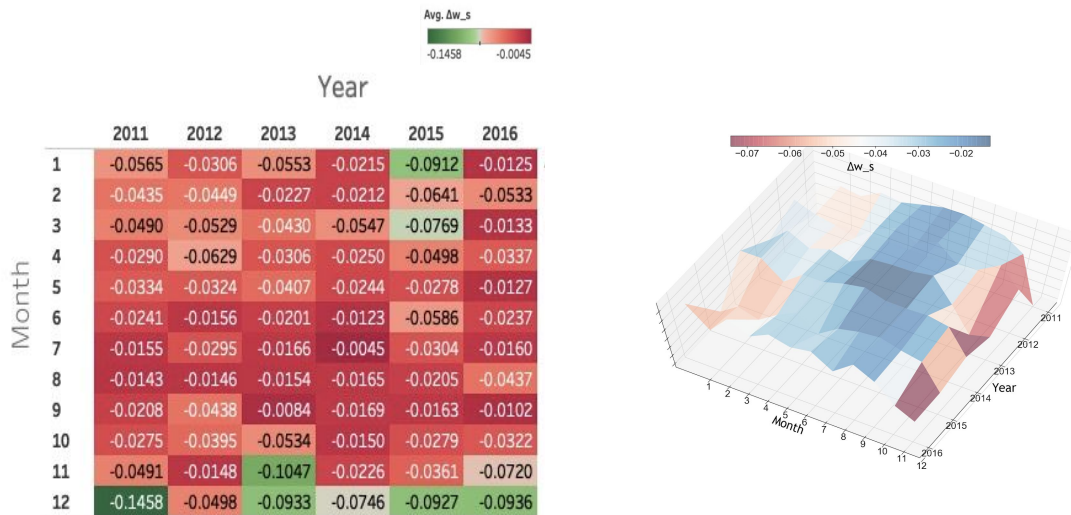


Fig. 6. Amplitude of Ramp behavior(Δw_s) heatmap

Fig. 7 shows the heat map and 3-d plot of angle of ramp behavior ($\theta(\Delta w_s)$). This angle can be recorded in the range of 0 to 1 in most times. We could observe the values were smaller in winter, but the trend is not so clear.

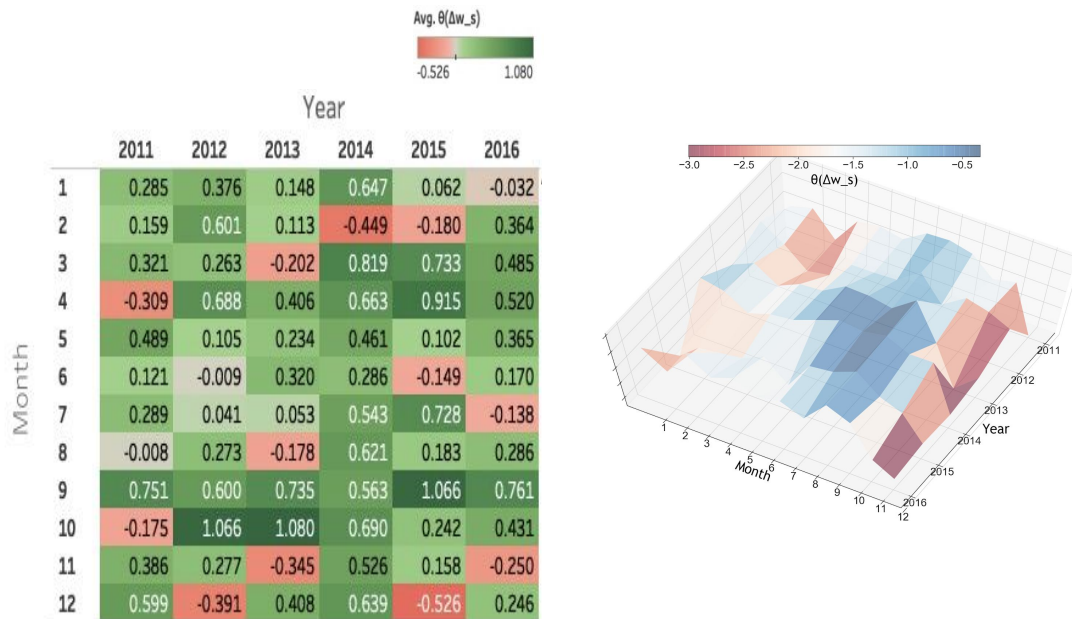


Fig. 7. Angle for amplitude of ramp behavior($\theta(\Delta w_s(t))$) heatmap

Fig. 8 shows a heatmap of Time when Ramp behavior happened(t). We have not prepared a 3-d plot about t because it is clear that the occurrence time of the ramp event increases proportionally.



Fig. 8. Time when Ramp behavior happened(t) heatmap

1.3 Input and Output data structure on Prediction

There are four model structures presented below.

- Multiple Input Multiple Output (MIMO)
- Multiple Input Single Output (MISO)
- Single Input Single Output (SISO)
- Single Input Multiple Output (SIMO)

We have chosen to demonstrate the MIMO and SISO structures in this thesis. However, MISO and SIMO can be implemented following similar fashion. Fig.9 shows the MIMO structure for the prediction. In prediction 1, the input data is WF and TEMP, then the output data is WF and TEMP in the same way. This means prediction 1 is 2 inputs \times 2 outputs prediction structure. In prediction 2, the input data is RBA parameters ($w_s(t)$, t , Δw_s , Δt , $\theta(\Delta w_s)$) and the output data is also the same. The prediction structure is 5 inputs \times 5 outputs.

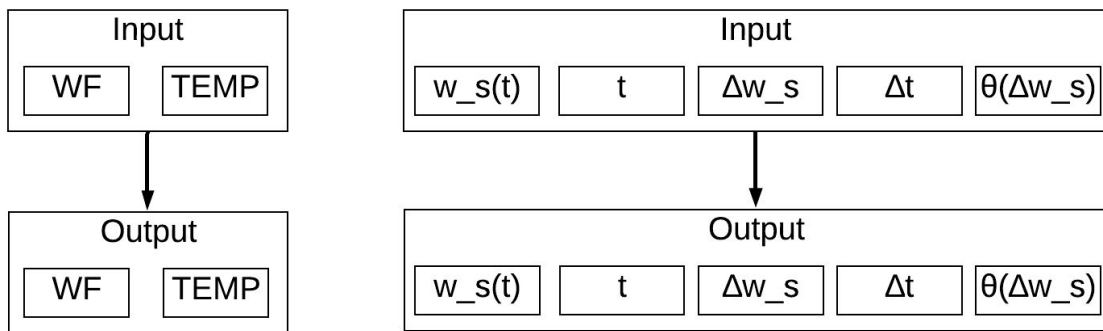


Fig. 9. MIMO for Prediction (Left: WF and TEMP, Right: RBA parameters)

1.4 Input data Filter

In [3] a wavelet filter is applied to improve the prediction efficiency of the LSTM network-based wavelet model. In that paper, It has been observed that the wavelet-based layer proposed in the study significantly improves the recognition performance of conventional networks. Using filter transforms the time-series into signal which is easier to adapt for the model.

1.5 Wavelet Transformation

We have used wavelet transformation to transform the data from time to frequency domain. Wavelet transformation was applied in the form of encoding before putting data into the model,

and the process of decoding the data output from the model and returning it to time series data was performed.

1.5.1 Basic principle of wavelet

Wavelets are a popular tool for computational analysis. They provide localization in both the temporal domain as well as in the frequency domain [5]. A prominent feature is the ability to perform a multiresolution analysis [6]. This method is key to the good performance of wavelets in applications such as data compression and denoising.

Wavelet transformation methods can be categorized as the discrete wavelet transform (DWT) and the continuous wavelet transform (CWT). The CWT allows wavelet transforms at every scale with continuous translation [7]. The CWT is widely used in pattern matching, such as discontinuity and chirp signal detection [8]. Mathematically, the CWT can be represented as [4].

The DWT operates over scales and positions based on the power of two components.

On decomposing the original signal $y(t)$, two components cA_1 and cD_1 are produced by convoluting the signal with a decomposition low pass filter and a decomposition high pass filter, respectively [9].

In [10], the original series in terms of components is represented as:

$$y_t = cA_j + \sum_{j=1}^J cD_j$$

y_t : Actual value at time (t)

cA : Approximation coefficient (j)

cD : Detail coefficients (j)

In this section, we used PyWave package called PyWavelet [11]. With this Package, we could apply continuous and discrete wavelet transforms for the datasets. In Fig. 10, we presented the sample data (count=1000) of original WF data, wave after CWT and wave after inverse CWT. In addition, you could observe the result of DWT method in Fig. 11.

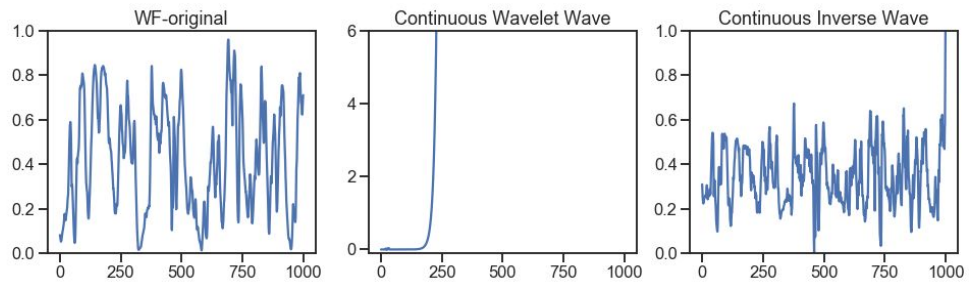


Fig 10.WF with continuous Wavelet transform (Left:Original WF, Middle:Wave after CWTt, RIGHT: Wave after inverse CWT)

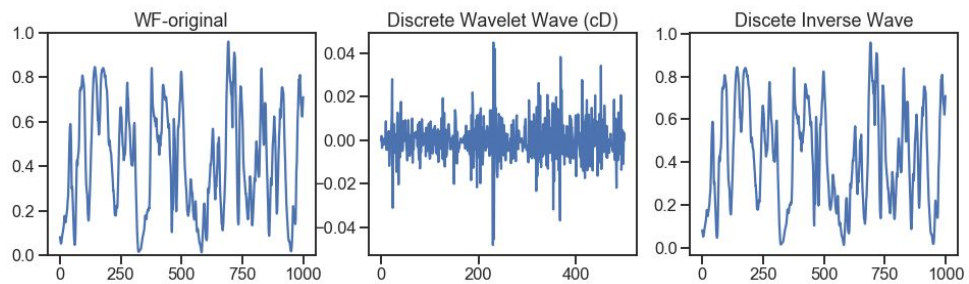


Fig 11.WF with discrete Wavelet transform (Left:Original WF, Middle:DWT component, RIGHT: Wave after inverse DWT)

Table.3 shows the statistic information of original WF, Inverse CWT and inverse DWT data. As Table. 3 shows, Discrete wavelet transform can transform and inverse the data mutually. Additionally, the average gap of continuous and discrete wavelet transform is around 19%, Inspired by that, we have chosen to pre-process with discrete wavelet.

Table. 3. Comparison between original, Inverse CWT and Inverse DWT

WF	original	Inverse CWT	Inverse DWT
count	1000	1000	1000
mean	0.42	0.34	0.42
std	0.24	0.13	0.24
min	0.01	0.00	0.01
0.25	0.21	0.25	0.21
0.5	0.42	0.33	0.42
0.75	0.62	0.44	0.62
max	0.96	1.00	0.96

Fig. 12 shows WF and TEMP wavelet transformations respectively. cD is the detail coefficient and cA is the approximation coefficient. WF and TEMP are divided into two components cA and cD that is input for the models.

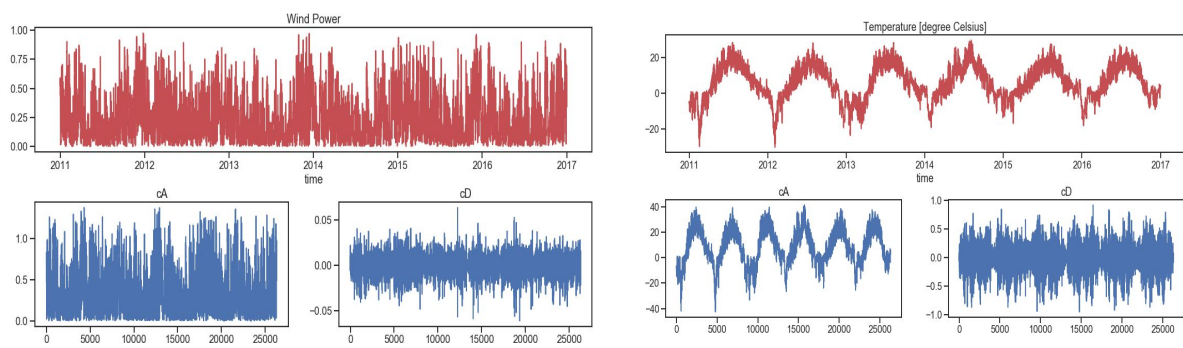


Fig 12.WF and TEMP with DWT

1.6 Fast Fourier Transformation (FFT)

Next, for the comparison, we applied Fast Fourier transformation for the dataset. This transformation computes the n-dimensional n-point discrete Fourier transform with the efficient Fast Fourier Transform algorithm. In [12], using FFT accelerates training and inference by a significant factor and can lead to a speedup of over an order of magnitude. You can see the method which we tried to implement in Fig. 13. As Fig. 13 shows, after applying FFT to the dataset, we have used min-max normalization which normalizes the data from 0 to 1 for training efficiently. Then, after training by the model, we used the inverse-normalized method and inverse-FFT for the data to get the predicted output.

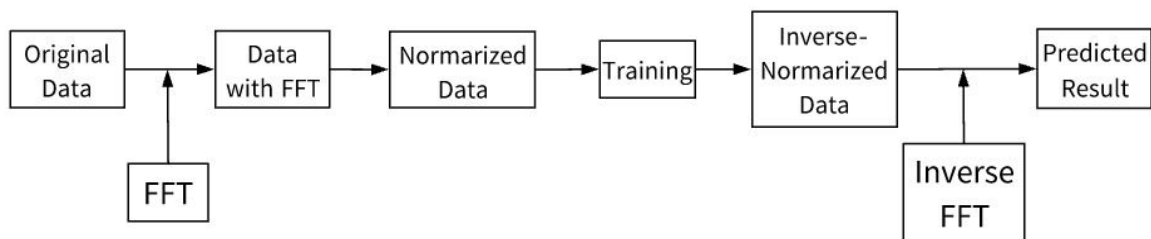


Fig. 13 Workflow of applying FFT

Fig. 14 shows the comparison of original data, data with FFT and data with inverse-FFT. As you can see, after the FFT, the signal can be reduced significantly. In Table. 4, we presented the gap between original data and inverse-FFT data. It completely returns to the original signal with the inverse-FFT.

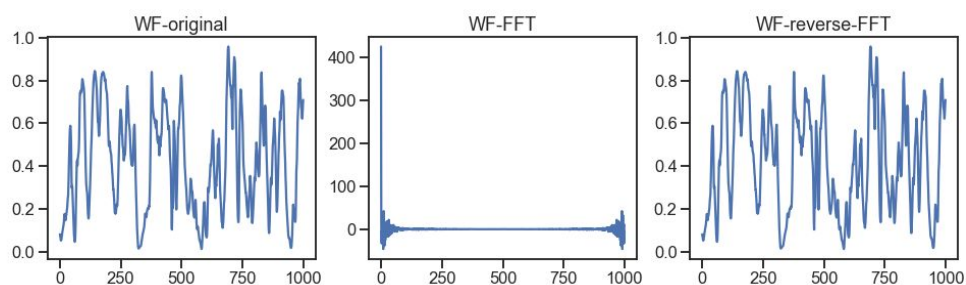


Fig. 14. WF with continuous FFT (Left:Original WF, Middle:Wave after FFT, RIGHT: Wave after inverse FFT)

Table. 4. Comparison between original, Inverse FFT

	original	Inverse-FFT
count	1000	1000
mean	0.42	0.42
std	0.24	0.24
min	0.01	0.01
0.25	0.21	0.21
0.5	0.42	0.42
0.75	0.62	0.62
max	0.96	0.96

In Fig. 16, you can see WF and TEMP with FFT respectively. The noise of each dataset can be reduced.

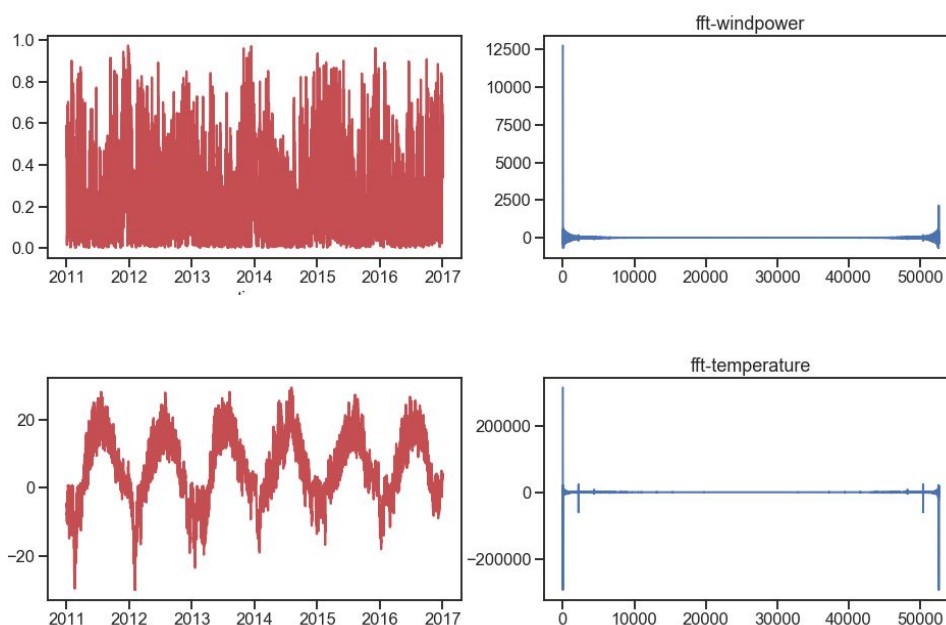


Fig 16. Temperature with FFT(Left: Original temperature data, Right: Temperature data after FFT)

2. Prediction structure

This chapter gives an overview of our Prediction structure. We explain the process of data analysis from the input data. First, we prepared the WF · TEMP time series data and Ramp event data as the input. These input datas are shaped under the following three conditions.

1. Original Data
2. Wavelet transformation filter (WT)
3. Fast Fourier transformation filter (FFT)

After this, we conducted the simulation using software called Jupyter notebook. Based on the simulation results, we could get the plots compared real and predicted value, the calculation of evaluation indices, save the data and model as excel files.

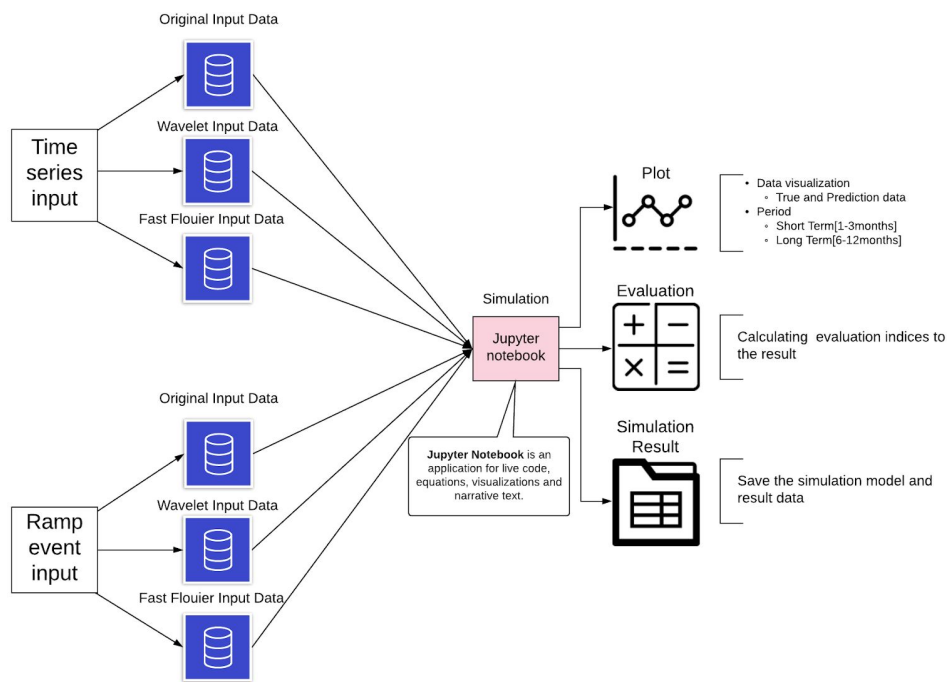


Fig. 17. Overview of the Prediction

2.0 Model description

In this study, we have tested 5 machine learning models:

- Deep Feedforward (DFF)
- Long-short Term Memory (LSTM)
- Attention mechanism (Attention)
- Deep Convolutional Network (DCN)
- indRNN

These models are implemented for the prediction of time series WF, TEMP data and Ramp behavior parameters. We referred the image from Fjodor van Veen of Asimov institute [13] to select these models . This section is begins with description of the each model architectures.

2.1 Deep Feedforward (DFF)

Deep Feedforward is the most popular neural network [14]. This neural network consists of neurons, that are ordered into layers which contain input , hidden and output. Each neuron in one layer is connected to every neuron on the next layer. Therefore, information is constantly feed forward from next.

The goal of a feedford network is to approximate some function $f(x)$ whose x is an input . A feedforward network defines a mapping the following equation.

$$y = f(x; \theta)$$

The feedforward network learns the input x and the value of the parameters θ that result in the best function approximation. In our case, we have placed the several feed-forward layers, therefore this can be called Deep Feedforward (DFF) network.

2.1.1 DFF Prediction structure

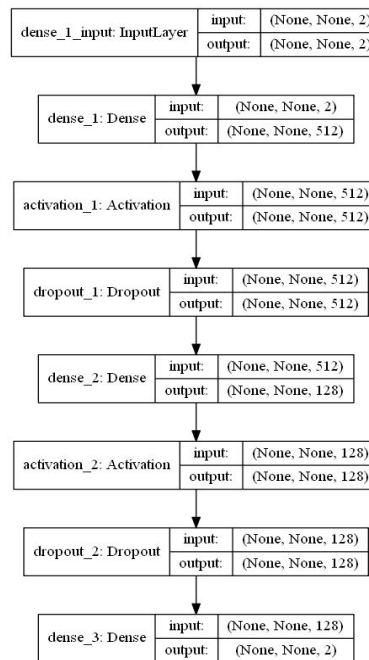


Fig. 18. DFF model structure

As Fig. 18 shows, firstly Dense layer can receive the input data and generate an output of 512 dimensions. Next, there is a activation layer which calculates a “weighted sum” of its input, then a Dropout layer which is a simple way to prevent neural networks from overfitting [5]. This process repeated in the next process as Fig. 10 is shown. Lastly, a dense layer adapts the multi outputs which have two dimensions to fit the output shape.

2.2 Long Short Term Memory Networks

In this section, firstly we will explain about the Recurrent Neural Network (RNN). RNN is a network with loops in that, allowing information to persist. In the diagram, a loop of neural network, A, looks at some input x_t and outputs a value h_t . A loop allows information to be passed from one step of the network to the next. One of the appeals of RNN is it can convey the previous information to the present task, but it also has the problem which is called long-term dependency problem. The problem is as the gap between the relevant information and the point where it is needed to grows, RNN become unable to learn to connect the incorrect information.

At this point, LSTM is explicitly designed to avoid the long-term dependency problem with four interacting layers [15].

2.2.1 LSTM Prediction structure

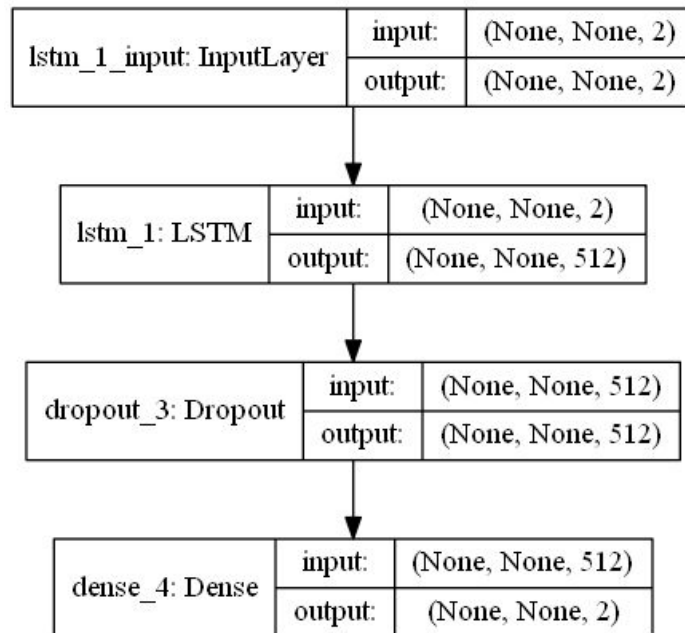


Fig 19. LSTM model structure

As Fig. 19 shows, LSTM layers can receive the input data and generate an output of 512 dimensions. Next, there is a Dropout layer which is a simple way to prevent neural networks from overfitting [5]. Lastly, a dense layer adapts the multi outputs which have two dimensions.

2.3 Attention Mechanism

Regarding Attention based models there is still very little in literature in terms of wind power forecast applications. However, a good introduction to the Attention based models specific properties can be found in [16]. Attention mechanisms are also introduced in [17] as deep learning methods able to improve the accuracy on many tasks, particularly within natural language processing and image recognition. The importance of attention mechanisms is also discussed in [18] as a promising method to better address the real world complexity. In addition, an automatic forecasting of time series data with Multifactor Neural Attention can be found in [19]. The novel methodology achieves a 23.9% improvement of forecasts in comparison to other neural networks proposed for time series forecasting to date.

2.3.1 Attention Mechanism Prediction structure

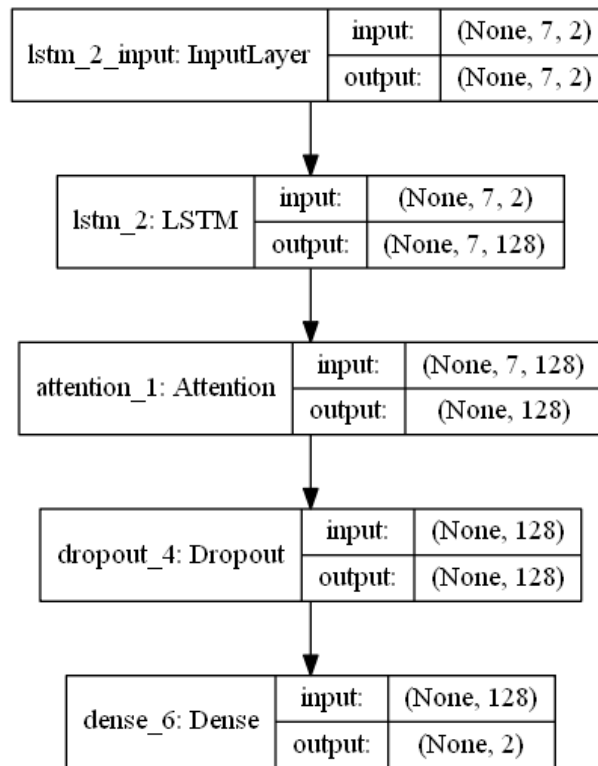


Fig. 20. Attention Mechanism model structure

As Fig. 20 shows, firstly LSTM layers can receive the input data and generate an output of 128 dimensions. Next, there is a Attention layer which can deal with the weight vector. After this layer, as the same to other models, there are Dropout and a dense layer adapts the multi outputs which have two dimensions.

2.4 Deep Convolutional Network(DCN)

Deep convolutional Network [20] is a Deep Learning algorithm which can take in an input and be able to differentiate one from the other. The processing required in a ConvNet. While in primitive methods filter are hand-engineerineered, with enough training, ConvNets have the ability to learn these filtered/characteristics.

2.4.1 DCN Prediction structure

Fig. 21 shows an used structure of DCN. There are two 1 dimensional Convolutional neural network (Con1D) from Input to the first layer. Con1d can derive the features from shorter of the overall dataset and where the location of the features within the segment is not of high relevance. Then, followed by the form of Output, it build the each layers that shapes the output data. The Pooling layer is a layer that performs element compression.

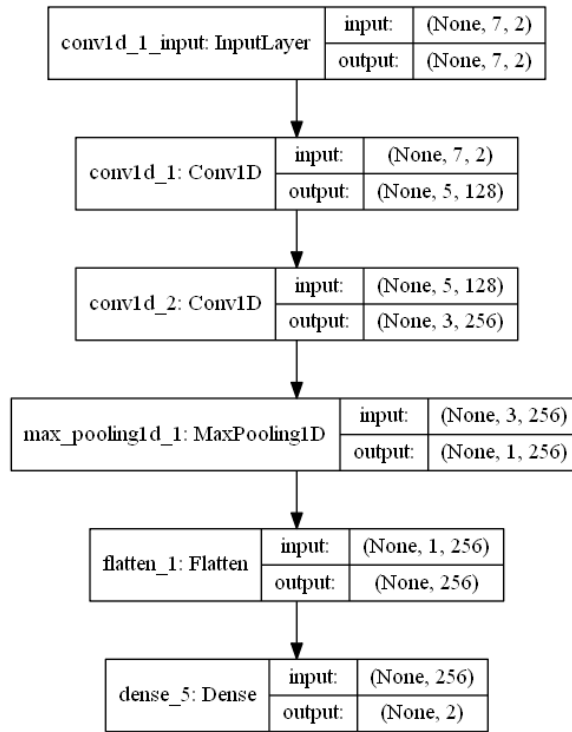


Fig 21. DCN model structure

2.5 indRNN Mechanism

In this section, we explain about an independently recurrent neural network (IndRNN). It can be described as:

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{u} \odot \mathbf{h}_{t-1} + \mathbf{b})$$

where recurrent weight \mathbf{u} is a vector and \odot represents Hadamard product. \mathbf{u} and \mathbf{b} are the weights for the current input and the recurrent input. \mathbf{x}_t and \mathbf{h} are the input and hidden state at time step t . σ is an element-wise activation function of the neurons. Each neuron in one layer is independent from others and connection between neurons can be achieved by stacking two or more layers of IndRNNs. In [21], The result of indRNN was better than that of LSTM as shown fig. 22. Therefore, we have chosen this model to test out datasets.

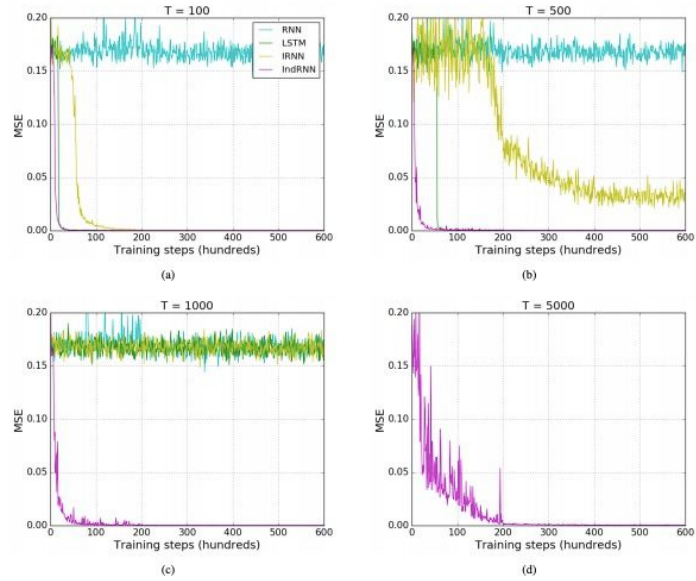


Fig. 22. result of comparison between IndRNN and LSTM [21]

2.5.1 indRNN Mechanism Prediction structure

Fig. 23 shows an used structure of indRNN. We replaced LSTM with IndRNN layer with the same dimensional number.

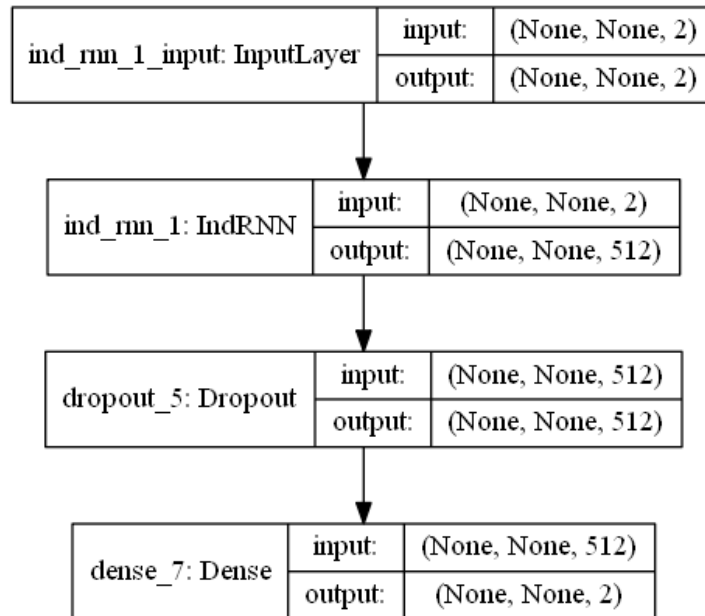


Fig 23. indRNN Mechanism model structure

2.6 Prerequisites for prediction

2.6.1 Initial parameter for each model

The parameters of all the models are given in Table 7. Batch indicated the number of training samples used in one iteration. For example, we can divide the dataset of 1680 into batches of 168 then it will take 10 iterations to finish 1 epoch which the number of passing through an entire dataset.

About activation function, we chosen ReLU function that can be defined as $y = \max(0,x)$ mathematically. In Fig. 24, you can see the sample plot of Tanh, Sigmoid and ReLU. ReLU is linear for all positive values, and zero for all negative values. This means ReLU function can take less time to train or run. With the limited compute resource, this point is important. There is a linear character which is the slope doesn't saturate, when x gets large. It doesn't have the vanishing gradient problem which can happen in other activation functions like sigmoid or tanh.

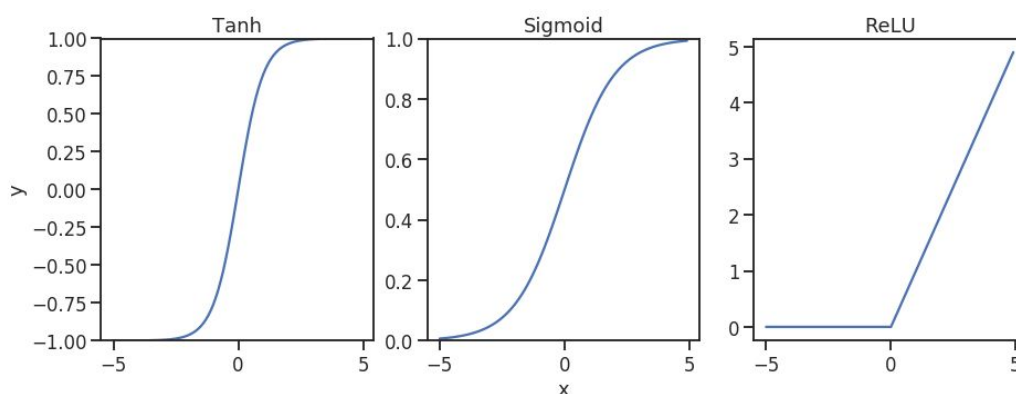


Fig. 24. Sample Plot of Tanh, Sigmoid and ReLU

We used Adaptive moment estimation(Adam) as a optimizer that is developed by Diederik P. Kingma in 2015[22]. This method is an improvement on AdaGrad[23], RMSprop and AdaDelta[24]. ADAM optimizer is an extension to stochastic gradient descent. ADAM is based on adaptive moment estimation. ADAM has two parts - Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). ADAM computes the individual adaptive learning rates for different parameters from estimates of first and second moments of the gradient.

Learning rate is a hyper-parameter that controls how much we are tuning the weights of our network with the loss gradient. The lower the value, the slower we pass through the downward slope. In Fig. 25, It shows the comparison of large and small learning rate. If learning rate is too small, the gradient of the loss can be slow. On the other hand, the learning rate is too large, it may be hard to converge. In our case, we have chosen $1.00E-03$ from [25].

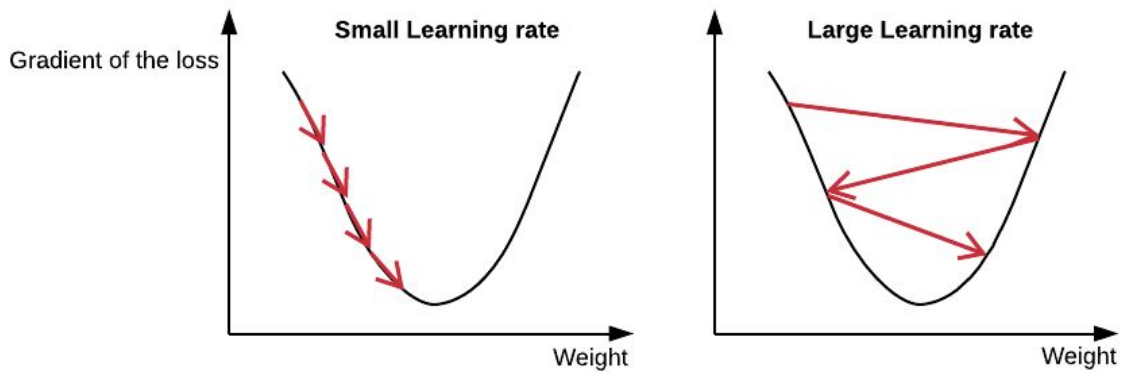


Fig. 25. Comparison of learning rate

Table 7. Initial parameter for prediction

Model	Batch Number	Epoch	Activation Function	Optimizer	Learning Rate
DFF	168	200	ReLU	Adam	1.00E-03
LSTM	168	200			
Attention	168	200			
DCN	168	200			
indRNN	168	200			

2.2.2 Machine spec

Table. 8 shows the specifications of the computer used this research.

Table 8. Prediction Machine Spec

Name	Detail
Windows edition	Windows 10 Pro
Processor	Intel(R) Core(TM) i7-3770 CPU@3.4GHz
Installed memory(RAM)	16GB
System type	64-bit operating system

2.7 Method for prediction evaluation

In this section, we will show Prediction results and considerations for each model of Wind power capacity factor and Temperature. Each result included short-term [1-3 months] and long-term [6-12 months] forecasts, and evaluation indices are shown in Table 9.

Table 9. Evaluation indices and the predicted period

Evaluation indices	Average gap
	Mean Square Error (MSE)
	Root Mean Square Error(RMSE)
	Root Mean Squared Log Error (RMSLE)
Prediction Period	1 month
	3 months
	6 months
	12 months

We adopted five evaluation indices to evaluate the prediction result: The average gap between real data and prediction data, the mean square error(MSE), Root Mean Square Error(RMSE) and the Root Mean Squared Logarithmic Error(RMSLE). The evaluation indices are described by Equation(1)-(3). MSE incorporates both the variance and the bias of the predictor. RMSE is the square root of MSE. In case of unbiased estimator, RMSE is the square root of variance, which is actually Standard Deviation. RMSLE takes the log of the predictions and actual values. RMSLE is particularly used to avoid penalizing huge differences in the predicted and the actual values when both predicted and true values are huge numbers. If both predicted and actual values are small then RMSE and RMSLE is same. If either the predicted or the actual values are big then $RMSE > RMSLE$. If both the predicted and the actual values are big the $RMSE > RMSLE$, thus RMSLE becomes negligible.

Average Gap

$$\frac{1}{n} \sum_{t=1}^n |y_t - y'_t| \quad (1)$$

MSE

$$\frac{1}{n} \sum_{t=1}^n (y_t - y'_t)^2 \quad (2)$$

RMSE

$$\sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - y'_t)^2} \quad (3)$$

3. Prediction Result

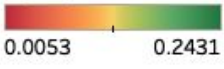
3.1 Result on Temperature and Wind power capacity factor prediction

In this section, we report on the result on Temperature and Wind power prediction which we compared the following 5 models: Attention, DCN, DFF, LSTM and indRNN. Each result was evaluated by the following viewpoints.

1. Comparison of model performance
 - Which model could get the best result for TEMP and WF prediction respectively
2. Short VS Long term prediction
 - Investigate the prediction period dependency

1. Model Comparison

Table. 10. Model Comparison of prediction 1

Avg Gap

 0.0053 0.2431

Data	Type	Attention	DCN	DFF	indRNN	LSTM
TEMP	Original	0.0334	0.0870	0.0167	0.0208	0.0238
	WT	0.0187	0.0818	0.0520	0.0442	0.0410
	FFT	0.0124	0.0329	0.0346	0.0498	0.1305
WF	Original	0.0766	0.2022	0.0164	0.0117	0.0331
	WT	0.0467	0.2431	0.1157	0.1021	0.0973
	FFT	0.0053	0.0775	0.0780	0.0676	0.0857

		Attention	DCN	DFF	indRNN	LSTM
TEMP	Original	1.00	2.60	0.50	0.62	0.71
	WT	1.00	4.38	2.79	2.37	2.20
	FFT	1.00	2.66	2.80	4.03	10.57
WF	Original	1.00	2.64	0.21	0.15	0.43
	WT	1.00	5.20	2.48	2.19	2.08
	FFT	1.00	14.76	14.85	12.87	16.31

We evaluated the performance of each model by averaging the period of 1, 3, 6 and 12 months. The results are shown in Table 10. Table. 10 consist of the average gap (Avg Gap) heat map and the table which compares the Attention model with other models. For example, The value of DCN with Original in TEMP is 2.6, which means the Avg Gap is 2.6 times than Attention model. We set the color of red for the number that exceeds the Avg Gap of Attention model.

In TEMP, it was the attention model's FFT filter that had the least gap with the true value. Avg gap is 0.0124, and there is a difference of 10% or more compared to the FFT of LSTM that had the most difference with true value. Also, comparing the computing time, the best result attention took a long time of about 1800 seconds, and it was 4.6 seconds of DFF that ended in the shortest. It can be seen that the DFF results are not as bad as 0.0167 in the Original data, and that TEMP produced reasonable results in the shortest time.

On the other hand, even with WF, the Attention model FFT filter gave the best result of 0.0053 gaps. For Attention and DCN, the result with the FFT filter was the best, but for DFF, IndRNN, LSTM, the unfiltered Original was the best. The DCN -WT has the worst result with an Avg Gap of more than 0.2. Also, we note that LSTM has the lowest RMSLE. Avg gap is inferior to Attention model, but we could find that there are few significant outliers.

2. Short VS Long term prediction

Table. 11. Comparison of prediction period

Data	Type	Attention				DCN				DFF				indRNN				LSTM			
		1	3	6	12	1	3	6	12	1	3	6	12	1	3	6	12	1	3	6	12
TEMP	Original	0.022	0.027	0.044	0.040	0.095	0.106	0.083	0.065	0.015	0.012	0.018	0.021	0.023	0.017	0.021	0.022	0.027	0.020	0.025	0.024
	WT	0.011	0.015	0.023	0.025	0.106	0.072	0.077	0.073	0.032	0.046	0.067	0.064	0.026	0.038	0.058	0.055	0.027	0.035	0.052	0.050
	FFT	0.008	0.005	0.017	0.020	0.018	0.027	0.045	0.042	0.023	0.028	0.045	0.042	0.069	0.044	0.046	0.041	0.187	0.149	0.099	0.087
WF	Original	0.072	0.075	0.076	0.083	0.192	0.217	0.192	0.207	0.015	0.013	0.019	0.019	0.011	0.009	0.013	0.014	0.034	0.033	0.033	0.034
	WT	0.039	0.043	0.051	0.054	0.300	0.249	0.212	0.210	0.115	0.116	0.110	0.122	0.100	0.101	0.099	0.108	0.098	0.096	0.093	0.102
	FFT	0.007	0.005	0.005	0.004	0.075	0.077	0.076	0.083	0.076	0.077	0.076	0.083	0.067	0.067	0.066	0.071	0.098	0.090	0.077	0.079
Data	Type	Attention				DCN				DFF				indRNN				LSTM			
TEMP	Original	1.00	1.24	1.98	1.83	1.00	1.11	0.87	0.68	1.00	0.81	1.21	1.39	1.00	0.75	0.91	0.98	1.00	0.75	0.92	0.87
	WT	1.00	1.42	2.18	2.37	1.00	0.68	0.73	0.69	1.00	1.43	2.09	2.02	1.00	1.49	2.25	2.14	1.00	1.31	1.93	1.85
	FFT	1.00	0.63	2.08	2.48	1.00	1.48	2.45	2.27	1.00	1.25	2.02	1.88	1.00	0.63	0.67	0.59	1.00	0.80	0.53	0.47
WF	Original	1.00	1.05	1.06	1.16	1.00	1.13	1.00	1.08	1.00	0.90	1.24	1.26	1.00	0.88	1.23	1.33	1.00	0.97	0.97	1.01
	WT	1.00	1.12	1.31	1.39	1.00	0.83	0.71	0.70	1.00	1.00	0.95	1.05	1.00	1.01	0.99	1.07	1.00	0.99	0.95	1.04
	FFT	1.00	0.76	0.72	0.66	1.00	1.03	1.01	1.10	1.00	1.02	1.00	1.09	1.00	0.99	0.97	1.05	1.00	0.91	0.78	0.80

Next, we compared the result of short (1-3 month) and long period (6-12 months) prediction. We got the result shown in the table. 11. In the Attention model, the performance of 1-month prediction was the best for all condition. We could discover that there is a gap of 50% when comparing Avg Gap in FFT of the Attention. IndRNN and LSTM have shown a slight difference in

between the short and long prediction period of Original and WT, but only long-term prediction with FFT had better results than the one from short-term.

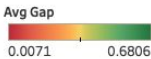
In WF, the result of Attention's FFT filter was remarkable. Although the results of the original and the WT were better in the short-term prediction, the results of the FFT as the longer prediction period was improved. In indRNN and LSTM, the trend is different from that observed at TEMP, the difference in Avg gap with time was not so massive.

3.2 Result on RBA parameters prediction

We report on the result on RBA parameters prediction which we have tested with 5 models: Attention, DCN, DFF, LSTM and indRNN. Each result was evaluated from the same viewpoints with prediction 1.

1. Model Comparison

Table. 12. Model Comparison of prediction 2



Data	Type	Attention	DCN	DFF	indRNN	LSTM
Δt	Original	0.0071	0.0228	0.0272	0.0182	0.0702
	WT	0.1043	0.1119	0.1002	0.1004	0.0877
	FFT	0.0527	0.0534	0.1473	0.3671	0.1345
Δw_s	Original	0.0145	0.0156	0.0210	0.0418	0.0643
	WT	0.1066	0.1145	0.1063	0.1056	0.0957
	FFT	0.0670	0.0542	0.1866	0.1160	0.1818
t	Original	0.0215	0.0114	0.0535	0.0267	0.0205
	WT	0.0367	0.0217	0.0416	0.0128	0.0160
	FFT	0.6790	0.6806	0.6531	0.3644	0.6624
$w_s(t)$	Original	0.0191	0.0215	0.0289	0.2476	0.0635
	WT	0.1583	0.1689	0.1696	0.1667	0.1394
	FFT	0.0457	0.0368	0.1896	0.1711	0.2321
$\theta(\Delta w_s)$	Original	0.0131	0.0219	0.0326	0.0350	0.0821
	WT	0.1303	0.1365	0.1228	0.1251	0.1135
	FFT	0.0498	0.0555	0.1714	0.1312	0.1408
		Attention	DCN	DFF	indRNN	LSTM
Δt	Original	1.00	3.20	3.81	2.56	9.86
	WT	1.00	1.07	0.96	0.96	0.84
	FFT	1.00	1.01	2.79	6.97	2.55
Δw_s	Original	1.00	1.08	1.44	2.88	4.44
	WT	1.00	1.07	1.00	0.99	0.90
	FFT	1.00	0.81	2.78	1.73	2.71
t	Original	1.00	0.53	2.48	1.24	0.95
	WT	1.00	0.59	1.13	0.35	0.44
	FFT	1.00	1.00	0.96	0.54	0.98
$w_s(t)$	Original	1.00	1.13	1.51	12.96	3.33
	WT	1.00	1.07	1.07	1.05	0.88
	FFT	1.00	0.80	4.15	3.74	5.08
$\theta(\Delta w_s)$	Original	1.00	1.67	2.49	2.67	6.28
	WT	1.00	1.05	0.94	0.96	0.87
	FFT	1.00	1.11	3.44	2.64	2.83

The result of model comparison in prediction 2 is shown in Fig. 12. In Δt , The best result was the original filtered Attention model. In DFF · indRNN · LSTM, FFT is the worst, but in Attention · DCN, Wavelet is not good. Comparison of computational time shows that it is shorter than Prediction 1 as a whole. This is because the number of data of the extracted RBA parameters is small. Looking at the times of each model, it can be read that the shortest is the DFF original data and the longest is the LSTM WT.

The same tendency as Δt was observed also for Δw_s . The Attention model results using the Original data are the best.

At t , it can be mentioned that the result of FFT is significantly worse. Also, until now the Attention model has been giving better results, but at t it can be seen that DCN is the best model.

In $w_s(t)$, the same tendency as $\Delta t \cdot \Delta w_s$ was observed. The difference is that in the indRNN, the result of the Original data was bad, and the result of the WT was good. About $\theta(\Delta w_s)$, The DCN had larger gaps in all data filters compared to the Attention model. However, WT filters in other models have improved about 5-15% over Attention models.

2. Short vs Long term prediction

The result of short-long term comparison in prediction 2 is shown in Table. 13.

Table. 13. Comparison of prediction period

Data	Type	Attention				DCN				DFF				indRNN				LSTM			
Δt	Original	0.0069	0.0073	0.0069	0.0074	0.0170	0.0195	0.0235	0.0311	0.0299	0.0278	0.0247	0.0263	0.0197	0.0183	0.0169	0.0180	0.0768	0.0715	0.0648	0.0678
	WT	0.1134	0.1079	0.0968	0.0990	0.1203	0.1165	0.1041	0.1065	0.1065	0.1032	0.0932	0.0979	0.1058	0.1034	0.0931	0.0991	0.0943	0.0907	0.0810	0.0846
	FFT	0.0230	0.0323	0.0542	0.1013	0.0248	0.0338	0.0536	0.1013	0.1340	0.1427	0.1414	0.1709	0.3456	0.3597	0.3725	0.3907	0.1085	0.1088	0.1175	0.2030
Δw_s	Original	0.0140	0.0140	0.0139	0.0161	0.0142	0.0143	0.0156	0.0183	0.0223	0.0185	0.0184	0.0246	0.0618	0.0386	0.0320	0.0347	0.0747	0.0618	0.0585	0.0623
	WT	0.1050	0.1079	0.1053	0.1083	0.1111	0.1145	0.1139	0.1185	0.1060	0.1075	0.1037	0.1080	0.1014	0.1079	0.1045	0.1086	0.0941	0.0967	0.0930	0.0989
	FFT	0.0651	0.0655	0.0693	0.0682	0.0554	0.0526	0.0545	0.0542	0.1780	0.1911	0.1876	0.1898	0.1089	0.1187	0.1157	0.1205	0.1916	0.1854	0.1771	0.1731
t	Original	0.0198	0.0203	0.0216	0.0244	0.0108	0.0105	0.0105	0.0136	0.0481	0.0516	0.0551	0.0591	0.0298	0.0283	0.0237	0.0249	0.0331	0.0186	0.0156	0.0148
	WT	0.0345	0.0348	0.0366	0.0407	0.0209	0.0200	0.0200	0.0258	0.0375	0.0393	0.0426	0.0469	0.0173	0.0111	0.0103	0.0125	0.0275	0.0144	0.0113	0.0107
	FFT	0.6708	0.6818	0.6871	0.6763	0.6737	0.6829	0.6889	0.6769	0.6475	0.6564	0.6593	0.6493	0.3446	0.3555	0.3692	0.3881	0.6803	0.6899	0.6763	0.6029
$w_s(t)$	Original	0.0177	0.0190	0.0195	0.0202	0.0148	0.0185	0.0231	0.0296	0.0319	0.0267	0.0267	0.0303	0.2805	0.2456	0.2186	0.2456	0.0709	0.0609	0.0587	0.0636
	WT	0.1592	0.1597	0.1525	0.1618	0.1731	0.1685	0.1611	0.1727	0.1778	0.1619	0.1596	0.1789	0.1701	0.1576	0.1599	0.1791	0.1486	0.1363	0.1310	0.1417
	FFT	0.0459	0.0458	0.0444	0.0467	0.0365	0.0370	0.0365	0.0371	0.2130	0.1818	0.1733	0.1903	0.1883	0.1644	0.1586	0.1729	0.2452	0.2289	0.2226	0.2316
$\theta(\Delta w_s)$	Original	0.0110	0.0119	0.0131	0.0163	0.0192	0.0217	0.0234	0.0232	0.0301	0.0319	0.0336	0.0348	0.0317	0.0332	0.0369	0.0380	0.0791	0.0795	0.0850	0.0849
	WT	0.1186	0.1295	0.1377	0.1355	0.1268	0.1324	0.1440	0.1429	0.1144	0.1235	0.1269	0.1262	0.1133	0.1254	0.1301	0.1314	0.1035	0.1137	0.1174	0.1195
	FFT	0.0506	0.0507	0.0476	0.0502	0.0546	0.0571	0.0551	0.0551	0.1625	0.1718	0.1737	0.1776	0.1242	0.1307	0.1337	0.1361	0.1445	0.1404	0.1400	0.1382

Data	Type	Attention				DCN				DFF				indRNN				LSTM			
		1	3	6	12	1	3	6	12	1	3	6	12	1	3	6	12	1	3	6	12
Δt	Original	1.00	1.06	1.00	1.07	1.00	1.15	1.38	1.83	1.00	0.93	0.83	0.88	1.00	0.93	0.86	0.91	1.00	0.93	0.84	0.88
	WT	1.00	0.95	0.85	0.87	1.00	0.97	0.87	0.89	1.00	0.97	0.88	0.92	1.00	0.98	0.88	0.94	1.00	0.96	0.86	0.90
	FFT	1.00	1.40	2.36	4.40	1.00	1.36	2.16	4.08	1.00	1.06	1.06	1.28	1.00	1.04	1.08	1.13	1.00	1.00	1.08	1.87
Δw_s	Original	1.00	1.00	0.99	1.15	1.00	1.01	1.10	1.29	1.00	0.83	0.83	1.10	1.00	0.62	0.52	0.56	1.00	0.83	0.78	0.83
	WT	1.00	1.03	1.00	1.03	1.00	1.03	1.03	1.07	1.00	1.01	0.98	1.02	1.00	1.06	1.03	1.07	1.00	1.03	0.99	1.05
	FFT	1.00	1.01	1.06	1.05	1.00	0.95	0.98	0.98	1.00	1.07	1.05	1.07	1.00	1.09	1.06	1.11	1.00	0.97	0.92	0.90
t	Original	1.00	1.03	1.09	1.23	1.00	0.97	0.97	1.26	1.00	1.07	1.15	1.23	1.00	0.95	0.80	0.84	1.00	0.56	0.47	0.45
	WT	1.00	1.01	1.06	1.18	1.00	0.96	0.96	1.23	1.00	1.05	1.14	1.25	1.00	0.64	0.60	0.72	1.00	0.52	0.41	0.39
	FFT	1.00	1.02	1.02	1.01	1.00	1.01	1.02	1.00	1.00	1.01	1.02	1.00	1.00	1.03	1.07	1.13	1.00	1.01	0.99	0.89
$w_s(t)$	Original	1.00	1.07	1.10	1.14	1.00	1.25	1.56	2.00	1.00	0.84	0.84	0.95	1.00	0.88	0.78	0.88	1.00	0.86	0.83	0.90
	WT	1.00	1.00	0.96	1.02	1.00	0.97	0.93	1.00	1.00	0.91	0.90	1.01	1.00	0.93	0.94	1.05	1.00	0.92	0.88	0.95
	FFT	1.00	1.00	0.97	1.02	1.00	1.01	1.00	1.02	1.00	0.85	0.81	0.89	1.00	0.87	0.84	0.92	1.00	0.93	0.91	0.94
$\theta(\Delta w_s)$	Original	1.00	1.08	1.19	1.48	1.00	1.13	1.22	1.21	1.00	1.06	1.12	1.16	1.00	1.05	1.16	1.20	1.00	1.01	1.07	1.07
	WT	1.00	1.09	1.16	1.14	1.00	1.04	1.14	1.13	1.00	1.08	1.11	1.10	1.00	1.11	1.15	1.16	1.00	1.10	1.13	1.15
	FFT	1.00	1.00	0.94	0.99	1.00	1.05	1.01	1.01	1.00	1.06	1.07	1.09	1.00	1.05	1.08	1.10	1.00	0.97	0.97	0.96

In Δt , Attention model is the best at the point of the gap. Compared the short and long period in the model, short term prediction was much better in original data. In other data types, the longer period prediction was better. The second best model was the IndRNN. In this model, prediction period did not make a difference in results for any data type. At Δw_s , the DCN results were improved. The values were almost the same as the Original of Attention, and the Short period was better. Looking at indRNN and LSTM, in Original, the long period prediction had a smaller gap. At t , it is clear that the result by the FFT filter is not good in the short and long period prediction. As for the tendency, similar to $\Delta t \cdot \Delta w_s \cdot t$ described above, the results of Short period prediction were better for Attention · DCN, and the results for Long period prediction were better for DFF · indRNN · LSTM. In $w_s(t)$, the same tendency as $\Delta t \cdot \Delta w_s$ was observed. The difference is that in the indRNN, the result of the Original data was bad, and the result of the WT was good. In $\theta(\Delta w_s)$, the same tendency as $\Delta t \cdot \Delta w_s$ was observed. The difference is that in the indRNN, the result of the Original data was bad, and the result of the WT was good.

4. Hyperparameter tuning

Hyperparameter tuning (HT) is a crucial step in machine learning practice. This process is often carried out by hand. In our cases, We have set the initial parameters manually as shown in Table. 7. At this point, there is a hypothesis that we could improve the result based on the hyperparameter tuning. In this section, we demonstrate how the prediction results improve after the hyperparameter tuning. Looking back to the result section, Attention model performs the best among the tested models. Therefore, we have tried the tuning process with the model and compare the result between before and after the tuning.

The methods which we used is Tree-structured Parzen Estimator Approach (TPE) that is one kind of Bayesian Optimization Algorithm [26]. We adopted an automatic hyperparameter optimization software framework: “Optuna”. It features an imperative, define-by-run style user API. The parameters which we have adjusted are batch size, epoch and learning rate. Table. 14 presents the range of the values which we have tested for HT.

Table. 14. Tested range of the hyperparameters

	Tested Range
Number of units	5 - 300
Learning rate	1e-5 - 1e-4
Epoch	50 - 200
Batch_size	32 - 128

We have experimented the HT based on the parameters in Table. 14. In table. 15, we have shown the comparison of the parameters before and after HT.

Table. 15. The comparison of the parameters

	Before	Prediction 1 (After HT)			Prediction 2 (After HT)		
		Original	WT	FFT	Original	WT	FFT
Number of units	128	70	297	244	189	167	232
Learning rate	1.00E-03	2.65E-05	1.16E-05	4.19E-05	5.41E-05	3.07E-05	6.90E-05
Epoch	200	56	106	70	149	107	135
Batch_size	128	52	105	91	64	49	109

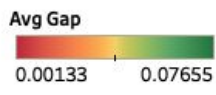
4.1 Prediction 1 hyperparameter comparison result

We have performed Prediction 1 and Prediction 2 again with Attention model using the parameters obtained after executing Hyperparameter tuning. The results are shown in Table 20 and Table 21.

1. Comparison of model with and without HT

If you look at Table 20, you can see the comparison of the results by Attention model without HT and Attention model with HT. There is no difference in tendency due to data filters (Original, WT, FFT). However, if you look at the table below, there is an improvement in the value of the Attention model with HT under almost all conditions. (It is considered as red when Attention is 1 and it is 1 if it is more than 1). This indicates that HT has performed parameter search and was optimized. Although there was no improvement in the values in TEMP WT, it is considered that this was caused by selecting appropriate parameters from the beginning.

Table. 20. Comparison of before/after HT in prediction 1(Attention: Without HT, Attention-h: With HT)



Data	Type	Attention	Attention-h
TEMP	Original	0.03343	0.03310
	WT	0.01865	0.01945
	FFT	0.01235	0.00605
WF	Original	0.07655	0.07628
	WT	0.04670	0.04545
	FFT	0.00525	0.00133

		Attention	Attention-h
TEMP	Original	1.00	0.99
	WT	1.00	1.04
	FFT	1.00	0.49
WF	Original	1.00	1.00
	WT	1.00	0.97
	FFT	1.00	0.25

2. Short vs Long term period

Next, the comparison in the short-long term is performed. Looking at Tables. 21, there was no difference in comparison between Attention and Attention-h as the Short term was superior. As

for Attention-h, except for the FFT of TEMP, the result of Short term prediction was better. (In each model, the result of 1 month prediction was taken as 1 and compared.)

Table. 21. Period comparison of before/after HT in prediction 1



4.2 Prediction 2 hyperparameter comparison result

This chapter compares Prediction 2 with and without HT.

1. Comparison of model with and without HT

Consider the comparison of Attention and Attention-h in Prediction 2. There was no difference in trend depending on the conditions. Under all conditions, the post HT parameters improved the results. There was also a result that exceeded but a difference of less than 5%, in which case it is considered that the parameters used without HT matched the data set.

Table. 22. Comparison of before/after HT in prediction 2

Data	Type	Attention	Attention-h
Δt	Original	0.0071	0.0048
	WT	0.1043	0.1069
	FFT	0.0527	0.0387
Δw_s	Original	0.0145	0.0088
	WT	0.1066	0.1123
	FFT	0.0670	0.0623
t	Original	0.0215	0.0153
	WT	0.0367	0.0056
	FFT	0.6790	0.6940
$w_s(t)$	Original	0.0191	0.0053
	WT	0.1583	0.1597
	FFT	0.0457	0.0200
$\theta(\Delta w_s)$	Original	0.0131	0.0077
	WT	0.1303	0.1350
	FFT	0.0498	0.0498

		Attention	Attention-h
Δt	Original	1.00	0.68
	WT	1.00	1.03
	FFT	1.00	0.73
Δw_s	Original	1.00	0.60
	WT	1.00	1.05
	FFT	1.00	0.93
t	Original	1.00	0.71
	WT	1.00	0.15
	FFT	1.00	1.02
$w_s(t)$	Original	1.00	0.28
	WT	1.00	1.01
	FFT	1.00	0.44
$\theta(\Delta w_s)$	Original	1.00	0.59
	WT	1.00	1.04
	FFT	1.00	1.00

2. Short vs Long term period

Let's look at the comparison of prediction periods. The trend of the results did not change, and almost all the results showed that the short period was better.

Table. 23. Period comparison of before/after HT in prediction 2

Data	Type	Attention				Attention-h			
		1	3	6	12	1	3	6	12
Δt	Original	0.0069	0.0073	0.0069	0.0074	0.0051	0.0046	0.0040	0.0056
	WT	0.1134	0.1079	0.0968	0.0990	0.1169	0.1106	0.0990	0.1012
	FFT	0.0230	0.0323	0.0542	0.1013	0.0101	0.0198	0.0412	0.0837
Δw_s	Original	0.0140	0.0140	0.0139	0.0161	0.0087	0.0086	0.0089	0.0088
	WT	0.1050	0.1079	0.1053	0.1083	0.1097	0.1137	0.1114	0.1142
	FFT	0.0651	0.0655	0.0693	0.0682	0.0608	0.0610	0.0643	0.0630
t	Original	0.0198	0.0203	0.0216	0.0244	0.0123	0.0136	0.0159	0.0192
	WT	0.0345	0.0348	0.0366	0.0407	0.0041	0.0043	0.0054	0.0087
	FFT	0.6708	0.6818	0.6871	0.6763	0.6860	0.6952	0.7005	0.6941
$w_s(t)$	Original	0.0177	0.0190	0.0195	0.0202	0.0055	0.0053	0.0052	0.0053
	WT	0.1592	0.1597	0.1525	0.1618	0.1604	0.1606	0.1533	0.1646
	FFT	0.0459	0.0458	0.0444	0.0467	0.0204	0.0199	0.0185	0.0212
$\theta(\Delta w_s)$	Original	0.0110	0.0119	0.0131	0.0163	0.0067	0.0072	0.0081	0.0086
	WT	0.1186	0.1295	0.1377	0.1355	0.1223	0.1340	0.1428	0.1408
	FFT	0.0506	0.0507	0.0476	0.0502	0.0497	0.0507	0.0484	0.0504
Δt	Original	1.00	1.06	1.00	1.07	1.00	0.90	0.78	1.10
	WT	1.00	0.95	0.85	0.87	1.00	0.95	0.85	0.87
	FFT	1.00	1.40	2.36	4.40	1.00	1.96	4.08	8.29
Δw_s	Original	1.00	1.00	0.99	1.15	1.00	0.99	1.02	1.01
	WT	1.00	1.03	1.00	1.03	1.00	1.04	1.02	1.04
	FFT	1.00	1.01	1.06	1.05	1.00	1.00	1.06	1.04
t	Original	1.00	1.03	1.09	1.23	1.00	1.11	1.29	1.56
	WT	1.00	1.01	1.06	1.18	1.00	1.05	1.32	2.12
	FFT	1.00	1.02	1.02	1.01	1.00	1.01	1.02	1.01
$w_s(t)$	Original	1.00	1.07	1.10	1.14	1.00	0.96	0.95	0.96
	WT	1.00	1.00	0.96	1.02	1.00	1.00	0.96	1.03
	FFT	1.00	1.00	0.97	1.02	1.00	0.98	0.91	1.04
$\theta(\Delta w_s)$	Original	1.00	1.08	1.19	1.48	1.00	1.07	1.21	1.28
	WT	1.00	1.09	1.16	1.14	1.00	1.10	1.17	1.15
	FFT	1.00	1.00	0.94	0.99	1.00	1.02	0.97	1.01

2. Weight allocation and classification of power network

2.0 Motivation

These days the network management is still developing and has more impacts to the present power operation. Since electricity grid is very huge system, it is hard to specify a consistent data set, easy to lost in detail and computationally expensive simulation cost. Therefore, with the reduced model, we are aimed to demonstrate the network analysis in Estonia. The salient features include multi-scale network information, classification of zones and ranking each power line in the network.

2.1 Research Strategy and Methods

Fig. 26 shows our method of the classification. It includes data generation, unsupervised and supervised clustering and ranking. The workflow we adopted is as the following process. First, We have prepared geographical data of Estonia such as latitude and longitude. Then, we used data generation algorithms (Markov-switching autoregressive model and Brownian motion) to replace the IEEE 14 [27] and IEEE 118 [28] bus values. As the result, it is possible to obtain a data set that integrates Estonian geographical and Bus data. Using the created four data sets (IEEE 14 based on Markov, IEEE 14 based on Brownian, IEEE 118 based on Markov and IEEE 118 based on Brownian), we tried to classify Arc into four classes by clustering using Unsupervised model. The simulation with the Supervised model was performed using the label data generated by the Unsupervised model. Lastly, we have calculated Power Transfer Distribution Factor and voltage angles for the buses in each Arc and rank them based on the calculated result.

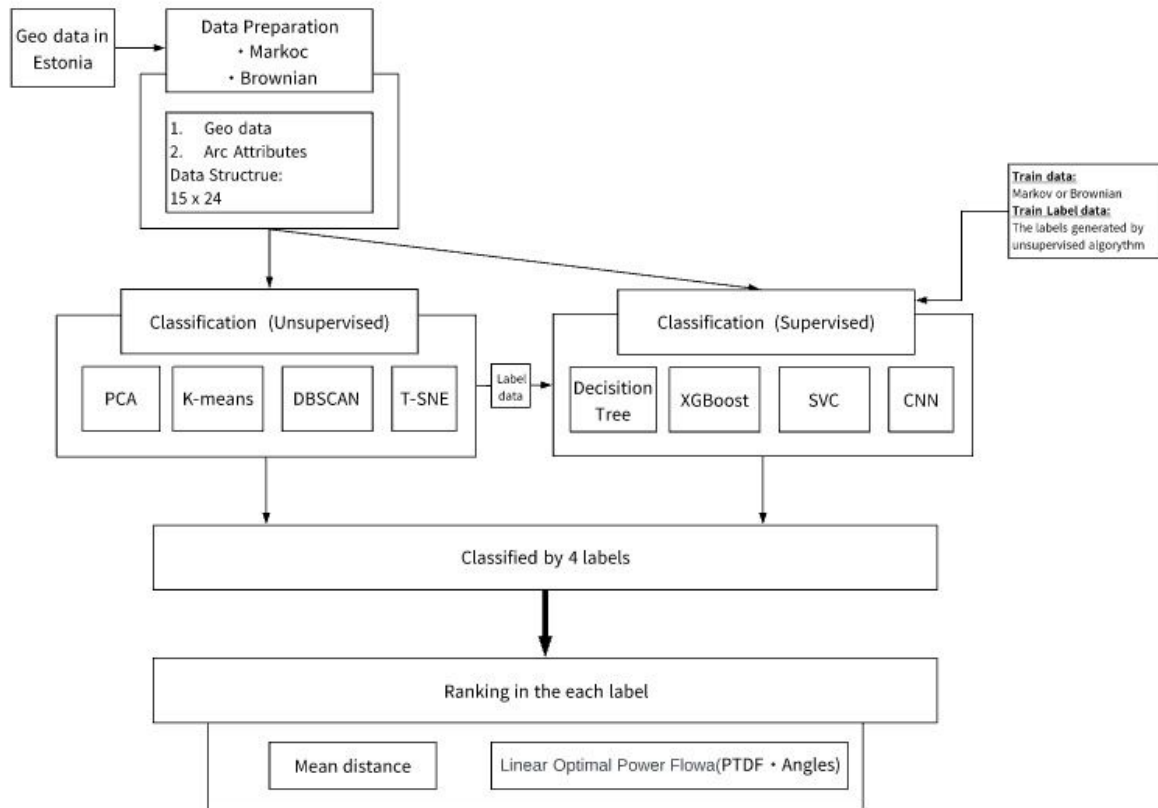


Fig. 26. Classification structure

2.1.1 Data Preparation

The basic data for the network analysis is IEEE 14 bus which represents a portion of the American Electric Power System (in the Midwestern US) as of February, 1962. A much-Xeroxed paper version of the data was kindly provided by Iraj Dabbagchi of AEP and entered in IEEE Common Data Format by Rich Christie at the University of Washington in August 1993. We have used two methods to replace the data for our classification test: Markov-Switching Autoregressive model (MS-AR) and Brownian motion.

Markov-Switching Autoregressive model (MS-AR)

In [29], MS-AR is a non-homogeneous model which is developed by Hamilton in 1989. The model is an autoregressive model of order 4 in which the mean of the process switches between two regimes.

Brownian motion

The brownian motion is the random portion of the equation. Each brownian increment is computed by multiplying a standard random variable from a normal distribution with mean - and standard deviation 1 by the square root of the rime increment [30].

2.2 Model description

In this section, we explain about the used model to classify the each arc. We have tested 4 unsupervised and supervised machine learning model respectively. There is a list of the models in Table. 24.

Table. 24. List of the models

Unsupervised	Supervised
DBSCAN	CNN
K-means	Decision Tree
PCA	Support Vector Machine
T-SNE	XGBoost

Next, We describe the each models used in this classification.

2.2.1 DBSCAN

In [31], they say DBSCAN is rely on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. This method can find core samples of high density and expands clusters from them. There are two parameters to the algorithm, number of minimum samples (the sample number which is in the defined radius) and epsilon (radius from the specific point) which needs to be defined as density. In this sense, Higher minimum samples or lower epsilon indicate higher density necessary to form one cluster.

2.2.2 K-means

K-means algorithm clusters data by trying to separate samples in n groups of equal variance, minimising the within-cluster sum-of -square. This algorithm needs to the number of clusters. In our case, we chosen 4 clusters for that. K-means aims to select centroids that minimise the inertia. [32]

2.2.3 Principal component analysis (PCA)

PCA is used to find the maximum amount of the variance. Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD.

It uses the LAPACK implementation of the full SVD or a randomized truncated SVD by the method of Halko et al. 2009, depending on the shape of the input data and the number of components to extract. [33]

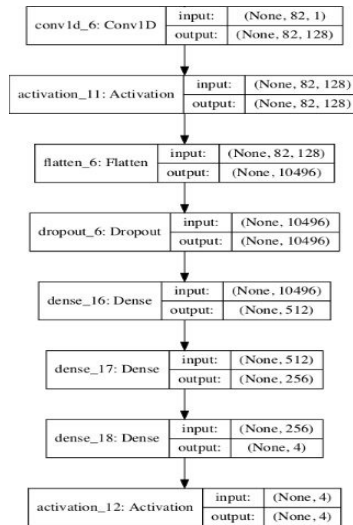
2.2.4 T-SNE

t-SNE can convert similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high dimensional data. [34]

2.2.5 Convolutional Neural Network (CNN)

We also used this algorithm for prediction part. This uses a system which contains a multilayer perceptron that has been designed or reduced processing requirements. The layer of CNN consists of an input layer, output layer and a hidden layers. In 2016, Kiranyaz et al. [35] proposed a novel way for patient-specific monitoring by using one-dimensional Convolutional Neural Network (1D-CNNs). The structure of CNN which we have used is shown in Fig. 27.

Fig. 27. Structure of CNN



2.2.6 Decision Tree (DCT)

Decision Tree is a non-parametric supervised learning method used for classification. This can archive to create a model that predicts the value of a target variable by learning simple decision rules from the data features. This can split the data set into subset based on the attributed value. This process is repeated on each derived subset in a recursive manner called recursive partitioning. [34]

2.2.7 Support Vector Machine (SVC)

Support Vector Machine constructs a hyperplane in multidimensional space to separate different classes. SVM creates optimal hyperplane in an interactive manner, which is used to minimize an error. The ,main idea of SVM is to find a maximum marginal hyperplane that can divide the dataset into classes in the best way. This method calculates the distance between the either nearest points which is the margin. The aim is to choose a hyperplane with the maximum possible margin between the support vectors in the dataset. [35]

2.2.8 XGBoost (XGB)

XGBoost is used for supervised learning problems, where we use the training data (with multiple features) to predict a target label. This is an ensemble Learning combining gradient boosting and random forests. [36]

2.3 Python for Power System Analysis (Pypsa)

Pypsa is a free software toolbox for simulating and optimizing modern electrical power systems over multiple periods. Pypsa contains IEEE14 and IEEE118 buses information. Hence, we have chosen to use this software to construct the network. In addition to that, this software allows user to calculate optimal power flow which are power transfer distribution factor (PTDF) and voltage angles. [37]

2.4 Ranking

Table. 25 shows the list of Ranking Methods. The dataset which is classified by Unsupervised / Supervised Machine learning algorithm can be ranked based on optimal power flow. The selected parameters of Optimal power flow are PTDF and voltage angles. These numbers are normalized from 0 to 1 and visualized ranked by Highly critical / Critical / Secure / Very secure.[38]

Table. 25. Ranking Method

Classification	Ranking
Class 0	Highly critical • Critical • Secure • Very secure
Class 1	
Class 2	
Class 3	

2.5 Classification and ranking Result

In this section, we show the result of the clustering and ranking method. We prepared the estonian map as the basemap which are terrain and population coloring map.

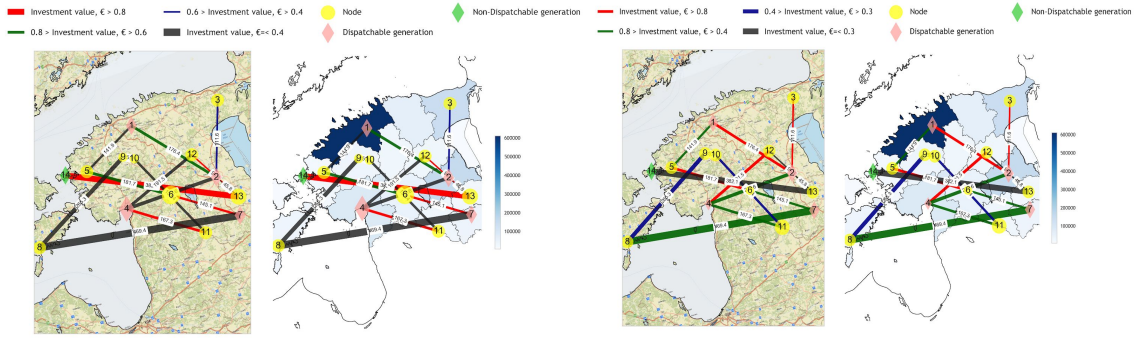


Fig. 28. IEEE14 Base network on Estonia(Left: Markov dataset, Right: Brownian dataset)

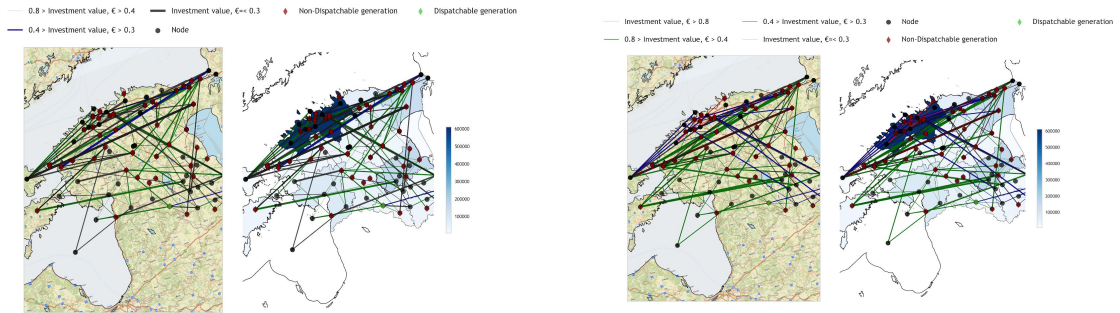


Fig. 29. IEEE118 Base network on Estonia(Left: Markov dataset, Right: Brownian dataset)

Fig. 28 and Fig. 29 show the network of IEEE14 and IEEE118 which are generated by using Markov switching-autoregressive and Brownian motion methods. These networks are subjected to clustering to make 4 classes of each Arc.

Table. 26. Classification result

Data	Supervised	Avg. test				Avg. crossvaridation				Avg. AUC			
		CNN	DCT	SVM	XGB	CNN	DCT	SVM	XGB	CNN	DCT	SVM	XGB
brown-14	DBSCAN	0.333	0.133	0.333	0.133	0.656	0.726	0.464	0.821	0.477	0.648	0.591	0.102
	kmeans	0.267	0.333	0.333	0.200	0.906	0.667	0.728	0.739	0.577	0.615	0.500	0.038
	pca	0.267	0.267	0.067	0.267	0.833	0.389	0.500	0.389	0.038	0.000	0.077	0.000
	tsne	0.267	0.200	0.267	0.267	0.573	0.131	0.083	0.262	0.500	0.500	0.398	0.318
markov-14	DBSCAN	0.333	0.200	0.067	0.333	0.552	0.345	0.476	0.464	0.239	0.477	0.261	0.466
	kmeans	0.067	0.800	0.800	0.733	0.896	0.700	0.822	0.767	0.500	0.643	0.750	0.679
	pca	0.333	0.333	0.000	0.067	0.865	0.517	0.838	0.727	0.536	0.750	0.286	0.214
	tsne	0.267	0.067	0.067	0.067	0.682	0.131	0.476	0.226	0.477	0.398	0.295	0.250
brown-118	DBSCAN	0.160	0.286	0.257	0.229	0.504	0.702	0.656	0.675	0.360	0.597	0.408	0.597
	kmeans	0.406	0.291	0.429	0.160	0.808	0.764	0.811	0.747	0.516	0.544	0.696	0.432
	pca	0.720	0.331	0.429	0.389	0.796	0.584	0.761	0.624	0.963	0.881	0.951	0.664
	tsne	0.063	0.389	0.251	0.406	0.713	0.690	0.599	0.723	0.118	0.750	0.690	0.810
markov-118	DBSCAN	0.189	0.291	0.234	0.257	0.564	0.314	0.292	0.281	0.458	0.527	0.386	0.531
	kmeans	0.503	0.669	0.451	0.606	0.874	1.000	0.657	1.000	0.766	0.840	0.817	0.938
	pca	0.011	0.000	0.000	0.000	0.956	0.972	0.932	0.955	0.038	0.000	0.027	0.000
	tsne	0.251	0.286	0.263	0.280	0.670	0.257	0.211	0.217	0.480	0.480	0.497	0.480

In table. 26, you can see the result of the classification. We have used three evaluation indices. The first indice is Accuracy using test data, second is cross validation score, the third is area under ROC curve. As the table shown, In IEEE14, it can be seen that kmeans in the data set by Markov stably produces a value close to 80% in any Unsupervised

algorithm. Moreover, in IEEE 18, since the number of data is large compared to IEEE 14, the values of the overall are stable. The best test score was PCA in the Brownian dataset.

Lastly, we select the two network which results the best in the classification part and show the network with ranking output. The selected networks are IEEE14-Markov-kmeans and IEEE18-Brownian-PCA.

Fig. 30. Result of IEEE14-Markov-kmeans

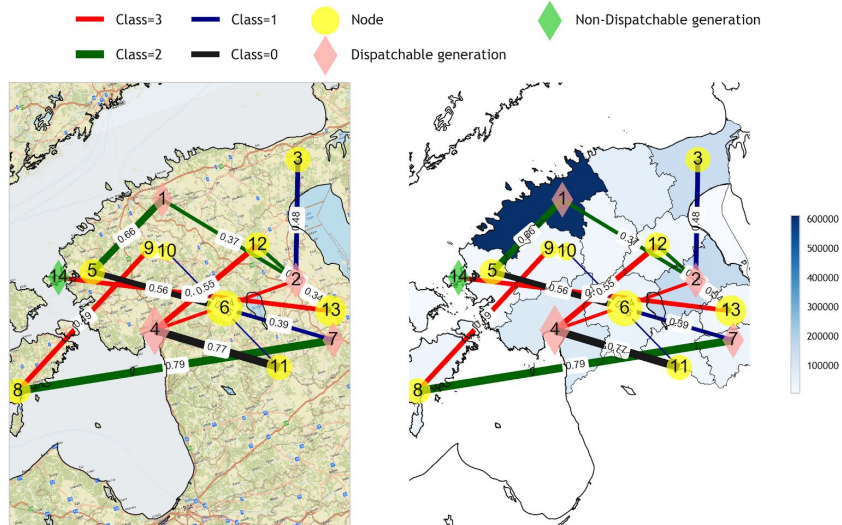


Fig. 30 shows the network that achieved the best result in IEEE14 network. 4 classes 0-3 are colored *red*, *blue*, *green* and *black*. The width of the line showed the result of ranking. It can be seen that each arc is classified and the importance is visualized by thickness in that class. However, the data amount is very small, so it is difficult to see the clustered class.

Fig. 31. Result of IEEE18-Brownian-PCA

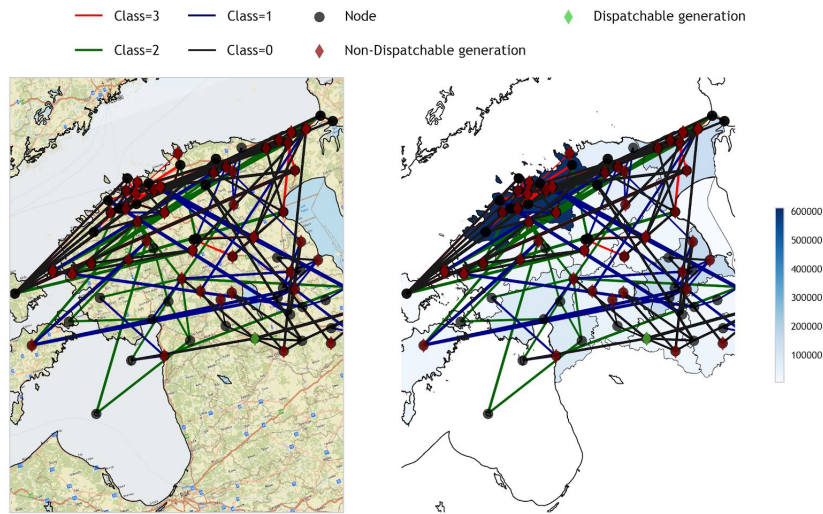


Fig. 31 shows the result of IEEE18-Brownian-PCA which outputs the best score. As the number of dataset increases, the result of classification shows the each class clearly. Class 3 is concentrated on Harju county, and Class 1 is observed in Arc connecting Tartu and Tallinn, Class 2 is in vertical Arc of Harju, and Class 0 is slightly closer to the sea than Class 2. Considering the power transfer distribution factors and line outage angles , variance values are calculated. These variance values are used as weight/score for each line. The thickness of the line depends on the weight of the line.

Future steps

As a part of the overall project development, we have planned the following tasks as in Table. 27.

Table. 27. Future task list

	Task Title	Time	Status
Prediction	Run the script in Taltech HPC cluster computer	June	Pending
	Hyper parameter model tuning (Apache Spark) based on the HPC	June	Pending
	Test more models such as RBM, NTM, GAN	June	Pending
Classification	Use Estonian data to connect with Pypsa project	June	pending
	Improve the ranking method by adding more conjunction parameters	June	pending
	Improve the visualization of classification result	June	pending

By completing these tasks, I believe that the contents of this time will be more meaningful and my contribution would be valuable to the power network research.

SUMMARY and Conclusion

The objective of this thesis is to compare the different machine learning methodologies and see the performance within two parts related with energy applications. The two systems are divided into prediction and classification part.

1) Prediction part:

We have predicted Temperature, Wind power and Ramp events extracted from wind power data in 2016. A ramp event can measure quantification of the wind power variations. In this prediction, the data from 2011 to 2016 is used for training data to predict the data in 2016. There are 5 machine learning models: DFF, DCN, CNN, LSTM and Attention which we tested. In addition to that, we prepared 2 kinds of filter to apply for our datasets. The one is Wavelet filter (WT) which can provide localization in both the temporal domain as well as in the frequency domain. Another one is Fast Fourier transform (FFT) which is able to convert the original data to a representation in the frequency domain. It means we have done (3 datasets) x (3 datasets) x (5 machine learning model) = 45 cases of simulations. As the result of these simulation, we could see Attention model is the best for the prediction of Wind power, Temperature and ramp events. In particular, Attention model with FFT performs very well compared with other models. In addition to that, we applied Hyperparameter optimization to the Attention model to improve the result more and more. By that tuning, we could see the drastic improvement of the result for most of the datasets. For the future steps, we are planning

to try to implement other kinds of machine learning methods to test and do the distributed optimization to reduce the computing time and increase the accuracy.

2) Classification part:

In this part, we have tested IEEE-14 and IEEE-118 bus data to implement machine learning method. We have adapted 4 unsupervised and supervised machine learning model respectively to categorize the buses into 4 classes. Since the network data is more sensitive and not open public, firstly we generated the basic data to store in each IEEE-14 and IEEE-118 buses. We chosen Markov-switching autoregressive model and Brownian motion to generate that. Based on the 4 datasets (IEEE-14 with markov and Brownian sampling, IEEE-118 with matkov and Brownian sampling), we embed the network on Estonian geographical data with population heat-map and terrain map. We could visualize the networks with IEEE datasets and confirm the arcs are categorized into 4 classes[0,1,2,3]. The most meaningful classification method was Markov-kmeans for IEEE-14 and Brownian-PCA for IEEE-118. Next, with the result of clarification, we have ranked the each arcs in the each class by using power transfer distribution factor (PTDF) and Voltage angles. These factors can be calculated in Pypsa and constitute the linear relationship between the active power flows on the lines and nodal active power balance. With these factors, we have done with the visualization of arc classification by color and the ranking by arc width. Next step is that we are using extracting the real data from Estonian government statistic data store and combine with Pypsa project to improve the ranking method.

Upon completion of the project, the related codes would be made available with a MIT license at the following link at GitHub: https://github.com/sambeets/RBA_Prediction. The codes are written in Python programming language with an open-science initiative.

LIST OF REFERENCES

- [1] M. Bauer and J.-L. Scartezzini, "A simplified correlation method accounting for heating and cooling loads in energy-efficient buildings," *Energy and Buildings*, vol. 27, no. 2, pp. 147–154, Apr. 1998.
- [2] Z. Li, L. Ye, Y. Zhao, X. Song, J. Teng, and J. Jin, "Short-term wind power prediction based on extreme learning machine with error correction," *Prot Control Mod Power Syst*, vol. 1, no. 1, p. 1, Jun. 2016.
- [3] S. Pfenninger and I. Staffell, "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, 2016.
- [4] S. Mishra, M. Leinakse, I. Palu, and J. Kilter, "Ramping Behaviour Analysis of Wind Farms," in 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Palermo, 2018, pp. 1–5.
- [5] H. Liu, H. Tian, and Y. Li, "Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms," *Energy Conversion and Management*, vol. 100, pp. 16–22, 2015.
- [6] S. Khan and M. K. Ahmad, "Some Results on Wavelet Frame Packets," *APM*, vol. 04, no. 11, pp. 601–609, 2014.
- [7] "WaveletTourChap1-2-3.pdf."
- [8] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, Sep. 2006.
- [9] B. Torr sani, "An Overview of Wavelet Analysis and Time-Frequency Analysis (a minicourse)," p. 27.
- [10] N. Nikolaou and I. A. Antoniadis, "Application of Wavelet Packets in Bearing Fault Diagnosis," 2001.
- [11] PyWaves, Object-oriented library for the Waves blockchain platform: PyWaves/PyWaves. 2019.
- [12] M. Mathieu, M. Henaff, and Y. LeCun, "Fast Training of Convolutional Networks through FFTs," arXiv:1312.5851 [cs], Dec. 2013.
- [13] "Fjodor van Veen, Author at The Asimov Institute." [Online]. Available: <http://www.asimovinstitute.org/author/fjodorvanveen/>. [Accessed: 10-May-2019].
- [14] "Deep Learning." [Online]. Available: <http://www.deeplearningbook.org/>. [Accessed: 27-May-2019].
- [15] Z. Li, L. Ye, Y. Zhao, X. Song, J. Teng, and J. Jin, "Short-term wind power prediction based on extreme learning machine with error correction," *Prot Control Mod Power Syst*, vol. 1, no. 1, p. 1, Jun. 2016.
- [16] A. Vaswani *et al.*, "Attention Is All You Need," arXiv:1706.03762 [cs], Jun. 2017.
- [17] Y. Niv *et al.*, "Reinforcement learning in multidimensional environments relies on attention mechanisms," *J. Neurosci.*, vol. 35, no. 21, pp. 8145–8157, May 2015.

- [18] "IDEAS/RePEc search." [Online]. Available: <https://ideas.repec.org/cgi-bin/htsearch?q=On+comparing+three+artificial+neural+networks+for+wind+speed+forecasting>. [Accessed: 27-May-2019].
- [19] H. Shao, X. Deng, and Y. Jiang, "A novel deep learning approach for short-term wind power forecasting based on infinite feature selection and recurrent neural network," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 4, p. 043303, Jul. 2018.
- [20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a Deep Convolutional Network for Image Super-Resolution," in *Computer Vision – ECCV 2014*, 2014, pp. 184–199.
- [21] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN," *arXiv:1803.04831 [cs]*, Mar. 2018.
- [22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014.
- [23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," p. 39.
- [24] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv:1212.5701 [cs]*, Dec. 2012.
- [25] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *arXiv:1206.5533 [cs]*, Jun. 2012.
- [26] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," p. 9.
- [27] "IEEE 14-bus modified test system." [Online]. Available: <https://www.kios.ucy.ac.cy/testsystems/index.php/dynamic-ieee-test-systems/ieee-14-bus-modified-test-system>. [Accessed: 27-May-2019].
- [28] Power Systems Test Case Archive [online]. Available: http://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm. [Accessed: 27-May-2019].
- [29] P. Ailliot and V. Monbet, "Markov-switching autoregressive models for wind time series," *Environmental Modelling & Software*, vol. 30, pp. 92–101, 2012.
- [30] "Delpini_tesi.pdf." .

- [31] M. Ester, H.-P. Kriegel, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," p. 6.
- [32] "k-means++: The advantages of careful seeding" Arthur, David, and Sergei Vassilvitskii, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics (2007)
- [33] Laurens van der Maaten, Geoffrey Hinton; 9(Nov):2579--2605, 2008.
- [34] "arXiv:1412.6980 PDF." .
- [35] S. Kiranyaz, T. Ince, R. Hamila, M. Gabbouj, "Convolutional Neural Networks for Patient-specific ECG Classification," in Proc. 37th Annual International Conf. of the Engineering in Medicine and Biology Society Conference(EMBC), Milano, 2015, pp. 2608-2611.
- [36] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, pp. 785–794, 2016.
- [37] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for Power System Analysis," Journal of Open Research Software, vol. 6, no. 1, p. 4, Jan. 2018.
- [38] I. Musirin and T. K. A. Rahman, "Hybrid neural network topology (HNNT) for line outage contingency ranking," in Proceedings. National Power Engineering Conference, 2003. PECon 2003., 2003, pp. 220–224.